



**SIEMENS**

Development and evaluation of a web based  
monitoring system

By

**Trond Undrum & Thomas Andersen**

**Master thesis in  
Information- and communication technology**

**Agder University College  
Faculty for technology**

**Oslo, May 2006**



## Summary

Siemens Business Services (SBS) Department 6211-CIC is a Customer Interaction Center (CIC). The CIC consists of Service Desk (SD) teams which helps customers with ICT problems, mostly over the phone. Each of the SD teams has a Team Leader who is in charge, and there is an Escalation Manager and a Department Manager in the department.

Each of SBS's customers has a Service Level Agreement (SLA) with SBS where their relationship is described.

Our task has been to develop and evaluate a web based monitoring system to help SBS get SLAs within target.

We used Contextual Design as our design method for the new system. Contextual Design is a User Centered Design method which focuses on making a usable system.

During the process of Contextual Design we interviewed Team Leaders and Escalation Managers, created models of their work today, designed work models for the new system, designed the new system, and created a paper prototype.

The Rational Unified Process was selected as the development process. The process focuses on UML, and we created UML documents based on the results we got from Contextual Design.

We programmed a nearly finished system before we used a heuristic evaluation method to evaluate the usability of the system.

The result from the evaluation was used to conduct a new round of programming where the usability problems and system shortages that were discovered were fixed, resulting in an iterative development process.

When the program was changed and the errors corrected we conducted a user test which resulted in more problems and new system shortages. The problems were fixed into a new version of the program and the documentation.

We conducted a second user test on the new version which indicated that the system has become more usable and that we had succeeded with adding the system shortages.

## Foreword

This report is a final master thesis in information- and communication technology at Agder University College, faculty for technology in Grimstad. The study and work that have been performed during this master thesis have carried out from January to May 2006.

We would like to thank Siemens SBS department 6211 including the Department Manager and all Team Leaders for providing us with an exciting and comprehensive task. Frank Lyder Hassel, senior system at SBS, for helping us with the database. A special thanks to the previous Escalation Manager at Siemens SBS department 6211, Bjørn Henrik Pedersen, for collaborating with us and providing us with input and guidance throughout the master thesis despite a busy work schedule.

Finally we would like to thank our supervisor, Lars Line, in this thesis for useful and helpful tips and advice.

Oslo 27. May 24, 2006

Trond Undrum and Thomas Andersen

# Content

Summary .....	III
Foreword.....	IV
Content .....	V
Figure list .....	VII
Table list .....	VIII
Table list .....	VIII
1 Introduction .....	1
1.1 Background .....	2
1.2 Statement of the problem .....	3
1.3 Research question .....	4
1.5 Delimitations and assumptions .....	4
1.6 The significance of the study.....	4
1.7 Definitions and terms.....	5
2 Theory.....	7
2.1 Design .....	7
2.1.1 User Centered Design .....	8
2.2 Development .....	10
2.2.1 Rational Unified Process.....	11
2.3 Evaluation.....	12
2.3.1 Usability.....	12
2.3.2 Selection of usability definition.....	14
3 Methods / Methodology .....	15
3.1 Sub problem 1 - Design.....	15
3.1.1 Data needed for sub problem 1 .....	15
3.1.2 The research methodology for sub problem 1 .....	16
3.1.3 Data collection strategies for sub problem 1.....	18
3.1.4 The specific treatment of data for sub problem 1 .....	18
3.2 Sub problem 2 - Development .....	18
3.2.1 Data needed for sub problem 2 .....	19
3.2.2 The research methodology for sub problem 2.....	19
3.2.3 Data collection strategies for sub problem 2.....	20
3.3 Sub problem 3 - Evaluation.....	21
Data needed for sub problem 3 .....	21
The research methodology for sub problem 3.....	22
Data collection strategies for sub problem 3.....	23
The specific treatment of data for sub problem 3 .....	24
3.3.1 Comparing the research methodologies.....	24
3.3.2 Methods of choice.....	25
4 Accomplishment .....	27
4.1 Design .....	27
4.1.1 Contextual Inquiry .....	27
4.1.2 Work Modeling .....	27
4.1.3 Consolidation .....	28
4.1.4 Work Redesign .....	28
4.1.5 User Environment .....	28
4.1.6 Mockup and Test with Customers .....	29
4.2 Development .....	30
4.2.1 Documents.....	30
4.2.2 Software development .....	31
4.3 Evaluation.....	33
4.3.1 Heuristic evaluation.....	33
4.3.2 User testing.....	34
5 Results.....	36
5.1 Design .....	36
5.1.1 Contextual Inquiry .....	36
5.1.2 Work Modeling .....	38
5.1.3 Consolidation .....	39
5.1.4 Work Redesign .....	39
5.1.5 User Environment .....	39

---

5.1.6 Mockup and Test with Customers .....	39
5.2 Development .....	40
5.2.1 Documents .....	40
5.2.2 Software development .....	41
5.3 Evaluation.....	43
5.3.1 Heuristic evaluation.....	43
5.3.2 User test 1 .....	48
5.3.3 User test 2.....	51
6 Discussion .....	55
6.1 The use of Contextual Design as a design method .....	55
6.1.1 Contextual Inquiry .....	55
6.1.2 Work Modeling .....	55
6.1.3 Consolidation .....	56
6.1.4 Work Redesign .....	57
6.1.5 User Environment Design.....	57
6.1.6 Mockup And Test With Customers .....	58
6.1.1 The use of Contextual Design as a design method .....	58
6.2 Experience from the development .....	59
6.2.1 The development documents .....	59
6.2.2 The database .....	59
6.2.3 The web application and the Windows application.....	59
6.2.4 The implementation .....	60
6.3 Experience from the evaluation.....	60
6.3.1 Experience from the heuristic evaluation.....	60
6.3.2 Experience from the user tests .....	62
6.3.3 Usability.....	63
6.3.4 The use of heuristic evaluation and user tests .....	68
7 Conclusion .....	69
7.1 Future directions/research .....	70
References .....	72
Appendix.....	73

## Figure list

Figure 1.1.1: Department 6211-CIC organization .....	2
Figure 2.2.1. The old water fall development process.....	10
Figure 2.2.2. Typical iterative flow. ....	10
Figure 2.3.1: Learning curves for a hypothetical system.....	12
Figure 3.3.1: The usability problems that are easiest to find.....	24
Figure 4.1.1: The paper prototype in use .....	29
Figure 4.1.2: Heuristic evaluation at Siemens.....	34
Figure 4.2.1: User testing at Siemens SBS department 6211. ....	35
Figure 5.2.1: The class diagram for the .cs classes.....	41
Figure 5.2.2: The class diagram for the .aspx files.....	41
Figure 5.2.3: Entity Relation Diagram.....	42
Figure 5.2.4: System overview .....	42
Figure 5.3.1: The figure shows which evaluators found which usability problems.....	44
Figure 5.3.2: Before the heuristic evaluation. ....	45
Figure 5.3.3: After the heuristic evaluation. ....	46
Figure 5.3.4 A: Before the heuristic evaluation. ....	46
Figure 5.2.4 B: After the heuristic evaluation. ....	46
Figure 5.3.5: Incomprehensible error message.....	46
Figure 5.3.6.A: Before the heuristic evaluation. ....	47
Figure 5.3.6 B: After the heuristic evaluation. ....	47
Figure 5.3.7: Inconsistent use of tables between the different pages. ....	47
Figure 5.3.9: The figure shows which test users found.....	49
Figure 5.3.10.A: Before user test 1.....	50
Figure 5.3.10.B: After user test 1.....	50
Figure 5.3.11: The response field before and after user test 1. ....	50
Figure 5.3.12.A: The drop-down-menu on the "Users" page before user test 1 .....	51
Figure 5.3.12.B: After user test 1.....	51
Figure 5.3.13: The figure shows which test users found which usability problems.....	52
Figure 5.3.14.A: The text next to the checkbox before user test 2.....	53
Figure 5.3.14.B: The text next to the checkbox after user test 2.....	53
Figure 5.3.15: The user could only view workload for a certain date. ....	53
Figure 5.3.16: A new calendar was added in the "Workload" page .....	54
Figure 6.1.1: Consolidation model.....	56

## Table list

Table 3.3.1: J. Nielsen's 5 point rating scale .....	24
Table 5.1.1: Contextual Inquiry question 1, with answers and the number of agreeing users.....	36
Table 5.1.2: Contextual Inquiry question 2, with answers and the number of agreeing users.....	37
Table 5.1.3: Contextual Inquiry question 3, with answers and the number of agreeing users.....	37
Table 5.1.4: Contextual Inquiry question 4, with answers and the number of agreeing users.....	37
Table 5.1.5: Contextual Inquiry question 5, with answers and the number of agreeing users.....	37
Table 5.1.6: Contextual Inquiry question 6, with answers and the number of agreeing users.....	38
Table 5.3.1: An overview of how many usability problems found .....	43
Table 5.3.2: An overview of how many usability problems found .....	48
Table 5.3.3: An overview of how many usability problems found .....	52
Table 6.3.1: Comparing time spent on each task.....	64
Table 6.3.2: Comparing time user test 1 vs. user test 2.....	65
Table 6.3.3: Comparing errors user test 1 vs. user test 2 .....	66
Table 6.3.4: Comparing system errors user test 1 vs. user test 2.....	66
Table 6.3.5: The calculated median for rating subjective satisfaction in the two user tests.....	67
Table 6.3.6: Comparing joy/frustration user test 1 vs. user test 2.....	68



# 1 Introduction

The background for this master thesis is based on the collaboration between Siemens Business Services department 6211-CIC, Agder University College, and ourselves. Siemens wants a web based system to monitor service performance at their customer interaction center (CIC). In Addition to develop this system, we will also design and evaluate the system based on a user centered design method and a usability engineering method. To view the complete task description, see appendix A.

When developing a web based monitoring system there are many things to consider. A monitoring system requires that the users of the system are able to quickly apprehend the information they are searching. For instance, in the case of this project, the users have to be able to rapidly perceive and understand that the customer response time is above the agreed Service Level Agreement (SLA). It also requires that the users are made aware of critical information by the monitoring system.

Time is an essence in such a system as a company loses money every second above the agreed SLA. For a large company with hundreds of calls every day, an implementation of a successful monitoring system could reduce the costs tremendously.

In order for relevant data to be found and critical information to be detected quickly, requires that data are organized well and that information are presented in such a way that the users can discover desired information in a glance.

In this project we will implement suitable methods into the development of a web based monitoring system. We will describe how we implemented the different methods and discuss the outcome of the methods used to develop the system.

This study is bound in chapter 1. The main problem is stated and divided into three sub problems.

In chapter 2 we will present different UCD methods for sub problem 1 and select an appropriate method. Two definitions on usability, ISO 9241-11 and J. Nielsen, is presented for sub problem 3, and we will choose one of this definitions to use it on the evaluation of the web based system.

In chapter 3 we will step into the methodologies used in the selected method (Contextual Design) and definition (J. Nielsen's [8] definition on usability) presented in chapter 2. Elements from the selected development process (RUP) is presented and described in sub chapter 3.2. The chapter describes how we *intend* to accomplish the three sub problems

Chapter 4 will describe the accomplishment of the methods that were selected in chapter 2. The accomplishment of the evaluation methods selected in chapter 3 will be described in sub chapter 4.3

In chapter 5 the results of the accomplishment from the selected methods is presented. Sub chapter 5.1 presents the results from the Contextual Design, the results from the development process is presented in sub chapter 5.2. In sub chapter 5.3 the measurements and results from the evaluation is presented.

In chapter 6 we will discuss the results found in chapter 5. Our experience with the accomplishment of the methods used will also be discussed.

Finally, in chapter 7 we will present the most significant parts of the discussion in chapter 6.

## 1.1 Background

Siemens Business Services is a wholly-owned subsidiary of Siemens and the Norwegian office is located at Linderud in Oslo. SBS has considerable experience in outsourcing and IT managed services, networking, large-scale installations, and end-to-end systems maintenance. With over 36 100 employees in 40 countries and annual sales for 4.7 billion € SBS is one of the ten largest IT companies in the world [7].

SBS in Norway has a Customer Interaction Center (CIC) which is organized in Service Desk (SD) teams and is called Department 6211-CIC. The different teams are serving different types of costumers such as retail customers (Stores and kiosks), schools, the military, and Siemens itself. Each customer has its own telephone number which connects the customer to the same SD team every time.

Each customer has a SLA with SBS, where they agree on maximum response time (normally 30 sec.), service hours (24/7 365 days a year), availability (99% uptime), reliability (the time between failures), and so on. Everything in the business relationship between SBS and the customer is described in the SLA.

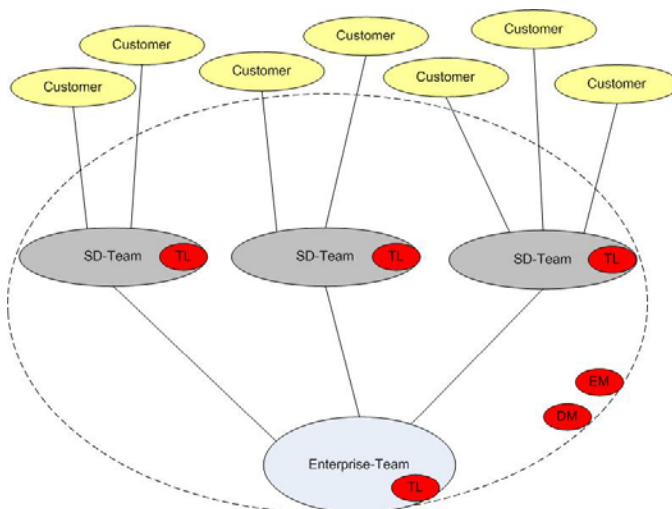


Figure 1.1.1: Department 6211-CIC organization

Department 6211-CIC is organized as figure 1.1. Each Service Desk team has a Team Leader (TL). The department has two superior leaders, one Escalation Manager (EM) and one Department Manager (DM). The Team Leader has the responsibility for the team's performance and is continuously monitoring the average response time per customer, the average response time for the whole team, open incidents in the database, and making sure that no one is in the phone queue for too long.

The two superior leaders are monitoring each team and the whole department. If a team, or the department, isn't performing good enough compared with an agreed SLA, the Team Leader is being noticed and have to do quick adjustments to get the SLA within target.

The SD teams are solving around 94% of the incidents reported. But there are not all incidents the SD teams as a first line instance are able to solve and these are sent on to

the Enterprise team. The Enterprise team is structured in the same way as a regular SD team but they are not responding to phone calls from customers. The Enterprise Team is the second line instance in the process of solving incidents.

Today the Team Leaders, Escalation Manager, and Department Manager are using several different GUIs and a lot of manual work to monitor the teams and the department. HighPath ProCenter is the phone system they use to get response times, OSMq is the tool used to treat information about incidents, Excel and SQL-Query Enterprise is used to get data, and a lot of calculations have to be done manually. This makes it difficult for all of them to make sure that an Agreed SLA is within target while they are doing their other tasks.

## 1.2 Statement of the problem

The purpose of this study is to design and develop a web based system to monitor service performance within department 6211-CIC at Siemens Business Services (SBS). We will evaluate the new system, its graphical user interface (GUI) and system functionality, in accordance with usability.

### Sub problems

The problem, which is stated above, consists of three sub tasks or sub problems. We have, in this chapter, separated out those three sub tasks and divided them into smaller and stepwise work tasks.

Each of the small work tasks needs to be conducted to solve the initial sub task. When all of the three initial sub tasks are solved, the initial problem has been answered.

1. Design and specify a web based system to monitor service performance within department 6211
  - a. Find out how the work is done today
  - b. Find ways to adapt the work to a new web based system and design the system in regard of usability
2. Develop a web based system to monitor service performance within department 6211
  - a. Create implementation documentation based on the new system design. E.g. Use case diagrams, data flow diagrams, class diagram, ER diagrams, and so on.
  - b. Implement system functionality
  - c. Implement GUI
3. Evaluate the new web based system in accordance with usability  
Usability will be considered in the design process and evaluated using different types of usability engineering methods such as; Heuristic evaluation, cognitive walkthrough, and user testing.

This report will emphasize the design and evaluation of the web based system as these two sub problems serves as a background for the research question stated in chapter 1.3 below. The development is a large part of this master thesis and will be mentioned throughout this report, but since it the development is not part of the research question

much of the information regarding the development process has been placed in the appendix section.

### **1.3 Research question**

Based on the problem statement in chapter 1.2 and its sub problems the following research question have been produced:

To evaluate the experience of applying a User Centered Design method (UCD) on sub problem 1 and usability engineering method on sub problem 2.

In chapter 2 we will present different UCD methods for sub problem 1 and select an appropriate method. Two definitions on usability, ISO 9241-11 and J. Nielsen, is presented for sub problem 3, and we will choose one of this definitions to use it on the evaluation of the web based system.

### **1.5 Delimitations and assumptions**

We have taken some assumptions and set delimitations for this study. Some were taken before the project started and some were taken during the project. The delimitations and assumptions we have taken are as follows:

We will not study the change, if any, in number of SLAs within target before and after the implementation of the web based system (SWS). Meaning, we will not perform any tests on the work tasks that are performed today to state whether the SWS is more usable than the old system. For instance, we will not test how fast the users perform certain work tasks at the department today and compare the measures with the test results on the new web based system. Such a study is considered out of context of this thesis due to limited time.

We will not state whether the system is good or not. However, we will use evaluation methods to discuss whether the web based system has become more usable during the project time.

We are assuming that the users of this system have higher computer skills than a novice user. We consider the users of the SWS to be expert users.

The Snapshot website can only be accessed from inside the SBS intranet.

We are assuming that the readers of this thesis have some computer knowledge.

We are assuming that everyone working on the SBS intranet are added to the Active Directory.

We are not using the Rational Unified Process to its full in this study. We have only used some elements, and adjusted those elements to this study.

### **1.6 The significance of the study**

This study could be very significant for the Department 6211-CIC, their leaders, and the SBS management. If the study is successful, and the web based monitoring system developed is usable, will;

- SBS have the possibility to see how the Department is currently performing
  - o Adjustments can be made if the performance is lower than what to expect.
- SBS have the possibility to go back and see the performance history.
  - o For instance, to use it as a reference to look ahead when they are setting up work plans.
- The Team Leaders, Escalation Mangers, and Department Manager will be able to get all information they need from one GUI.
  - o All the manual work done in Excel and other programs today will be unnecessary. Thus, the users will save time.

SBS has also considered taking this study further and deploy the web application to other locations which uses HPPC.

This study has also been very important for us. It is our Master Thesis, and our performance is graded by Agder University College (HIA).

We did also profit on this study our selves: We Learnt C# and ASP.net, we were allowed to sit in a regular work environment and see how the daily work in a company as big as SBS is, and we got a feeling of the real work life. We did also feel the pressure of being our own bosses and create our own work day, creating our own tasks, and be strict to our selves.

It has been very inspiring that SBS has been satisfied with our solution. They have talked with us several times about what they need to do for rolling the SWS out on another location, and if it would be possible to sell the SWS to other companies.

## 1.7 Definitions and terms

We will in this chapter go through the different definitions and terms we have used in this thesis.

HIA	Høgskolen I Agder <i>[<a href="http://www.hia.no">www.hia.no</a>]</i>
SBS	Siemens Business Services <i>[<a href="http://www.sbs.siemens.no/">http://www.sbs.siemens.no/</a>]</i>
SD	Service Desk
CIC	Customer Interaction Center
DM	Department Manager
EM	Escalation Manager
TL	Team Leader
SLA	Service Level Agreement: An agreement between SBS and a customer on what kind of service SBS are to deliver. For Department 6211-CIC the major factor of the SLA is the response time from a phone starts to ring until it is picked up. Different departmens have different factors to take regard to in the SLA. <i>[<a href="http://en.wikipedia.org/wiki/Service_Level_Agreement">http://en.wikipedia.org/wiki/Service_Level_Agreement</a>]</i>
SWS	Snapshot WebSite
HPPC	HighPath ProCenter: A Siemens developed phone system for Service desks.
OSMq	Open Source Message Queue: The system SBS uses for registering Incidents, Orders, and other incoming requests. <i>[<a href="http://www.osmq.org/">http://www.osmq.org/</a>]</i>
XML	eXtensible Markup Language

---

HTML	<a href="http://en.wikipedia.org/wiki/XML">[http://en.wikipedia.org/wiki/XML]</a> Hyper Text Markup Language <a href="http://en.wikipedia.org/wiki/HTML">[http://en.wikipedia.org/wiki/HTML]</a>
UML	Unified Modeling Language <a href="http://en.wikipedia.org/wiki/Unified_Modeling_Language">[http://en.wikipedia.org/wiki/Unified_Modeling_Language]</a>
RUP	Rational Unified Process <a href="http://en.wikipedia.org/wiki/RUP">[http://en.wikipedia.org/wiki/RUP]</a>
ASP	Active Server Pages <a href="http://en.wikipedia.org/wiki/Active_Server_Pages">[http://en.wikipedia.org/wiki/Active_Server_Pages]</a>
ASP.net	Active Server Pages using the Microsoft .net platform <a href="http://en.wikipedia.org/wiki/ASP.NET">[http://en.wikipedia.org/wiki/ASP.NET]</a>
C#	C sharp, A C-programming language. <a href="http://en.wikipedia.org/wiki/C_Sharp">[http://en.wikipedia.org/wiki/C_Sharp]</a>
ER	Entity Relationship diagram <a href="http://en.wikipedia.org/wiki/ER_diagram">[http://en.wikipedia.org/wiki/ER_diagram]</a>
UCD	User Centered Design <a href="http://en.wikipedia.org/wiki/User_Centered_Design">[http://en.wikipedia.org/wiki/User_Centered_Design]</a>
LDAP	Lightweight Directory Access Protocol <a href="http://en.wikipedia.org/wiki/LDAP">[http://en.wikipedia.org/wiki/LDAP]</a>
HCI	Human Computer Interaction <a href="http://en.wikipedia.org/wiki/Human-Computer_Interaction">[http://en.wikipedia.org/wiki/Human-Computer_Interaction]</a>
SQL	Structured Query Language <a href="http://en.wikipedia.org/wiki/SQL">[http://en.wikipedia.org/wiki/SQL]</a>
OLE-DB	Object Linking and Embedding for Data Bases <a href="http://en.wikipedia.org/wiki/OLE_DB">[http://en.wikipedia.org/wiki/OLE_DB]</a>
ODBC	Open Database Connectivity <a href="http://en.wikipedia.org/wiki/ODBC">[http://en.wikipedia.org/wiki/ODBC]</a>
GUI	Graphical User Interface <a href="http://en.wikipedia.org/wiki/GUI">[http://en.wikipedia.org/wiki/GUI]</a>
DTS	Data Transformation Service <a href="http://en.wikipedia.org/wiki/Data_Transformation_Services">[http://en.wikipedia.org/wiki/Data_Transformation_Services]</a>

## 2 Theory

This chapter is divided into three main parts based on the sub problems presented in chapter 1.2. We will, in this chapter, define in which research areas this thesis comprises for each of the three sub problems.

We will present different methods that are relevant to accomplish sub problem 1, design (chapter 2.1.1.1), and choose one appropriate method which we will use in the design of the web based system.

Chapter 2.2 presents RUP as a reference method for sub problem 2, the development phase of the project.

In chapter 2.3.1 we present two definitions on usability which can be used to evaluate the SWS as sub problem 3 states. We will choose one of these definitions in chapter 2.3.2.

### 2.1 Design

Human-computer Interaction (HCI) is an important area when designing a new computer system. ACM has described HCI as a field concerning; the structure of communication between human and machine, human capabilities to use the machines (including the learnability of interfaces), algorithms and programming of the interface, and design.

IBM defines HCI as: “HCI is an area where the work is concentrated around the behavioral, aesthetic, and value-sensitive aspects of the design of interactive systems” [4].

ACM defines HCI as: “Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them” [3].

HCI is concerned with [23]: Methodologies and designing interfaces (i.e. optimizing for a desired property such as learnability or efficiency of use), methods for implementing interfaces, techniques for evaluating and comparing interfaces, and developing new interfaces and interaction techniques.

Many methodologies outlining techniques for HCI interaction design exist. “Modern models tend to focus on a constant feedback and conversation between the users, designers, and engineers and push for technical systems to be wrapped around the types of experiences users want to have, rather than wrapping user experience around a complete system” [23]. Such a model is appropriate for developing the Snapshot Website (SWS) as we are dependent on information and feedback from the users of the web based system throughout the development (here: development includes all of the three sub problems stated in chapter 1.2).

User-centered design (UCD) is such a model. “UCD is a modern design philosophy rooted in the idea that users must take center-stage in the design of any computer system” [23]

We will in the following subchapter, chapter 2.1.1, look further into User-centered design.

## 2.1.1 User Centered Design

Henry and Grossnickle define UCD as: “a user interface design process that focuses on usability goals, user characteristics, environment, tasks, and workflow in the design of an interface” [1].

In ISO 13407 is UCD defined as: “an approach to interactive system development that focuses specifically on making systems usable. It is a multidisciplinary activity” [6].

### 2.1.1.1 User Centered Design Methods

There are many UCD methods which are used today. We will not mention all of them in this sub chapter, but we will present three of the UCD methods which follow the ISO 13407 standard.

“*Cooperative design* involves designers and users on an equal footing. This is the Scandinavian tradition of IT artifacts and it has been evolving since 1970” [24].

“*Participatory design (PD)*, North American term for about the same, inspired in Cooperative Design, focusing on the participation of the users” [23].

*Contextual Design* has some ideas from Participatory Design. Beyer and Holtzblatt [2] says: “Contextual Design is a state-of-the-art approach to designing products directly from a designers understanding of how the customer works...The best product designs result when the product’s designers are involved in collecting and interpreting customer data and appreciate what real people need. Contextual Design gives designers the tools to do just that”.

### 2.1.1.2 The selection of a UCD method

Contextual Design is the method of choice. This method focuses on involving the users of the system throughout the process of developing the SWS. The method also follows the ISO 13407 standard [6] which is suitable for this thesis as the standard focuses specifically on making systems usable during the development. Beyer and Holtzblatt [2] states: “According to the Contextual Design approach, data gathered from customers is the base criterion for deciding which needs to address, what the system should do, and how it should be structured”. The approach corresponds to our situation where we have to base the development of the SWS from the information given by the users of the system.

In the next chapter, chapter 2.1.1.3, we will give a more detailed description of the selected UCD method, Contextual Design.

### 2.1.1.3 Contextual Design

Beyer and Holtzblatt[2] state that Contextual Design is a User-centered Design method in six steps:

*Contextual Inquiry*: Contextual Inquiry is an explicit step for understanding who the customers really are and how they work to day.



The design team conducts one-on-one field interviews at the customer's workplace. The interviewer observes how the customer works and asks about the user's actions step by step to understand their motivations and strategy. Through discussion the interviewer and customer develops an interpretation of the work.

*Work Modeling:* Work models show the work of individuals and organizations in diagrams.

People's work is complex and full of detail. There is no good way to write down or talk about work practice. Design teams seldom have the critical skill of seeing the structure of work done by others, looking past the surface detail to see the intents, strategies, and motivations that control how work is done.

Different models could be used to show different perspectives of the work: The flow models shows communication and coordination, the cultural model shows culture and policy, the sequence model shows detailed steps to accomplish a task, the physical model shows the physical environment as it supports the work, and the artifact model shows how artifacts are used and structured during the work.

*Consolidation:* Consolidation collects data from individual customer interviews so the team can see common patterns and structure without losing individual variation.

Consolidated work models bring together each different type of work model separately to reveal common strategies and intents while retaining and organizing individual differences. This gives the design team a single picture of the customer population a design will address.

*Work Redesign:* Work Redesign brings the designers together to discuss the consolidated data and how technology can improve the work.

This focuses the conversation on how technology helps people get their jobs done, rather than on what could be done with technology without considering the impact on people's real lives. If this is done right the system would be more effective than the old one.

*User Environment:* The User Environment Design shows the floor plan of the new system. It shows each part of the system, how it supports the user's work, exactly what function is available in that part, and how the user gets to and from other parts of the system, without tying this structure to any particular user interface.

The User Environment Design will design the system in a sensible way for the user. This will make the system easier to learn, and a system easy to learn is often easy to remember. The interaction between the human and the computer (HCI) will mostly be taken care of in this part of the system.

*Mockup and Test with Customers:* it's important to test and iterate a design early, before anyone becomes invested in the design and before anyone spends time writing code.

Paper prototyping develops rough mockups of the system using Post-it® notes to represent windows, dialog boxes, buttons, and menus. The design team tests these prototypes with users in their workplace, replaying real work events in the proposed system. When the user discovers problems, they and the designers redesign the prototype together to fit their needs.

Each user has during this step the possibility to give feedback on interaction between the user and the system. This will help the designers to adjust any international problems so that the users will find the system structure pleasant.

One of the main goals by using user participation and user orientation in design is to achieve usability.

## 2.2 Development

In “the old days” a sequential software development approach, like the water fall method (Figure 2.2.1), was often used. Each step was finished up before a new step was started. That approach has a big weakness. The users are hardly involved in the process before deployment. If the user’s requirements are changed during development, it is a big risk that the user could end up with the wrong product. [26]

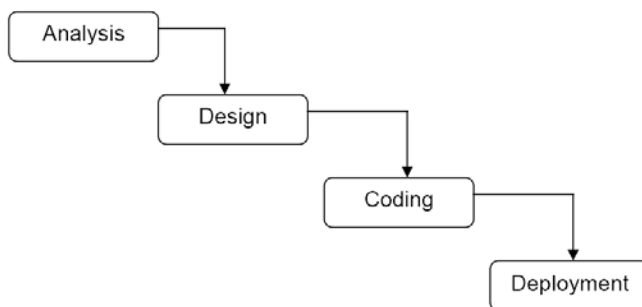


Figure 2.2.1. The old water fall development process [26]

As an alternative to the waterfall method we have the newer iterative development processes. The iterative development process goes through the different steps several times (figure 2.2.2). As an example, one time for each main system part.

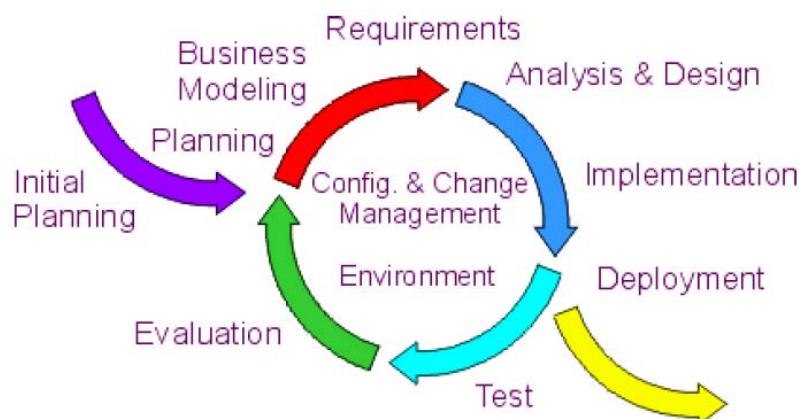


Figure 2.2.2. Typical iterative flow. The work will go in “circles” where the different steps could be done several times. [27]

In the next chapter, chapter 2.2.1, we will look at an iterative development process, Rational Unified Process (RUP).

## 2.2.1 Rational Unified Process

In the 1990's, Rational Software, now a division of IBM, started a process to unify various existing methodologies for object oriented analyze and design into a "Unified Method". This was done in two steps. The first step; Design and publish the Unified Modelling Language (UML)[28] as a notation for any kind of software modeling result, and the second; complement UML with a process description, RUP. [29]

The Rational Unified Process (RUP) is a complete software-development process framework. It is an iterative development process that goes through four phases; Inception, Elaboration, Construction, and Transition. The main focus of RUP is to reduce risks when developing new software. This is done by using six best practices. [27]

The RUP is an incremental process. The overall project is broken down into phases and iterations. The iterations are risk driven and oriented towards mitigation of those risks. Each of the iterations should be executable software that is testable against the project requirements and use cases.

The developing team must decide on the priorities and start with the most important functionality. Feedback from this could be used to refine the requirements, and identify problems before moving on to the next step. This process is repeated, until all requirements are implemented. [26]

### 2.2.1.2 Rational Unified Process In Practice

The six best practices that were mentioned, by Pollise, [27] and Wikipedia [30] the free online encyclopedia.

#### **The six best practices to RUP are:**

*Develop Software iteratively:* When developing large scale sophisticated systems it is not possible to develop it in one step.

*Manage Requirements:* If the requirements are managed in the right way the correct product are generated and all the right features are added.

*Use Component-Based Architecture:* Systems built with component-based architecture are easy to extend, intuitively understandable and promotes software reuse.

*Visually Model Software:* An effective way to get a good overview of your solution is to abstract your code and presenting using graphical building blocks

*Verify Software Quality:* Quality assessment is the most common failing point of all software projects, since it is often handled by a different team.

*Control Changes to Software:* This is very important in lager projects.

## 2.3 Evaluation

In this chapter we will present two definitions on usability which can be used to evaluate the SWS as sub problem 3 states. We will choose one of these definitions in chapter 2.3.2.

### 2.3.1 Usability

In ISO 9241-11 usability is defined as: "Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." [5].

Effectiveness is defined as: "Accuracy and completeness with which users achieve specified goals", efficiency is defined as: "Resources expended in relation to the accuracy and completeness with which users achieve goals", and satisfaction is defined as: "Freedom from discomfort, and positive attitudes towards the use of the product".

Nielsen state that it is important to realize that usability is not a single, one dimensional, property of an interface [8]. Usability has multiple components and is traditionally associated with these five usability attributes:

*Learnability:* The system should be easy to learn so that the user can rapidly start getting some work done with the system.

Learnability is in some sense the most fundamental usability attribute, since the first experience most people have with a new system is that of learning to use it.

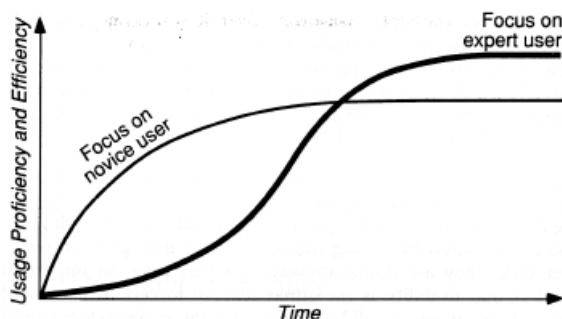


Figure 2.3.1: Learning curves for a hypothetical system that focuses on the novice user, being easy to learn but less effective to use, as well as one that is hard to learn but highly effective for expert users.

Ease of learning refers to the novice user's experience on the initial part of the learning curve, as shown in figure 2.3.1. Highly learnable systems have a steep incline for the first part of the learning curve and allow reaching a reasonable level of usage proficiency within a short time.

Learning is probably also the easiest of the attributes to measure, with the possible exception of subjective satisfaction. New users can be set to do tasks and the time they use to reach a goal is measured. Of course, the test persons should be representatives of the intended users of the system.

The most common way to express the specified level of proficiency is simply to state that the users have to be able to complete a certain task successfully. Alternatively, one can specify that users need to be able to complete a set of tasks in a certain, minimum time before one will consider them as having "learned" the system.

*Efficiency:* The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.

Efficiency refers to the expert user's steady state level of performance at the time when the learning curve flattens out (Figure 2.3.1).

For measuring efficiency of use for experienced users, one obviously needs access to experienced users. For systems that have been in use for some time, "experience" is often defined somewhat informally, and users are considered experienced either if they say so themselves or if they have been users for more than a certain amount of time.

Efficiency can be measured by letting users do a task for some time and measure the workload they have been able to do.

*Memorability:* The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.

It is important to have an interface that is easy to remember. Most users use the interface to identify the system functions.

Interface memorability is rarely tested as thoroughly as the other usability attributes, but there are in principle two main ways of measuring it. One is to perform a standard test with a casual user who has been away from the system for a specified amount of time, and measure the time they spend performing some typical test task. Alternatively, it is possible to conduct a memory test with users after they are finished with a test session on the system and ask them to explain the effect of various commands or name the command (or draw an icon) that does a certain thing.

*Errors:* The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.

*Subjective Satisfaction:* The system should be pleasant to use, so that the users are subjectively satisfied when using it; they like it.

Subjective satisfaction can be an especially important usability attribute for systems that are used on a discretionary basis in a nonworking environment, such as home computing. For some such systems, their entertainment value is more important than the speed with which things gets done.

Subjective satisfaction can be measured as psycho physiological measures such as EEGs, pupil dilation, heart rate, skin conductivity, blood pressure, and the adrenaline in the blood.

Alternatively, subjective satisfaction may be measured by simply asking the users for their subjective opinion.

The ISO definition of usability is very focused on the goal. Nielsen on the other hand focuses more on learnability and memorability of the system. There are several similarities between the two definitions, e.g. they both state that effectiveness is an important issue of usability.

### **Website usability**

Murray states that: "Usability engineering for the Web grew out of the software development discipline of Human Computer Interaction (HCI). However, the Web is different from software, and the nature of the Web poses new challenges to designers and developers who are trying to incorporate usability into their sites" [9]. Murray also gives a set of general properties for a usable Web interface; accessible, appealing, consistent, clear, simple, navigable, and forgiving of user blunders.

The properties for a usable web interface can be recognized from some of the usability attributes from Nielsen [8]. An appealing website will also be subjectively pleasant. If a website is clear, consistent, simple, and navigable there are good chances for that the website also is effective for the user and easy to learn. A system that is easy to learn and is simple would be easier to remember than a complex and hard to learn system.

Before employing any usability techniques, the challenge is to define the Website's target audience. Whether the site is at the beginning stage of development, at the user testing stage, or due for a redesign, it is critical for the designers to have as much knowledge and understanding as possible of the site's users.

This could be set in connection with the Consolidation from Contextual Design where the designer will get a single picture of the customer population the design will address.

Murray also states: "The key to building a useful and usable Website is to involve the user in the development process from the beginning. In recent software development, user-centered design has generally led to improvements in software interfaces and a higher likelihood that the software will actually deliver what the user wants" which corresponds with the definition of Contextual Design [9], [2].

Beyer and Holtzblatt [2] say in their definition that Contextual Design is a state-of-the-art approach for designing products from what a designer understands of how the customers work. If we take their definition of Contextual Design and uses Murray [9] statement of: "The key to building a useful and usable Website is to involve the user in the development process from the beginning" we can conclude that Contextual Design used in a design process could increase the usability of a website.

### **2.3.2 Selection of usability definition**

The ISO definition of usability is very focused on the goal. Nielsen on the other hand focuses more on learnability and memorability of the system. There are several similarities between the two definitions, e.g. they both state that effectiveness is an important issue of usability.

We will in this study use Nielsen's definition for usability and take the attributes he described in regard when designing and developing the system. To achieve this usability we will use Contextual Design as a User Centered Design method and apply evaluation methods during the development of the SWS.

## 3 Methods / Methodology

In this chapter we will step into the methodologies used in the selected method (Contextual Design) and definition (J. Nielsen's [8] definition on usability) presented in chapter 2. Elements from the selected development process (RUP) is presented and described in sub chapter 3.2

We have divided this chapter into three parts, chapter 3.1, 3.2, and .3.3, where each part corresponds to one of the sub problems presented in chapter 1.2.

This chapter describes how we *intend* to accomplish the three sub problems.

### 3.1 Sub problem 1 - Design

The methodology used in the Contextual Design method which was selected to be our design method is presented in this sub chapter.

In sub chapter 3.1.2 we look into how we intend to accomplish the different steps of the Contextual Design method

#### 3.1.1 Data needed for sub problem 1

The data needed as input in the Contextual Inquiry is concerned around the users. The user's explanations of how each work task is performed, a description of how the user performs the work task, which GUIs the users are using for each of the task, and what manual calculations and the formula that is used will be gathered through the interviews.

The interviews will be summarized in digital form, for easier treatment later.

For visualizing the interviews in the Work Modeling the summary of the interviews for each user will be used as background.

There will be created flow models for each work task for each user. The flow models will show the work flow, interactions between the user and the different GUIs, the manual calculations the user does, interactions between the user and other persons, and the underlying databases and system.

After the models are done it is time for the Consolidation. All of the models from Work Modeling are used to summarize each work task will into one single flow model for all the users. Common patterns in the work models will be taken in regard, and the same will individual variations.

When the Consolidation is finished it is time to redesign the work.

The new model will only have one single GUI for all interactions between the system and the user. Interactions between the user and other persons and the underlying system are important parts of the new model.

When the work is redesigned a floor plan must be created. The floor plan is designed under the step for User Environment and will show all interactions from the user to the

system, the different parts of the system, and how the user can move around to those different parts.

At the end, the only thing left is the Mockup and Test with Customers. A paper prototype will be used to get the users impression of the system. Notes about how he solves the tasks and notes possible complains from the customer will be taken.

### **3.1.2 The research methodology for sub problem 1**

Design and specification of the system is done with Contextual Design [2]. The method consists of six steps;

#### *Contextual Inquiry*

We will interview each and one of the Team Leaders, the Escalation Manager, and the Department Manager. The interviews will be done at the user's work place.

A set of work tasks will be defined in cooperation with the Escalation Manager. The tasks are regular work tasks that are supposed to be done every day. All interviews will use the same set of tasks to make sure that the models developed are concise.

There are two ways to get input from the user, through conversation or demonstration. We will focus on making notes of what the user demonstrates, that is the basis for the creation of models later.

The interviews will be done with pen and paper, and after the interviews all information is rewritten and made digital for easier treatment. We could have used a video camera to record the user's actions, but since most of the work is done on the computer it could be hard to get good video.

#### *Work Modeling*

When the interviews are completed models of the work can be created. Each work task each user performs will be one model.

We have chosen to use a flow model in this study. The models will show the different GUIs the user uses, what manual calculations the user does, and what other persons the user interacts with.

We will take the notes from the interview and directly transform them into flow models. Components the users demonstrated during the interview will be added step by step to the model. When all information from the interview is visualized in the model, underlying databases and systems are added to make the model complete.

The models are going to be created with Microsoft Visio.

#### *Consolidation*

Consolidation models are the same type of flow models as used in the work modeling, but with only one model for each work task for all the users.

The work models for each work task will printed out on paper laid next to each other and compared. Different colored pencils will be used to mark out common patterns and individual variations.



A single new model will be created on the basis of the colors. The new model will have all common patterns and individual variations if they are not too big or contradicting each other. If the individual variations create problems, several models must be made for the same work task.

The models are going to be created with Microsoft Visio.

### *Work Redesign*

Work Redesign models are the same the same type of flow model used in Work Modeling and Consolidation, but they show new ways how the work could be done.

When creating Work Redesign models we are going to use one single web based system as GUI. This GUI will do the job for all the other GUIs used today. All GUI components and manual work is replaced with one new GUI component in the model. Interaction with other persons will not be changed.

Work Redesign is easiest constructed in Microsoft Visio. The components can be deleted and replaced on the fly on the digital model.

### *User environment*

A User Environment model describes interactions between the system parts. The model is showing how the user can move around the system and get from one part to another.

The User Environment models will not take regard for any specific GUI.

We will create one model for the whole system showing how the user can move around. This will mostly show how the user can access different info pages through the menu. It is possible for the user to access the admin system to make adjustments on the system.

The models are going to be created with Microsoft Visio.

### *Mockup and Test with Customers*

The users are again involved in the design of the system by testing a prototype. The prototype is not a developed computer system, but a model created by physical objects to give the users the same impression.

We will create a model of the system with post it notes and paper sheets. The post it notes will function as menu buttons and dialog boxes, and the paper sheet will be the page showing information.

The users will be sat to do the same tasks as specified in the contextual inquiry. We will stand beside the user while he is using the paper prototype and take notes during the whole test. The users are asked to comment everything they do, and ask question if there is something they don't understand.

If there are complaints to the system, work models could be changed and the User Environment and Mockup and Test with Customer steps will have to be done again. If the customers are satisfied the system is ready for development.

### 3.1.3 Data collection strategies for sub problem 1

Data is collected from the user in two different ways; talking with the user and observing what the user does.

The interview is going to be performed as a combination of questions and conversation with the user.

An example interview:

- The designer asks the user how he makes sure that an agreed SLA is within target.
- The user explains that he uses HPPC to get response times for each customer and OSMq to get a list of open incidents.
- The designer takes notes and asks how HPPC is used and if the user can show him how he uses HPPC.
- The user explains about HPPC and how information about each team's response time and each customer's response time is displayed. Then he runs a demonstration on how to get the response times.
- The designer takes notes and makes small sketches of how the user works. During the demonstration the designer stops the user several times to make sure he is able to write down everything.
- The designer asks about OSMq, how it is used, and how lists of open incidents are retrieved.
- The user explains about OSMq and gives a demonstration about the use.
- The designer has to stop this several times to make sure he has everything. But there was something he didn't understand during the demonstration and he asks if the user can show this again.
- The user shows and gives further explanations.

The interview is a strategy that will be used in the Contextual Inquiry and the Mockup and Test with Customers.

The models are developed by using the interviews or other earlier created models as background and can not be said to be collected.

### 3.1.4 The specific treatment of data for sub problem 1

The data collected from the Contextual Inquiry are:

- Rewritten and made digitally for easier use at a later time

The data collected from the Mockup and Test with Customers:

- Rewritten and made digitally
- Positive and negative comments are separated
  - o Negative comments could be used for a new round of work redesign

## 3.2 Sub problem 2 - Development

Develop a web based system to monitor Service Level Agreements within department 6211

### 3.2.1 Data needed for sub problem 2

Use Case diagram will be created on basis of the user requirement specification system description from SBS, and the same will the Data Flow diagrams

The Data Flow diagrams will be used as background for the Class diagrams together with the floor plan form Contextual Design.

The Class Diagram will be created on basis of the user requirement specification from SBS, and the Floor Plan from Contextual Design.

Entity Relation (ER) diagrams will be created based on the user requirement specification system description from SBS, Data Flow diagrams, and Class diagrams.

### 3.2.2 The research methodology for sub problem 2

The development of the system will be a three step process. Develop documentation, develop the system functionality, and develop GUI. The two last steps will be done more or less contemporary.

#### 1. Development documentation

The first step in the development process is creation of development documents. We will create different types of documents; Use Case diagrams, Data flow diagrams, Class diagrams, and ER diagrams.

##### *Use Case diagram:*

The Use Case diagram will be created with basis in the different modules SBS have specified in their description of this study.

The different modules are; Main, Response time, Agent availability, Incidents, Deviations, and Work load. All modules except Main will have a present and a history module.

There will be Use Case diagrams for the whole system which expands out to each of the modules and again to their sub modules if there are any and so on.

The Data Flow diagrams will be created according to the UML specification.

##### *Data Flow diagrams:*

Data Flow models are similar to those models used during Contextual Design, but focus on processes rather than user interaction.

The Data Flow diagrams will describe the user's interaction with the system and interactions within the system. The Data Flow diagrams will be created according to the system specification from SBS.

When the user asks for the average response time for in 2004, the system modules that are used and the data sent between them will be in the flow chart. All databases and everything used in the system will be a part of that Data Flow diagram.

The Data Flow diagrams will be created according to the UML specification.

### *Class diagrams:*

The foundation for the programming is the Class diagrams. The Class diagram illustrates the system architecture with classes and interfaces.

There will be several classes regarding treatment of information, one class is needed for handling GUI, and there will be at least one interface towards the databases.

The Class diagram will be created according to the UML specification.

### *ER diagrams:*

There will be a need for a local database storing the local information. This database will be designed using ER modeling. The ER model shows entities and relations between the entities.

The database will store history, information about which customers that are connected to what team, information about the websites appearance, and other information that can't be retrieved from other databases.

The database can also be used as the main storage for data accessed by the web based system. Data from the other databases can be collected and stored in the local for less system activity.

The ER diagram will be created with Modulator.

## **2. Development of the system functionality**

The system will be developed in Visual Studio 2005 and will be an ASP.net and C# application. The application will be developed for the web so everyone that wants to use it only needs a web browser.

The Class diagram will be the foundation for the development. Each class will have functionality from the other diagrams. Data Flow diagram will be used to create input and output and sequence diagrams will be used for relations between the classes.

The local database will be a MS SQL 2000 server with stored procedures for easier access.

## **3. Development of the GUI**

The GUI in this system will be a regular HTML GUI. A Cascading Style Sheet (CSS) will be used for specifying the graphical elements.

We will use Visual Studio 2005 for most of the development of the GUI in cooperation with a regular text editor.

### **3.2.3 Data collection strategies for sub problem 2**

Data will be retrieved through Contextual Design (described under sub problem 1) and a description of the study handed to us from SBS.

### 3.3 Sub problem 3 - Evaluation

To evaluate the usability of the developed (or partly developed) system one can use either real end users or competent persons with knowledge on usability and user interface to perform the evaluation of the system.

There are four basic ways of evaluating user interfaces [15]; automatically (usability measures computed by running a user interface specification through some program), empirically (usability assessed by testing the interface with real users), formally (using exact models and formulas to calculate usability measures), and informally (based on rules of thumb and the general skill and experience of the evaluators).

We can divide the research methodologies, or evaluation methods, into two categories, inspection and user testing. There are many different inspection methods such as: Heuristic evaluation, cognitive walkthroughs, formal usability inspection, pluralistic walkthroughs, feature inspection, consistency inspection, and standard inspection.

In this sub chapter we will first look at what data is needed on sub problem 3. Then we will present different research methodologies which can obtain the needed data. There are a variety of usability engineering methods that are of interest in the study of usability such as: Heuristic evaluation, cognitive walkthrough, heuristic walkthrough, prototyping, task analysis, guidelines, inspection, and user tests. The heuristic evaluation, the cognitive walkthrough, and user testing will most likely be used.

Further on we will describe how we plan to collect needed data for some of the most relevant methods.

Finally the specific treatment of data which will be obtained by the different methods is presented.

In sub chapter 3.3.1 we will compare the different ways of evaluating an interface, and in sub chapter 3.3.2 we will choose the evaluation methods which will be used in this study.

#### Data needed for sub problem 3

This section presents an overview of the data/measurements that are needed in relation to Nielsen's definition on usability.

The heuristic evaluation and the cognitive walkthrough will output possible problems with the web based system.

Quantifiable usability measurements needed that can be obtained by the user testing:

- a. *The time users take to complete a specific task.*
- b. *The number of tasks (or the proportion of a larger task) of various kinds that can be completed within a given time limit.*
- c. *The ratio between successful interaction and errors.*
- d. *The time spent recovering from errors.*

- e. *The number of user errors.*
- f. *The number of commands or other features that are never used by the user.*
- g. *The number of system features the user can remember during a debriefing after the test.*
- h. *The frequency of use of the manuals and/or the help system, and the time spent using these system elements.*
- i. *How frequently the manual and/or help system solved the user's problem.*
- j. *The proportion of user statements during the tests that are positive versus critical toward the system.*
- k. *The number of times the user expresses clear frustration (or clear joy).*

### **The research methodology for sub problem 3**

This section presents the research methods that are relevant to use in the evaluation of the SWS.

*Heuristic evaluation* can be used to identify usability problems in the developed web based system. This evaluation will not include real end users (real end users in this project would include the Team Leaders, the Escalation Manager, and the Department Manager at SBS department 6211), but will be performed by competent persons with knowledge on user interface and usability [11]. Nielsen recommends using three to 5 evaluators since one does not gain that much additional information by using larger numbers. This evaluation method would be suited during the development of the system as the competent persons would inspect the user interface rather than performing tasks on the system. Hence, the method can be used as part of an iterative development process.

*Cognitive walkthrough* is used as a usability inspection method focusing on evaluating whether the web based system is easy to use [12]. Cognitive walkthrough uses a detailed procedure to simulate a user's problem-solving process at each step through the dialogue, checking if the simulated user's goals and memory content can be assumed to lead to the next correct action. This method will be performed, like the heuristic evaluation, by competent persons with knowledge on usability. They will act as evaluators while we will have the role as observers.

*Formal usability inspections* [15] use a six-step procedure with strictly defined roles to combine heuristic evaluation and a simplified form of cognitive walkthroughs.

*Pluralistic walkthroughs* [15] are meetings where users, developers, and human factors people step through a scenario, discussing each dialogue element.

*Feature inspection* [15] lists sequence of features used to accomplish typical tasks, checks for long sequences, cumbersome steps, steps that would not be natural for users to try, and steps that require extensive knowledge/experience in order to assess a proposed feature set.

*Consistency inspection* [15] has designers representing multiple projects inspect an interface to see whether it does things in the same way as their own designs.

*Standards inspection* [15] has an expert on some interface standard inspect the interface for compliance.

*User testing* would involve real end users. This method would be carried out by observing real end users using the developed web based system to perform representative tasks [14]. This testing method would be carried out at SBS department 6211 where the web based system is intended to be used. The end users will be the Team Leaders, the Escalation Manager, and the Department Manager at SBS department 6211. We will act as observers during the user testing.

### **Data collection strategies for sub problem 3**

This section describes how we are planning to carry out the relevant (for this project) research methods presented in the previous section.

*Heuristic evaluation* can be conducted either as an individual evaluation or as a group evaluation [17]. Since an individual evaluation picks up more usability problems [17] we will prefer this method even though it takes more time to complete. Also, since the evaluators are situated at different places (Grimstad and Oslo) makes a group evaluation difficult to carry out as all evaluators need to be assembled. The individual heuristic evaluation is performed by having each individual evaluator inspect the interface alone. Only after all evaluations have been completed are the evaluators allowed to communicate and have their findings aggregated [11]

The evaluators will be given different task scenarios which they must carry out. During each evaluation the evaluator will go through the web based system several times and inspects the different system elements and compares them with the list of ten usability heuristics given by Nielsen, see appendix 4. If the evaluators ask questions on how to perform certain tasks using the web based system we, as observers, will give them guidance.

The results of the evaluation will be recorded either as written reports from each evaluator or we, as observers, will write down comments from each observer during an evaluation session.

The heuristic evaluation will be performed during the development of the web based system to insure an iterative development process. Each heuristic evaluation session for an individual evaluator typically lasts one or two hours.

The *user testing* will be carried out the following way: Each real end user will be brought to the lab where the developed web based system is intended to be used. The user will then be given a list of tasks he/she needs to perform on the system without help. The data collection technique using this method can be performed by using a stopwatch [8]. When using a stopwatch we, as observers, can measure the time or the users could report the time themselves in a diary. I.e. efficiency of use can be quantified as the average time it takes users to perform a certain number of specified tasks. After the user tests the test users will be given a short questionnaire as part of a debriefing session.

### The specific treatment of data for sub problem 3

In this section we will describe how we will treat the obtained data from each of the relevant methods.

The output from using the *heuristic evaluation* method will be a list of usability problems in the web based system with references to those usability principles that were violated by the design in each case in the opinion of the evaluator.

Each problem identified will then be ranked according to Nielsen's five point scale, see table 3.3.1. This scale will give us the opportunity to dismiss whatever an evaluator or user means could be a problem. The five point scale will be used both in the heuristic evaluation, the cognitive walkthrough, and in the user tests.

0 = I don't agree that this is a usability problem at all

1 = Cosmetic problem only: need not be fixed unless extra time is available on project

2 = Minor usability problem: fixing this should be given low priority

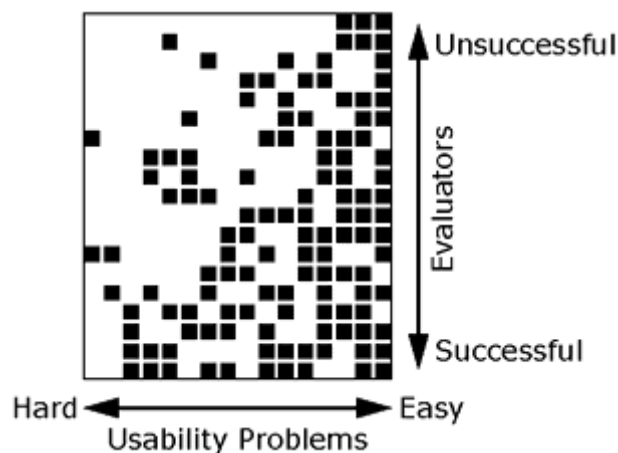
3 = Major usability problem: important to fix, so should be given high priority

4 = Usability catastrophe: imperative to fix this before product can be released

*Table 3.3.1: J. Nielsen's 5 point rating scale to rate the severity of usability problems*

The output data could then be organized in a table categorizing the problems. The problems could then easily be identified and corrected if necessary.

The output data collected during the different methods can also be organized as shown in figure 3.3.1 indicating overlapping of the different problems:



*Figure 3.3.1: The usability problems that are easiest to find are to the left and the ones most difficult to find are to the right. In regards to the user tests, the evaluators will be replaced by the users.*

#### 3.3.1 Comparing the research methodologies

J. Nielsen says: "Under the current state of the art, automatic methods do not work and formal methods are very difficult to apply and do not scale up well to handle larger user interfaces" [15].

Further on Nielsen comments: "Empirical methods are the main way of evaluating user interfaces, with user testing probably being the most commonly used method. Often, real users can be difficult or expensive to recruit in sufficient numbers to test all aspects of all



the versions of an evolving design, leading to the use of inspection as a way to “save users” [15].

J. Nielsen also states: “Several studies have shown that usability inspection methods are able to find many usability problems that are overlooked by user testing but that user testing also finds some problems that are overlooked by inspection, meaning that the best results can often be achieved by combining several methods” [15].

### 3.3.2 Methods of choice

When using a monitoring system as the SWS the user communicates in a small degree with the system by means of user input. The communication between the user and the system will mostly happen through a visual picture of the system, meaning that the system displays wanted information to the user when asked. Thus, the organization of data and information is a contributing factor to how fast the users of the system find the data they are searching for and how quickly the users discover critical data.

We can divide the data and information in the user interface into two categories according to the severity a usability problem might have. Minor usability problems would involve the presentation of data that are not critical and would not stop users from completing their tasks. For example, the use of inconsistent typography in different parts of the system would be considered minor usability problems. Although we use the term “minor problems” it does not mean that these problems are not important as the presentation of data will affect the usability of the system. Easy noticeable problems could be an equal term to use as minor problems. Major usability problems would involve data that are of importance to the users, the content on a page. Every usability problem that would slow down the user or prevent the user from doing a task will be considered a major usability problem.

We will in this project make use of two evaluation methods that would involve both outside inspectors and real end users. As J. Nielsen states: “Users are not designers and designers are not users”. Meaning that usability problems overlooked in an inspection method can be found in an empirical method and vice versa.

As an inspection method the heuristic evaluation will be applied as it requires few resources in terms of time and expertise. Heuristic evaluation is also “a good method for finding both major and minor problems in a user interface” [19]. Case studies by J. Nielsen showed that usability problems found by heuristic evaluation will tend to be dominated by minor problems. Since the evaluators used in the heuristic evaluation in this project are not real end users, but competent persons on usability and design, the heuristic evaluation will mostly serve as an evaluation on the graphical user interface (how data and information is presented) rather than an evaluation on critical data (the content).

The empirical method applied will be user testing. The SWS is domain-dependent system and the evaluators used in the heuristic evaluation have little domain expertise. Therefore, the evaluators are likely to overlook many usability problems, particularly the major problems. The user testing involves real end users and by observing the users performing task scenarios on the system we can see how they perceive data presented to them, how they solve the tasks, how quickly they solve the tasks, and make note of positive and negative comments.

Iterative design will be used when applying the heuristic evaluation and the user tests. By correcting the usability problems found in the first evaluation we will avoid finding the same usability problems in the next evaluation. Also, the two methods have been shown to find fairly distinct sets of usability problems, and supplement each other rather than lead to repetitive findings [19]. The heuristic evaluation will be applied first. As J. Nielsen states: "Typically, one would first perform a heuristic evaluation to clean up the interface and remove as many "obvious" usability problems as possible. After a redesign of the interface, it would be subjected to user testing both to check the outcome of the iterative design step and to find remaining usability problems that were not picked up by the heuristic evaluation" [19].

## 4 Accomplishment

In this chapter we will describe the accomplishment of the methods that were selected in chapter 2. The accomplishment of the evaluation methods selected in chapter 3 will be described in sub chapter 4.3

### 4.1 Design

We will in this part look further into what we did during the steps six steps of Contextual Design [2].

#### 4.1.1 Contextual Inquiry

We were two system designers who interviewed four Team Leaders, one Escalation Manager, and one Escalation Manager acting as Department Manager. They are all skilled computer users and have much knowledge about the work in Department 6211.

Before the interviews we prepared a set of common work tasks for Department 6211. Each work task was formulated into a question where the answer was the user's explanation on how the task was performed. The same set of question was used for everyone interviewed.

The process of Contextual Design states that each user should be interviewed by one designer at the time, and that each user should be interviewed by all system designers. We did not have the possibility to do this at SBS, so each user was interviewed by both system designers at the time.

The interviews was carried out at the users own work place and the user had access to everything used during regular work.

When a user was asked a question, the answer was given orally and at the same time demonstrated. As designers we had to take notes of what the user said and did. All the notes were summarized into a text document for each user.

All questions and answers from the users can be found in appendix C.

#### 4.1.2 Work Modeling

The work models were created in two steps.

We started out by using Information from the interviews, such as GUIs used, interaction with other persons, and manual calculations, to create the first flow models. The creation of models was an iterative process where we asked the users frequently if the models we created were right, and corrected what they pointed out as wrong.

The first models were defective and we had to use the escalation manager to make them complete. What made the models incomplete was information the regular user did not mention during the interviews, but we felt it necessary to give a full system overview. The Escalation Manager provided us with underlying system information such as databases, software, and GUI information.

All work models were created with Microsoft Visio 2003 and can be found in appendix D.

### 4.1.3 Consolidation

We started out by printing all work models for one work task (question) and laid them next to each other on the table. We took pencils in different colors and marked all different patterns on one work model with different colors before we started on the next. If the same pattern appeared on two or more models it was colored in the same color. In the end all different patterns on a set of work models was colored in different colors.

Each of the different colored patterns was copied out and placed on a new model. We tried to avoid using the same element more than one time in each model. In some cases two instances of the same element had to appear to make it understandable.

All consolidation models were created with Microsoft Visio 2003 and can be found in appendix E.

### 4.1.4 Work Redesign

The user requirement specification was used as background when we found out what we needed in the new system.

We came up with three main points for creation of the work models:

- All the different GUIs should be replaced with one single GUI, the Snapshot website (SWS).
- There should be a local database which retrieves store necessary information from other databases to make it easier to access and for historical reasons.
- The Snapshot website (SWS) should provide all information the user need, so that all manual operations and calculations are unnecessary.

With the consolidation models as background new models were created where there was only one GUI between the user and the system. The new GUI got an own database which a windows service was responsible for updating. In front of the "old" databases we added a Windows service which was responsible for updating the SWS's local database with data from the "old" databases.

All Work Redesign models were created with Microsoft Visio 2003 and can be found in appendix F.

### 4.1.5 User Environment

The new system will have three major parts; the user system, the Management system, and the Admin system, which we got from the user requirement specification. Each of the system parts has its own modules which also were written out of the user requirement specification.

We chose a kind of square in squares model to display the system parts and modules. Each module was created as a big square, and the modules belonging to that part were created as smaller squares inside the big system part square.

From the user requirement we got the modules in each system part, but we did not get any clear information on how to connect them together.

Our solution to this was to create a menu module in each system part. The menu was not in the user requirement, but is an important part of a webpage.

The menu module has also given the function for controlling access to the different system parts and modules.

When we added modules into a system part they were given a little description and arrows were used to show communication between them.

After adding all modules and all internal links in the different system parts external we created external links between the menu modules and the login-module. The menu modules could link to lower menu but not to higher modules. And login linked to higher modules. (1.1 is a lower system part than 1.2 and so on).

The User Environment Floor Plan was created with Microsoft Visio 2003 and can be found in appendix G.

#### 4.1.6 Mockup and Test with Customers

Before any paper prototype was created we started out by the website. We created a structure with the menu as page tabs and information in "the page" below.

To make the paper prototype easy to create and readable for the users we used A3 paper. On a paper sheet, which became the webpage background, we drew a static header and the regular close-, maximize-, and minimize "buttons" from windows. We did also draw borders for the menu and the page content.

The idea of the A3 page was to make it look like a web page window without the menu and without any information (content).

We created A4 paper sheets for each different sub pages on the website.

The user would get a feeling that this is a dynamic website where the content and the menu is changing while the background and header is static.

Some of the A4 pages also needed dynamic features and we used post it notes to give that effect.

The menu was created with small post it notes in another color than the background. At the start there were only two buttons showing, the main and the login button. After a user logged other buttons appeared. The buttons that appeared after a login was determined by the user's privileges.

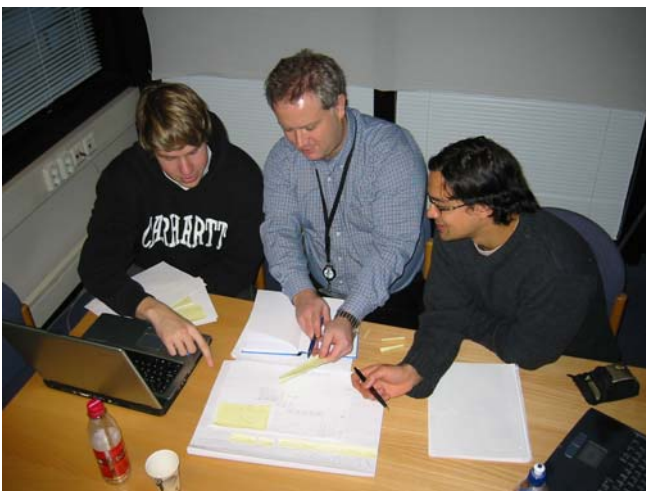


Figure 4.1.1: The paper prototype in use

After creating the paper prototype, we used it for user testing. Each of the users we interviewed during contextual inquiry was asked to “us” the paper prototype. We acted as “the system” and responded by changing the A4 pages or changing the menu when a user pushed a menu button.

The users gave us replies on system functionality and design on the website. We did not get any replies that acquired us to go back to the Work Modeling step and start over.

## **4.2 Development**

We will in this chapter look further into what we did during the development of the system. As a background for the development we used some elements from the Rational Unified Process (RUP) supplemented with experiences from previous projects we have been involved in.

For more detailed information, see appendix J

### **4.2.1 Documents**

We decided to create was use case diagram, data flow diagram, class diagram, and entity relationship diagram based on the RUP.

All documents were created, or started on, in the beginning, and changed during the development process.

#### **4.2.1.1 Use Case diagram**

The use cases were developed iteratively during the project. At first they were constructed based on the findings from the Contextual Design. As some requirements were changed during the project a change to the use cases were changed as well. The use case models were designed based on our experiences and knowledge from previous education on UML. To view the use case diagrams see appendix I.1, I.2, and I.3.

#### **4.2.1.2 Data flow diagram**

We used the Floor Plan and the Work Redesign models developed during the Contextual Design as background to design the data flow diagrams. The data flow diagrams were designed based on our experiences and knowledge from previous education on UML. To view the data flow diagrams see appendix I.4 and I.5.

#### **4.2.1.3 Class diagram**

We were both new at ASP.net development, and started with finding examples for ASP.net class diagrams. It was not easy to find any good examples on ASP.net class diagrams. All of those we could were only showing code classes that where not a part of the ASP.net pages.

Our application is a type of application where most of the logic is in the aspx.cs files in the web section.

We found out that a class diagram with all code files (classes) from the web section, would give us much more meaning.

We created our own type of class diagram containing all the code files, and the connections between them, and we created a class diagram for the ASP.net files showing the relationship between them and the master page. See appendix I.6 and appendix I.7.

For more detailed information, see appendix J.1

#### **4.2.1.4 Entity relationship diagram**

The entity relationship (ER) diagram was probably the most important diagram we created before we started the development.

We started off by creating two different diagrams. One diagram holding all information gathered from HPPC and OSMq, which became snapshot database diagram, and one diagram holding user information, which was a small diagram with only three tables. The two diagrams was put together in the new ER diagram.

The snapshot database diagram was created in close cooperation with SBS. If we made errors in the ER diagram it could have big consequences later in the development process. A change in the ER diagram, which means a change in the database, could result in big changes to the program code.

The ER diagram could be found in appendix I.8.

### **4.2.2 Software development**

The software development was a big part of this thesis which consumed most time. We started by developing a prototype early in the thesis and kept on developing and bug fixing until the thesis was about to be delivered.

#### **4.2.2.1 Prototype**

We created a prototype of the web application in static HTML right after the Contextual Design. This prototype did not have any functionality but helped us getting an impression of what SBS really wanted.

SBS on the other hand used the prototype to create a new user requirement specification which was more detailed and explaining.

#### **4.2.2.2 Web application**

The web application was the biggest developing task, and we did not have any experience with C# or ASP.net before we started on this thesis. We created several small test applications with an objective to test and learn how different features worked before we used it in the web application.

We started to develop the main page, which was the only page in the user system. It was developed as a single website only containing one information table. When the main page was finished we started the first page in the management system, and so on, until all pages in the management system was finished.

We created a masterpage with GUI and a login page with a username and password prompt.

All authorized users gets a user name and password which from the page administrator.

We had discussed this matter around passwords with them during the user requirement specification. But after talking with the SBS database expert we were told that they had a rule against saving password and username other places than in the Active Directory. We had to redo and change our login system.

There were three admin pages, and were created simultaneously, since they were using the same type of logic. We did not use the same method as before, with creation in a single page, instead they were created directly into the masterpage system.

For more detailed information, see appendix J.2

#### **4.2.2.3 Windows application**

A Windows Service was supposed to be used, in the original system design, for collecting data from the HPPC real time and the OSMq copy databases and store this at the snapshot database.

We stated to develop the Windows Service against a backup copy of the OSMq database and the HPPC database.

The Windows Service worked well against the OSMq copy database, but we had problems with the HPPC database. We could be held responsible for any crashes or other incidents that could occur. If the HPPC system was offline for a short period of time, it would cost SBS a lot of money. The Data was retrieved from the HPPC database with a DTS, see chapter 4.2.2.4, or appendix J.5.

When we were about to deploy the Window Service out the server, we were told that Windows Services weren't very popular at SBS. They wanted us to create a GUI less Windows application which could be started as a scheduled program instead. Since we had the most of the data retrieval code ready, this wasn't any big job. The porting was finished in less than one day.

For more detailed information, see appendix J.4.

#### **4.2.2.4 Database, stored procedures and DTS**

The first version of the database was created soon after the Snapshot ER diagram was proclaimed OK by the SBS database expert.

The database was written in SQL as a creation script which could be executed every time we wanted a fresh instance of the database.

To access data in the database SBS recommended use of stored procedures [20]. This would make it easy to reuse code, and it would let them see what kind of data we used in the database.

To retrieve data from the HPPC real time database, as we mentioned earlier, created some problems. But we were lucky and the SBS database expert helped us. He had a DTS [21] job going for retrieving data from the real time database and to the copy database once a day. He changed this DTS so that it would run every 15<sup>th</sup> minute and included the data we needed to the Snapshot database.

For more detailed information, see appendix J.5.



## 4.3 Evaluation

The evaluation of the Snapshot Website was conducted in three rounds. First we performed a heuristic evaluation. After modifying the SWS pursuant to the usability problems found during the heuristic evaluation, a user test was performed at Siemens. Subsequent to the first user test new modifications were carried out. The last corrections on the SWS were performed shortly after user test 2.

We will in this chapter describe how the heuristic evaluation, user test 1, and user test 2 were carried out.

### 4.3.1 Heuristic evaluation

#### Preparation

For the heuristic evaluation we prepared three documents. The first document (appendix L) consisted of a system overview, describing the objectives and target audiences [18]. The system overview also consisted of a set of goals of the system, according to the heuristic evaluation guide by N. Danino [16], in order to allow the evaluators to develop their own tasks as they would explore the SWS.

To assist the evaluators of finding usability problems we provided a heuristic evaluation checklist where the ten heuristics by J. Nielsen was stated, see appendix L. We also prepared a separate sheet where the evaluators could write down additional usability problems and comments. The SWS had not been implemented into the Siemens system at this stage, and the admin system had not been implemented into the SWS. Thus, we needed to prepare test data on a local database on our laptops. We also needed to run the SWS prototype on a laptop since the heuristic evaluation needed to be performed at each of the evaluators' workplace.

#### The evaluators

For the heuristic evaluation we wanted evaluators with competence and knowledge on usability and user interface. We also needed to find between three to five evaluators to follow the recommendations of J. Nielsen [11]. Two of the evaluators were from HIA, one professor and one 1<sup>st</sup> amanuenses, and the third evaluator, a graphical designer, was from Siemens. None of the evaluators had any experience with using the SWS prototype before the evaluation, and only the evaluator at Siemens had knowledge on the work tasks at Siemens SBS department 6211 and Siemens' specifications. The evaluators from HIA have good knowledge on usability and the evaluator from Siemens have experience and knowledge on user interface design.

#### The heuristic evaluation accomplishment

The heuristic evaluation was conducted having each individual evaluator inspect the SWS prototype alone. Each evaluator was given 10 minutes to get to know the SWS prototype by him/her self. We acted as observers during the heuristic evaluations. During the heuristic evaluation the evaluators had the opportunity to use the system overview sheet, appendix I.9, to explore the SWS prototype. The checklist was used by the evaluators to assist them of finding usability problems. Additional usability problems found and comments the evaluators had, were written down on a separate sheet of paper. We, as observers answered every question and provided guidance whenever the evaluators needed assistance. We also took notes of any comments and input the evaluators had.

The three heuristic evaluation sessions was conducted slightly different from one another since we wanted the evaluators to decide on their own how they wanted to proceed with

evaluating the SWS prototype according to J. Nielsen [11]. The two evaluators at HIA used the heuristic checklist to find usability problems. They did also comment the SWS prototype as they were exploring it, the comments were written down by the observer(s). The evaluator at Siemens chose to conduct the evaluation slightly different. The evaluator used the goals of the system as a guide to explore the SWS prototype. We, as observers, wrote down the usability problems the evaluator found and the comments she made.

The accomplishment of the heuristic evaluation will be discussed in chapter 6.



Figure 4.1.2: Heuristic evaluation at Siemens

### 4.3.2 User testing

#### Preparation

For the user tests we prepared a set of task scenarios, see appendix M.1 and appendix N.1, the test users needed to perform. Task scenario 8 was slightly modified before user test 2. A stopwatch and a digital video camera were provided for the user tests. The SWS was modified according to the user problems found in the heuristic evaluation before user test 1, and the SWS was also modified after user test 1 and after user test 2. At this point the SWS had not been implemented into the Siemens system. Hence, the same test database used in the heuristic evaluation was provided for user test 1 and user test 2.

#### The test users

The test users were real end users at Siemens SBS department 6211. In user test 1 three test persons participated, all team leaders at SBS. None of the test users had any experience with the SWS beforehand, but they all have knowledge and plenty experience with the work tasks at Siemens SBS department 6211. The test users are also considered to be expert users in the field of information technology. In user test 2 a fourth test user participated. The test user is the escalation manager at the department and has the same background as the other test users. Test user 4 is also our supervisor at Siemens.

### The user tests accomplishment

The user tests were carried out at Siemens SBS department 6211 where the SWS is to be used. Each test user conducted the user test alone. Before the user test, ten minutes were given to the test users to get to know the SWS. The test users were then given a set of task scenarios which they needed to solve. They were asked to read through and understand one question at a time, when they were ready we (the observers) started the stopwatch. The clock was not stopped before the test user had written down the correct answer. The test user could then read and solve the next task. A digital video recorder was placed behind the test users and the laptop, recording only the laptop screen and the comments made during the user test.

We (as observers) sat beside the test users and made note of the time spent on each task, the number of system errors discovered on each task, how many mistakes the test users made on each task (and were the errors fatal or recoverable with additional hints and information), and the number of times the test users expresses clear frustration or clear joy on each task. We, the observers, were reserved giving help or hints to solve the tasks in correspondence to what J. Nielsen states: "For traditional user testing, one normally wants to discover the mistakes the users make when using the interface; the experimenter are therefore reluctant to provide more help than absolutely necessary" [11]. After the test users solved all of the tasks, we calculated the average time it took the users to perform the ten task scenarios.



Figure 4.2.1: User testing at Siemens SBS department 6211.

## 5 Results

In this chapter the results of the accomplishment from the selected methods is presented. Sub chapter 5.1 presents the results from the Contextual Design, the results from the development process is presented in sub chapter 5.2. In sub chapter 5.3 the measurements and results from the evaluation is presented.

### 5.1 Design

We will in this subchapter look further into the results from Contextual Design. We will look the result from each of the six steps in form of models, prototypes, and written interviews. We will also look at what we as designers learnt from each step.

All figures are found in appendix C – H.

#### 5.1.1 Contextual Inquiry

We did get a set of answers for each of the question we asked the users during the Contextual Inquiry.

We have summarized the answers for each question in a table and counted the number of agents giving a each answer. Notice that one user could give more than one answer in each table.

All answers in full text are found in appendix C.

<b>Question 1: How is it possible to discover that you are not within target of an agreed SLA?</b>	
<b>Answer:</b>	<b>Number of users:</b>
HPPC is used to get the average response time for each customer	6
OSMq is used manually to search after old open incidents	4
Manually check against the SLA's to see if they are within target	2
Use the Excel macro to find response time history	2
Customers complains if their problem has not been solved	2
The activity and the stress of the agents is used as an indication that the SLA could be out of target.	1
Collect the number of new incidents and the number of closed incidents	1
Almost all SLA's are similar	1
There are no clear SLA's defined at the moment	1

*Table 5.1.1: Contextual Inquiry question 1, with answers and the number of agreeing users.*

<b>Question 2: How do you monitor the response time?</b>	
<b>Answer:</b>	<b>Number of users:</b>
HPPC is used to get the response time for each customer	6
The Escalation Manager tells the Team Leader if the response time is to high	1

Excel is used to see response times per agent, response times per agent, a history of the response time, and total delivery	1
-----------------------------------------------------------------------------------------------------------------------------	---

Table 5.1.2: Contextual Inquiry question 2, with answers and the number of agreeing users.

<b>Question 3: How do you monitor the agents?</b>	
Answer:	Number of users:
HPPC shows the status for each agent	5
HPPC is used to get response times	2
HPPC Reporter generates a history of the agents states	2
OSMQ is used to get reports about open incidents	1
Everyone is telling the Team Leader what they do	1
The service desk requires people to leave their seat and go around. This makes it hard to monitor everyone	1

Table 5.1.3: Contextual Inquiry question 3, with answers and the number of agreeing users.

<b>Question 4: How do you monitor open incidents?</b>	
Answer:	Number of users:
OSMq is used to search for open incidents	6
A web client is used to search for open incidents	3
The Escalation Manager tells Team Leaders about open incidents for his/hers team	2
15-20 open incidents in queue are OK.	1

Table 5.1.4: Contextual Inquiry question 4, with answers and the number of agreeing users.

<b>Question 5: How do you find deviations between the number of OSMq incidents and the number of phone calls?</b>	
Answer:	Number of users:
Manually check the number of incoming calls from HPPC and the number of registered incidents from OSMq and calculate the deviation	4
Get all answered calls with an Excel macro, all wrap-up reason reports from the HPPC-reporter, get all registered incidents from OSMq, and calculate manually	2
Have the possibility to use an Excel macro to calculate	2
There could be registered wrap-up reason reports if a customer calls a second time on the same incident, but this is seldom done	1
A lot of incidents are forwarded to the team from other teams, and E-mails are coming but not taken regard for in the calculation	1

Table 5.1.5: Contextual Inquiry question 5, with answers and the number of agreeing users.

<b>Question 6: How do you measure the workload and plan how many that are going to work?</b>	
Answer:	Number of users:
OSMQ is used to find the number of registered incidents	3

Routines and experience. Example in the end of every month when the stores are counting goods	3
The work plan is used to see how many that were working compared wit the number of answered phones	2
Cutsomers are telling about speciall circumstanes or events	2
HPPC reporter is used to see the response time at different times at the day to se how many that are working. Not possible longer than 40 days back.	1
SQL Enterprise Manager is used to get the number of incoming calls in sequences	1
Excel is used to get the number of incoming calls	1
No tool is used	1
Talks with agents that worked the same time last year if there are special circumstanes or events comming up	1

Table 5.1.6: Contextual Inquiry question 6, with answers and the number of agreeing users.

The number of answers to each of the questions is between three and nine, and there are several answers to all questions which only one user has replied. The reason for the number of answers could be many.

Where there is only one user giving that answer there could be the result of access rights to different systems and work tasks that are special for the Escalation Manager and the Department Manager.

There are questions where all users are replying the same answer as well, but there isn't any explanation to why all users replied the same here. There are no written work routines or special work task they have to perform.

We got to meet and talk to the people working in the department; to see what they did, how they did it, and an we got an explanation on why they did it. Our knowledge around the business did also increase a lot during Contextual Inquiry.

### 5.1.2 Work Modeling

The Work Modeling gave us a set of flow models describing the answers from the Contextual Inquiry. The models consisted of more information than the answers dough. Deeper system parts, such as databases, GUIs, and communication between those parts were also included.

The flow models could not be validated against any standard, and there was no need to either, since they were only used to represent an old system and making it viewable for people.

The chance for having a correct model was pretty good since they were created with help from SBS. They helped us when we asked, and they checked if the models were correct.

Our knowledge of the business was increased more during the Work Modeling. We got to see which parts the system consisted of, how they cooperated, and we started to understand why they had to do the manual job as they did.

### **5.1.3 Consolidation**

The result of the Consolidation was a set of the same type of flow models as we had from the Work Modeling.

The Consolidation Models were larger than the Work Models and they represented all the different the answers to one question, from Contextual Inquiry, with only with one model. This was right for the five first questions. But for the last one there were two models, one model representing the work of the Team Leaders and the other representing the work of the Escalation Manager and the Department Manager. The two last questions required two models because of the access to use the right tools.

We increased our business understanding because we went through the Work Models several times in this step.

### **5.1.4 Work Redesign**

From Work Redesign we got a set of new models which wasn't related with the way of working before this thesis.

There was only one model per question, as in Consolidation, but the new models were a lot smaller. The Work Redesign Models separated them selves from the Consolidation Models by the number of paths. The Consolidation Models had one path for each of the different answers to a question, from Contextual Inquiry, while the Work Redesign models have only one path.

The work redesign models do not separate the users by their job. All users are allowed to perform the same actions.

We learnt even more about the business organization and their technology during this step. We had to find out how we could create new models based on their existing data storages and find new ways to collect data from them.

### **5.1.5 User Environment**

The Floor Plan was a new model for us. We had no examples, so it became a model we designed ourselves. The model did give the first complete system overview with system parts, modules and communication. The Floor Plan consisted of modules placed inside system parts. Communication between each module was done through the menu modules that could be found in all system parts.

We started for the first time to see how big and comprehensive the system really was. We had to try to be far-seeing and create the models we thought the real system would have.

### **5.1.6 Mockup and Test with Customers**

The Mockup and Test with Customers was done with a paper website prototype and we got a lot of feedback. However, most of feedback was regarding the user interface.

Here is a summarizing the feedback:

- One Escalation Manager wanted response times to be viewed per team and with the possibility to click on that team to see the customers related to that team and their response times separately.

- One Team Leader wanted just to select a date for when the open incidents should be calculated and not a week number. It was much easier to just have a control over dates and not remember week number.
- One Escalation Manager wanted the possibility to have flexible periods for which the work load was calculated over.
- One Escalation Manager wanted the possibility to select which modules each user of the management system was allowed to see (This was in our model but the Escalation Manager did not understand it).
- One Escalation Manager wanted a possibility for team configuration where team could be created or deleted and agents could be added or removed from a team.

During the creation of the paper prototype we found out a lot about the future layout of the system. We learnt what the users wanted the layout, what they didn't want, and how the features should work.

## **5.2 Development**

We will in this subchapter look further at the result we got from the development process.

### **5.2.1 Documents**

The different documents we created were created for different purposes. We will now look further into the result we got from each document.

#### **5.2.1.1 Use Case Diagram**

We will not describe the use case diagrams further in this chapter. Appendix I.1, I.2, and I.3 shows the use case diagrams.

#### **5.2.1.2 Data flow Diagram**

We will not describe the data flow diagrams further in this chapter. Appendix I.4, and I.5 shows the data flow diagrams.

#### **5.2.1.3 Class Diagram**

The class diagram had very little influence on our development, but the process of creating it had. We learnt and found out how an ASP.net web application was structured and how we were going to create our solution when we created the class diagram. Mostly because of the background information we read about ASP.net and class diagrams.

We will in the rest of this subchapter talk about the class diagram result as two diagrams; one showing C# relations (Figure 5.2.1) and the other showing ASP.net relations ( Figure 5.2.2).

Each of the classes in the C# relationship diagram are specified in smaller diagrams showing attributes and methods, see appendix I.7. There was a misunderstanding from one person at SBS on why we took each class into a smaller diagram, but a short explanation, where we said it was because of the size and overview, he understood. Everyone else that we showed the diagrams to, understood point of them almost right away (we did only show them to people with some background in development).



The class diagram showing ASP.net relations was not that easy to explain to someone that hadn't worked on the system. Several of the people we showed the diagram to, at SBS, could not see the point of it. We got reactions that a website map could give the same effect.

But after an explanation around master pages and their relationship with ASP.net pages, they agreed the diagram gave meaning, especially in the context of further development.

We did also get a comment on the lines (communication lines) between the classes on the diagrams, specially the diagram showing C# relations. We could have made the lines different in colors or in size so that they would be easier to separate them, but this would not be according to what we have learnt about UML.

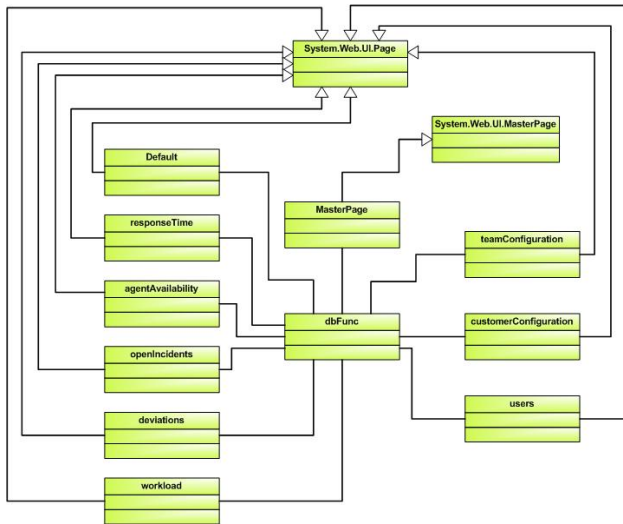


Figure 5.2.1: The class diagram for the .cs classes. A larger version could be found in appendix I.7.

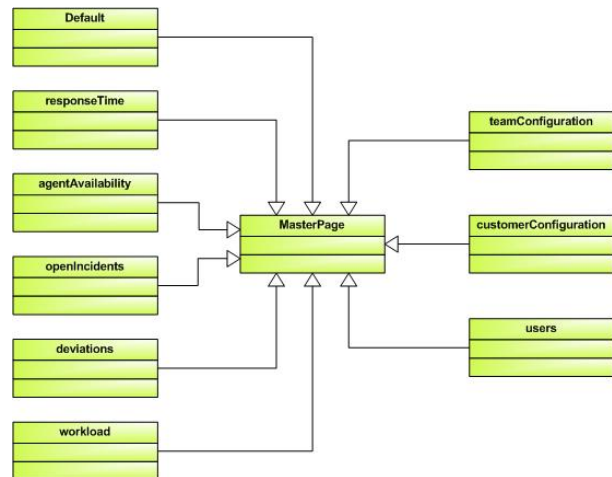


Figure 5.2.2: The class diagram for the .aspx files. A larger version could be found in appendix I.6.

#### 5.2.1.4 Entity relationship diagram

The ER diagram was the only background we had for writing the SQL create queries for the Snapshot database. We did also use the ER diagram as an important piece in the process where we explained the database to the SBS database expert. That would have been very hard to do without the ER diagram.

SBS are probably going to add new modules or system parts to the system at a later point, and they would need to change the database. It is much easier to reconstruct, add tables, or just add fields if they have a graphical view of the database.

### 5.2.2 Software development

We produced three software products during this thesis; the Snapshot web application, the Windows application, and the database with stored procedures. The people using this system will only see the web application, as we can see in figure 5.2.4. The Windows application and the database will be hidden elements only performing system tasks.

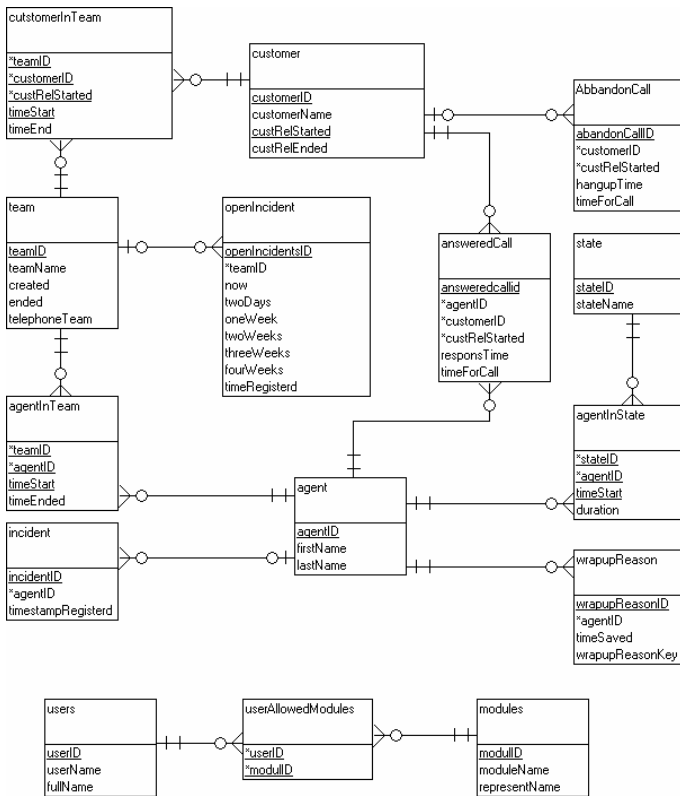


Figure 5.2.3: Entity Relation Diagram, a larger version could be found in appendix I.8

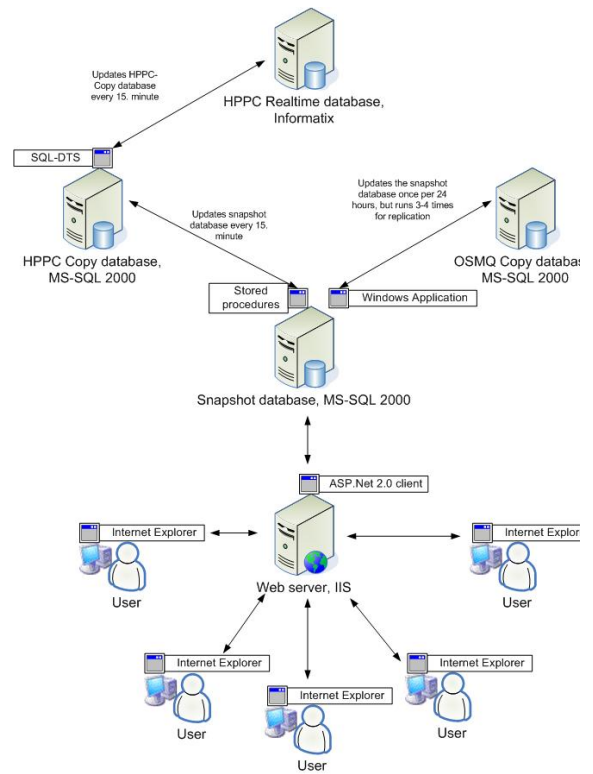


Figure 5.2.4: System overview. A full size picture are in appendix I.9.

### 5.2.2.1 Web application

The web application is the main software product of this thesis. It is a website) on the SBS intranet, showing department monitoring real time and history data, and will be used by the Department Manager, the Escalation Manager, Team Leaders, agents, the management, and everyone else interested in the performance of department 6211. It can be accessed through any ordinary web browser, but it is optimized for Internet Explorer 6 which is standard at SBS.

The web application works as intended and all main features from the user requirement specification is implemented. There are some small features missing, such as the possibility to switch from graphs to table mode at the response time table, but they will probably be added at a later point by SBS.

Screenshots from the web application can be found in appendix O.

“The web application makes our work much easier”, cited by the previous escalation manager.

### 5.2.2.2 Windows Application

The windows application has no GUI and can not be seen running by a regular user. There will be no user interactions with this application since it only updates the Snapshot database with incident data from OSMq.

The windows application has several good advantages; it is easier to administrate than a Windows Service, it is also easier to move than a Windows service, and it can be started manually if the scheduling doesn't work.

### 5.2.2.3 Database

The database is running on a SBS SQL 2000 database server and is working as intended. The stored procedures are also running without errors.

The regular user will not feel any interaction with the database. It will be invisible when it is working, and is accessed through the web application.

The database will provide the web application with the data and it will get data from the windows application and the DTS by stored procedures.

## 5.3 Evaluation

In this sub-chapter we will first present the results from the heuristic evaluation. Further on, the results from user test 1 and user test 2 are presented. The problems found in the heuristic evaluation and the user tests have been categorized into six categories; Menu/navigation, system response, presentation of data/information, consistency, other problems, and input. The usability problems have then been ranked according to the rating scale provided by J. Nielsen (Table 3.3.1) to rate the severity of the usability problems.

### 5.3.1 Heuristic evaluation

Three evaluators conducted the heuristic evaluation. Evaluator 1 and evaluator 2 were from HIA, and evaluator 3 was from Siemens. The background and competence of the evaluators can be found in sub-chapter 4.3.1.

Table 5.3.1 shows the number of problems in each ranking category each of the evaluators identified during the heuristic evaluation. The total number of usability problems identified in each ranking category and the total number of problems identified by the evaluators are also shown.

Rank	Number of problems identified			
	Evaluator			Total
	1	2	3	
0	2(1)	1	1(1)	3(2)
1	2	1	2	2(0)
2	3(2)	1	4(3)	6(5)
3	2(1)		4(3)	5(4)
4	7(3)	3	6(2)	10(5)
<b>Total</b>	16(7)	6(0)	17(9)	26(16)

Table 5.3.1: An overview of how many usability problems found by the different evaluators in each ranking category. The numbers in parentheses indicate how many of the usability problems that are uniquely identified.

Figure 5.3.1 gives an overview of the usability problems found in each ranking category. The overview illustrates which usability problems that are uniquely identified and the overlap of other usability problems found.

Rank: 0

U ← Evaluator → S	3			
	2			
	1			
		3.2	5.3	3.3
Hard ← Problem → Easy				

Rank: 1

U ← Evaluator → S	2		
	3		
	1		
		5.2	2.5
Hard ← Problem → Easy			

Rank: 2

U ← Evaluator → S	2						
	1						
	3						
		2.1	2.3	3.7	3.9	3.10	5.1
Hard ← Problem → Easy							

Rank: 3

U ← Evaluator → S	2					
	1					
	3					
		3.4	3.6	3.8	3.11	1.1
Hard ← Problem → Easy						

Rank: 4

U ← Evaluator → S	2										
	3										
	1										
		2.4	2.6	2.7	3.5	3.12	1.2	2.2	4.1	4.2	3.1
Hard ← Problem → Easy											

Figure 5.3.1: The figure shows which evaluators found which usability problems in each ranking category. Each square shows whether the evaluator (represented by the row) found the usability problems (represented by the column): The square is orange if this is the case and white if the evaluator did not find the problem. The most successful (S) evaluators are placed at the bottom and the unsuccessful (U) placed

on the top. The problems that were hardest to find are placed to the left and the problems easiest to find are placed to the right.

Further on in this sub-chapter we will present the usability problems found that were the most severe in relation to the SWS, problems rated to 3 and 4 in the severity rating scale, and that were identified by two or three evaluators. See appendix L.3 to view the usability problems found in the heuristic evaluation.

## Menu/Navigation

Two of the evaluators commented that not all of the pages in the SWS had a single main header. They felt that a main header which is explanatory and eye catching can help the users to navigate. The “Team And Agent Configuration” page (see figure 5.3.2) did not have a single main header which could make it hard for the user to quickly understand the function of this page.

The screenshot shows a web interface with a navigation bar at the top containing the following tabs: Main, Response Time, Agent Availability, Open Incidents, Deviations, Workload, Team And Agent Configuration, and Custd. The main content area is divided into four distinct sections, each with its own header and form elements:

- Create Team:** Features a text input field for 'Team Name', a checked checkbox for 'Is telephone team', and a 'Create Team' button.
- Delete Team:** Features a dropdown menu currently showing 'Team 1', a checkbox for 'Delete this team and its customer and team relat', and a 'Delete Team' button.
- Add Agent To Team:** Features two columns: 'Available Agents' and 'Available Teams'. Under 'Available Agents', there are radio buttons for 'Team 1' (selected) and 'Team 2'. Under 'Available Teams', there are radio buttons for 'Team 1' and 'Team 2'.
- Remove Agent From Team:** Features two columns: 'Teams' and 'Agents in the Team'. Under 'Teams', there are radio buttons for 'Team 1' (selected) and 'Team 2'. Under 'Agents in the Team', there are radio buttons for 'Aki-Petteri Berg' (selected) and 'Alexander Overhkin'.

Figure 5.3.2: Before the heuristic evaluation. The “Team And Agent Configuration” page didn’t have a single main header.

The evaluator at Siemens suggested that we should add a single main header, which is explanatory, to the “Team And Agent Configuration” page. The existing headers (Create Team, Delete Team, Add Agent To Team, and Remove Agent From Team) should be considered as sub-headers and made smaller. The change was made after the heuristic evaluation resulting in a single main header and sub-headers with a smaller font as seen in figure 5.3.3.

<b>Main</b>	<b>Response Time</b>	<b>Agent Availability</b>	<b>Open Incidents</b>	<b>Deviations</b>	<b>Workload</b>	<b>Team/Agent Conf</b>	<b>Cu</b>
-------------	----------------------	---------------------------	-----------------------	-------------------	-----------------	------------------------	-----------

---

### Team And Agent Configuration

---

Delete or Create a Team and Add or Remove Agents from a Team

<p><b>Create Team</b></p> <hr/> <p>Team Name: <input type="text"/></p> <p><input checked="" type="checkbox"/> Is telephone team</p> <p><input type="button" value="Create Team"/></p>	<p><b>Delete Team</b></p> <hr/> <p>Team 1 <input type="button" value="v"/></p> <p><input type="checkbox"/> Delete Team.</p> <p><input type="button" value="Delete Team"/></p>
<p><b>Add Agent To Team</b></p> <hr/> <p>Available Agents:                      Available Teams:</p>	<p><b>Remove Agent From Team</b></p> <hr/> <p>Teams:                                      Agents in the Tea</p>

Figure 5.3.3: After the heuristic evaluation. A single main head was provided for the “Team And Agent Configuration” page.

The same evaluators also felt that there was too little space between the menu and the logo in such a way that the menu was hard to notice. As a result, the empty space between the menu and the logo was increased (see figure 5.3.4.A and B).



Figure 5.3.4 A: Before the heuristic evaluation.      Figure 5.2.4 B: After the heuristic evaluation.

### System response

Both evaluators at HIA noticed that the SWS prototype presented incomprehensible error messages to a potential user. We considered this usability problem to be a usability catastrophe, according to the severity rating scale by J. Nielsen (Table 3.3.1), and corrected the problem at once.

### Department 6211

```
System.IndexOutOfRangeException: There is no row at position 3. at System.Data.RBT
(Int32 index) at System.Data.DataRowCollection.get_Item(Int32 index) at _Default.gener
avgTeamResponsetime) in f:\Thomas\VSprojects\snapshotForTesting-V2\Default.aspx.cs:
V2\Default.aspx.cs:line 76
```

Figure 5.3.5: Incomprehensible error message

### Presentation of data/information

All of the evaluators commented on the use of colors on the main page. They all felt there were too many strong colors that could make it difficult for the user to notice critical information as the red color (signaling an alarm) used did not stand out. The web designer at Siemens helped us select new colors. The colors were replaced with lighter colors as seen in Figure 5.3.7.A and B.

<b>Team 1</b>	Customer 1	00:20	Available	3
	Customer 2	00:28	Unavailable	2
			Work	1
	Avg. response time	00:24	Talk	1
<b>Team 2</b>	Customer 3	00:37	Available	4
	Customer 4	00:24	Unavailable	2
			Work	0
	Avg. response time	00:30	Talk	1

Figure 5.3.6.A: Before the heuristic evaluation. Too many strong colors on the main page could remove the users' attention towards critical information enclosed in the color red.

<b>Team 1</b>	customer 1	14	Available:	2
	customer 2	28	Unavailable:	0
	customer 3	39	Work:	0
	customer 4	22	Talk:	3
	<b>Avg. Response time</b>	<b>27</b>		
<b>Team 2</b>	customer 5	34	Available:	0
	customer 6	42	Unavailable:	1
	customer 7	30	Work:	0
	<b>Avg. Response time</b>	<b>36</b>	Talk:	3

Figure 5.3.6 B: After the heuristic evaluation. The lighter colors used should make it easier for a user to notice critical and important information.

### Consistency

Two of the evaluators, one from HIA and one from Siemens, commented that the use of tables were inconsistent between the different pages. The evaluator from HIA said that the inconsistent use of tables between the different pages could make the users feel that they were not using the same system when switching between the pages.

### Workload

Status at: 2006-02-20 00:00:00

Team	Type	05:00-09:00	09:00-13:00
<b>Team 1</b>	Incomming Calls	6	11
	Answered Calls	2	7
	Agents Logged In	0	0
<b>Team 2</b>	Incomming Calls	2	13
	Answered Calls	2	8
	Agents Logged In	0	0

### Agent Availability

Status at: 2006-02-20 15:56:50

Team	Agent Name	Current State	Seconds In Current State
<b>Team 1</b>	Aki-Petteri Berg	0	0
	Alexander Ovechkin	0	0
	Saku Koivo	0	0
	Peter Forsberg	0	0
	Milan Hejduk	0	0

Figure 5.3.7: Inconsistent use of tables between the different pages. The use of background color on every other row was not used on every table.

### 5.3.2 User test 1

After the heuristic evaluation the SWS prototype was modified according to the usability problems found in the inspection.

Three test users participated in user test 1, all of them are team leaders at Siemens SBS department 6211. The background and competence of the test persons can be found in sub-chapter 4.3.2.

Table 5.3.2 shows the number of problems in each ranking category each of the test users identified during user test 1. The total number of usability problems identified in each ranking category, the total number of problems identified by the test users, and the total time spent on the task scenarios are also shown.

Rank	Number of problems identified			
	Test user			Total
	1	2	3	
0				0
1	1		1	1(0)
2				0
3	2(1)	2	5(3)	6(4)
4	3	3	4	5(0)
<b>Total</b>	<b>6(1)</b>	<b>5</b>	<b>10(3)</b>	<b>12(4)</b>
<b>Time used</b>	<b>742 sec</b>	<b>755 sec</b>	<b>848 sec</b>	<b>2345 sec</b>

Table 5.3.2: An overview of how many usability problems found by the different test users in each ranking category. The numbers in parentheses indicate how many of the usability problems that are uniquely identified. Time used indicates how many seconds a test user spent on the task scenarios.

Figure 5.3.9 gives an overview of the usability problems found in each ranking category. The overview illustrates which usability problems that are uniquely identified and the overlap of other usability problems found.



Rank: 1

U ← Test user → S	2	
	3	
	1	
		5.1
Hard ← Problem → Easy		

Rank: 2

U ← Test user → S	1					
	2					
	3					
		1.3	3.1	3.2	6.3	6.1
Hard ← Problem → Easy						

Rank: 4

U ← Test user → S	1					
	2					
	3					
		1.1	1.2	3.4	3.3	6.2
Hard ← Problem → Easy						

Figure 5.3.9: The figure shows which test users found which usability problems in each ranking category. Each square shows whether the test user (represented by the row) found the usability problems (represented by the column): The square is orange if this is the case and white if the test user did not find the problem. The most successful (S) test users are placed at the bottom and the unsuccessful (U) placed on the top. The problems that were hardest to find are placed to the left and the problems easiest to find are placed to the right.

We will now present the usability problems found that were the most severe in relation to the SWS, problems rated to 3 and 4 in the severity rating scale, and that were identified by two or three test users. See appendix M.2 to view the usability problems found in user test 1.

### Menu/navigation

In task scenario 1 (see task scenarios for user test 1 in appendix M.1), two of the test users (test user 1 and 3) didn't use the "Main" page to solve the problem. The two test users solved the task by using the "Response time" page. The same problem appears in task scenario 2 and task scenario 5 where test user 2 and test user 3 don't use the "Main" page straight away to solve the tasks. Even though task scenario 2 can be solved either way, it would be preferred that the "Main" page is being used whenever possible to save time. Test user 2 and test user 3 spends respectively 68 and 90 seconds on task 2. Test user 1, which uses the "Main" page saves time and uses 64 seconds. On task 5, test user 2 and test user 3 are going back and forth through the different pages (not the Main page) instead of going straight to the Main page. As a result they spend 158 and 55 seconds on the task. Test user 1 which solves the task by using the "Main" page spends only 4 seconds.

It was obvious that either the purpose of the "Main" page wasn't clear to the users (the "Main" page should be used to get quick and critical present data) or the "Main" menu tab

wasn't noticeable compared to the other menu tabs as the size of the "Main" menu tab was smaller than the others.



Figure 5.3.10.A: Before user test 1. The "Main" tab was smaller than the other tabs and the headline wasn't self-explanatory.

To correct the problem, the "Main" page got a new headline and an explanatory text under the headline. The size of the "Main" tab was also increased.



This is an overview department 6211's performance so far today. It could be up to 15 minutes

Figure 5.3.10.B: After user test 1. The "Main" tab was increased, a new headline and an explanatory text under the headline were implemented.

### System response

Two of the test users had problems noticing the response messages from the SWS. Therefore, the response field was enlarged and the font used was bolded out.



Figure 5.3.11: The response field before and after user test 1.

### Presentation of data/information

All of the users had problems with distinguishing the "Team and Agent configuration" page and the "Users" page. It seems it is not intuitive for the users where the different functions can be found between the three pages ("Team and agent configuration", "Customer configuration", and "Users") of the Admin system.

When solving task scenario 10 (appendix M.1), two of the users, test user 2 and 3, did not find the section for deleting a team and the team's relations right away in the "Team and Agent configuration" page. Instead the two test users started to delete the agents belonging to the team they were told to delete on the "Users" page. Then they deleted the team's customers on the "Customer configuration" page, before they finally deleted the team on the "Team and Agent configuration" page. Test user 1 used the section for deleting a team and its relations, spending only 9 seconds on the task. User 2 and 3, respectively, spent 55 and 90 seconds on the task.

The problems above were corrected by adding explanatory text under the main header on the "Team and agent configuration", "Customer configuration", and "Users" page.

## Input

All of the users were startled over the drop-down-menu used to add a new user on the Users page. They expected a text box to write the name of the new user. They didn't know that the new user already existed in the Siemens system, but was needed to be added in the SWS using the drop-down-menu.

### Add a new User

Figure 5.3.12.A: The drop-down-menu on the "Users" page before user test 1

The problem was corrected by adding text above the drop-down-menu describing its functionality.

### Add a new User

Select a user from the Siemens ORS user base (LDAP logins) and determine what the new user is allowed to use. Then add the user

Figure 5.3.12.B: After user test 1. Text was added above the drop-down-menu describing its functionality.

All of the users discovered that the function for deleting a customer from a team did not work on this version of the SWS prototype. A condition in a store procedure updating the screen was wrong and was fixed right away.

### 5.3.3 User test 2

The SWS prototype was corrected after user test 1 according to the usability problems found.

The same test users from user test 1 participated in user test 2. In addition a fourth test user, an escalation manager, took part in user test 2.

Table 5.3.3 and Figure 5.3.13 will not be explained as similar tables and figures are explained in previous sub-chapter.

Rank	Number of problems identified				
	Test user				Total
	1	2	3	4	
0		1	1		1(0)
1					0
2		1	1		1(0)
3	1(1)			1(1)	2(2)
4		1(1)		2(2)	3(3)
<b>Total</b>	<b>1(1)</b>	<b>3(1)</b>	<b>2(0)</b>	<b>3(3)</b>	<b>7(5)</b>
<b>Time used</b>	<b>252 sec</b>	<b>422 sec</b>	<b>369 sec</b>	<b>445 sec</b>	<b>1488 sec</b>

Table 5.3.3: An overview of how many usability problems found by the different test users in each ranking category in user test 2

Rank: 0

U ← Test user → S	4	
	1	
	3	
	2	
		1.1
Hard ← Problem → Easy		

Rank: 2

U ← Test user → S	4	
	1	
	3	
	2	
		3.2
Hard ← Problem → Easy		

Rank: 3

U ← Test user → S	3		
	2		
	4		
	1		
		3.1	4.1
Hard ← Problem → Easy			

Rank: 4

U ← Test user → S	3			
	1			
	2			
	4			
		3.3	5.1	5.2
Hard ← Problem → Easy				

Figure 5.3.13: The figure shows which test users found which usability problems in each ranking category in user test 2

Further on in this sub-chapter we will present the usability problems found that was rated as most severe, problems rated to 4 in the severity rating scale. To view the usability problems found in user test 2, see appendix N.2.

**Presentation of data/information**

Test user 2 finds the “Delete team” checkbox and the “Delete team” button, in the “Team and Configuration” page, confusing as both items has the same text. The text was changed on all ”Delete Check boxes” to illustrate that they are there for confirmation reasons and not function as a tool for removal, as shown in figure 5.3.14.B.

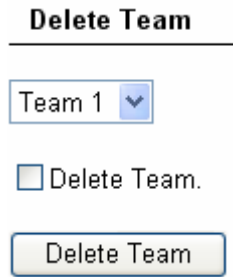


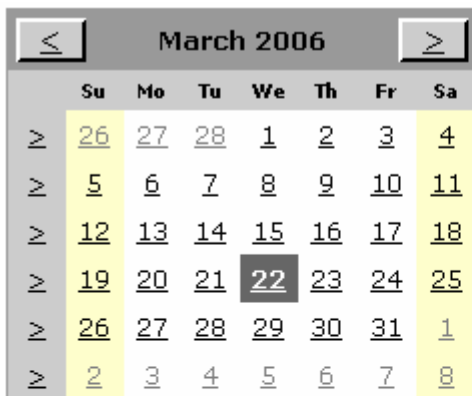
Figure 5.3.14.A: The text next to the checkbox before user test 2



Figure 5.3.14.B: The text next to the checkbox after user test 2

**Other problems**

User 4 discovers that a user can not select a period on the calendar to get information within this period; a user can only get history information for a specific date as seen in figure 5.3.15.



Resume for: March, 2006

Team	Type	05:00-09:00	09:00-13:00
Team 1	Avg.Incomming Calls	0	0
	Avg.Answered Calls	0	0
	Avg.Agents Logged In	0	0
Team 2	Avg.Incomming Calls	0	0
	Avg.Answered Calls	0	0
	Avg.Agents Logged In	0	0

Figure 5.3.15: The user could only view workload for a certain date.

This problem was needed to be fixed before we could deploy the SWS as this requirement was stated in the requirement specification earlier in the project. To fix this shortcoming a new calendar on the “Workload” page was added. There are now one calendar representing the “From Date” and another calendar representing a “To Date” as seen in figure 5.3.16.

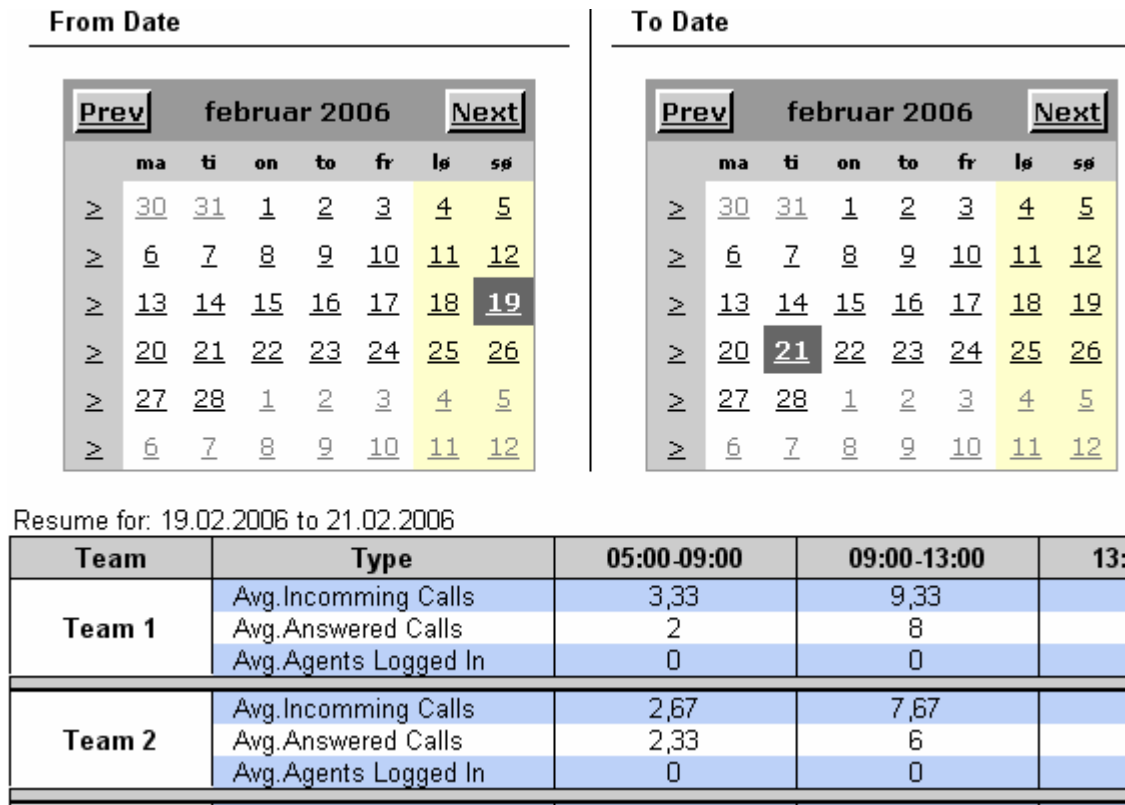


Figure 5.3.16: A new calendar was added in the “Workload” page so that a user could select a “from date” and a “to date”.

## 6 Discussion

In this chapter we will discuss the results found in chapter 5. Our experience with the accomplishment of the methods used will also be discussed.

### 6.1 The use of Contextual Design as a design method

We will in this subchapter discuss each of the steps in Contextual Design. What we did during each of those steps, what could have been done different, and the results of each step. We will also discuss the use of Contextual Design as a design method.

#### 6.1.1 Contextual Inquiry

The Contextual Inquiry was conducted by two of us and one user at the time. We felt at the time that this was a smart way to do the interviews because of the time we saved and the possibility to compare and discuss the results.

Now that we are finished we can see that it also could be negative to be two interviewers. During several of the interviews, especially two, we felt it like the user was a bit insecure. It took some time before the explanations became any good. Two interviewers and one user could be uncomfortable.

When we interviewed the last users we started to notice that they seemed more insecure than the first users we interviewed. A reason for this could be that they saw us conduct the other interviews, and felt that we were testing them out instead of interviewing them.

The users became more relaxed as the time went by, and we didn't comment any of their answers. They started to talk more freely, and did often end up talking about things outside the question we asked.

They told us about the agents working in the teams, how they distributed work in the teams, what they wanted to change in the department, and how they liked or did not like what they were doing.

All this talk around the questions did sometimes confuse us, and we had to ask them to answer some of the questions again.

The users we interviewed came and asked us when they wanted additional features in the phone system, and we were asked to attend to meetings representing the department as technical personnel. We felt that we were accepted in the department, and that they probably thought that we knew more about the systems they had than we really did.

The questions we created could have been formulated more open. We knew what kind of answer we wanted, and we steered the user in that direction with the questions. This did work out quite well, and we didn't get that many very different answers to each question, but we could have created the question more open to let the user do more of the thinking themselves.

#### 6.1.2 Work Modeling

We got several positive replies for our choice of models from SBS, even though they didn't validate to any standard. We used UML Flow model as basis, but had to adjust them to fit our system. The models we created used both computer and humans, and data was flowing around the whole system at the time.

We could have chosen a different type of model, maybe a sequence diagram. A sequence diagram could give a more detailed description of what the user actually did when the task was performed.

We had some superficial knowledge about the system from Contextual Inquiry, but we did not know anything about the database and their connections to the different systems used. We had a dialog going with SBS when we created the models to make sure that they were correct.

We are satisfied with the Work Models. They gave us deeper insight in the user's work than the interviews, and added knowledge of the databases, systems, and communication.

### 6.1.3 Consolidation

The Consolidation models were the most time consuming models we created during Contextual Design.

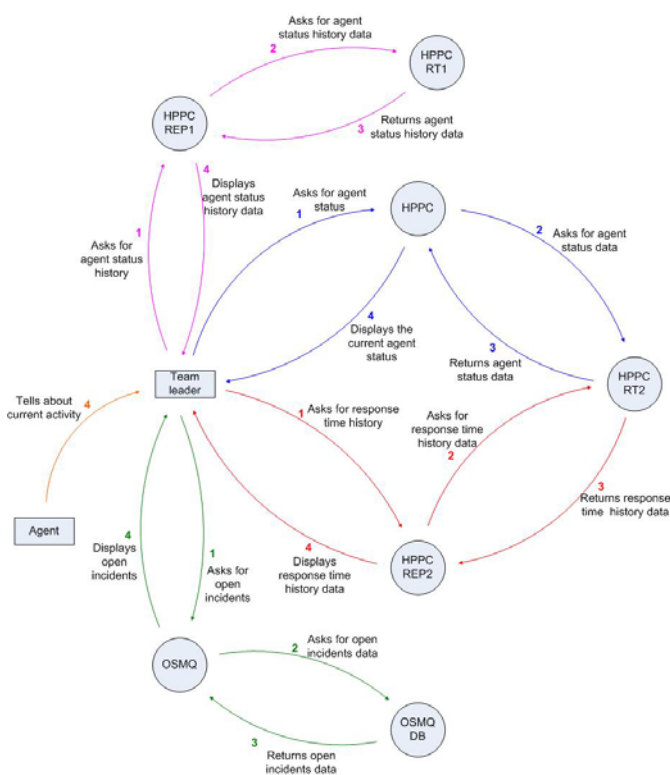


Figure 6.1.1: Consolidation model

They became large and confusing. We could not really see the point of these models in a small project like this. We didn't feel that we had any use for the Consolidation models later in the project.

The most positive we got out of the Consolidation was that we went through the Work Models again, and discovered several small mistakes we could correct.

This step could be removed from Contextual Design in this project without having anything to say for the outcome.



### 6.1.4 Work Redesign

We had several meetings, before we started the design process, where we got a description of the assignment. This information was used, together with the experiences from the Work Modeling and the Contextual Inquiry, to create the Work Redesign models. SBS didn't want to give us any directions on how to design the new system, so we had to take our own decisions and designed it the way we felt was the best.

The models we used, flow models, were the same type of models we had used during Work Modeling. We could have used sequence diagrams here too, but since we had used flow models in the Work Modeling, we continued with that.

To create these models was a very positive experience. We had the feeling that we made a difference for the users with a new system.

The comments we got after designing the Work Redesign models was very motivating. They gave us only positive replies and the models we created were exactly what they wanted.

We should have done more research before we started to create the models. We did not check with the server administrators if we were allowed to have a Windows Service running on their servers, and if we could access their databases the way we wanted to.

We ended up with Work Redesign models that seemed very good at the moment, but they weren't. They could have been correct with only small adjustments, which could have been done when the system was finished.

The Work Redesign models (in appendix F) will give a pointer to how we thought the system would be, not the way it became.

### 6.1.5 User Environment Design

The floor plan was something new for us. We had never seen any examples of a floor plan before, so what we created was our own model.

We did create the Floor Plan mostly with basis in the user requirement specification, and it became a link from the design to the final product. The system design was in our thoughts the whole time during the creation of the figure, and the user requirement specification gave us a lot of ideas.

We didn't check enough with SBS around their login and password policy. We discussed our password / login solution with them, and got a clear to go signal. It was a system with our own login page which used own password and usernames (nothing from their system). If we had asked some SBS programmers or system administrators we would have found out that they did not like a solution with own passwords in the system. They wanted all their logins and passwords controlled by the Active Directory.

This mistake had consequences during the whole project. We had to reprogram the authentication and login from our solution to use the Active Directory before we deployed the system.

The most positive thing with the floor plan was that it made us think. We thought about modules, and system parts (which also got from the user requirement specification). It was during this process we created the system design we used later.

The floor plan was very good to have when we started the development, and that gave it us a kick start.

### **6.1.6 Mockup And Test With Customers**

The prototype we created during Mockup And Test With Customers was a good experience, and it was fun creating.

The prototype became pretty close to our final solution, if we disregards the login function. We could have chosen to use different colored paper to separate the pages, but we used white to simulate the colors we decided to use on the real webpage.

Post it notes was clever. We got the idea from Byer and Holtzblatt [2]. It gave the page some colorful dynamics, and it was easier to control than the loose pages we used to create the other dynamics with.

Testing with the customers gave us a really good indication of what they wanted. We were given a lot of constructive replies, and we used it during the development.

We could have created the paper prototype better by just using the Siemens website style guides for layout, but we had not found the guides yet when we created the prototype.

The users which tested the prototype together with us said it was very fun, and that they got the impression of using a real website.

### **6.1.1 The use of Contextual Design as a design method**

When we started the design phase, we didn't know anything about Contextual Design. We didn't even know what a design method was. When we first heard about Contextual Design, we thought it to be a development method, like RUP.

When we started to work on this thesis we got a paper by Beyer and Holtzblatt [2] on Contextual Design from one of our professors. It described the method clearly and strict. Everything that needed to be done, and the way to do it, was described stepwise. We felt that we got the hang of the method just by reading.

Contextual Design was a very good opportunity for us to learn more about the department, their business, and to get to know the employees. The profit we gained in form of business knowledge from Contextual Design was unique. We had probably never learnt the things we did about the user's perspective of the department, their ways of working, and their problems in the daily work if it hadn't been for Contextual Design.

The new models we created (Work Redesign, Floor Plan, and The Paper Prototype) was models we used during the development process, some more than others. We got a kick start from the models which helped us a lot to get started on the development.

We are very satisfied with Contextual Design as a design method for our project. We got very good business knowledge, and we designed the system at the same time. The method was easy to conduct, the descriptions were easy to follow, and it made sense. Everything except the Consolidation had a purpose. We are satisfied with Contextual Design as a design method.

## **6.2 Experience from the development**

We will in this subchapter discuss the development process and the results for the creation of development documents and the database, and the software development. We will also discuss the implementation of the system.

### **6.2.1 The development documents**

We ended up with two different class diagrams after development, which both could have been very useful during the development.

The class diagram that we did create before, and had during, the development was useless.

We can see now that we shouldn't have spent time trying to create a class diagram before the development as we did. We didn't this, due lack experience with ASP.net web development. The Floor Plan, which we created during the Contextual Design, worked as a kind of substitute for the class diagram during the development.

The entity relationship diagram was finished before the software implementation of the database started and it worked as our negotiation document against SBS when we tried to convince them about our database. This is a diagram we are satisfied with, and there are not many things that could have been done different to fit SBS's needs.

There are though one thing on the diagram (and in the database), which has nothing to do with functionality, that should have been updated. The names used in the different tables are not consistent. We should have used the same name in all tables for fields that contained the same data. For example timeEnded, ended, and timeEnd is used for a field which contains the time and date for an ended relationship. This should be done earlier in the process, before we started any development.

### **6.2.2 The database**

We had an easy job creating the database since the entity relationship document was good. We could just write the SQL queries right of the diagram, and that saved us a lot of time. We could have used a tool to create the database, but since the diagram we had was so helpful, was it easier to just write the queries manually.

We didn't do any big changes on the database during the development, and we are satisfied with the database we created. It is easy to extend, it is normalized correct, and the structure is created for further development.

### **6.2.3 The web application and the Windows application**

The software that we created works as intended (see appendix P, for a SBS comment on the software).

The web application turned out almost exactly how we planned, for both GUI and functionality, during Contextual Design. We had to change the login feature we created to fit the SBS password policies dough. We should have discovered the SBS password policy earlier, during Contextual Design, or at least before we started to code. We did not see the possibility that SBS could have rules against locale saving of passwords after we got our first clear to go signal.

The Windows application did also go through some reprogramming. It started out as a Windows Service and ended up as a GUI less Windows application.

The Windows Service had never been an issue if we had checked with the server administrators during Contextual Design.

The reprogramming was no big problem. It took less than one day, and we discovered a few smaller bugs that would have appeared sooner or later. It is not certain that the reprogramming did set us back at all.

We got a lot of positive comments on the web application and that was very motivating for doing it even better. There was never any question if we should choose the easy way, or the hard but better way. All the positive comments made us always select the robust solution. We could have saved time several places, but we did really want to give SBS what they wanted, and they motivated us through the whole process.

#### **6.2.4 The implementation**

We had some problems in the end of the development phase. Both the SBS databases we used had duplication errors. We discovered the errors for SBS since they created errors during our tests. We rewrote the Windows application to handle the error on the OSMq copy database, but we had to wait for the error on the HPPC database to be fixed. The DTS job would start to fill our database up with information before the data was concise. Because of this was the implementation of the system a bit delayed.

We were asked if we were willing to help them with the system after our thesis was finished, which we agreed on. They wanted to set up a Work Shop where we were going to give a short introduction to the system and show some functionality. We have also said agreed to help them if they find bugs or want to add new features later.

### **6.3 Experience from the evaluation**

In this subchapter we will first discuss our experience with the heuristic evaluation as an inspection method. In subchapter 6.3.2 our experience with user testing is discussed. In subchapter 6.3.3 we will discuss how these two evaluation methods, the heuristic evaluation and the user tests, affected the SWS in terms of usability. The advantage of employing both heuristic evaluation and user tests to further develop our system is discussed in subchapter 6.3.4.

#### **6.3.1 Experience from the heuristic evaluation**

##### **Discussion on the execution of the heuristic evaluation**

We conducted the heuristic evaluation with three evaluators, two evaluators from HIA and one from Siemens. The two first heuristic evaluations were performed at HIA and the last heuristic evaluation was executed at Siemens. We had no experience on conducting a heuristic evaluation beforehand, and as our experience grew the execution of the heuristic evaluation got better.

The first heuristic evaluation did not function as well as we would have hoped. One of the evaluators expressed displeasure on the large number of questions on the heuristic checklist. Looking back, we may have reduced the number of questions on the heuristic evaluation checklist. J. Nielsen recommends splitting up the evaluation into

several smaller sessions, each concentrating on a part of the interface, rather than having one long evaluation session [11]. However, in our situation where it was difficult to get appointments with the evaluators and two of the evaluators were situated in Grimstad, it would have been better to reduce the number of questions. The evaluator also commented that it was difficult to find usability problems as the evaluator wasn't familiar with different terms used at Siemens, as discussed in subchapter 6.3.4.

The second and third heuristic evaluation went well as many (26) usability problems were discovered between the evaluators, where 16 usability problems were discovered by the evaluator 2 and 17 usability problems discovered by evaluator 3.

The checklist which was provided was used extensively throughout the second heuristic evaluation and it contributed to find plenty of usability problems found. The checklist was also used in the first and third heuristic evaluation, but not to any great extent. In the third heuristic evaluation, a separate sheet was used to make note of usability problems and comments from the evaluator. We felt it was helpful for the evaluators and us (as observers) to use both the checklist and the separate sheet to find usability problems and other comments.

"The goals of the system" as a guidance to the system was hardly used by any of the evaluators and it can be discussed whether the use of task scenarios may have contributed to find more usability problems. However, we chose to not give the evaluators task scenarios as the SWS is a domain-dependent system and it may have been difficult for the evaluators to carry out the tasks given to them. With the SWS, we felt it more suitable to let the evaluators get to know the system by providing them with information of what the system is meant to do, and letting them discover each page of the system while approaching us with questions when needed.

A tape recorder or a video camera could have been used in the heuristic evaluation. We assumed, with the use of both the checklist and a separate sheet, that we could register every comment usability problem. Although we registered most of the comments and problems, we believe it would be easier to look back on the heuristic evaluations if we had them recorded.

### **Discussion on the heuristic evaluation results**

The three inspectors found 26 usability problems in total, where 16 of the problems were uniquely identified. All of the uniquely identified problems were discovered either by evaluator 2 or evaluator 3. These two evaluators have different backgrounds (one evaluator from HIA with knowledge on usability, and one evaluator from Siemens with experience and expertise on web design), which could serve as an explanation why 62 % of the usability problems found were uniquely identified.

The evaluators found, as we expected, mostly minor problems (or easy noticeable problems, as explained in chapter 3.3.2) as they had no domain-expertise. We could have used a couple of more evaluators in order to find more and major usability problems. Also, studies by J. Nielsen [11] have found that three evaluators will identify approximately 50 % of all usability problems and five evaluators will identify approximately 75 %. Although we may have found more usability problems and more major problems with the use of two more evaluators with domain-expertise during the heuristic evaluation, we believe that the usability problems (and major problems) that were not found during the heuristic

evaluation were found during the user tests as the heuristic evaluation and the user tests supplement each other (J. Nielsen [19]).

### 6.3.2 Experience from the user tests

#### Discussion on the execution of the user tests

We used three test users in the first user test, and four test users in the second test. We wanted to use more test users in user test 2 so that we could obtain substantial measures on whether the system had become easier to remember after user test 1. Due to the time constraint, both for this project and to fit the user tests into a tight working schedule for the test users at Siemens, were we only able to provide four test users.

Before the user tests we said to the test users that the purpose of the user tests is to test the system rather than them to avoid any discomfort. To our surprise the test users found the test situation as enjoyable. We also felt that the test users tried to solve the tasks as fast as possible as they asked each other and compared the time they used.

The task scenarios worked well in the user tests. The use of the Contextual Design method in the design of the SWS gave us good insight in the work tasks performed at the department and contributed to the making of the task scenarios. The task scenarios helped us find usability problems and functions which were not implemented as required. In particular, during task 7 in user test 2 test user 4 discovered a major lack in the SWS. The user wanted a flexible time period (“to- “and “from date”) when selecting to view historical work load data. The lack in the system was a requirement in the requirement specification developed earlier in the project, but the requirement had low priority. Since the user, during user test 2, very much wanted this requirement implemented we added this function after user test 2.

The use of a video camera was useful when evaluating the user tests. Filming the laptop display while the test users solved the tasks made it easier for us to remember how they solved the tasks, how they moved between the different pages in the SWS, and when they had to stop to ask for hints.

The user tests were performed at the test users' work place, that is, in an open work landscape. This resulted in much background noise which made it difficult to hear what was said during the user tests. Based on our experience, we would recommend the use of a closed of office when performing similar tests to avoid background noise. The use of an open work landscape to perform the user tests also resulted in the test users being distracted by work related questions and the sound of mobile phones.

We used only one observer during the user tests. Much data had to be recorded in a short period of time during each of the tasks in the user tests. Thus, it was not easy for the observer to detect all of the information that presented it self during each of the tasks. The use of two evaluators could have made it easier to pick up information; alternatively one can consider using a second video camera to record the user as long as it is ensured that background noise is avoided.

#### Discussion on the user tests results

12 usability problems were found in user test 1, 11 of them were rated 3 or 4 in the severity rating scale. Hence, 92 % of the usability problems found are categorized as major usability problems. The reason for this is that most of the problems found in user

test 1 are problems which we believe will affect how quickly a user solves a task or how fast a user apprehends data that are displayed. Time is an important factor in our web based monitoring system (the SWS). Also in user test 2 most of the problems found are major usability problems, 5 out of 7 problems (71 %).

As expected, the number of usability problems found decreases from user test 1 to user test 2, indicating the iterative process of applying evaluation methods throughout the development. Only 7 usability problems were found in user test 2, lowering the number of usability problems found with 42 % from user test 1 to user test 2. The test users had an evidently positive attitude towards the SWS before the user tests which may have contributed in fewer usability problems found compared to if they had a more skeptical attitude towards the SWS.

### 6.3.3 Usability

In this subchapter we will discuss whether the usability of the SWS has improved based on the measured data obtained from the two user tests. We will not discuss whether the SWS is more usable compared to the old system. J. Nielsen [8] states: "To determine a system's overall usability on the basis of a set of usability measures, one normally takes the mean value of each of the attributes that have been measured and checks whether these means are better than some previously specified minimum... Since users are known to be very different, it is probably better to consider the entire distribution of usability measures and not the mean value". Thus, we will consider all measures obtained in user test 1 and user test 2 when comparing how the SWS has evolved. The five usability attributes, stated in chapter 2.3.1, are used to evaluate the usability of the SWS version before user test 1 and before user test 2.

#### Easy to learn

To measure learnability J. Nielsen [8] states: "One simply picks some users who have not used the system before and measures the time it takes to reach a specified level of proficiency in using it". In our case, we will not use a specified level of proficiency, but we will compare the time it takes to complete the tasks for the test users in user test 1 with the time measures for user 4 in user test 2.

User 1, user 2, and user 3 spent respectively 742, 755, and 848 seconds to complete all of the tasks in user test 1, as seen in table 6.3.1 below. None of the users had used the SWS before the user test. In user test 2, user 4 (which had never used any version of the SWS before user test 2) spent 445 seconds on the tasks. The measurements obtained indicate that the SWS have become easier to learn as the tasks in the two user tests were identical (except from task 8 where the task had been slightly modified).

In table 6.3.1 we have compared the time spent on each task for test user 1-3 in user test 1 with the time spent on each task for user 4 in user test 2. As we can see, the time spent on each task for user 4 in user test 2 are considerably lower than the times spent on each task for user 1-3 in user test 1. Only on task 5 and 10 have user 4's time been beaten by user 1.

Task	Time used on each task in user test 1			Time used on each task in user test 2
	User 1	User 2	User 3	User 4
1	113	50	34	16
2	64	68	90	64
3	47	55	46	33
4	76	78	240	24
5	4	158	55	22
6	220	42	45	29
7	68	67	54	44
8	74	83	72	55
9	67	99	122	134
10	9	55	90	24
Total	<b>742</b>	<b>755</b>	<b>848</b>	<b>445</b>

Table 6.3.1: Comparing time spent on each task for test user 1-3 in user test 1 with the time spent on each task for user 4 in user test 2.

However, as we were only able to include one new test user in user test 2, we can not conclude with certainty that the SWS has become more easy to learn after the corrections had been implemented after user test 1. We can only say that the SWS may have become easier to learn.

### Efficient to use

To measure efficiency of use we have followed J. Nielsen's [8] recommendation of measuring the time it takes the users to perform some typical test tasks. We will not state whether the SWS is more efficient to use compared with the work tasks performed today as this is not within the scope of this thesis.

Table 6.3.2 shows the total time spent on the two user tests, as a total and for each user. In appendix M.3 we can see how much time each user spent on each task. The measurements indicate that the SWS has become more efficient to use as all the task scenarios are identical in the two user tests (task 8 was slightly modified before user test 2 to avoid possible misunderstandings). All of the test users that participated in both user tests improved their efficiency significantly. User 1 reduced the total time spent on the tasks by 66 %, user 2 by 44 %, and user 3 by 56 %. Although the three test users did not have the same starting point before each of the two user tests as the three users had gained some experience of using the SWS during user test 1, the measurements do indicate that the SWS version after user test 1 is more efficient to use compared to the



SWS before user test 1. Also, user 4 spent considerably less time on user test 2 compared to the three test users in user test 1.

	User 1	User 2	User 3	User 4	Total time
Total time spent on user test 1 (seconds)	742	755	848		2345
Total time spent on user test 2 (seconds)	252	422	369	445	1043 *
Time reduction (seconds)	490	333	479		1302
Time reduction (%)	<b>66 %</b>	<b>44 %</b>	<b>56 %</b>		<b>56 %</b>

\* Total time user test 2 – total time user 4

Table 6.3.2: Comparing time user test 1 vs. user test 2

### Easy to remember

J. Nielsen [8] mentions two ways of measuring memorability. “One is to perform a standard user test with casual users (people who are using a system intermittently rather than having the fairly frequent use assumed for expert users) who have been away from the system for a specified amount of time, and measure the time they need to perform some typical test tasks. Alternatively, it is possible to conduct a memory test with users after they finish a test session and ask them to name the command (or draw an icon) that do a certain thing. The number of correct answers will then be a way to measure memorability.”

We will use the first method of measuring memorability as J. Nielsen states that the situation where users are away from the system for a period of time is most representative of the reason we want to measure memorability. Preferably it would have been better (in the context of measuring memorability) to perform two user tests, with some time between them, with the same test users on the last SWS version. Unfortunately, with the limited time provided for this project such a test situation could not be applied. Therefore, changes on the SWS were needed to be implemented right after the inspection and each of the user tests.

However, since data is placed almost at the same place in both version of the SWS (before and after user test 1), we can use the measurements from the two user tests to give us a certain indication of the memorability of the SWS. Table 6.3.3 in previous section may indicate that the SWS is easy to remember as less time is spent on user test 2. We have also recorded the number of errors (minor errors) the users performed during the user tests, to see if the users remember the fastest way to solve a task. As we can see from Table 6.3.3, the error rate performed by the users drops with 81 % after user test 1.

	User 1	User 2	User 3	User 4	Total errors
Total errors performed in user test 1	6(6)	3(3)	7(7)		16(16)
Total errors performed in user test 2	0(0)	2(2)	1(1)	1(1)	3(3) *
Error reduction	6(6)	1(1)	6(6)		13(13)
Error reduction (%)	<b>100 %</b>	<b>33 %</b>	<b>86 %</b>		<b>81%</b>

\* Total errors performed user test 2 – total errors performed user 4

Table 6.3.3: Comparing errors user test 1 vs. user test 2

### Few errors

J. Nielsen [8] defines an error in two ways. An error is defined as “any action that does not accomplish the desired goal, and the system’s error rate is measured by counting the number of such actions made by users while performing some specified task”. These errors are considered minor errors as they do not prevent the users from accomplishing their tasks. Major errors are defined by J. Nielsen as “...more catastrophic in nature...making them difficult to recover from. Such catastrophic errors should be counted separately from minor errors, and special efforts should be made to minimize their frequency”.

We have chosen to measure the number of major errors (system errors that are preventing the user from doing work related tasks) discovered by the test users. Table 6.3.4 below shows how many major system errors the test users found in the two user tests. To view in which tasks the system errors were discovered please view appendix M.5 and N.5.

	User 1	User 2	User 3	User 4	Total system errors
Total system errors in user test 1	1(0)	1(0)	2(1)		4(1)
Total system errors in user test 2	0(0)	0(0)	0(0)	2(2)	0(0) *
System error reduction	1(0)	1(0)	2(1)		4(1)
System error reduction (%)	<b>0 %</b>	<b>100 %</b>	<b>100 %</b>		<b>100 %</b>

\* Total system errors user test 2 – total system errors user 4

Table 6.3.4: Comparing system errors user test 1 vs. user test 2

In user test 1, user 1 discovers an error with the calendar. When double-clicking on the calendar a new window pops up with the error message “the page can not be displayed”. User 2 and 3 discovers that the “delete customer” function does not work. User 3 also detects one has to select team 7 (team 7 is used in task 9) every time an agent is added to that team.

In user test 2, no major system error was identified by the test users that participated in both user tests. Test user 4 identified two major errors. The present information on the department as a whole was missing on the main page. User 4 also discovers that a user

can not select a period on the calendar to get information within this period; a user can only get history information for a specific week or month.

When comparing the results of the two user tests in Table 6.3.4 we can see that the number of errors has declined. If we only consider the test users which participated in both user tests, the number of major system errors has dropped 100 %. If we include test user 4, the number of major system error reduction will drop 50%. In either case, the SWS version after user test 1 has fewer errors than the SWS version before user test 1.

### Subjectively pleasing

“Subjective satisfaction may be measured by simply asking the users for their subjective opinion”, J. Nielsen [8]. Nielsen continues: “To ensure consistent measurements by a short questionnaire that is given to users as part of the debriefing session after a user test”.

After each of the user tests the test users were asked to rate the system, according to a semantic differential scale where 1 is considered the best and 7 the worst. The calculated mean of the ratings for the individual answers serves as a final rating for subjective satisfaction. As Nielsen mentions: “If subjective satisfaction ratings are available for different versions of the same system, it is possible to consider the ratings in relation to the others and thus to determine which system is the most pleasant to use”. Table 6.3.5 gives an overview of the results (to view all of the results, see appendix M.6 and N.6).

	User 1	User 2	User 3
Median User test 1	3,4	4,0	5,0
Median User test 2	1,8	2,4	1,8

Table 6.3.5: The calculated median for rating subjective satisfaction in the two user tests.

As we can see from table 6.3.5, the test users seems to find the SWS version after user test 1 more pleasant to use than the SWS version before user test 1. However, many factors may influence the users when answering such rating scale. For instance, the most difficult episode a user experienced is the most memorable for that user. And as J. Nielsen states: “Users normally know that the people who are asking for the ratings have a vested interest in the system being measured and they will tend to be positive unless they have had a really unpleasant experience”. We can therefore argue that one cannot rely solely on user ratings if the goal is to improve system performance.

To add to the discussion whether the SWS version after user test 1 is more pleasant to use compared to the SWS version before user test 1, we recorded the number of times the test users expressed clear joy or clear frustration during the user tests. Even though it is difficult to measure clear joy or frustration without for instance using psycho physiological [8] measures such as EEGs, pupil dilation, heart rate, blood pressure, and level of adrenaline in the blood to estimate users’ stress and comfort levels, it will give us a certain idea of how the users react to the system.

In table 6.3.6 we have compared the results from the two user tests (for a full view of the results, see appendix M.6 and N.6).

	User 1	User 2	User 3	User 4	Total
Nr of times user expresses clear joy/frustration user test 1	3/2	1/0	1/3		5/5
Nr of times user expresses clear joy/frustration user test 2	7/0	1/1	1/0	10/2	9/3 *
Joy/frustration change	+4/-2	0/+1	0/-3		+4/-2

Total joy/frustration user test 2 – total joy/frustration user 4

Table 6.3.6: Comparing joy/frustration user test 1 vs. user test 2

As we can see from table 6.3.6, the users' reactions when performing tasks on the two versions of the SWS suggests that the users are more positive towards the SWS when changed has been according to the usability problems found in user test 1.

### 6.3.4 The use of heuristic evaluation and user tests

One of the evaluators at HIA commented that it was difficult to find usability problems as the evaluator wasn't familiar with the all of the different terms used at Siemens SBS department 6211. This situation matches J. Nielsen's statement on domain-dependent systems; "Evaluators are probably especially likely to overlook usability problems if the system is highly domain-dependent and they have little domain expertise" [19]. Thus, the use of both an inspection method and user testing was necessary in order to find usability problems with user testing that have been overlooked with the heuristic evaluation.

As the user tests were conducted at Siemens during work hours we had to occupy as little of the test users time as possible. By performing a heuristic evaluation on the SWS prototype before the first user test, and correcting the usability problems found, we removed many "obvious" minor usability problems. The test users could then use their limited time on finding the remaining usability problems that were not discovered during the heuristic evaluation.

Applying heuristic evaluation and user tests resulted in many usability problems found. By applying the heuristic evaluation before the user tests, not only did this procedure save resources (test users' time, in this case), but it also functioned as an iterative design method where the system improved gradually.

## 7 Conclusion

Contextual Design helped us to understand the users work situation. We got to see everything from the user's perspective, and added more meat on the bone every time the processes went through a new phase.

We started out by talking to the users them selves. We learnt about their work through answers on questions we gave them, and through small talk around the questions.

We could have formulated the questions we asked in way where the answer wasn't that obvious as it was in our case. We got an introduction to the system, and we knew what SBS wanted as a result of the development.

This knowledge colored the questions, and they were written in way that we could predict the answers to a certain point.

We were allowed to design the new ways of working, through Work Redesign models, just the way we wanted. SBS did not answer any of our questions regarding how the agents should work, and how they wanted the new system.

The Floor Plan became our connection from the design phase and the development. System parts and modules were put together with use of the requirement specification and the Work Redesign models. We started to see the outlines of the new system.

With all the information we got in form of prototypes, models, and interviews did we create the web application and the Windows Service which became a Windows application and we got help to create the DTS job.

The whole new system consisting of three system parts was designed based on the users' work tasks, and it was adapted and adjusted to include all users' needs. Contextual Design as a User Centric Design method helped us to get the information we needed, and to design the system around the user. It was an easy method to follow, and the documentation about the method is good. There could have been better explanations around the choice of models, and the consequences of choosing one type over another. Contextual Design did the development easier for us, since the floor plan worked as a light substitute for the class diagram.

Contextual Design worked very well as a User Centric Design method for us.

The use of Contextual Design as a design method when designing the SWS gave us a good overview of the users' work tasks performed at the department today. We were therefore able to produce accurate task scenarios for the user tests. The task scenarios worked well during the user test.

The number of questions on the checklist used in the heuristic evaluation may have been reduced as one of the evaluators expressed displeasure in the huge number of questions. Even though we may have provided the evaluators with too many questions, we felt that the checklist contributed to find many of the usability problems during the heuristic evaluation sessions. The evaluator also commented that it was difficult to find usability problems as the evaluator wasn't familiar with different terms used at Siemens. The heuristic evaluation was thus used to mostly find minor usability problems.

"The goals of the system" as a guidance to the system was hardly used by any of the evaluators. It may be possible that the use of task scenarios could have contributed to find more usability problems.

62% of the usability problems found during the heuristic evaluations were uniquely identified. The evaluators had different background and can serve as a possible explanation for this result.

The use of a video camera was a useful attribute when evaluating the user tests. We filmed the laptop display while the test users solved the tasks and it made it easier for us to evaluate the user tests.

The user tests were conducted at the department at Siemens in an open work landscape. The background noise from the work place made it difficult to hear what was said on the recordings during the user tests. The test user also got distracted by work related questions from co-workers and the sound of mobile phones while solving the tasks. Thus, we would recommend conducting similar user tests within a closed environment rather than in an open work landscape.

Both user tests found mostly major usability problems. One reason for this is that many minor usability problems had been picked up by the heuristic evaluation. Another reason is that most of the problems found in the user tests are problems which we believe will affect how quickly a user solves a task or how fast a user apprehends data that are displayed. And time is an important factor for our web based monitoring system.

The use of both heuristic evaluation and user testing contributed to an iterative design process, correcting the SWS for usability problems and shortcomings in three iterative phases. By applying the heuristic evaluation before the user tests the evaluators removed many of minor "obvious" thus saving resources, in the form of the test users' time, when applying the user tests. The test users could then use their limited time to find new usability problems that were not picked up by the heuristic evaluation.

J. Nielsen's [8] five usability attributes were used to evaluate the web based monitoring system. The results from the user tests indicate that the SWS has become more usable during the development process based on J. Nielsen's definition on usability.

## 7.1 Future directions/research

There were a few requirements with very low priority which we didn't have time to implement and during the development we found several features that could be added in the future.

Here is a list of things which we feel that should be implemented in the future.

- A summarizing of the whole department on the main page. Today there are only possible to see the info at team level, but it could be interesting to see how the whole department is performing.
- The possibility to export history, and real time, data to Excel without copy and past as a manual job.
- The possibility to change view from graphs to tables on the pages for response time and open incidents.

- XML logging support. A XML-file is created with information on what to log (log level) and where to log to. This is for the Windows application, not the web application.
- The agents removed from the HPPC database should get an end date. Today they will be in the database for ever.
- The import of incidents from OSMQ should be done real time and not as now. The Deviations real time display will always state 0 as registered incidents today, since the last update was done by retrieving data from yesterday.
- Separate open incidents into: Service requests, incidents, and service requests. This will give a more accurate picture of the departments actual open incidents which they have control over.
- All tables on each website could be sorted. This could be done by the use of DOM objects which will make all sorting distributed to the users computers.

Regarding the usability of the system. It would have been interesting to measure how much time the users spend on their work tasks today without the use of the SWS, and obtain measurements from the use of the implemented version of the SWS at Siemens. We could then compare the results to see whether the new system (the SWS) is more effective to use compared to how work is done today. Other measurements besides efficiency would also be interesting to obtain, but the efficiency in the system would be preferable as this is an important attribute in such a monitoring system used at Siemens Business Services, department 6211-CIC.

## References

- [1] S. L. Henry and M. Grossnickle, Accessibility in the User-centered Design Process, Georgia Tech Research Corporation, [www.UIAccess.com/AccessUCD/](http://www.UIAccess.com/AccessUCD/), 2004
- [2] H. Beyer and K. Holtzblatt, Contextual Design, 1997
- [3] Hewet et al. "ACM SIGCHI Curricula for Human-Computer Interaction", ACM, <http://sigchi.org/cdg/cdg2.html>, 2004
- [4] Human Computer Interaction, IBM Research Center, <http://domino.research.ibm.com/comm/research.nsf/pages/r.hci.html>, 2005
- [5] ISO 9241-11 - Guidance on usability, International Organization for Standardization, Part 11, 1998 first edition
- [6] *ISO 13407 - Human centered design processes for interactive systems*, International Organization for Standardization, 1999
- [7] Siemens Business Services, [www.sbs.siemens.no](http://www.sbs.siemens.no), 2005
- [8] J. Nielsen, *Usability Engineering, Pages 23-48, 1993*
- [9] G. Murray and T. Costanzo, on the web, an overview, 1999
- [10] M.B.H Laugen, Anvendbarhet av PDA løsning i et produksjossystem for landbruket, Master thesis HiA 2005
- [11] J. Nielsen, How to Conduct a Heuristic Evaluation, [http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)
- [12] J. Nilsen and R. L. Mack, Usability Inspection Methods, 1994
- [13] G. Cockton, D. Lavery, and A. Woolrych, The human-computer interaction handbook. fundamentals, evolving technologies, and emerging applications, 2003
- [14] J. Rubin, Handbook of usability testing: how plan, design, and conduct effective tests, 2003
- [15] J. Nielsen, Usability inspection methods, Conference companion on Human factors in computing systems, April 1994
- [16] N. Danino, Heuristic Evaluation – a step by step guide, September 2001, <http://www.sitepoint.com/article/heuristic-evaluation-guide>
- [17] <http://www.stcsig.org/usability/topics/articles/he-checklist.html>
- [18] M.D. Levi & F. G. Conrad, A heuristic evaluation of a World Wide Web prototype, [http://stats.bls.gov/ore/htm\\_papers/st960160.htm](http://stats.bls.gov/ore/htm_papers/st960160.htm)
- [19] J. Nielsen, Characteristics of usability problems found by heuristic evaluation, [http://www.useit.com/papers/heuristic/usability\\_problems.html](http://www.useit.com/papers/heuristic/usability_problems.html)
- [20] Wikipedia, stored procedures [http://en.wikipedia.org/wiki/Stored\\_procedures](http://en.wikipedia.org/wiki/Stored_procedures)
- [21] Wikipedia, DTS [http://en.wikipedia.org/wiki/Data\\_transformation\\_service](http://en.wikipedia.org/wiki/Data_transformation_service)
- [22] Wikipedia, User Centered Design [http://en.wikipedia.org/wiki/User\\_centered\\_design](http://en.wikipedia.org/wiki/User_centered_design)
- [23] Wikipedia, HCI [http://en.wikipedia.org/wiki/Human-Computer\\_Interaction](http://en.wikipedia.org/wiki/Human-Computer_Interaction)
- [24] Greenbaum & Kyng (eds), Design At Work – Cooperative design of Computer Systems, 1991
- [25] Schuler & Namiok, Participatory Design, 1993
- [26] Luke Richey (Tomenta Software), The Rational Unified Process, 2004
- [27] Garry Pollise, RUP and XP in small projects, 2001
- [28] Gary Boch, Ivar Jacobsen, and James Rumbaugh, Unified modelling language, 1996
- [29] B. Henderson-sellers, G. Collins, R. Due, et. Al, A qualitative comparison of two processes of object-oriented software development, 2001
- [30] Wikipedia, Rational Unified Process [http://en.wikipedia.org/wiki/Rational\\_Unified\\_Process](http://en.wikipedia.org/wiki/Rational_Unified_Process)



# Appendix

Appendix A: Task description .....	75
Appendix B: Requirement specification.....	76
Appendix C: Contextual Design - Results from the Contextual Inquiry .....	79
Appendix D: Contextual Design – Work modeling .....	85
Appendix E: Contextual Design: Consolidation.....	121
Appendix F: Work Redesign.....	133
Appendix G: User Environment Design.....	139
Appendix H: Mockup and test with customers .....	141
Appendix I: Diagrams from the development .....	142
I.1 Use Case 0.....	142
I.2 Use Case 1:.....	143
I.3 Use Case 2:.....	144
I.4 DFD 0: .....	145
I.5 DFD 1: .....	146
I.6 Class diagram - ASPX.....	147
I.7 Class diagram – C#.....	148
I.8 Entity Relationship diagram .....	153
I.9 System Overview .....	154
Appendix J: Detailed Development Description .....	155
J.1 Class diagram.....	155
J.2 Web application .....	156
J.3 GUI .....	157
J.4 Windows application.....	157
J.5 Database, stored procedures and DTS.....	159
J.6 Bug fixing.....	160
Appendix K File Lists .....	161
K.1 Web Application .....	161
K.2 Windows Application.....	163
K.3 Database.....	164
Appendix L: Evaluation – Usability inspection.....	168
L.1 Heuristic evaluation – Hand out .....	168
L.2: Heuristic evaluation checklist .....	172
L.3: Results from the inspection .....	190
Appendix M: Evaluation - User test 1 .....	195
M.1 Task Scenarios user test 1.....	195
M.2: Results user test 1.....	196
M.3 Time spent on user test 1:.....	199
M.4 Number of errors performed in user test 1:.....	200
M.4 Number of errors performed in user test 1:.....	200
M.5 Number of system errors identified by the user in user test 1:.....	201
M.6 Number of times users express clear joy/frustration in user test 1:.....	202
M.7 Semantic differential scale user test 1 .....	202
Appendix N: Evaluation - User test 2.....	204
N.1 Task scenarios user test 2.....	204
N.2 Results user test 2.....	205
N.3 Time spent on user test 2: .....	207
N.4 Number of errors performed in user test 2: .....	208
N.5 Number of system errors identified by the test users in user test 2: .....	209
N.6 Number of times users expresses clear joy/frustration in user test 2:.....	210
N.7 Semantic differential scale user test 2.....	210
Appendix O: Screenshots.....	212
O.1 Main.....	212
O.2 Response Time .....	213
O.3 Agent Availability .....	214
O.4 Open Incidents .....	215
O.5 Deviations.....	216
O.6 Workload .....	217

O.7 Team/Agent Conf. ....	218
O.10 Customer Conf. ....	219
O.11 User Conf. ....	220
Appendix P: Testimonial from SBS .....	221
Appendix Q: User Manual .....	222

## Appendix A: Task description

<b>Oppgave til mastergraden</b>		<b>Våren 2006</b>	
<b>Oppdragsgiver:</b> Siemens Business Services Øster Akervei 90 Postboks 2 0613 Oslo		<b>Kontaktperson:</b> Bjørn Henrik Pedersen <a href="mailto:Bjoern.h.pedersen@siemens.com">Bjoern.h.pedersen@siemens.com</a> +47 920 95 972	
<b>Kontaktperson/veileder(HiA):</b> Lars Line Lars.line@hia.no		<b>Passer for studieretning:</b> Master IKT	
<b>Oppgavens arbeidstittel:</b> Development and evaluation of a web based monitoring system.			
<b>Tentativ oppgavebeskrivelse:</b>  <p>This thesis will be done at Siemens Business Services (SBS) in Oslo. SBS in Norway has a Customer Interaction Center (CIC), department 6211-CIC, which is organized in Service Desk (SD) teams. Each team is responsible for a set of customers and are performing ICT support for those customers.</p> <p>SBS wants a web based system, Snapshot Website (SWS), for monitoring department 6211 which will be used by Escalation Managers, Team Leaders, and regular CIC Agents. The SWS will give current and historical department performance status.</p> <p>This study will involve design and development of the system, and will be a qualitative study on the usability of the new system.</p>			
<b>Studenter:</b> Thomas Andersen og Trond Undrum			

## Appendix B: Requirement specification

Nr	Requirement	Description	Priority
<b>Delivery</b>			
1	<b>The SWS must be finished before the 1<sup>st</sup> of May 2006</b>	The SWS must be finished for delivery at SBS before the 1 <sup>st</sup> of May 2006 and ready for use. It would be preferred if it was done even earlier.	1
<b>Modules</b>			
2	<b>The SWS will be developed in modules</b>	The SWS will be developed in three modules; The User system, the Management system, and the Admin system.	1
2.1	<b>First module</b>	The first module to be finished is the User system (requirement 3 - 3.1)	1
2.2	<b>Second module</b>	The second module to be finished is the Management system (requirement 4 - 5.6)	1
2.3	<b>Third module</b>	The third module to be finished is the Admin system (requirement 6 - 7.3)	1
<b>Module 1: The User system</b>			
3	<b>The user system will have one main page</b>	The SWS will have a main page which summarizes all the real time attributes from the other pages and displays this in one single main page.	1
3.1	<b>Open system</b>	The User system will not need any login or authentication other than that the user is on the SBS intranet.	1
<b>Module 2: The Management system</b>			
4	<b>The Management system is a login system</b>	There is only possible to access the Management system by logging in through the user system.	2
4.1	<b>Pages displayed</b>	Each user logging into the Management system will be authorized to view some or all of the pages. The pages the user is authorized to view will be displayed.	3
5	<b>The Management system will have pages displaying detailed information</b>	The Management system will have pages for different measures. Each of the pages will be in two modules, one showing detailed information about the present and one for history.	1
5.1	<b>There is a page for response time</b>	The Management system will have a page for response time which presents the average response time today, this week, and this month. There will also be a possibility to select an earlier month and see the response time that month.	1
5.1.1	<b>Change of view in response time</b>	The page for response time will have the possibility to change view from graph to tables.	4
5.2	<b>There is a page for Agent Availability</b>	The Management system will have a page for Agent Availability showing how much time each agent spends on "talk time", "work time", "available time", and "unavailable time". There will also be possible to select an earlier month and see the time was spent that month.	1
5.3	<b>There is a page for Open Incidents</b>	The Management system will have a page for Open Incidents which will display how many open incidents department 6211-CIC and each team has in their backlog this week, Older than 2 days, older than a week, older than 2 weeks, older than 3 weeks, and 4 weeks or older. There will also be possible to select an earlier week and see the open incidents that week.	1
5.4	<b>There is a page for</b>	The Management system will have a page for	1

	<b>Deviations</b>	Deviations which will display the number of answered phone calls, the number of wrap up reason reports, the number of registered incidents, and show the deviation between answered calls and registered incidents this day. There will also be possible to select a start date and an end date and see the deviations in this period.	
5.5	<b>There is a page for Workload</b>	The Management system will have a page for workload which will display the number of incoming phone calls, answered phone calls, and the number of agents logged in. The information will be displayed in periods of 4 hours throughout the day. There will also be possible to select an earlier week and see the workload that week.	1
5.5.1	<b>Flexible time period</b>	The page for work load will have the possibility for manually configuration of the time period used to display the workload.	4
5.6	<b>There is a page for user information</b>	The Management system will have a page where logged in users can change their user information and password.	3
<b>Module3: The Admin system</b>			
6	<b>There is a login for Admin</b>	The only way to get into the Admin system is to log in trough the user system. If the user is authorized, the Admin system will appear in the same way as the Management system.	1
7	<b>The Admin system will have pages for configuration</b>	The Admin system has several pages Configuration. Each of the different pages will allow different configurations.	2
7.1	<b>There is a page for giving access</b>	The Admin system has a page to give certain users access to the Management system or Admin system. The Admin system also provides a username and password.	2
7.1.1	<b>Access to pages</b>	The page for giving access to users will have the possibility to give access to certain pages for certain users.	3
7.2	<b>There is a page for customer relationships</b>	The Admin system has a page to create relationships between customers and teams.	1
7.3	<b>There is a page for team configuration</b>	The Admin system has a page to create teams and add or delete agents on those teams.	1
<b>Software</b>			
8	<b>Specified software</b>	SBS will specify what kind of software that is used an on what kind of software the applications will run.	1
8.1	<b>Database server</b>	The database used for storage will be an MS SQL 2000 Database	1
8.2	<b>Web server</b>	The web server used will be an MS IIS web server.	1
8.3	<b>Application for gather data</b>	The application used on the web server for gathering data from the other databases will be a windows service which will collect data in given intervals.	2
8.4	<b>Web application</b>	The web application will be a ASP.Net C# application.	2
<b>Data collection</b>			
9	<b>Data collection</b>	There will be collected data at given times at the different databases so that they are not used to frequently.	1
9.1	<b>Data collection from HPPC real time database</b>	Data from the HPPC real time database will be collected every 10 <sup>1</sup> minutes.	1
9.2	<b>Data collection from the</b>	Data from the HPPC history database will be	1

9.3	<b>HPPC history database Data collection from the OSMQ database</b>	collected one time. Data from the OSMQ database will be collected every 20 <sup>th</sup> minutes.	1
<b>Coherence</b>			
10	<b>Coherence with old databases</b>	The SWS will be coherent with the OSMQ database, HPPC real time database, and the HPPC history database.	1
11	<b>Coherence with web browser</b>	The SWS will be coherent with the web browser used as default on the agent's computers.	1
12	<b>Coherence with screen size</b>	The SWS will be coherent with the screen size on the agent's computer.	3
<b>Appearance</b>			
13	<b>Layout</b>	The SWS will have the same kind of layout structure as the new FAQ website.	3
14	<b>Logo</b>	The SWS will use the same logo used in the FAQ website.	3
15	<b>Colors</b>	The SWS will use the same tint as used in the FAQ website but the colors could be different.	3
<b>Tools</b>			
16	<b>SBS provides tools</b>	The tools used to create the SWS are supported and provided by SBS.	1

Priority grading: 1: high, 2: medium, 3: low, 4: very low

## Appendix C: Contextual Design - Results from the Contextual Inquiry

In this appendix we will present the questions asked to the team leaders and escalation managers and the answers obtained.

<b>Name</b>	<b>: Anders Refsdahl</b>
<b>Title</b>	<b>: Team Leader Team 4</b>
<b>1. How is it possible to discover that you are not within target of an agreed SLA?</b>	
<ul style="list-style-type: none"> <li>- There is an Excel macro which finds the response time history.</li> <li>- Uses HPPC to get the average response times per customer real time.</li> </ul>	
<b>2. How do you monitor the response time?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the queue of incoming calls and displays a graphical view of the response time for each customer</li> <li>- The Escalation Manager tells if the response times are too high.</li> </ul>	
<b>3. How do you monitor the agents?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the status for each agent. But doesn't use that too much. No use of history.</li> </ul>	
<b>4. How do you monitor open incidents?</b>	
<ul style="list-style-type: none"> <li>- Uses OSMQ to search for open incidents.</li> </ul>	
<b>5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?</b>	
<ul style="list-style-type: none"> <li>- Takes a report from HPPC about incoming calls and a report from OSMQ with incidents and check the numbers.</li> </ul>	
<b>6. How do you measure the workload and plan how many that is going to work?</b>	
<ul style="list-style-type: none"> <li>- No tool is used.</li> <li>- Talks with the agents who worked last year if there are special circumstances.</li> </ul>	

<b>Name</b>	<b>: Andreas Watz</b>
<b>Title</b>	<b>: Team Leader Team 5</b>
<b>1. How is it possible to discover that you are not within target of an agreed SLA?</b>	
<ul style="list-style-type: none"> <li>- Response time from the HPPC phone system</li> <li>- Collecting queues of open incidents from OSMQ and see how long they have been in the system</li> <li>- Customers complains if their problem haven't been solved</li> <li>- There could be used alarms in OSMQ, but this is not done</li> </ul>	
<b>2. How do you monitor the response time?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the queue of incoming calls and displays a graphical view of the response time for each customer</li> </ul>	
<b>3. How do you monitor the agents?</b>	
<ul style="list-style-type: none"> <li>- HPPC is used to get response times. Nothing else is used in this process because of the projects going on right now.</li> </ul>	
<b>4. How do you monitor open incidents?</b>	
<ul style="list-style-type: none"> <li>- Manually go into OSMQ and search for open incidents</li> <li>- The Escalation Manager searches for open incidents and reminds the Team Leader</li> <li>- Uses web interface against OSMQ to get open incidents</li> </ul>	
<b>5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?</b>	
<ul style="list-style-type: none"> <li>- Manually check the number of incoming calls from HPPC and the number of OSMQ incidents.</li> <li>- There could be registered in HPPC if a customer calls several times, but this is often forgotten.</li> </ul>	
<b>6. How do you measure the workload and plan how many that is going to work?</b>	
<ul style="list-style-type: none"> <li>- Check OSMQ for registered incidents</li> <li>- Routines</li> <li>- Information from customer about special circumstances</li> </ul>	



<b>Name</b>	<b>: Henrik Marteliusson</b>
<b>Title</b>	<b>: Team Leader Team 1 - Siemens</b>
<b>1. How is it possible to discover that you are not within target of an agreed SLA?</b>	
<ul style="list-style-type: none"> <li>- Response time from the HPPC phone system</li> <li>- Collecting queues of open incidents from OSMQ and see how long they have been in the system</li> <li>- Customers complains if their problem haven't been solved</li> <li>- There are no clear SLA's defined at the moment</li> </ul>	
<b>2. How do you monitor the response time?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the queue of incoming calls and displays a graphical view of the response time for each customer</li> </ul>	
<b>3. How do you monitor the agents?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the status of each agent graphically</li> <li>- Everyone is telling the team leader what they do</li> <li>- The helpdesk requires people to leave the service desk and go there fast, which makes it hard to follow everyone</li> </ul>	
<b>4. How do you monitor open incidents?</b>	
<ul style="list-style-type: none"> <li>- Manually go into OSMQ and search for open incidents</li> </ul>	
<b>5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?</b>	
<ul style="list-style-type: none"> <li>- Manually check the number of incoming calls from HPPC and the number of OSMQ incidents.</li> <li>- A lot of incidents are forwarded to other teams, and emails are coming in but not registered in this calculation. This makes it inaccurate.</li> </ul>	
<b>6. How do you measure the workload and plan how many that is going to work?</b>	
<ul style="list-style-type: none"> <li>- Uses OSMQ and check the traffic last year</li> <li>- Experience from special occurrence</li> </ul>	

<b>Name</b>	<b>: Raymond Wiik</b>
<b>Title</b>	<b>: Team Leader Team 3 - Retail</b>
<b>1. How is it possible to discover that you are not within target of an agreed SLA?</b>	
<ul style="list-style-type: none"> <li>- Response time from the HPPC phone system</li> <li>- Collecting queues of open incidents from OSMQ and see how long they have been in the system</li> <li>- Customers complains if their problem haven't been solved</li> <li>- Almost all the SLAs are similar</li> </ul>	
<b>2. How do you monitor the response time?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the queue of incoming calls and displays a graphical view of the response time for each customer</li> </ul>	
<b>3. How do you monitor the agents?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the status of each agent graphically</li> </ul>	
<b>4. How do you monitor open incidents?</b>	
<ul style="list-style-type: none"> <li>- Manually go into OSMQ and search for open incidents</li> <li>- The Escalation Manager searches for open incidents and reminds the Team Leader</li> <li>- 15-20 open incidents are allowed in the queue.</li> </ul>	
<b>5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?</b>	
<ul style="list-style-type: none"> <li>- Manually check the number of incoming calls from HPPC and the number of OSMQ incidents. There is also a Excel macro tool but that's not used because of the time.</li> </ul>	
<b>6. How do you measure the workload and plan how many that is going to work?</b>	
<ul style="list-style-type: none"> <li>- Routines. In the end of every month more people is needed.</li> <li>- Customers telling about special circumstances.</li> </ul>	

<b>Name</b>	<b>: Attila Kalmar</b>
<b>Title</b>	<b>: Escalation Manager, vicarious Department Manager</b>
<b>1. How is it possible to discover that you are not within target of an agreed SLA?</b>	
<ul style="list-style-type: none"> <li>- The activity and the stress of the agents is used as an indication that the SLA's could be out of target</li> <li>- OSMQ is used for manually search after old open incidents</li> <li>- HPPC is used to get the average response times per customer real time.</li> <li>- Manually checking against the SLA's to see if they are within target</li> </ul>	
<b>2. How do you monitor the response time?</b>	
<ul style="list-style-type: none"> <li>- HPPC is used to get the response times for each customer</li> </ul>	
<b>3. How do you monitor the agents?</b>	
<ul style="list-style-type: none"> <li>- HPPC shows the status for each agent.</li> <li>- OSMQ generate reports over opened incidents</li> <li>- HPPC reporter generates reports over the agents history</li> </ul>	
<b>4. How do you monitor open incidents?</b>	
<ul style="list-style-type: none"> <li>- Use OSMQ to manual search for open incidents</li> <li>- Use the web client to get lists over open incidents</li> </ul>	
<b>5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?</b>	
<ul style="list-style-type: none"> <li>- Gets all answered calls from Excel (could use HPPC-reporter, but takes to much time)</li> <li>- Gets all Wrap up reasons from HPPC- Reporter</li> <li>- Gets all reported incidents from OSMQ</li> <li>- Calculates the deviation manually</li> <li>- Have the possibility to use an EXCEL macro to calculate</li> </ul>	
<b>6. How do you measure the workload and plan how many that is going to work?</b>	
<ul style="list-style-type: none"> <li>- HPPC reporter to se the response time at different times at the day to se how many incoming calls there were per agent.</li> <li>- OSMQ to find the number of opened cases</li> <li>- The work plan is used to se how many that were working</li> </ul>	

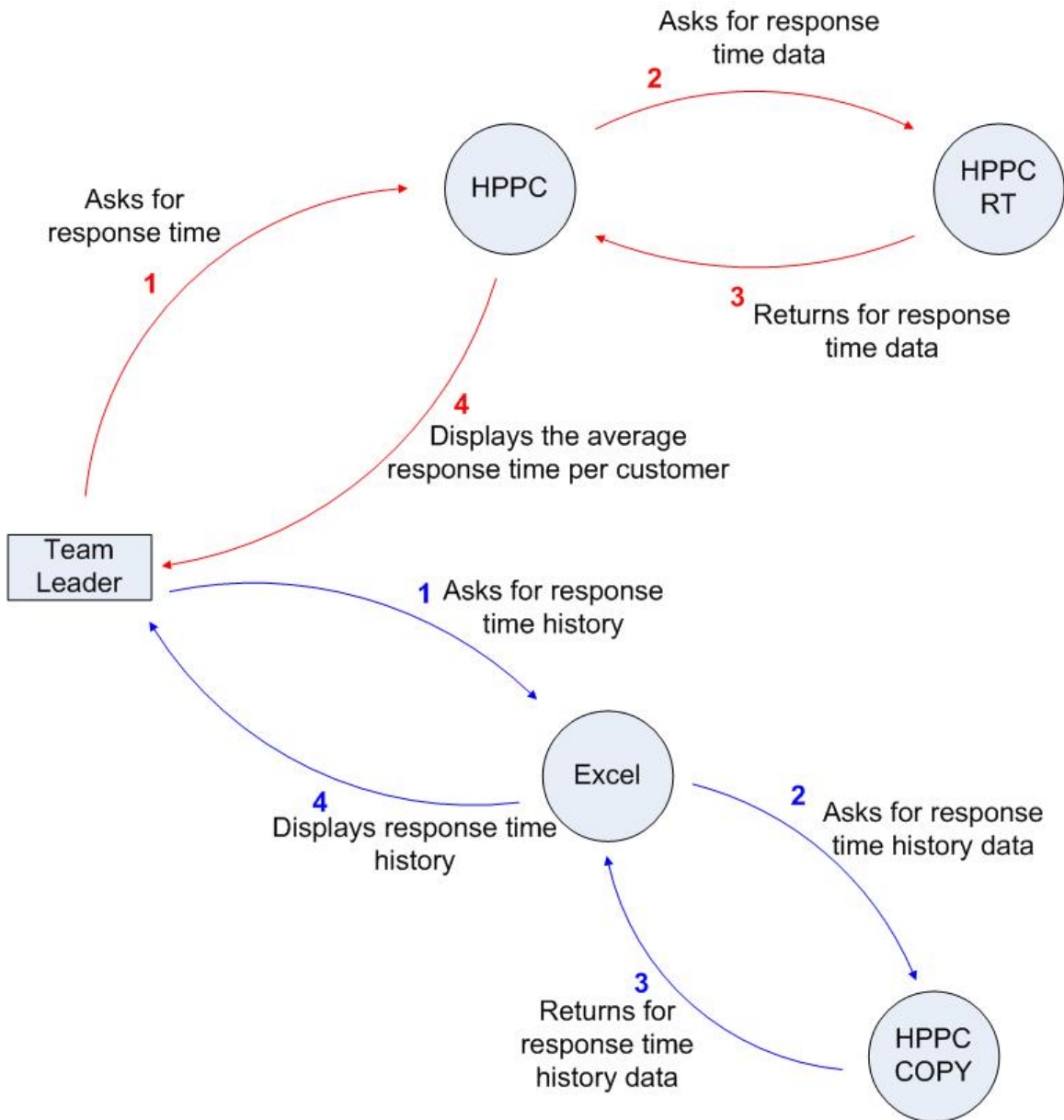
<b>Name</b>	<b>: Bjørn Henrik Pedersen</b>
<b>Title</b>	<b>: Escalation Manager</b>
<b>1. How is it possible to discover that you are not within target of an agreed SLA?</b>	
<ul style="list-style-type: none"> <li>- Response time from the HPPC phone system</li> <li>- Response time history from Excel</li> <li>- Collects number of new incidents and number of incidents closed in OSMQ</li> <li>- Compare these results with the SLAs manually. The Escalation Manager has to know each SLA.</li> </ul>	
<b>2. How do you monitor the response time?</b>	
<ul style="list-style-type: none"> <li>- Get the average response time per customer from HPPC</li> <li>- Uses Excel to see times per agent, history and team. Total delivery.</li> </ul>	
<b>3. How do you monitor the agents?</b>	
<ul style="list-style-type: none"> <li>- Real time monitoring from the HPPC</li> <li>- History could be collected from HPPC-Reporter</li> </ul>	
<b>4. How do you monitor open incidents?</b>	
<ul style="list-style-type: none"> <li>- Collects reports from OSMQ</li> <li>- Collects from Website</li> </ul>	
<b>5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?</b>	
<ul style="list-style-type: none"> <li>- Gets all answered calls from Excel (could use HPPC-reporter, but takes to much time)</li> <li>- Gets all Wrap-up reasons from HPPC- Reporter</li> <li>- Gets all reported incidents from OSMQ</li> <li>- Calculates the deviation manually</li> </ul>	
<b>6. How do you measure the workload and plan how many that is going to work?</b>	
<ul style="list-style-type: none"> <li>- Gets reports from HPPC about response time sequences by SQL inquiry <ul style="list-style-type: none"> <li>o Manually compares response times and sequences</li> </ul> </li> <li>- Excel is used to fined the number of incoming calls</li> <li>- Uses work plan and compare with earlier events</li> </ul>	

## Appendix D: Contextual Design – Work modeling

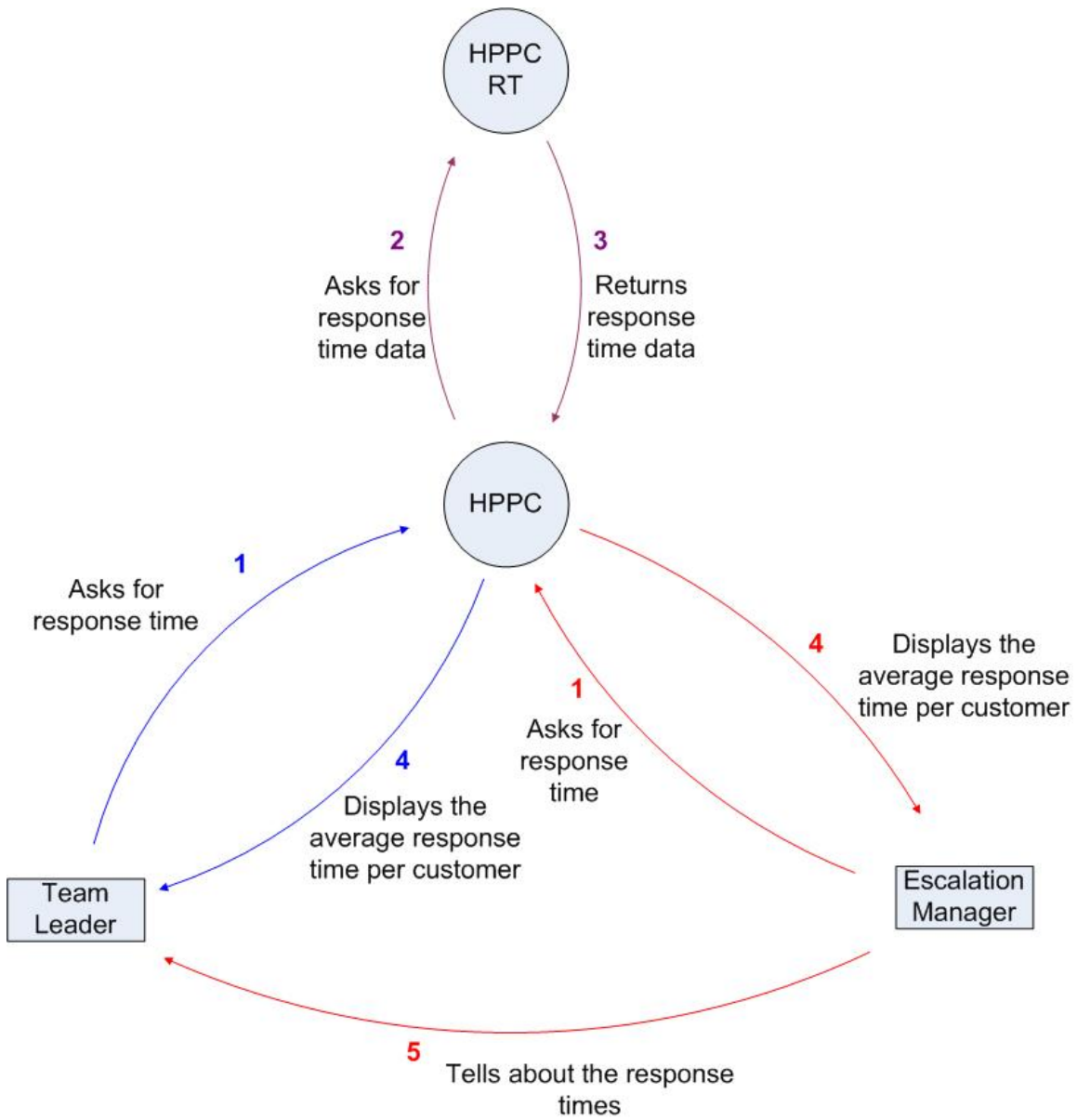
Here we present the models created based on the Contextual Inquiry in appendix C.

**Name** : Anders Refsdahl  
**Title** : Team Leader Team 4

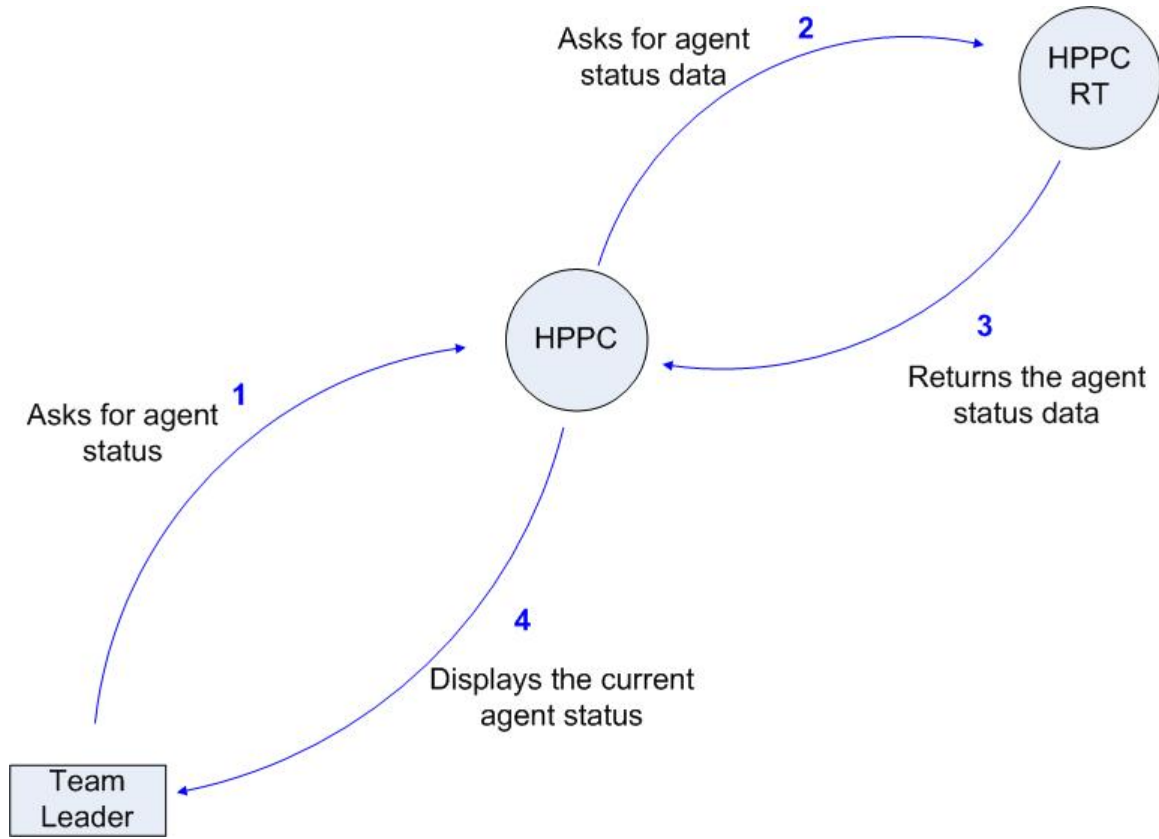
### 1. How is it possible to discover that you are not within target of an agreed SLA?



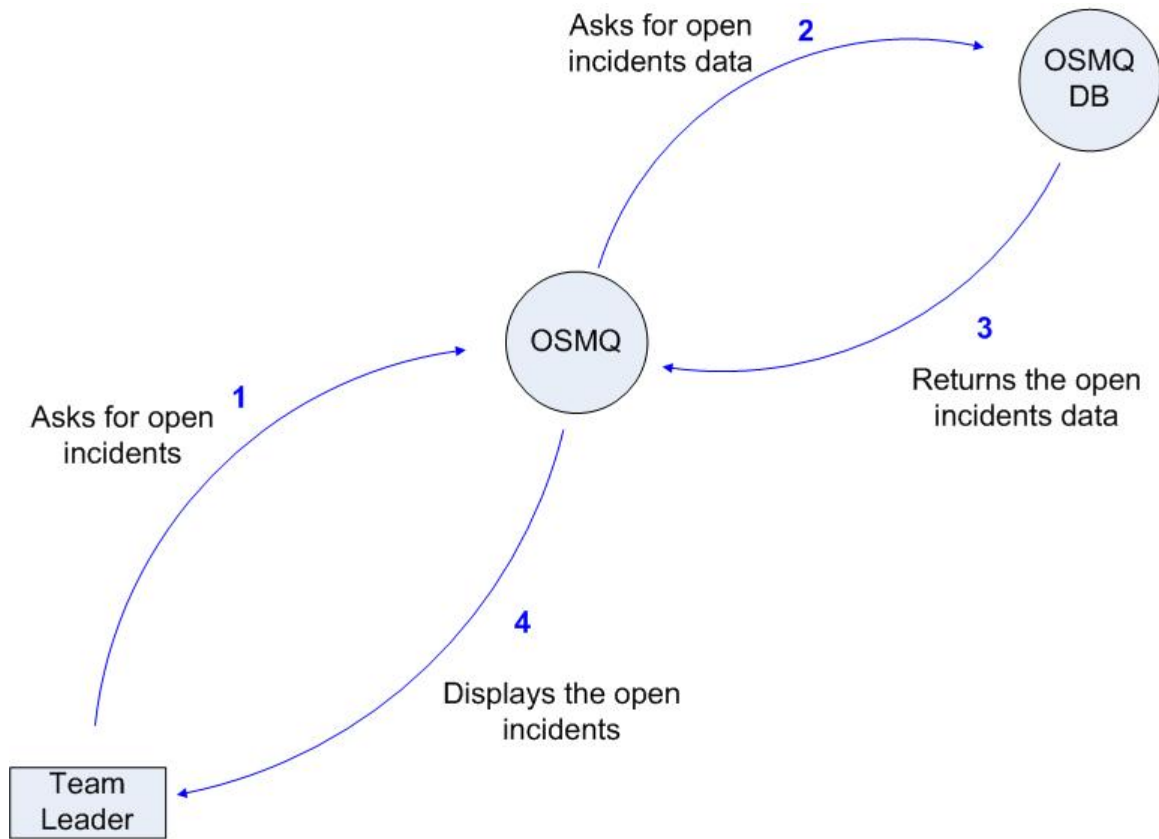
## 2. How do you monitor the response time?



### 3. How do you monitor the agents?

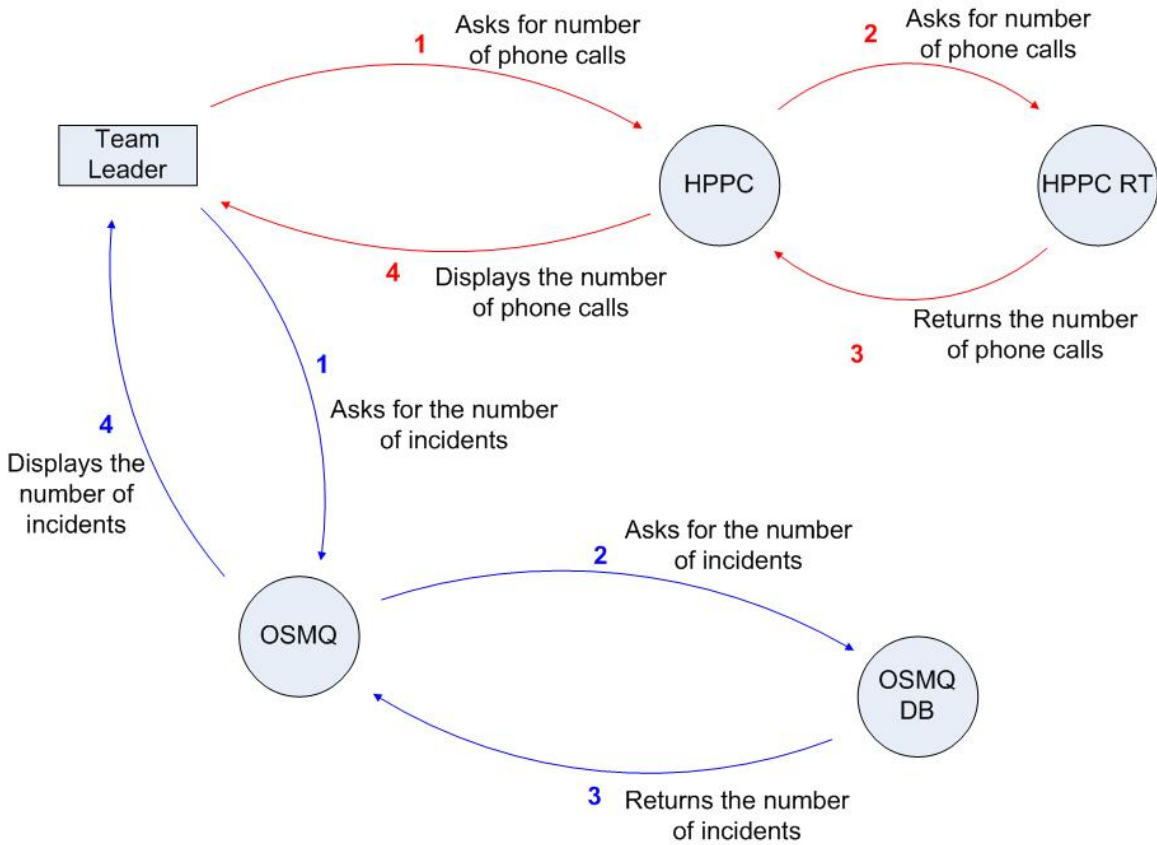


#### 4. How do you monitor open incidents?

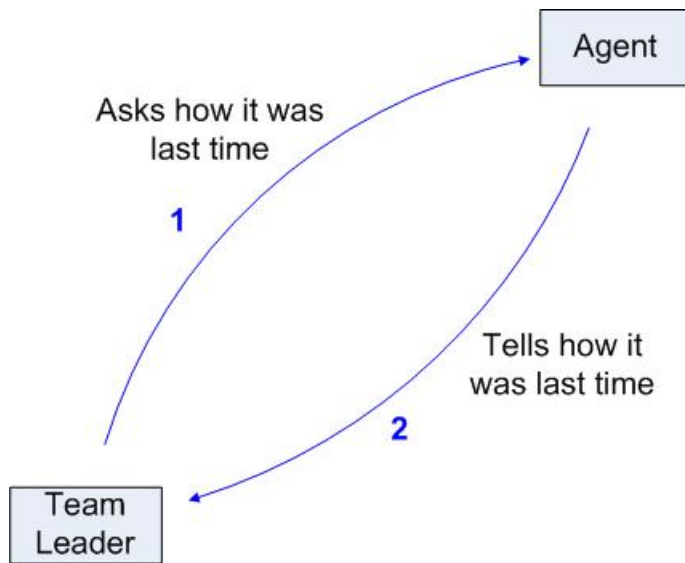




### 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?

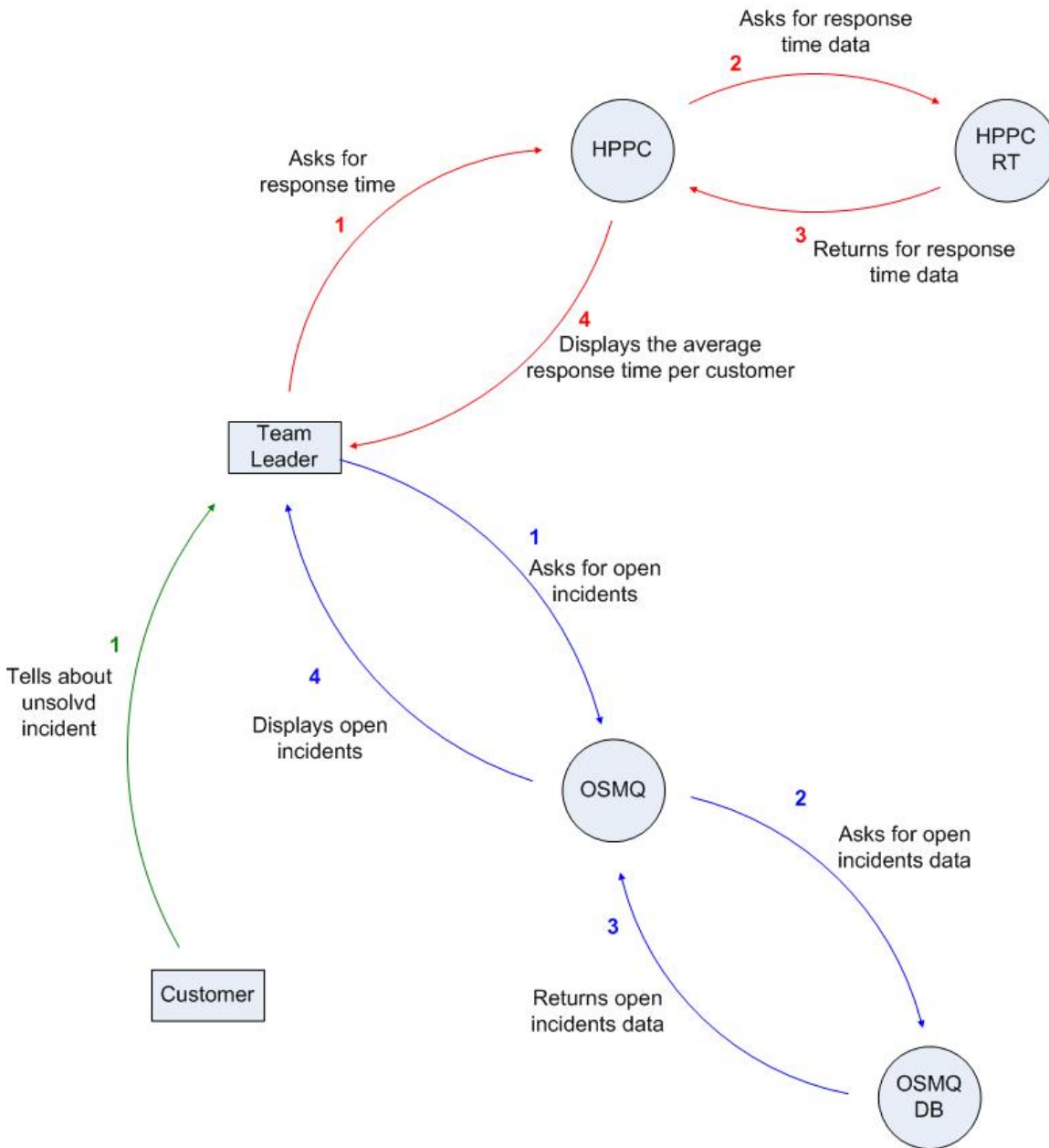


### 6. How do you measure the workload and plan how many that is going to work?

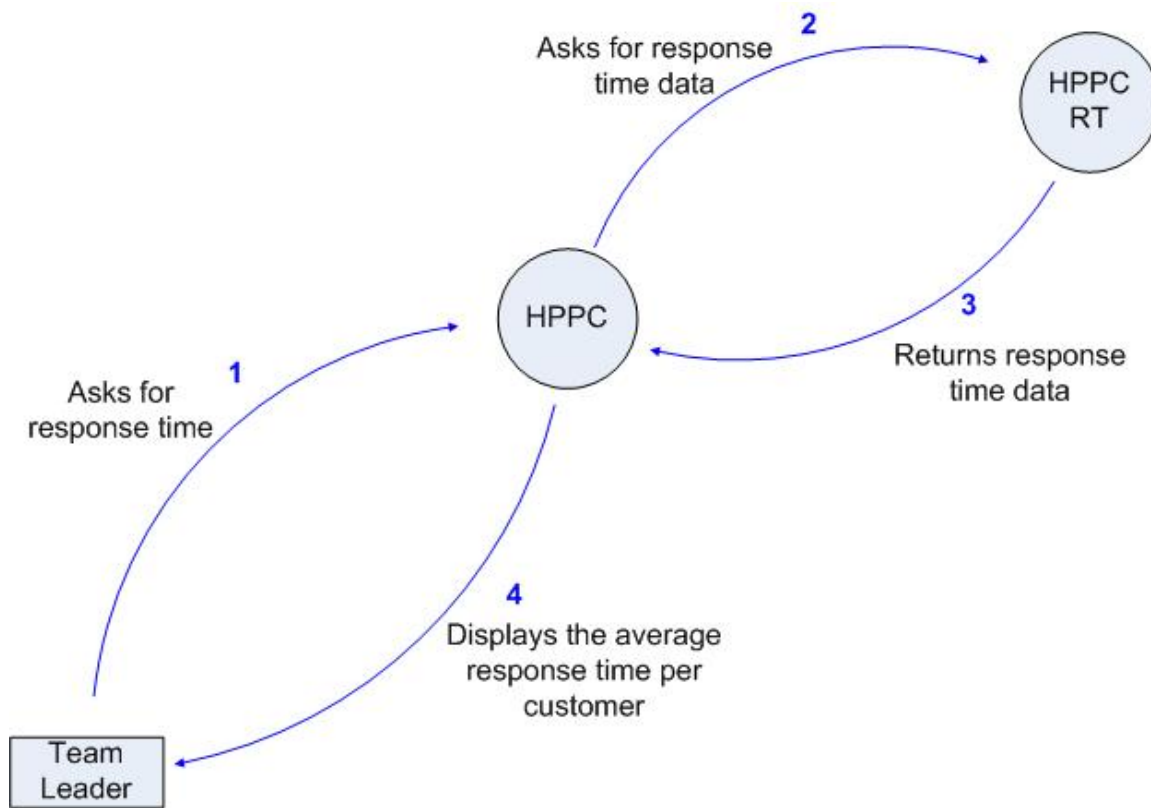


**Name** : Andreas Watz  
**Title** : Team Leader Team 5

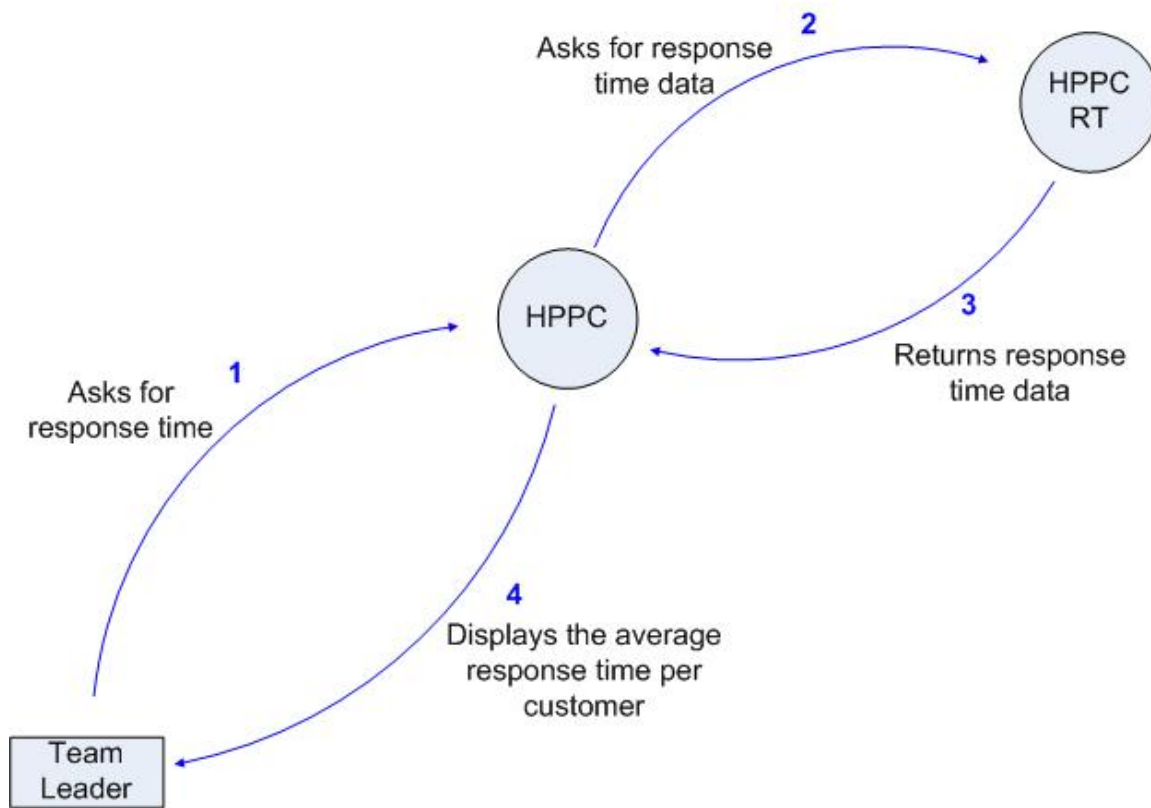
**1. How is it possible to discover that you are not within target of an agreed SLA?**



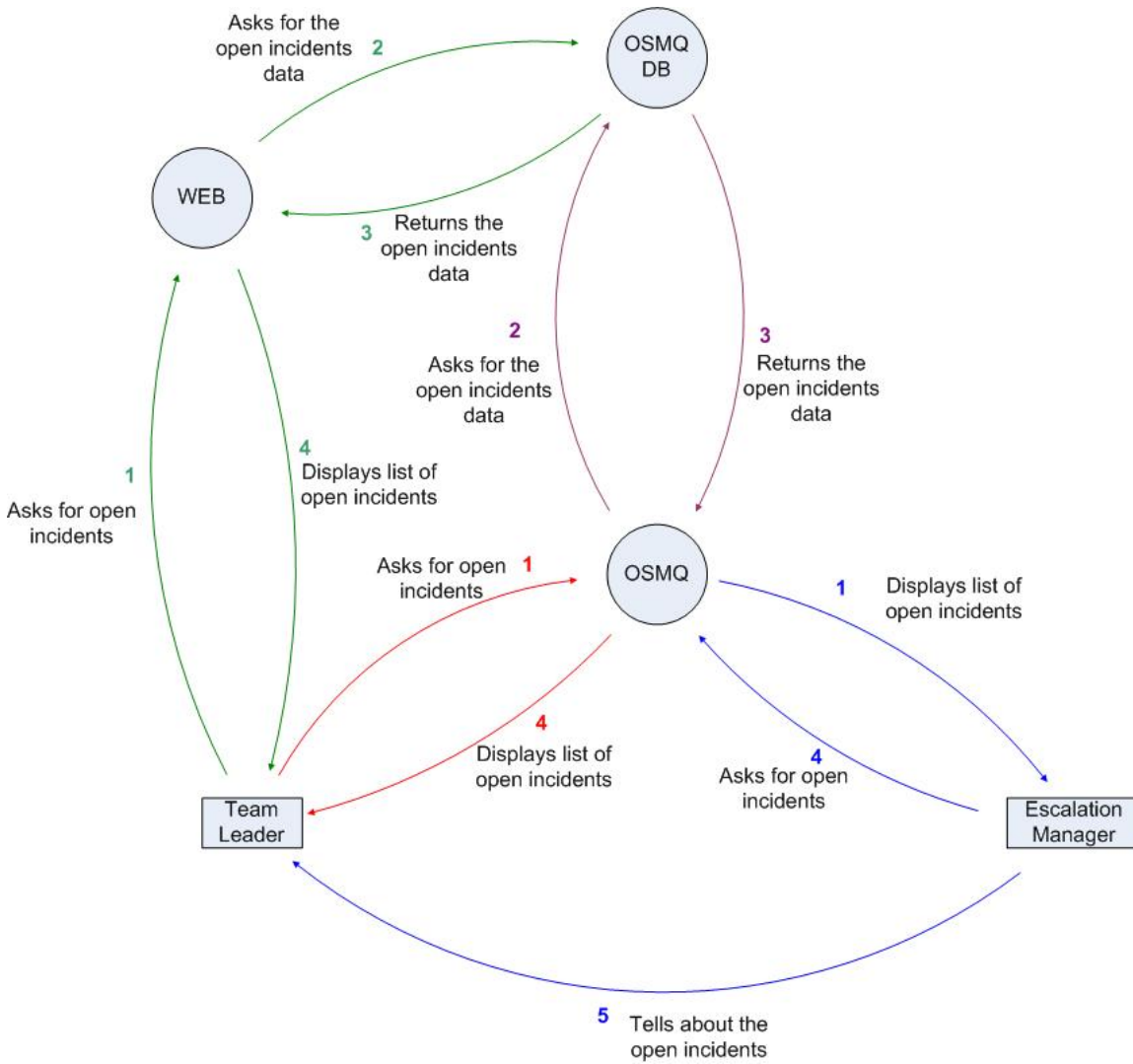
## 2. How do you monitor the response time?



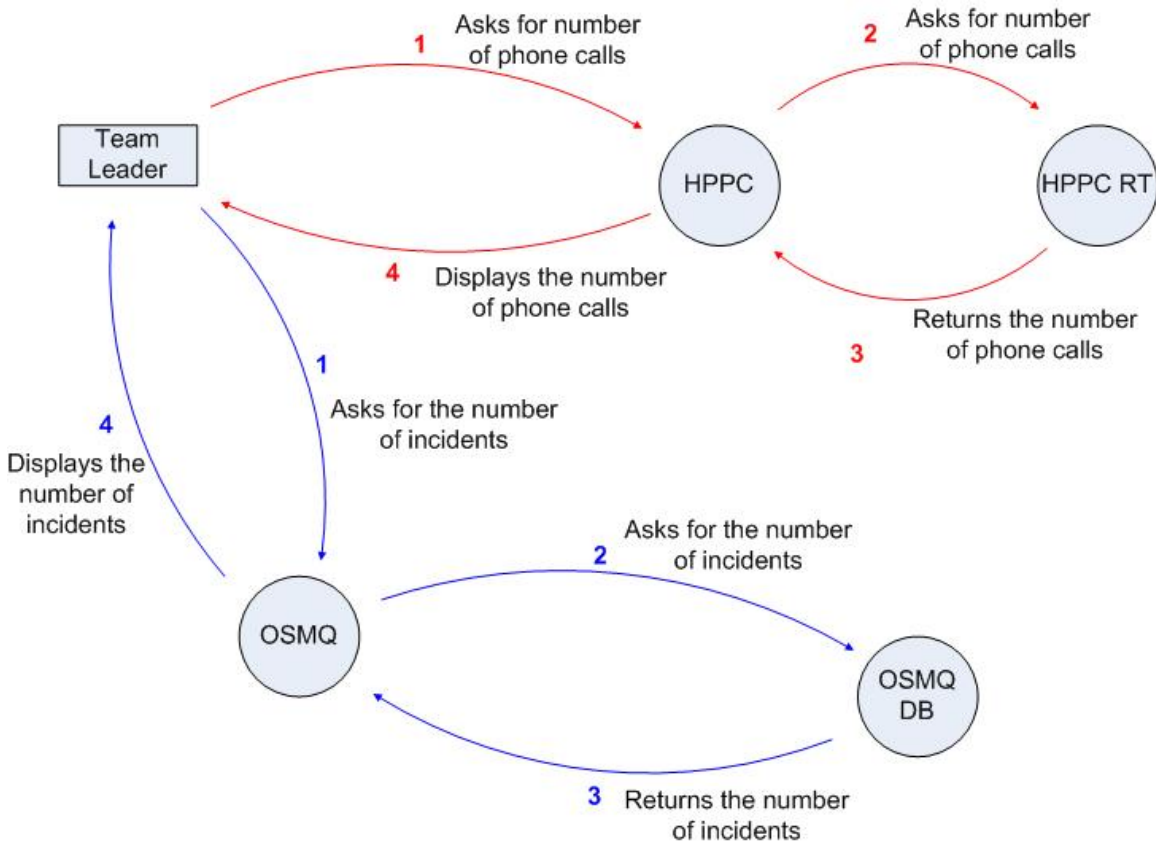
### 3. How do you monitor the agents?



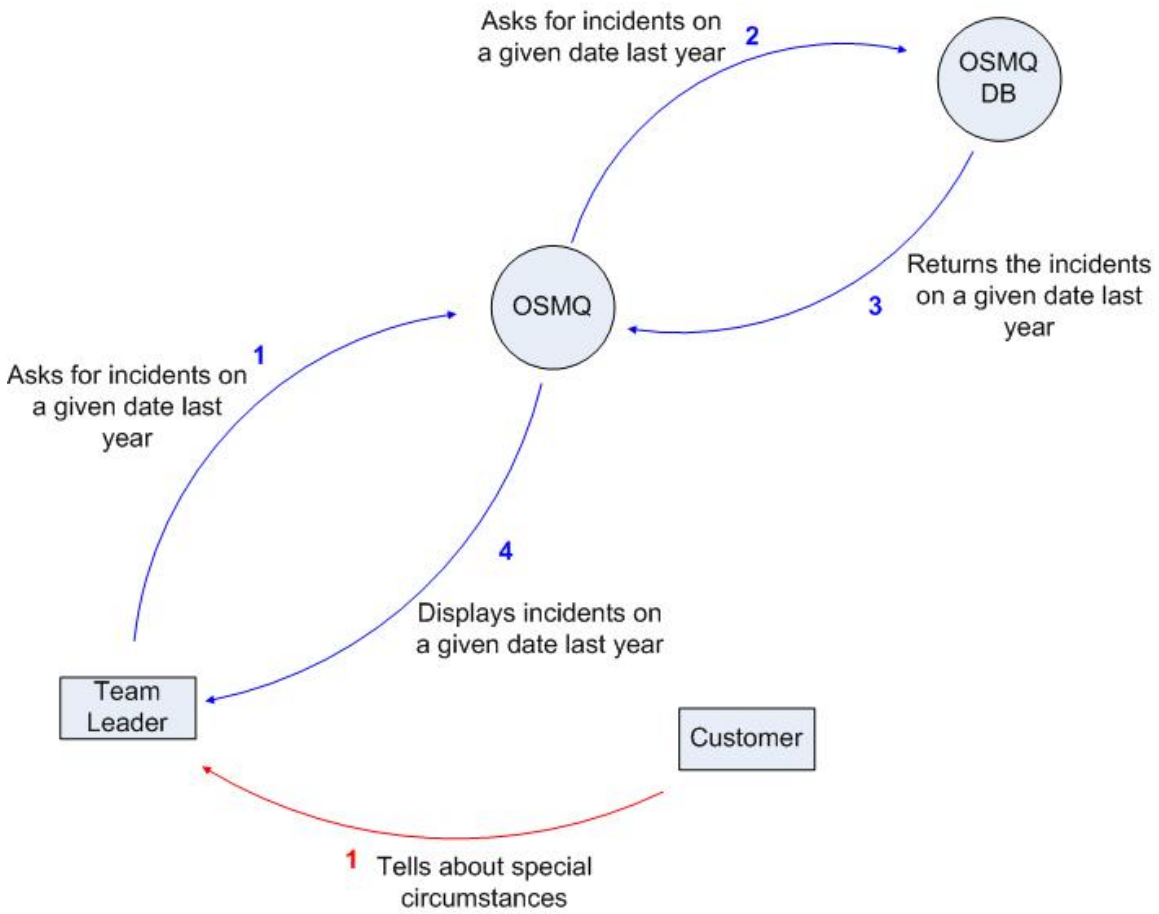
### 4. How do you monitor open incidents?



### 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?



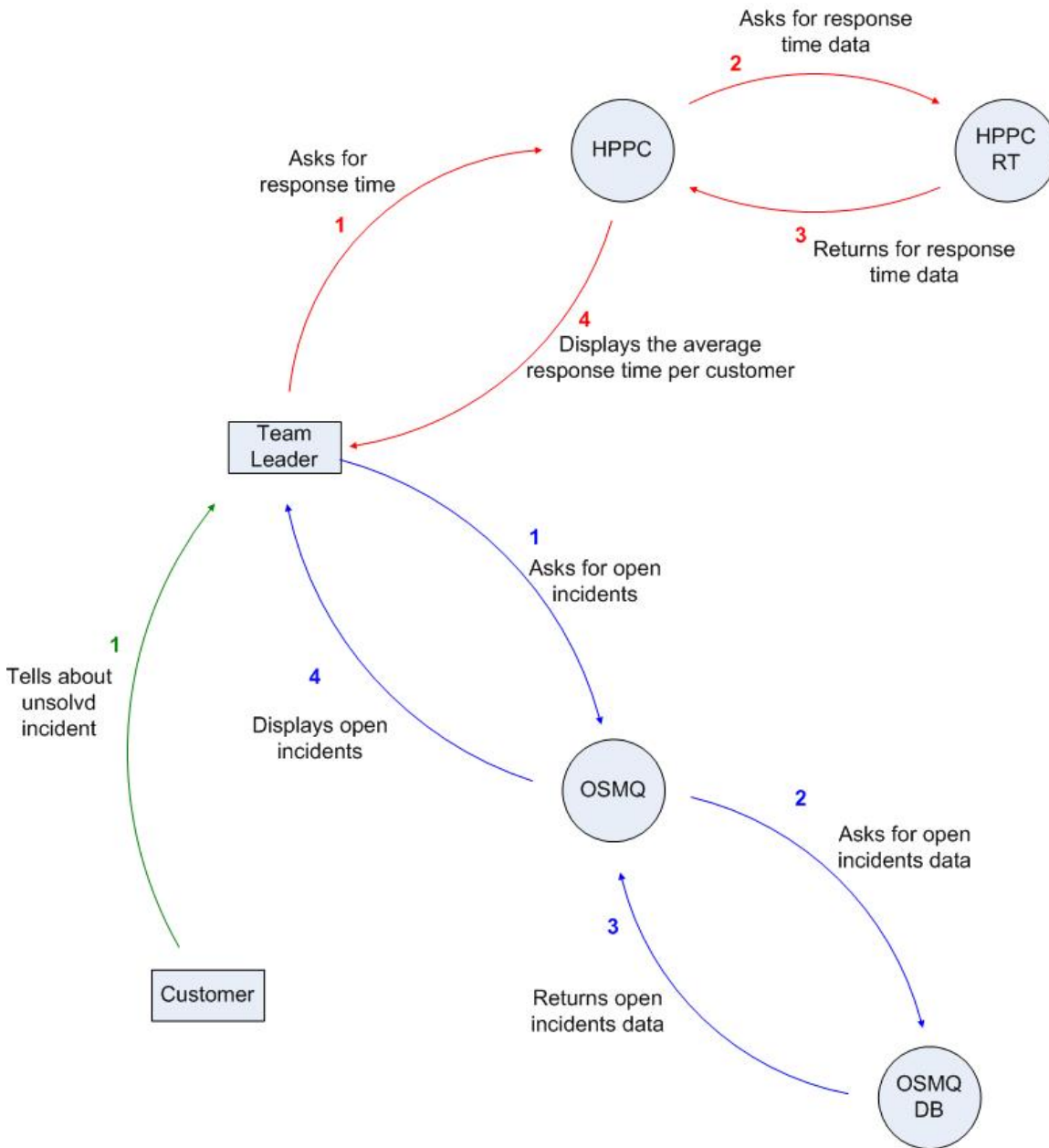
### 6. How do you measure the workload and plan how many that is going to work?



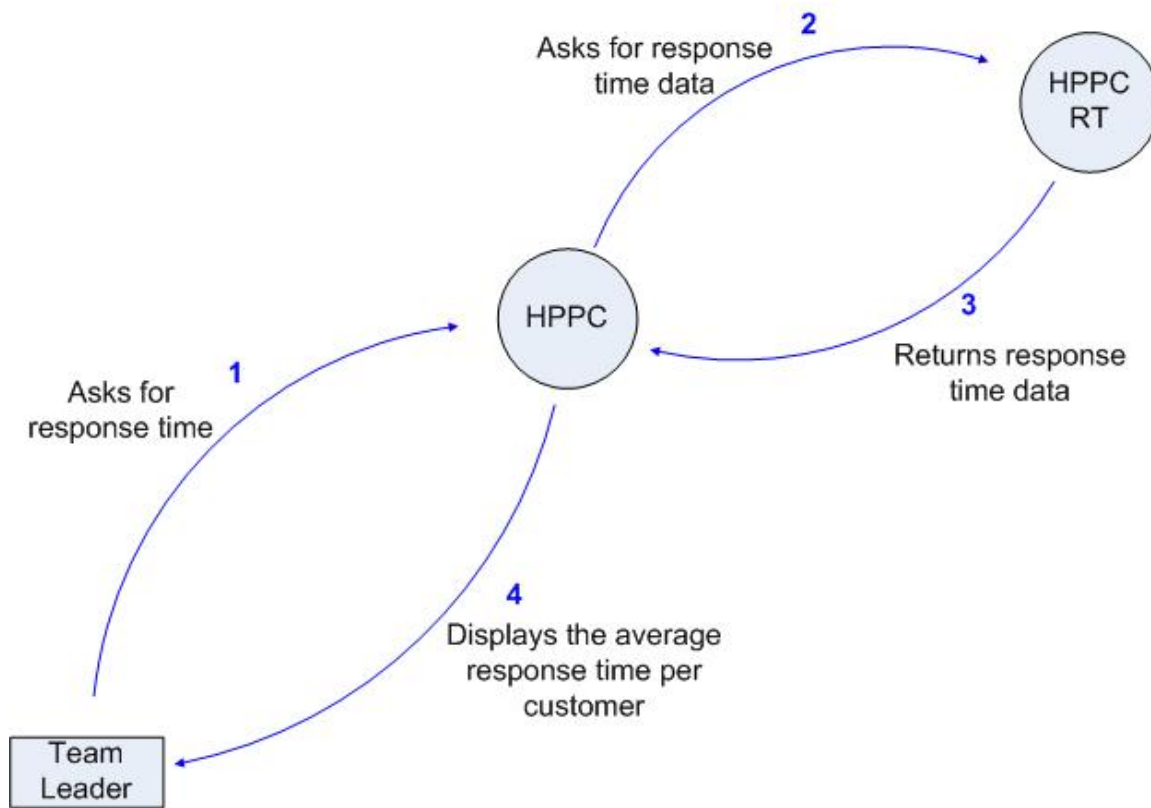


**Name** : Henrik Marteliusson  
**Title** : Team Leader Team 1 - Siemens

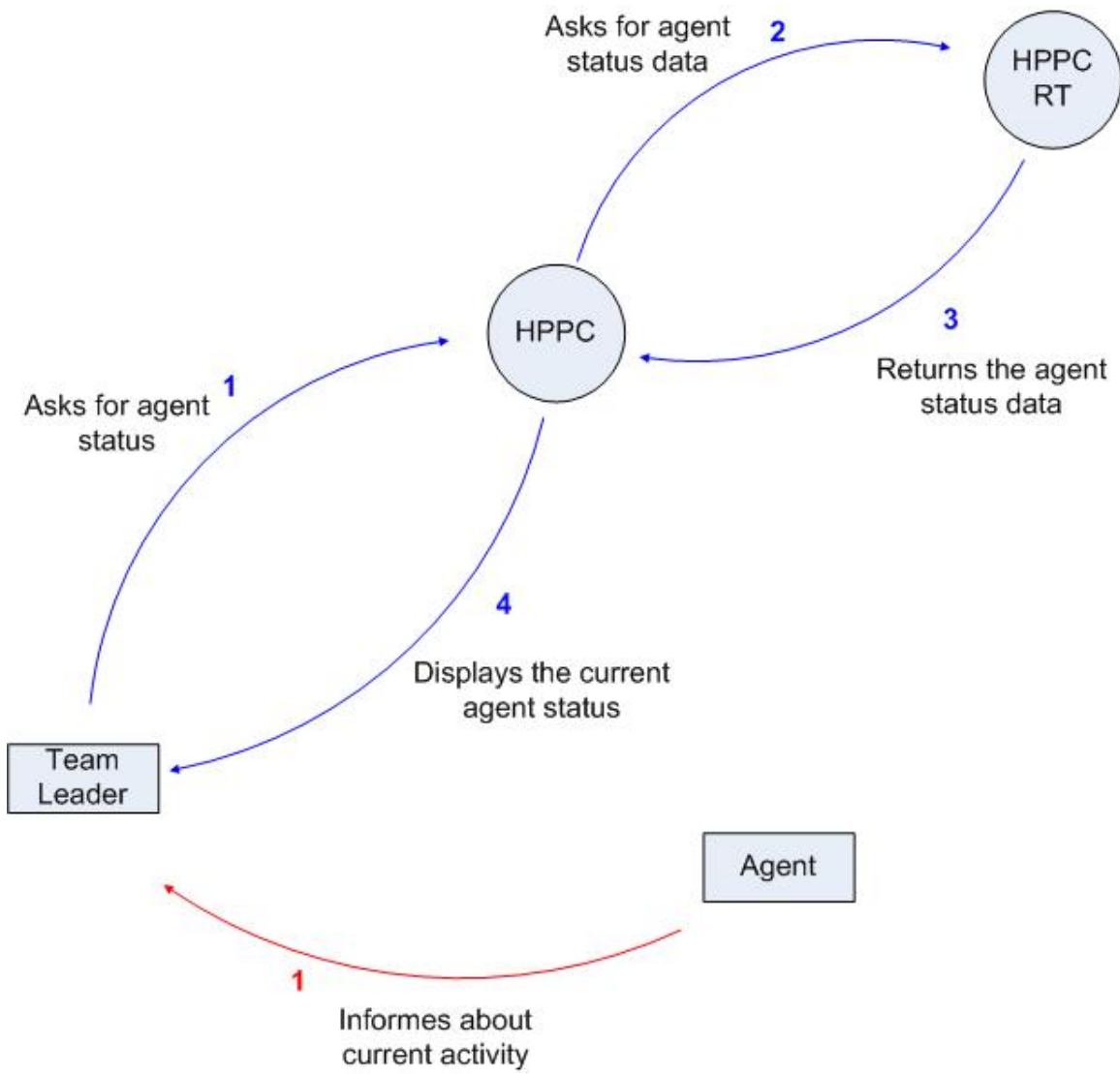
**1. How is it possible to discover that you are not within target of an agreed SLA?**



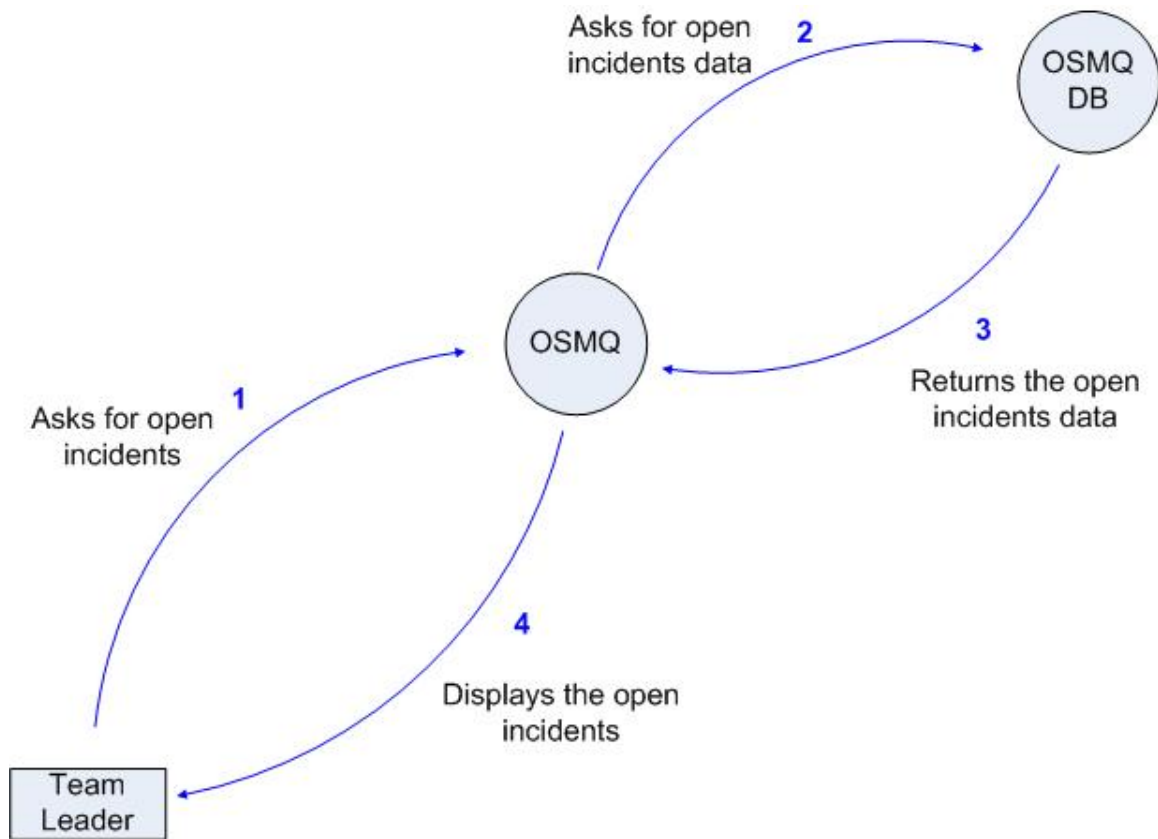
## 2. How do you monitor the response time?



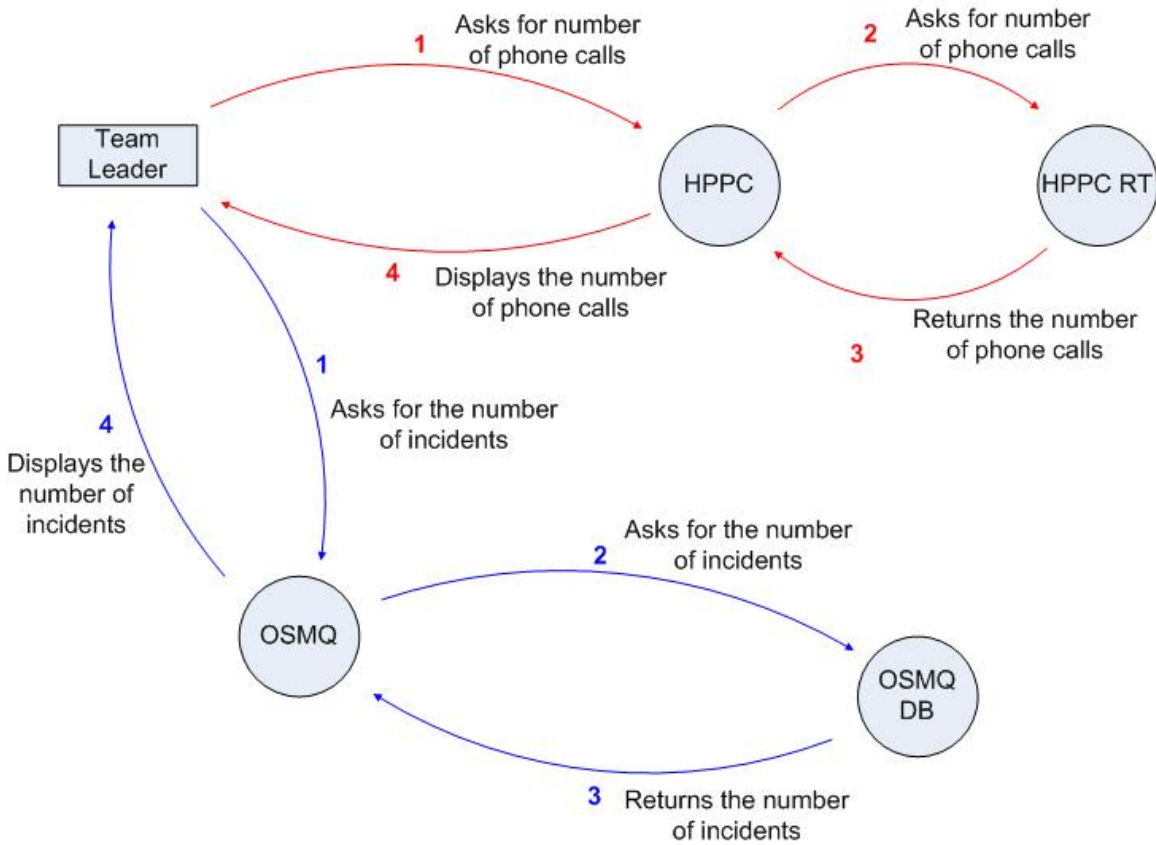
### 3. How do you monitor the agents?



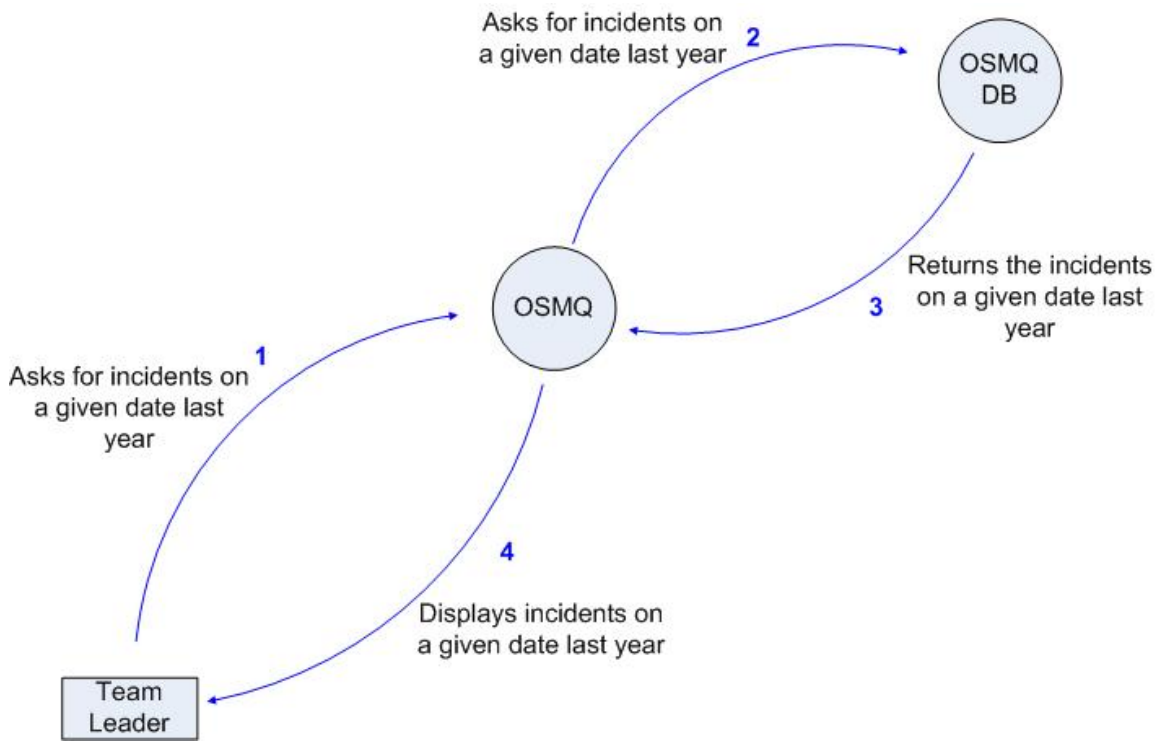
#### 4. How do you monitor open incidents?



### 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?

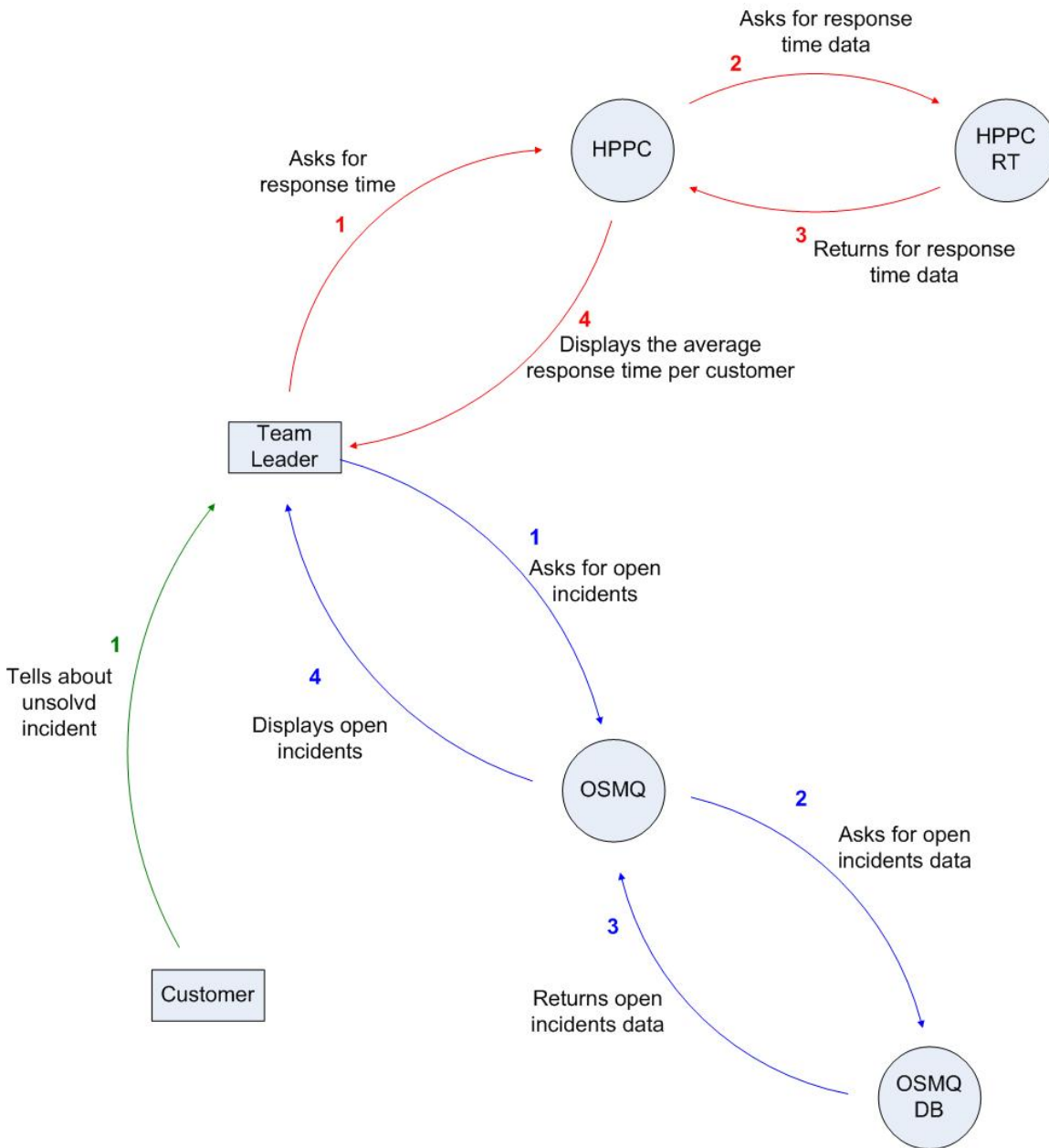


### 6. How do you measure the workload and plan how many that is going to work?

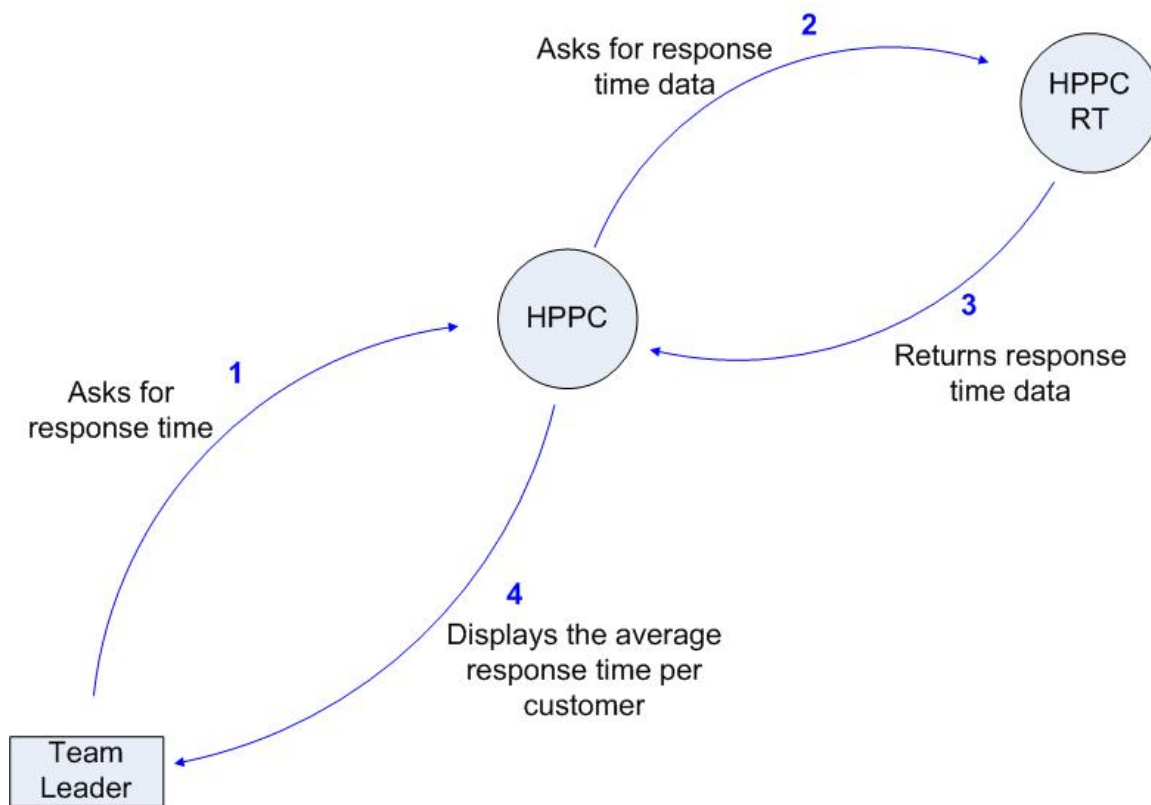


**Name** : Raymond Wiik  
**Title** : Team Leader Team 3 - Retail

**1. How is it possible to discover that you are not within target of an agreed SLA?**

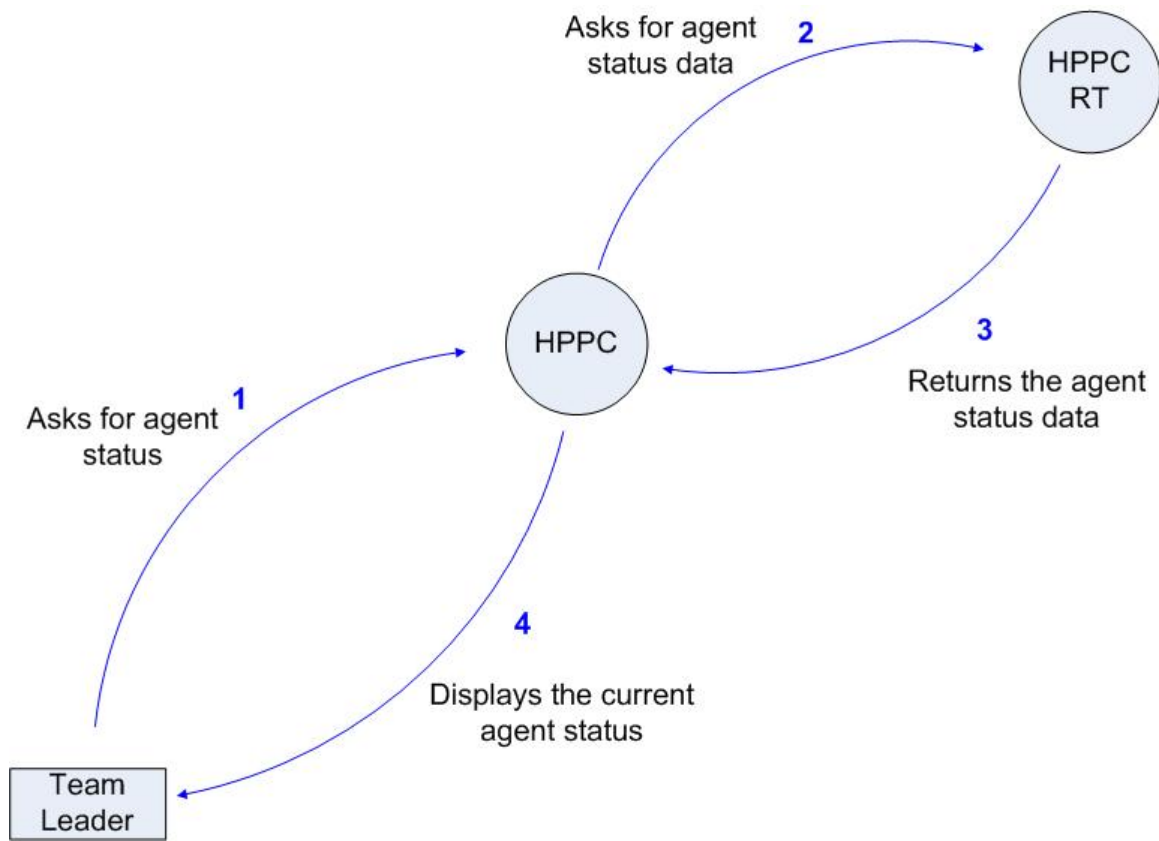


## 2. How do you monitor the response time?

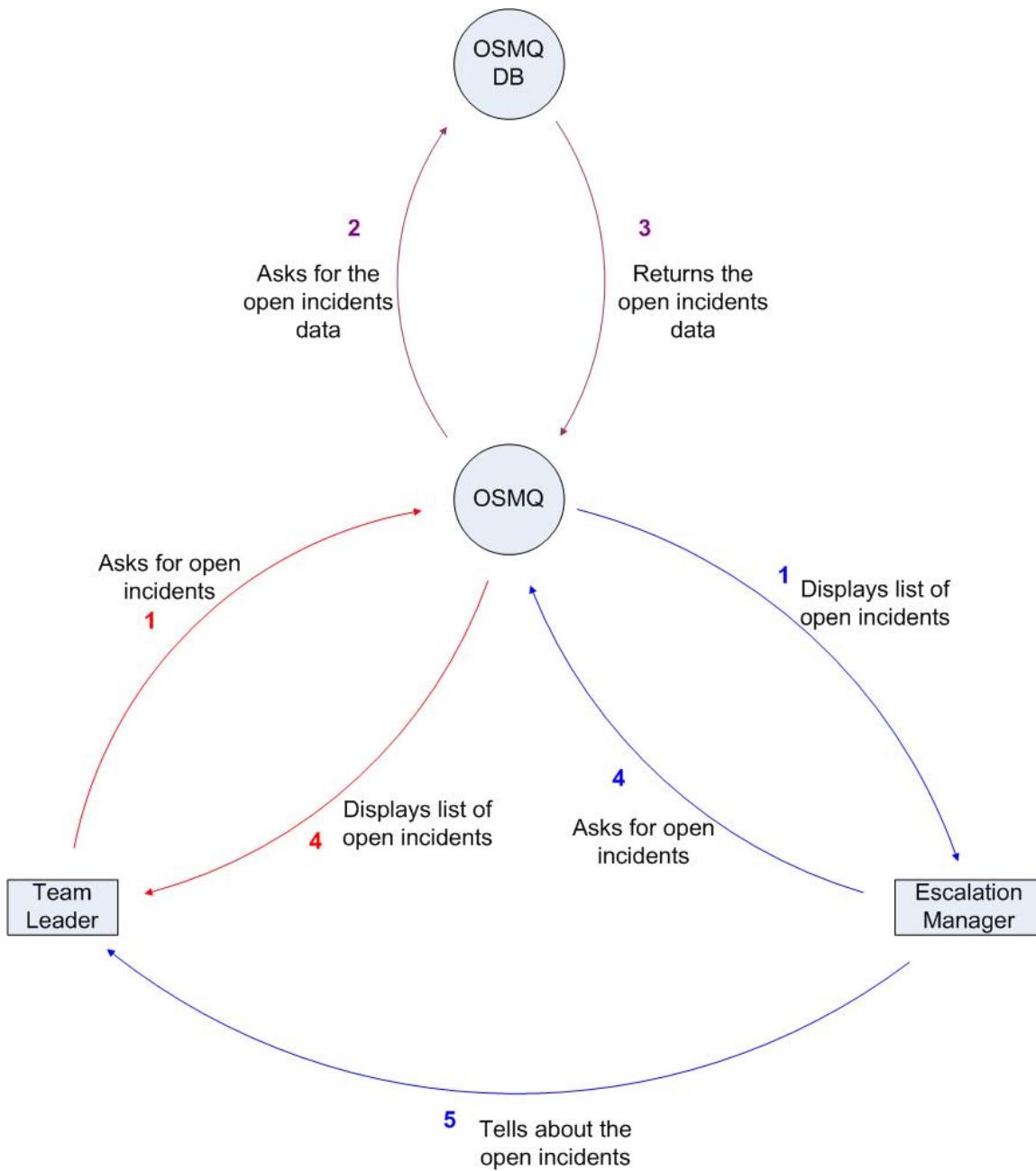




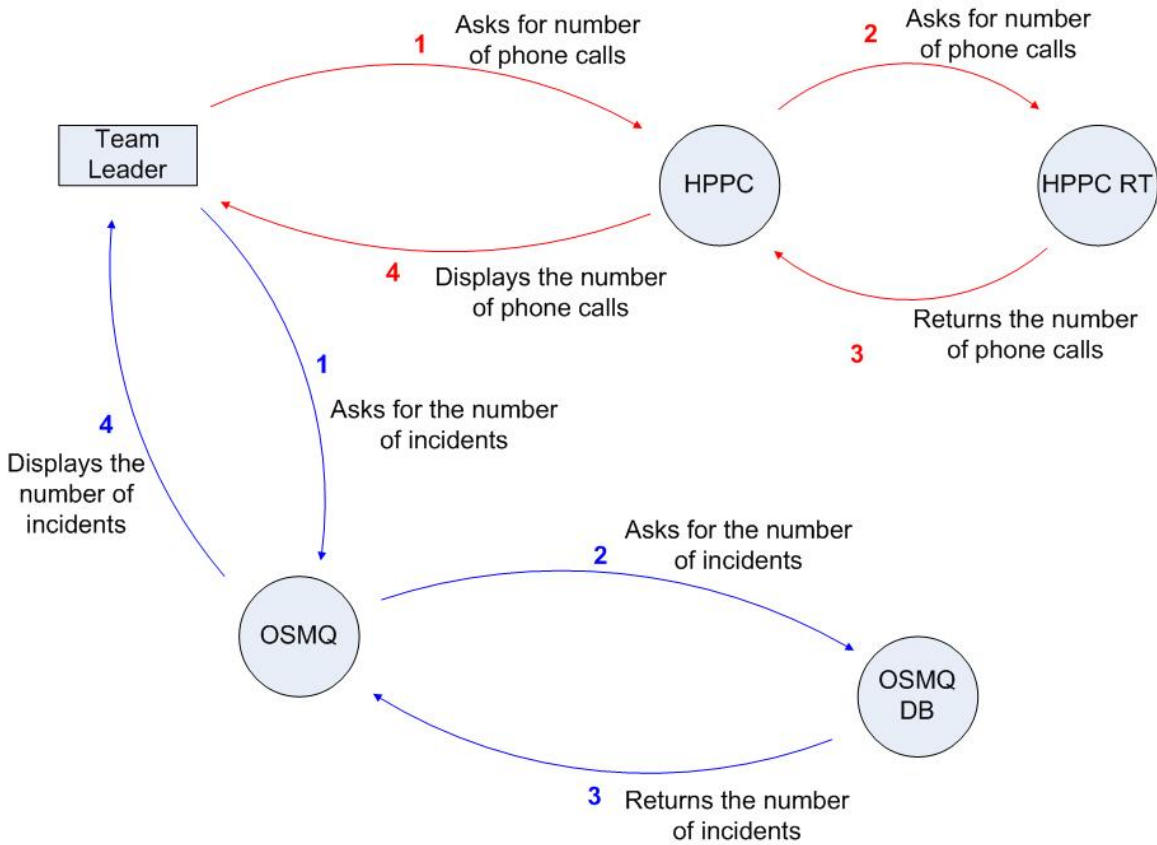
### 3. How do you monitor the agents?



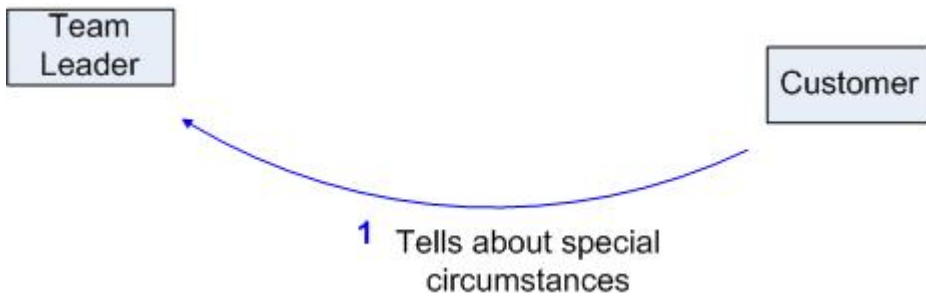
### 4. How do you monitor open incidents?



### 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?

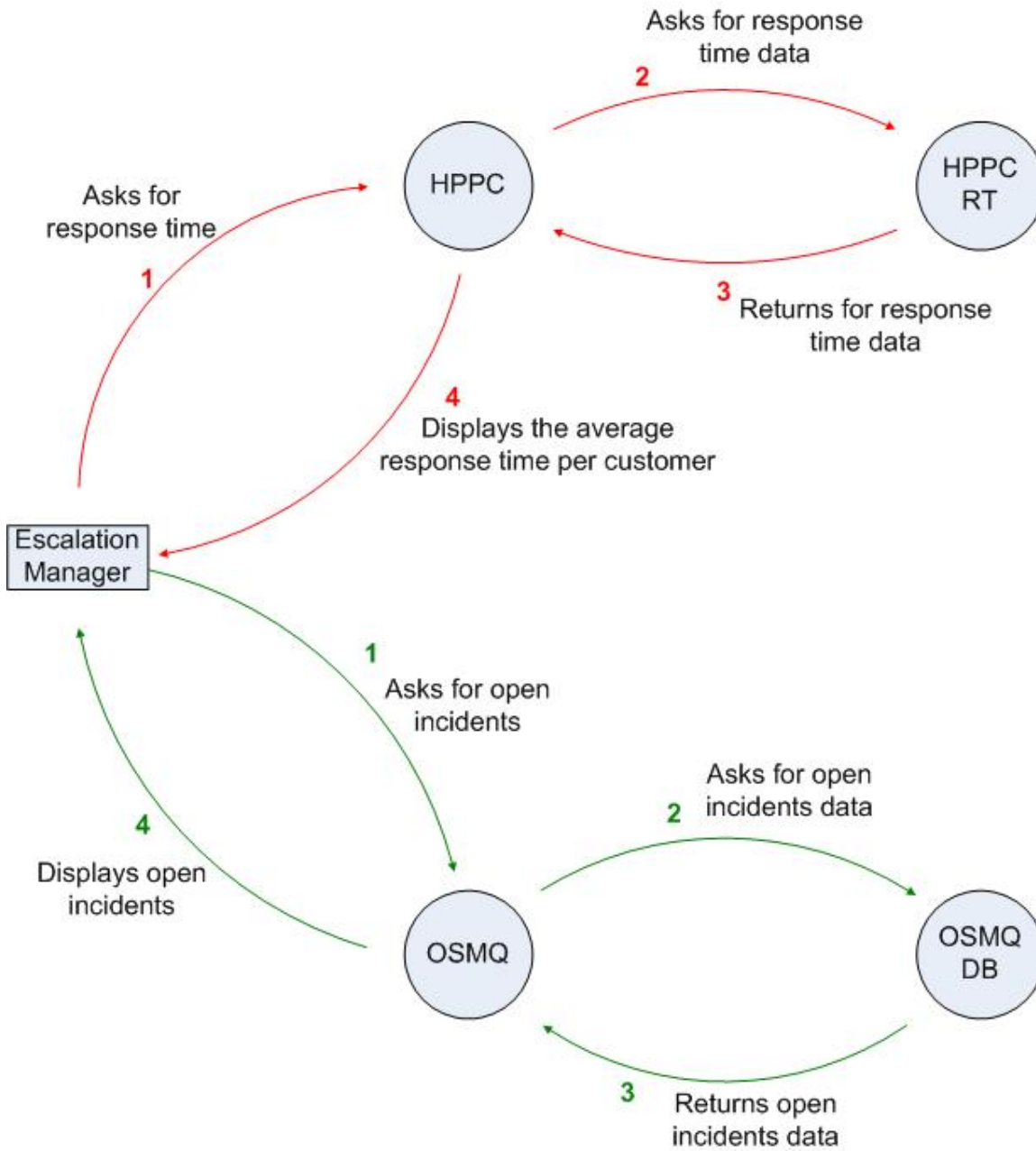


## 6. How do you measure the workload and plan how many that is going to work?

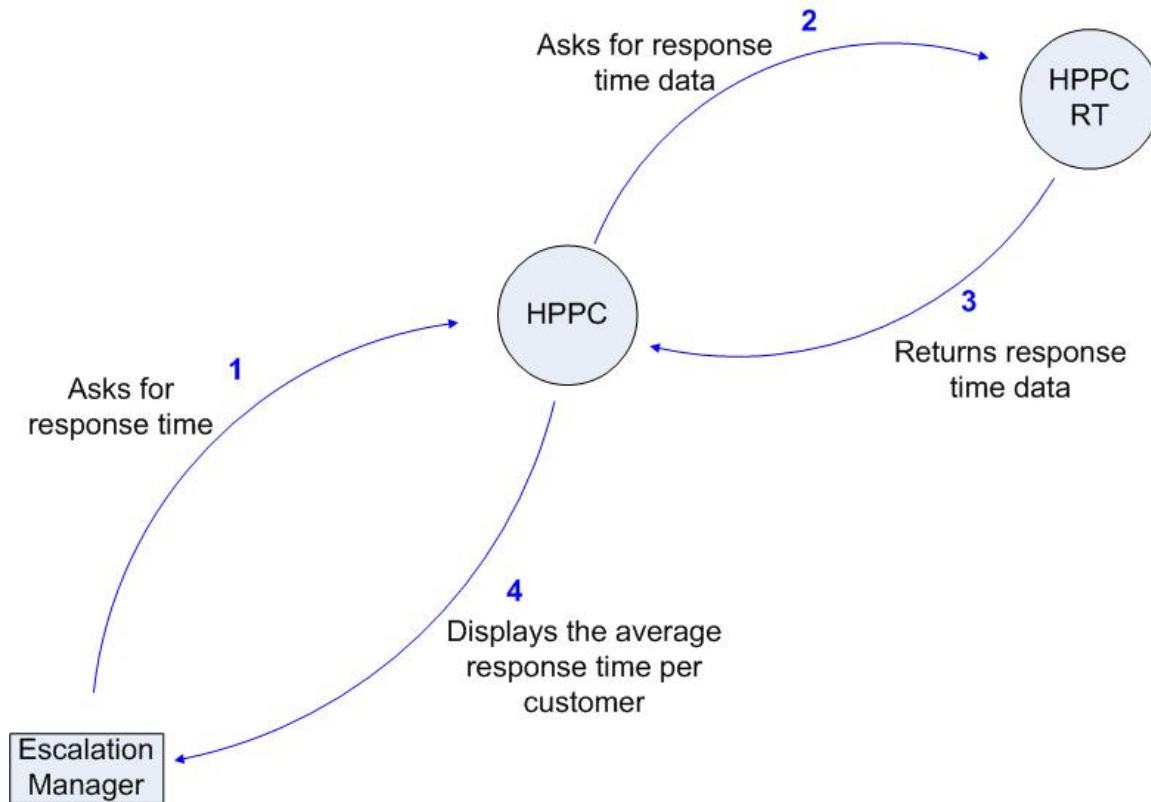


**Name** : Attila Kalmar  
**Title** : Escalation Manager, vicarious Department Manager

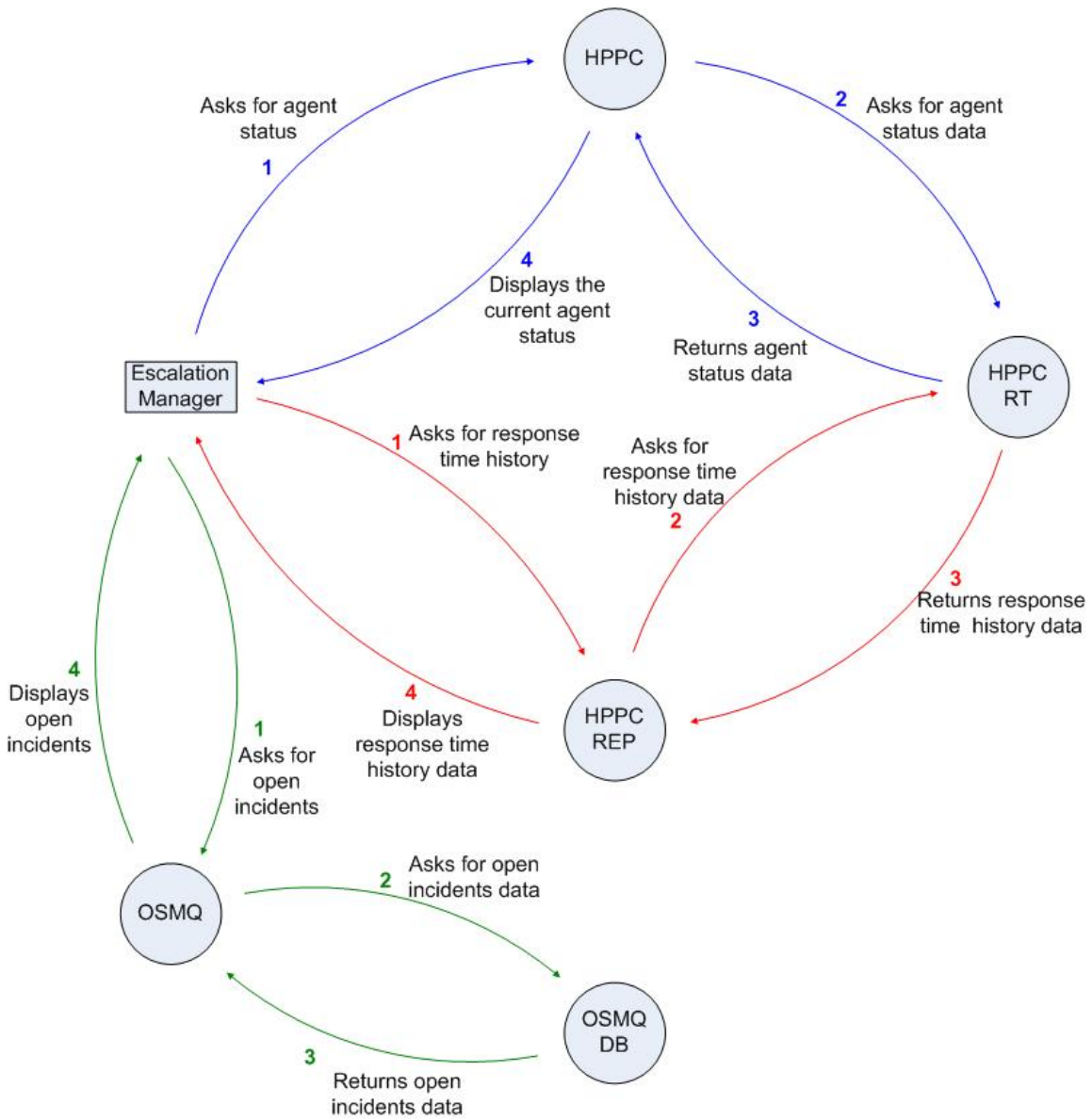
**1. How is it possible to discover that you are not within target of an agreed SLA?**



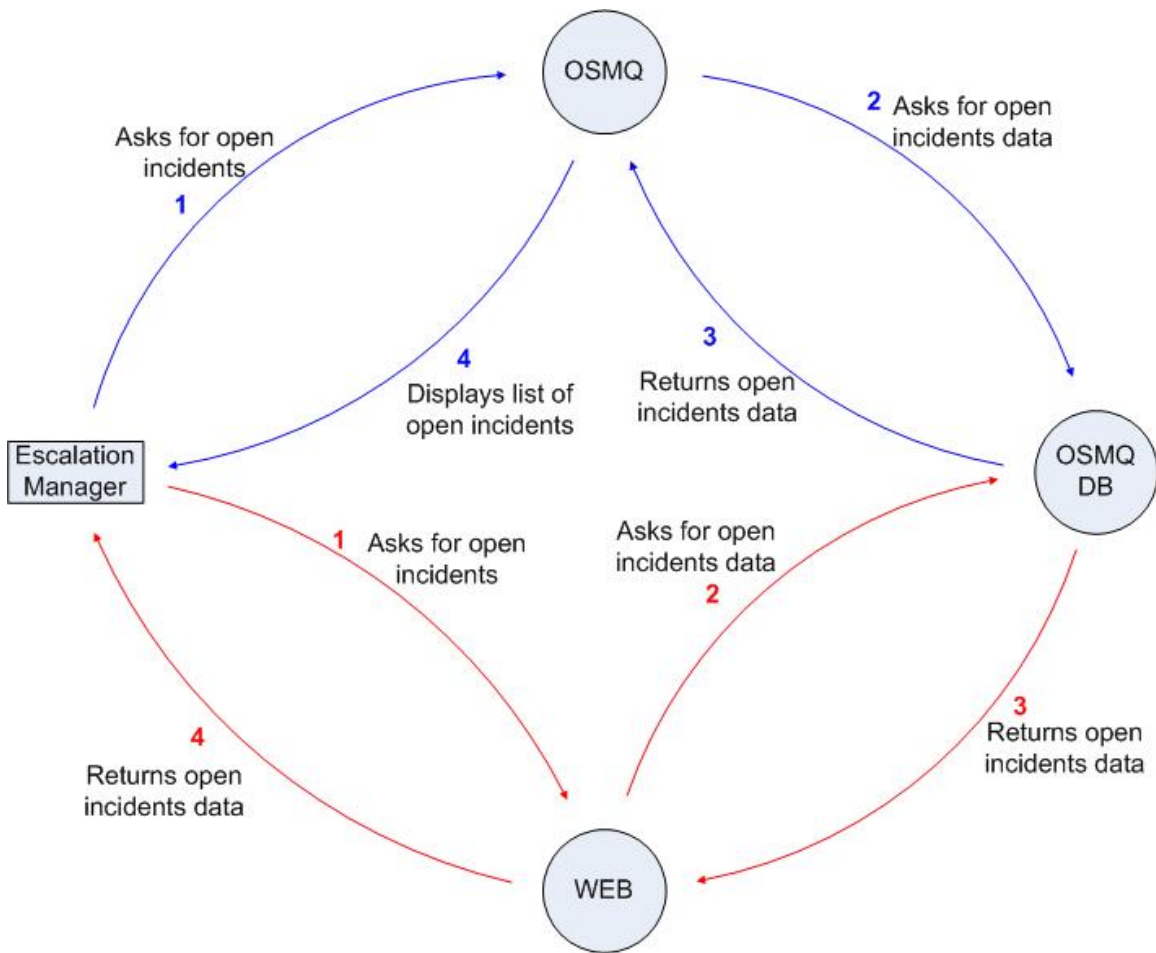
## 2. How do you monitor the response time?



### 3. How do you monitor the agents?

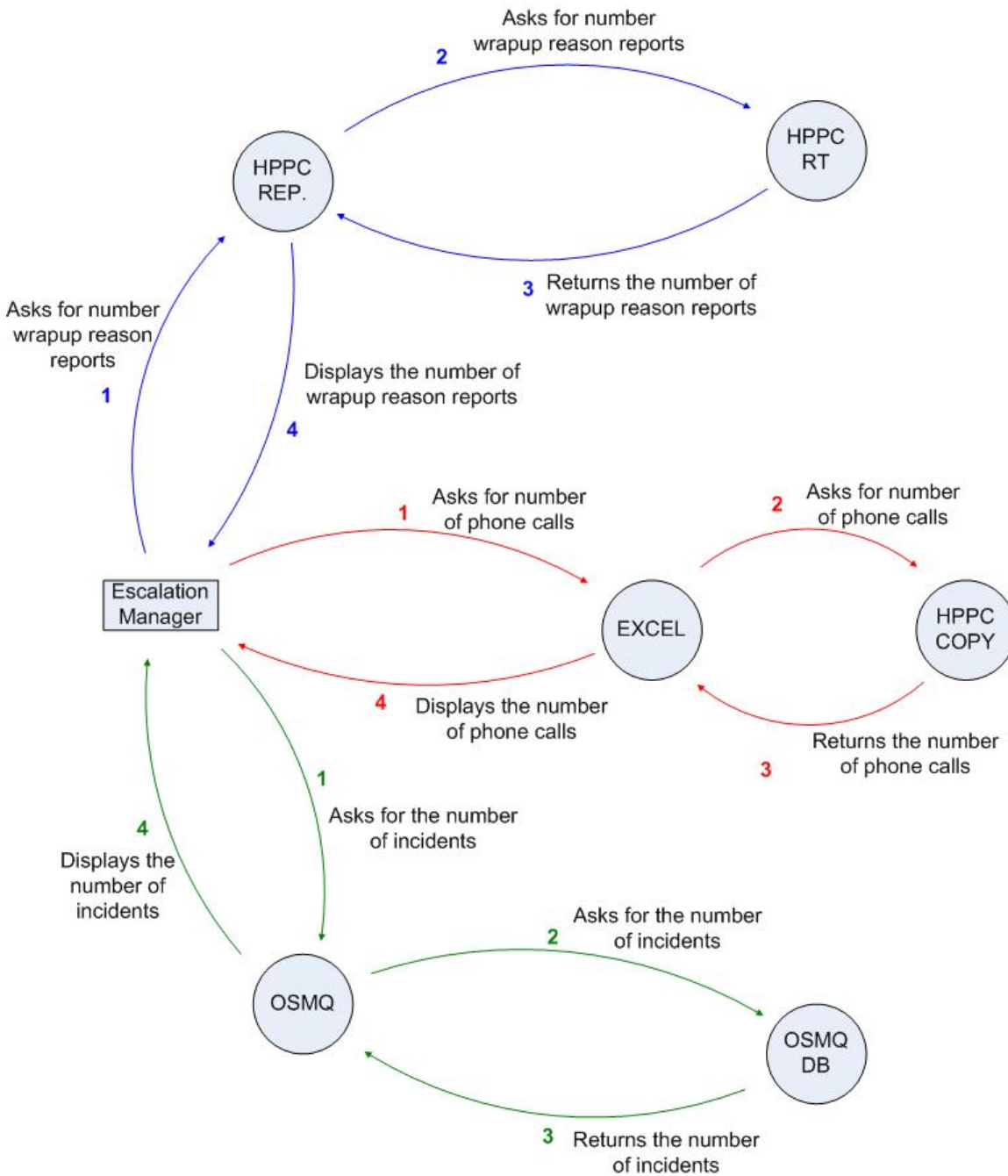


### 4. How do you monitor open incidents?

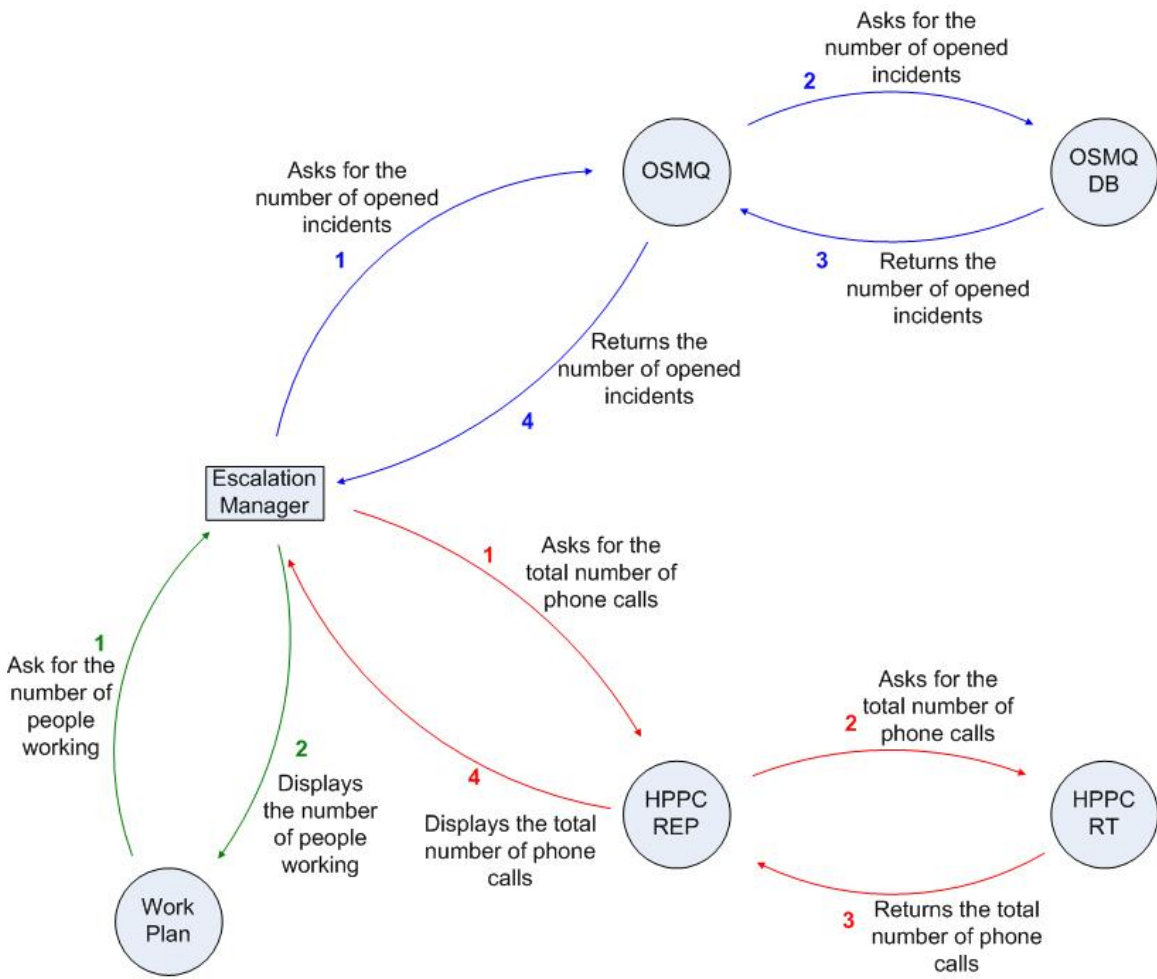




### 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?

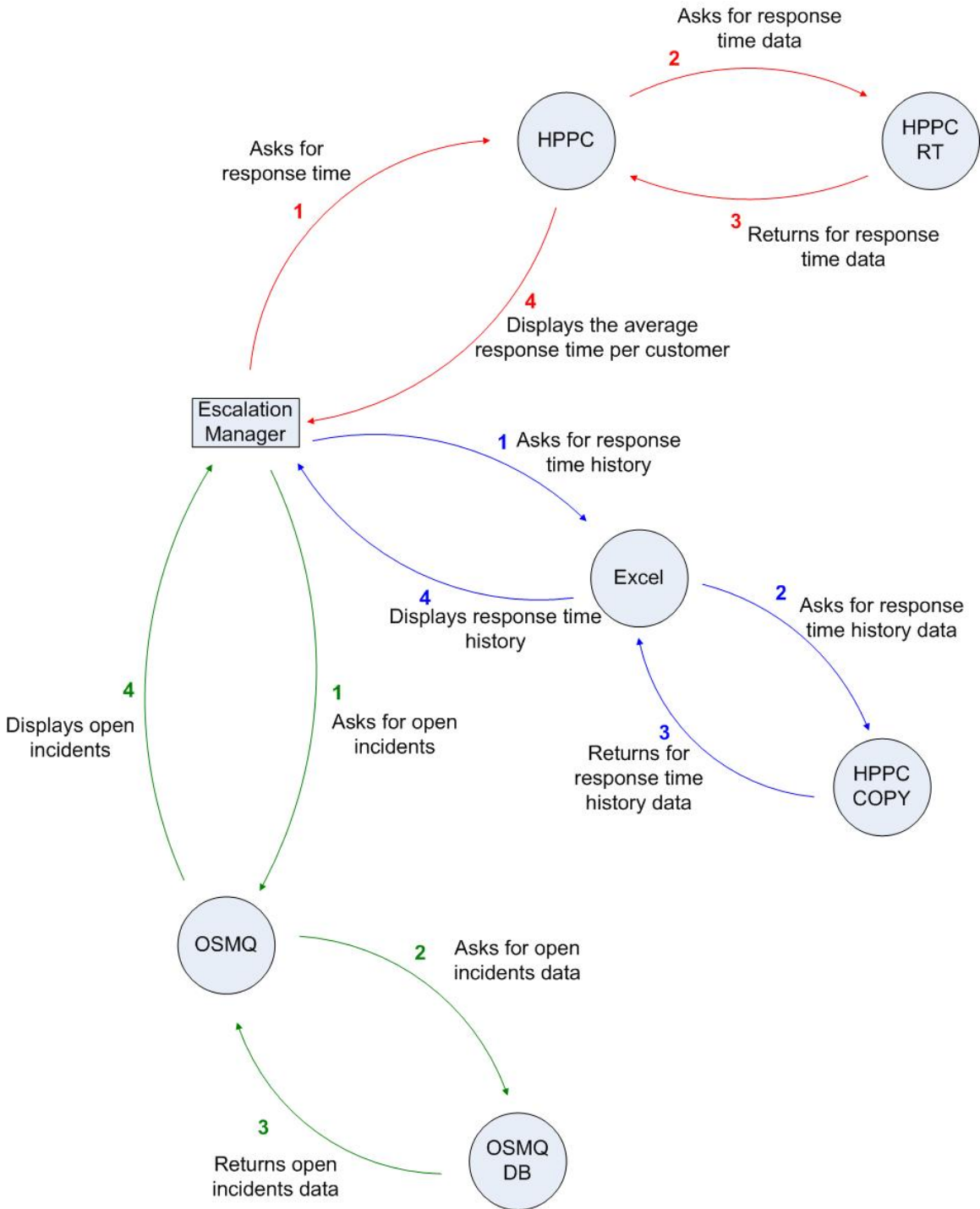


### 6. How do you measure the workload and plan how many that is going to work?

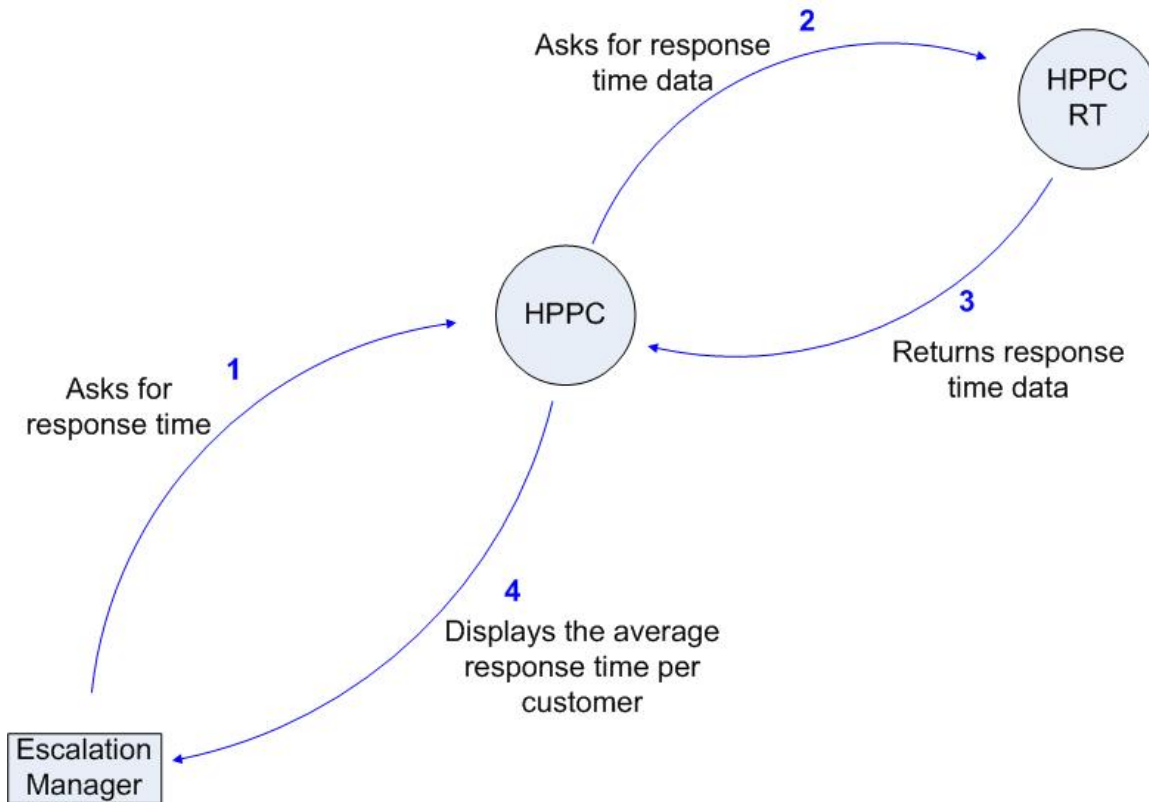


**Name** : Bjørn Henrik Pedersen  
**Title** : Escalation Manager

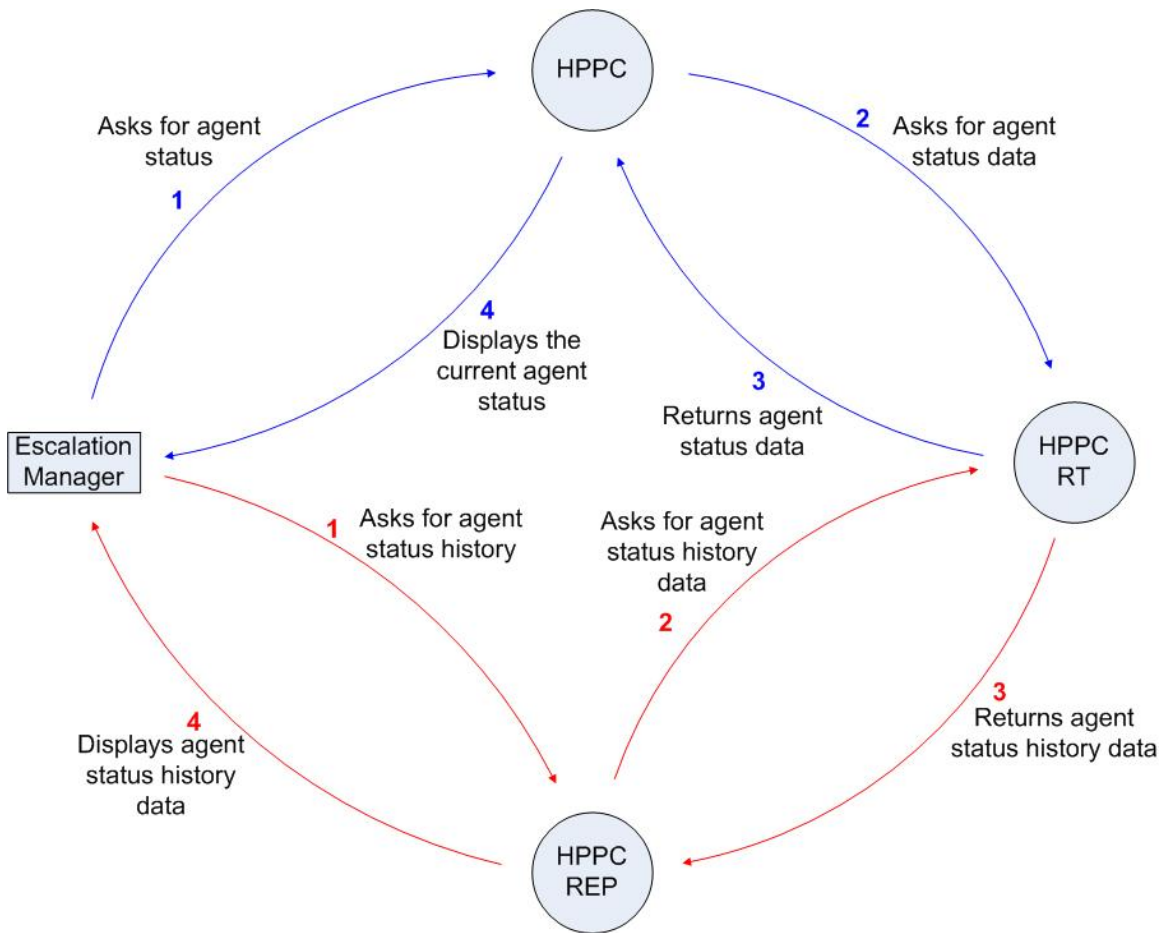
**1. How is it possible to discover that you are not within target of an agreed SLA?**



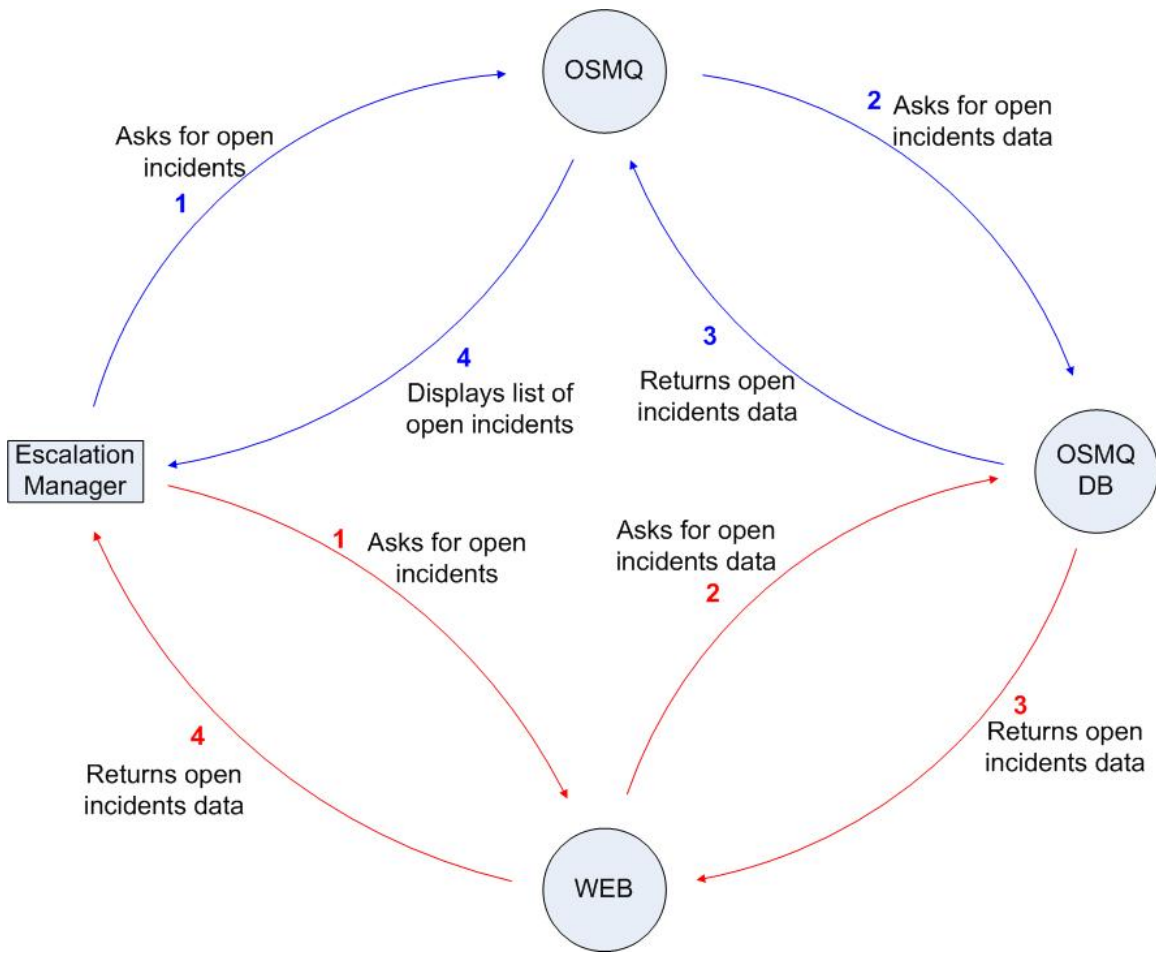
## 2. How do you monitor the response time?



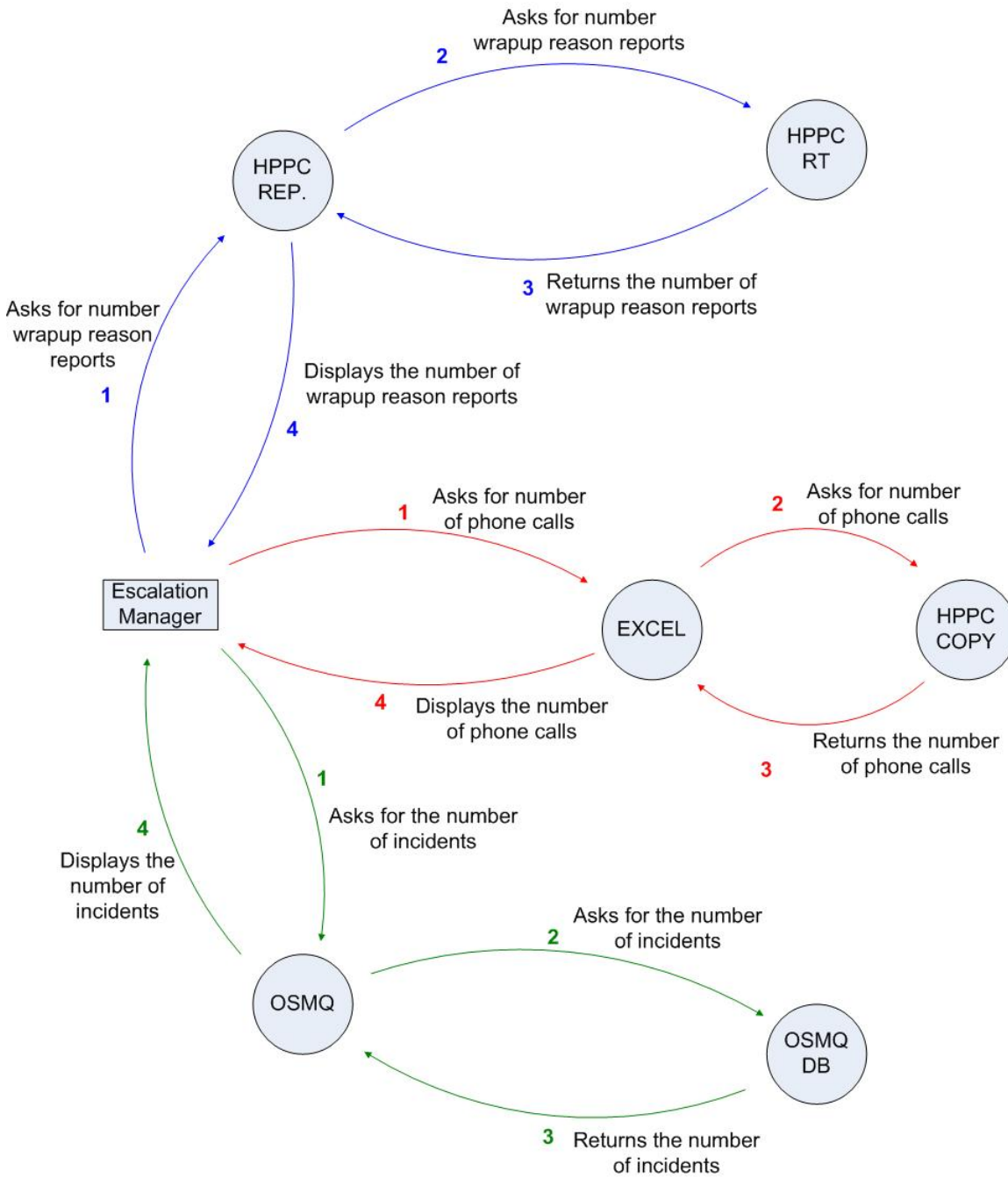
### 3. How do you monitor the agents?



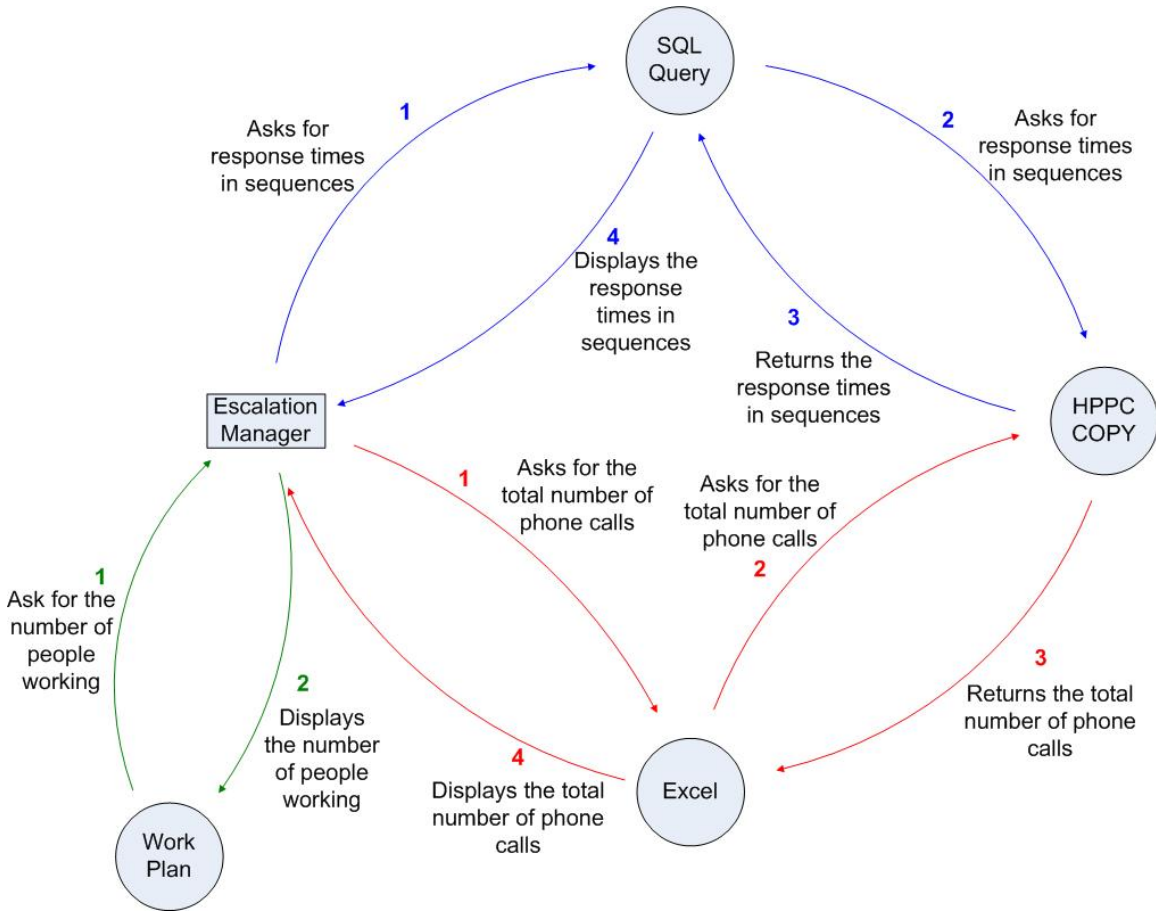
### 4. How do you monitor open incidents?



### 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?



### 6. How do you measure the workload and plan how many that is going to work?





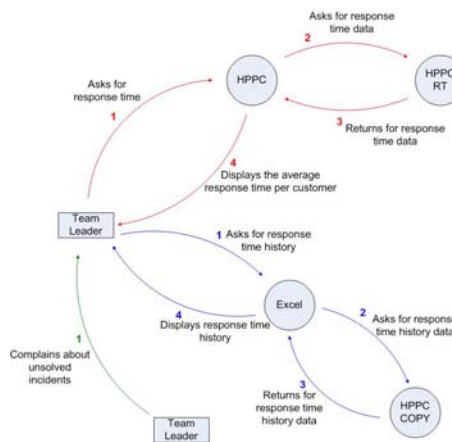
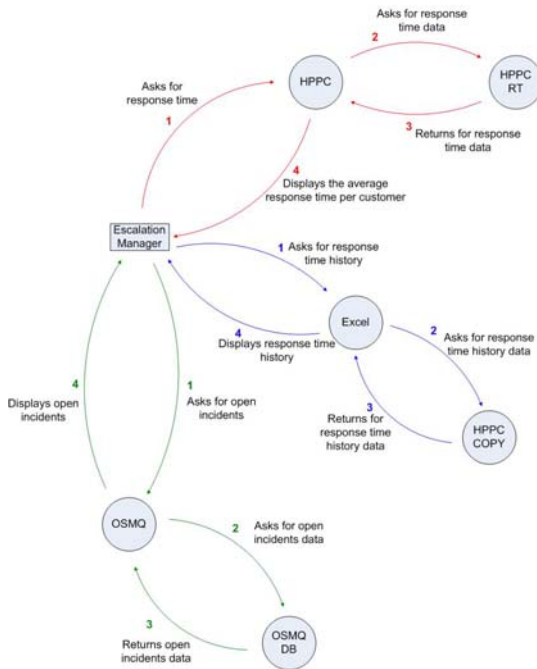
# Appendix E: Contextual Design: Consolidation

A single consolidation model of the different work models for each question from the Contextual Inquiry is presented in this section.

## 1. How is it possible to discover that you are not within target of an agreed SLA?

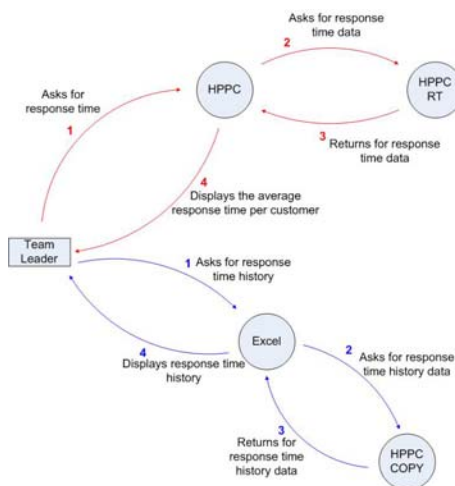
Bjørn Henrik Pedersen (Escal. Mng.)

Raymond Wiik (Team Leader)



Anders Refsdal (Team Leader)

Summary

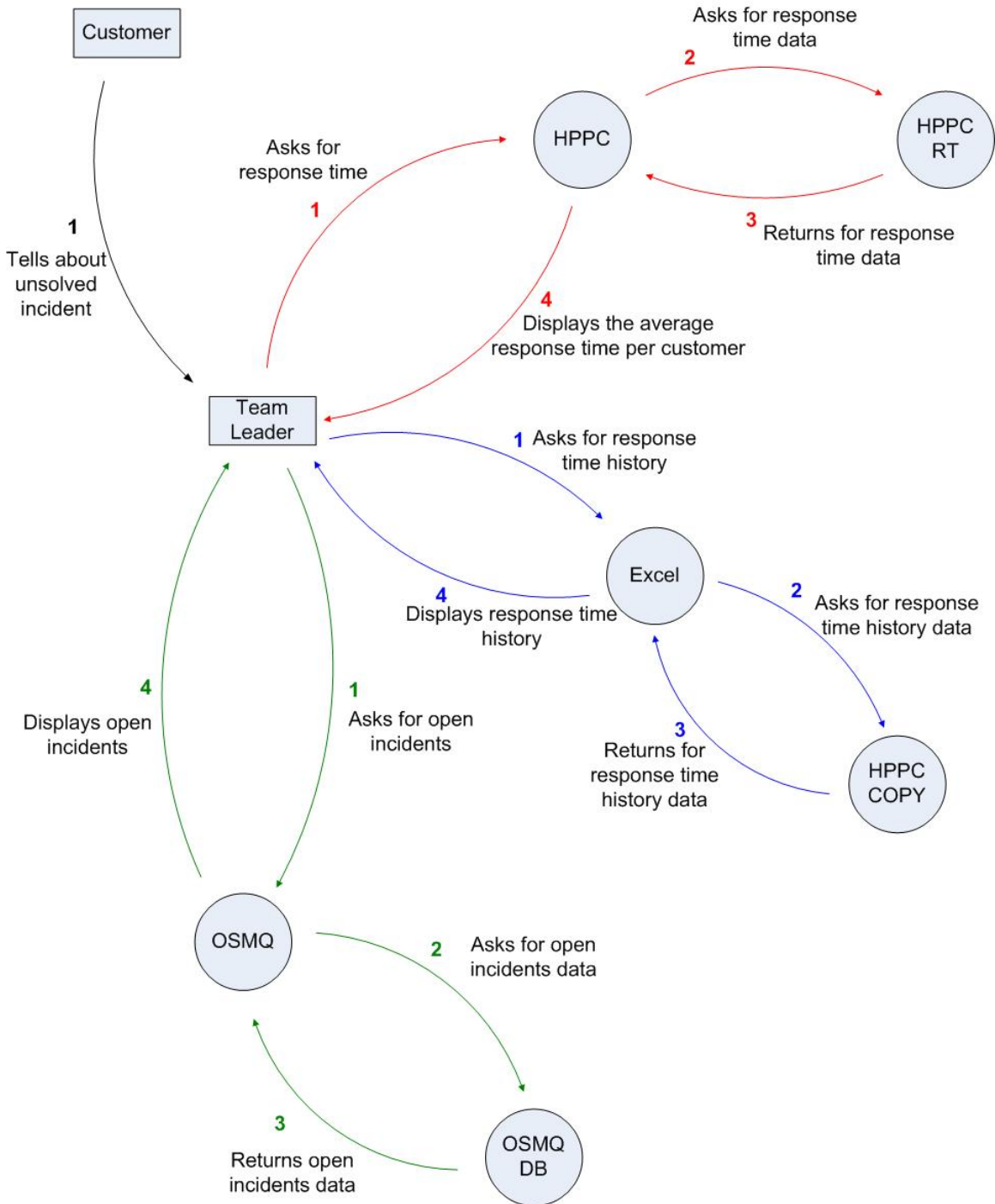


- All participants uses HPPC to see real time response times
- The Escalation Manager and one Team Leader use OSMQ to look for open incidents
- The Escalation Manager and one Team Leader uses Excel to get response time history.

### Assumptions:

- Both Team Leaders will get phone calls from unsatisfied customers
- Both Team Leaders should use OSMQ to look for open incidents
- Both Team Leaders should use Excel to get response time history.

Consolidation model:

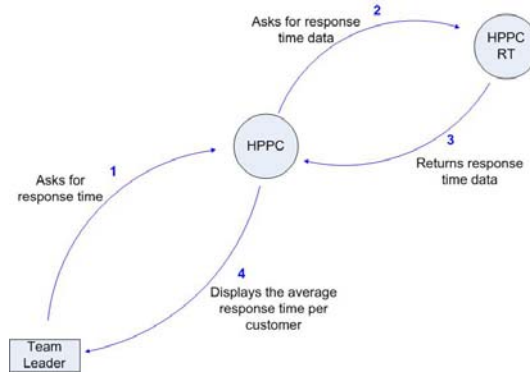
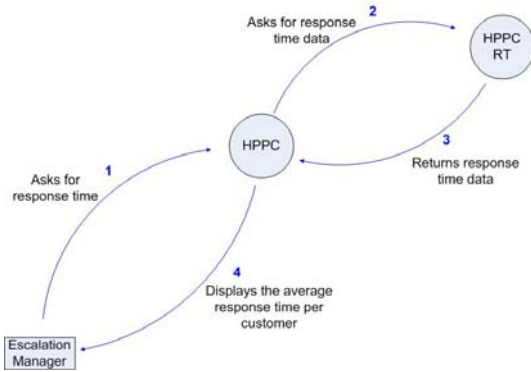


\*Team Leader in the model could also be Escalation Manager.

## 2. How do you monitor the response time?

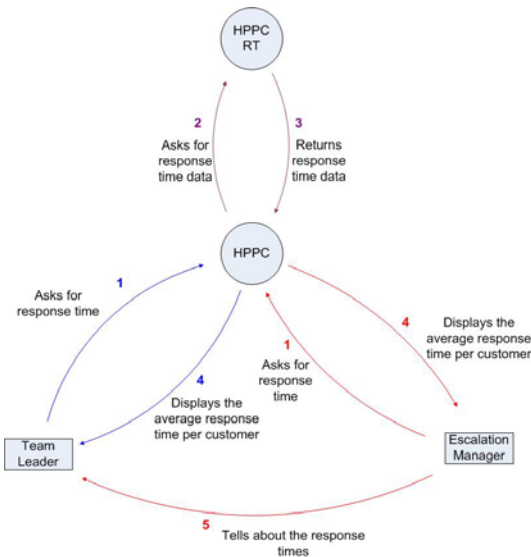
Bjørn Henrik Pedersen (Escal. Mng.)

Raymond Wiik (Team Leader)



Anders Refsdal (Team Leader)

Summary

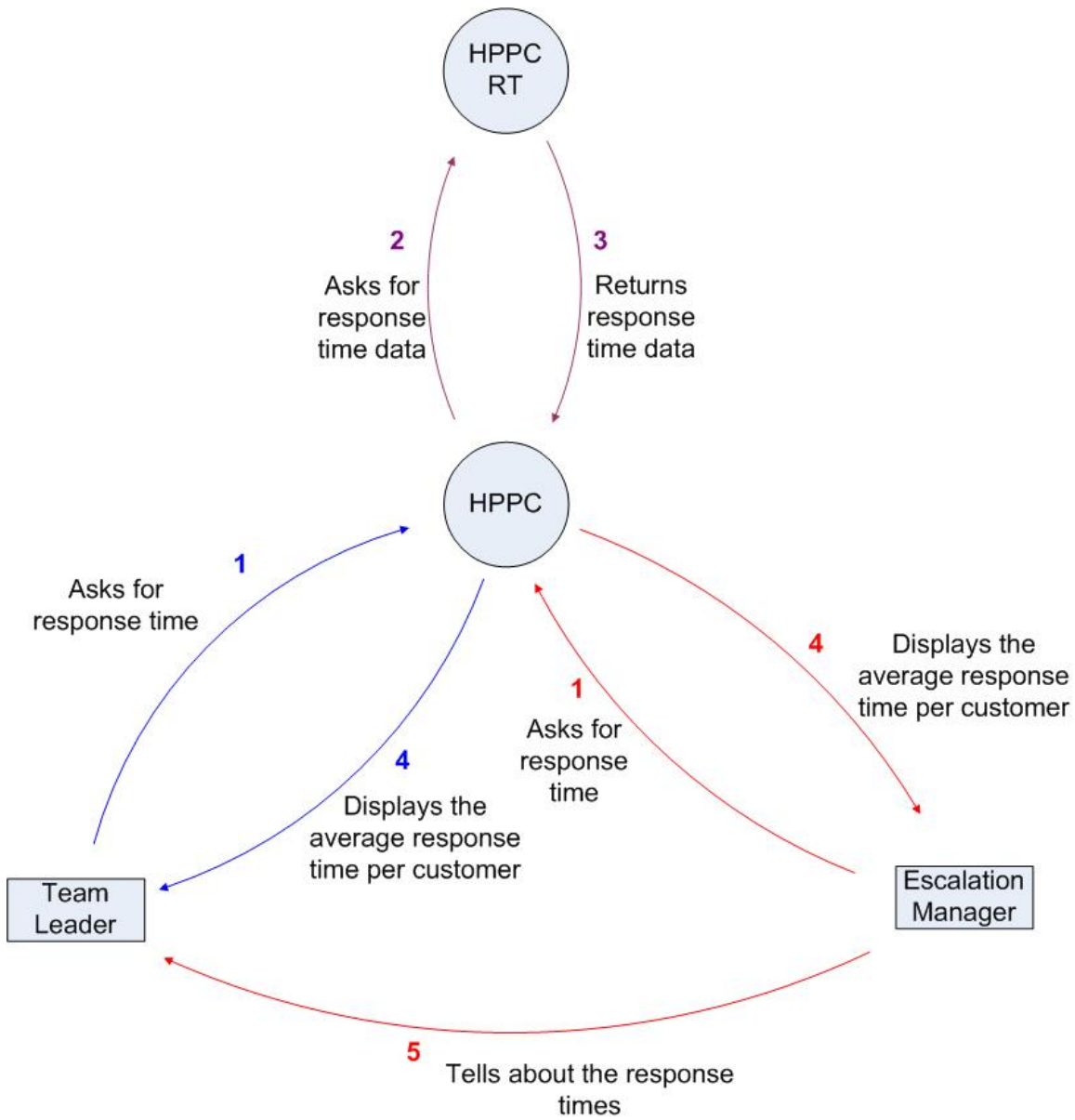


- All participants asks HPPC for response times
- One of the Team Leaders is told by the Escalation Manager if the response times are to high

### Assumptions

- Both Team Leaders will be notified by the Escalation Manager if the response times are to high

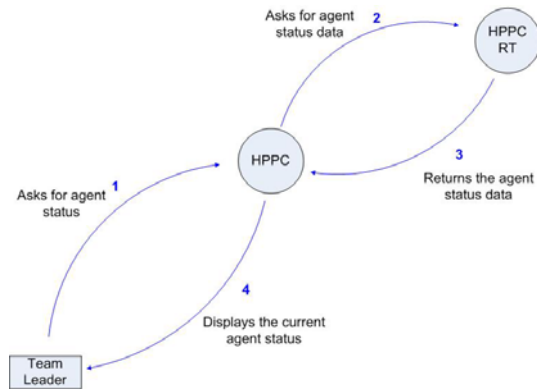
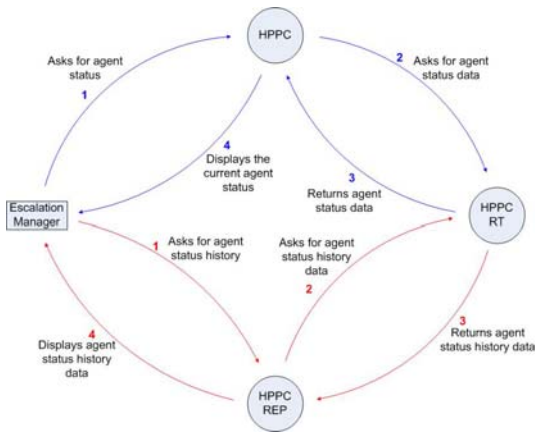
Consolidation Model:



### 3. How do you monitor the agents?

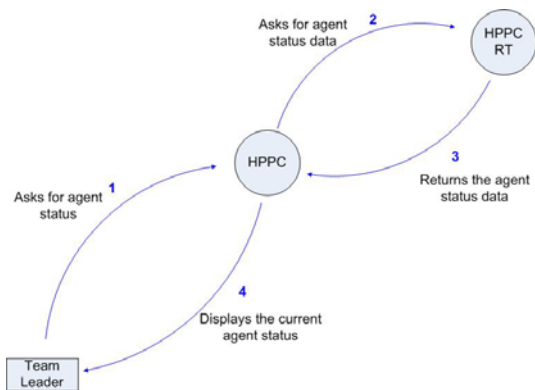
Bjørn Henrik Pedersen (Escal. Mng.)

Raymond Wiik (Team Leader)



Anders Refsdal (Team Leader)

Summary

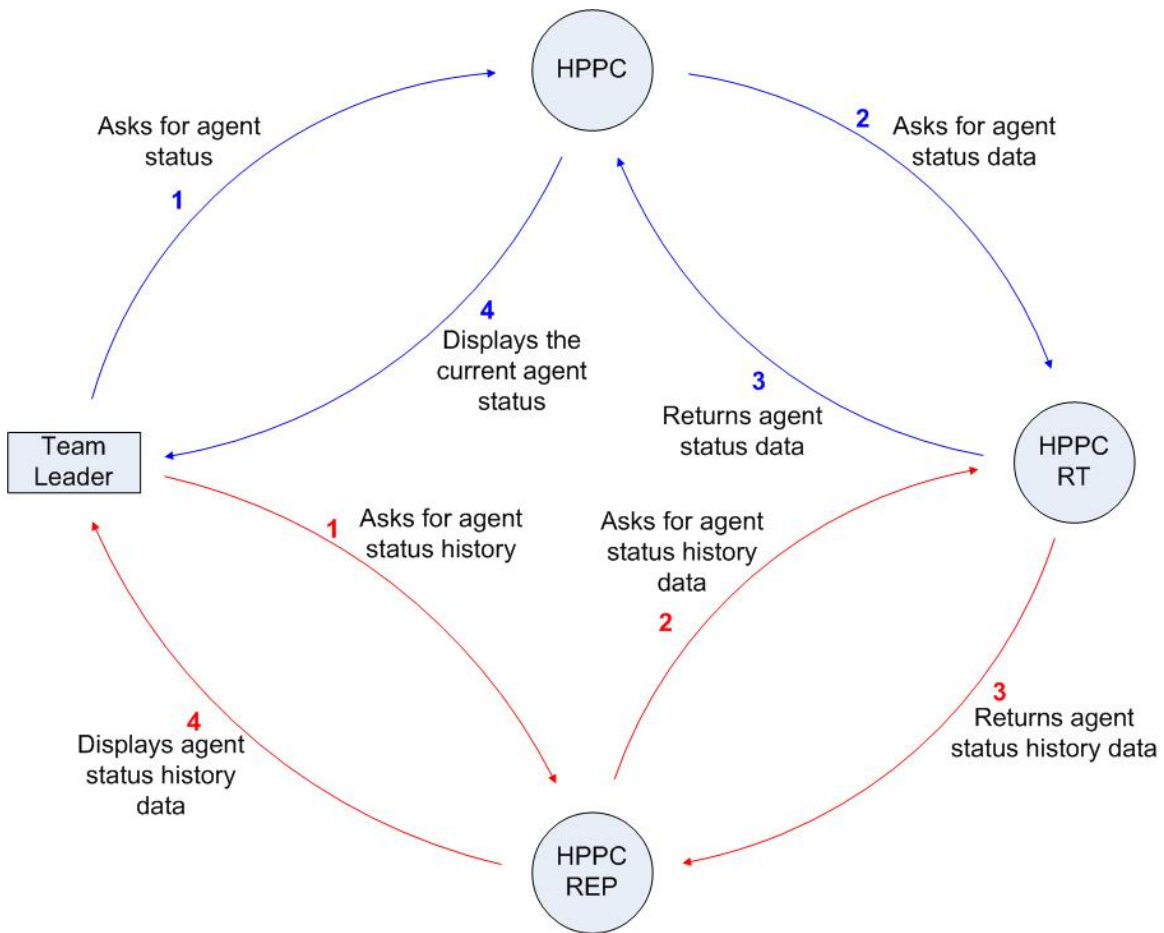


- Both the Team Leaders and the Escalation Manager uses HPPC for monitoring current agent status.
- The Escalation Manager uses HPPC Reporter for making reports over agent status history

### Assumptions

- Both Team Leaders could have use for agent status history

Consolidation Model:

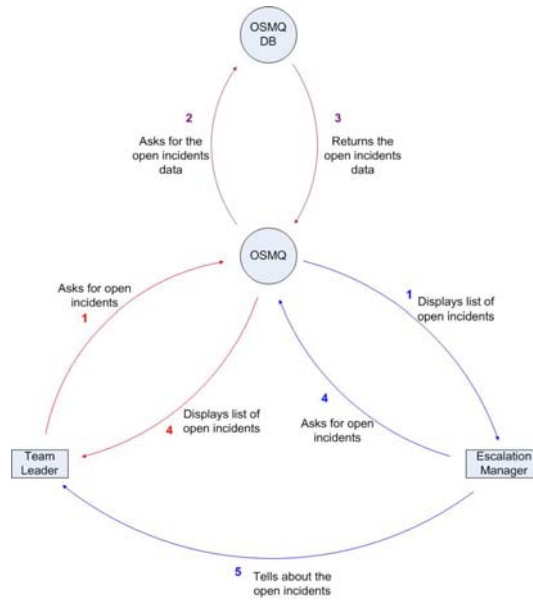
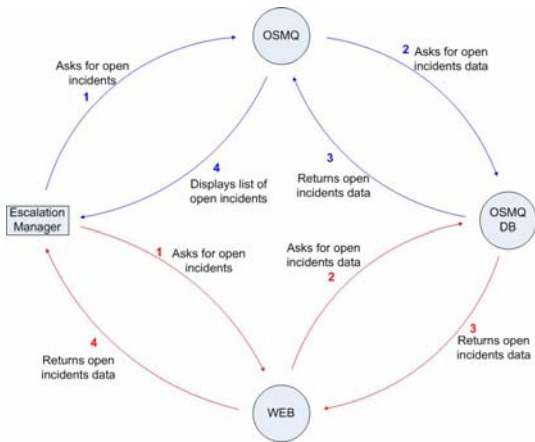


- Team Leader in the model could also be Escalation Manager.

### 4. How do you monitor open incidents?

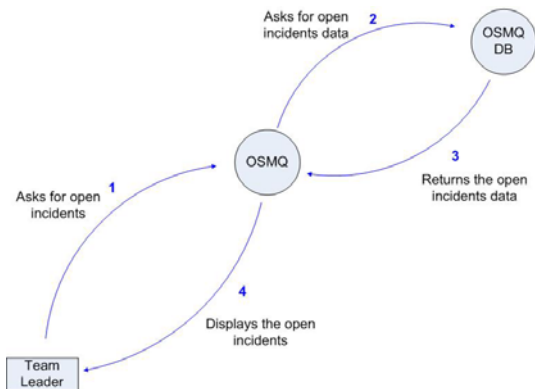
Bjørn Henrik Pedersen (Escal. Mng.)

Raymond Wiik (Team Leader)



Anders Refsdal (Team Leader)

Summary

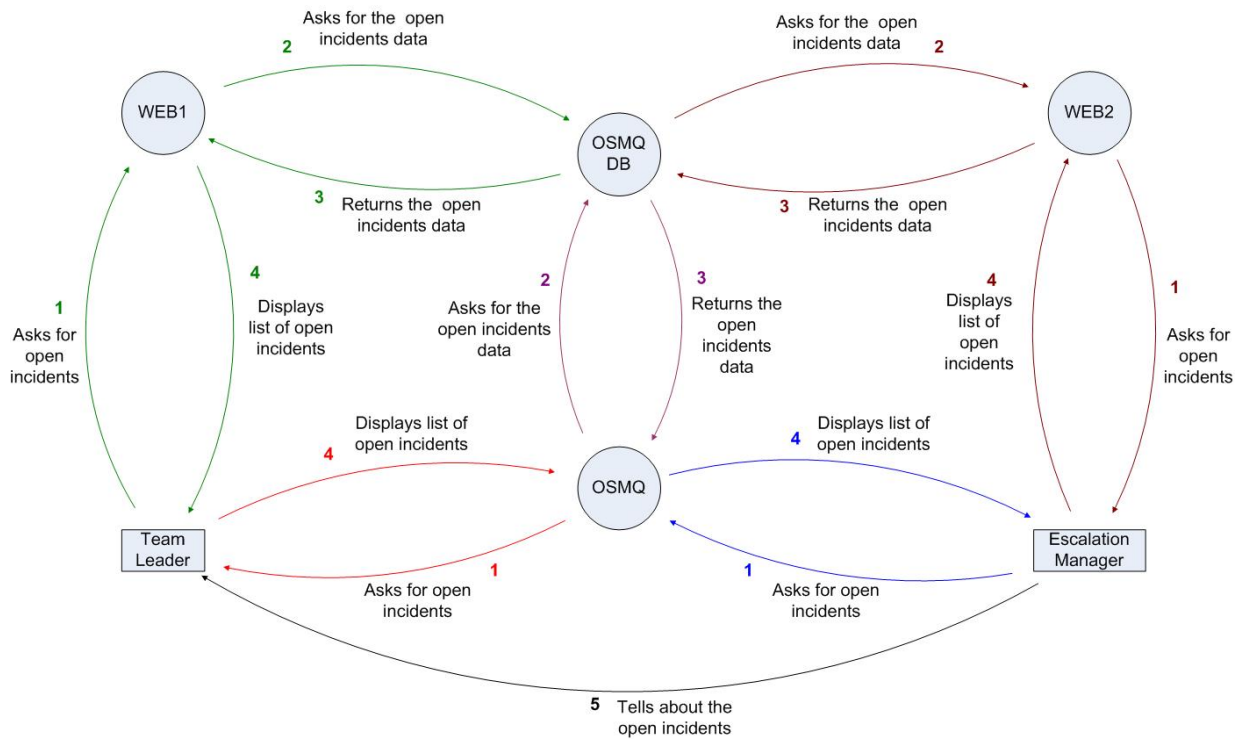


- Only the Escalation Manager is using the Web interface
- One of the Team Leaders are told by the Escalation Manager about open incidents
- 

### Assumptions

- Both Team Leaders could use Web interface
- The Escalation manager tells both Team Leaders about open incidents

Consolidation Model:



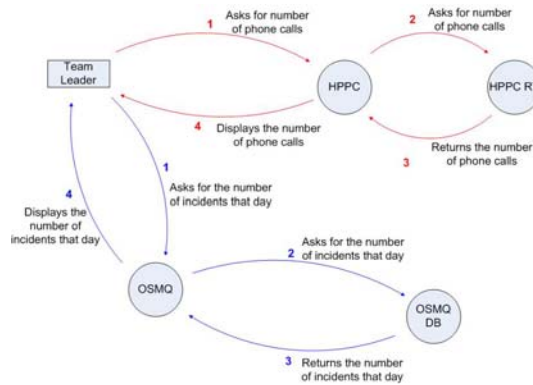
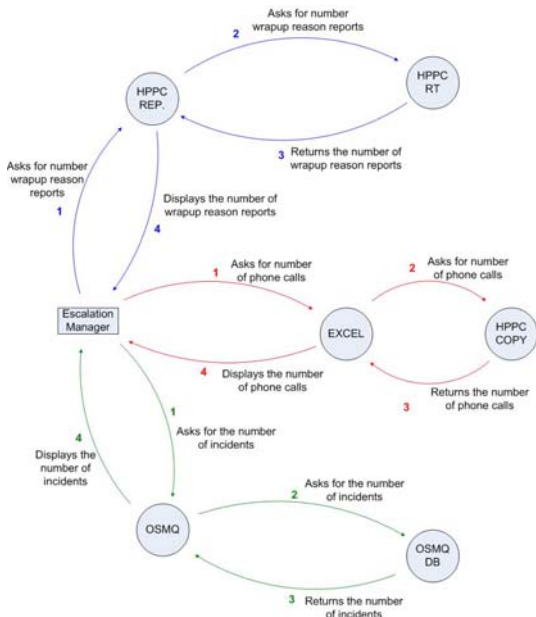
- WEB1 and WEB2 are the same process but divided in the figure to make it more understandable



### 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?

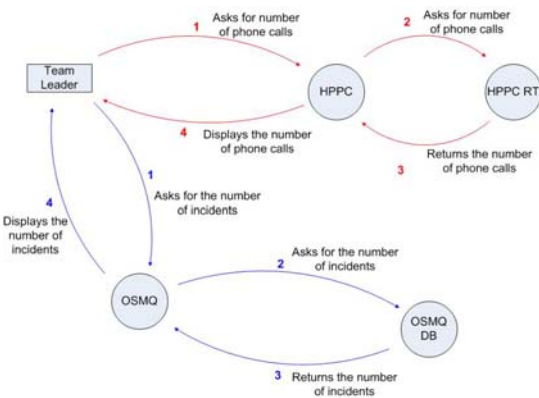
Bjørn Henrik Pedersen (Escal. Mng.)

Raymond Wiik (Team Leader)



Anders Refsdal (Team Leader)

Summary

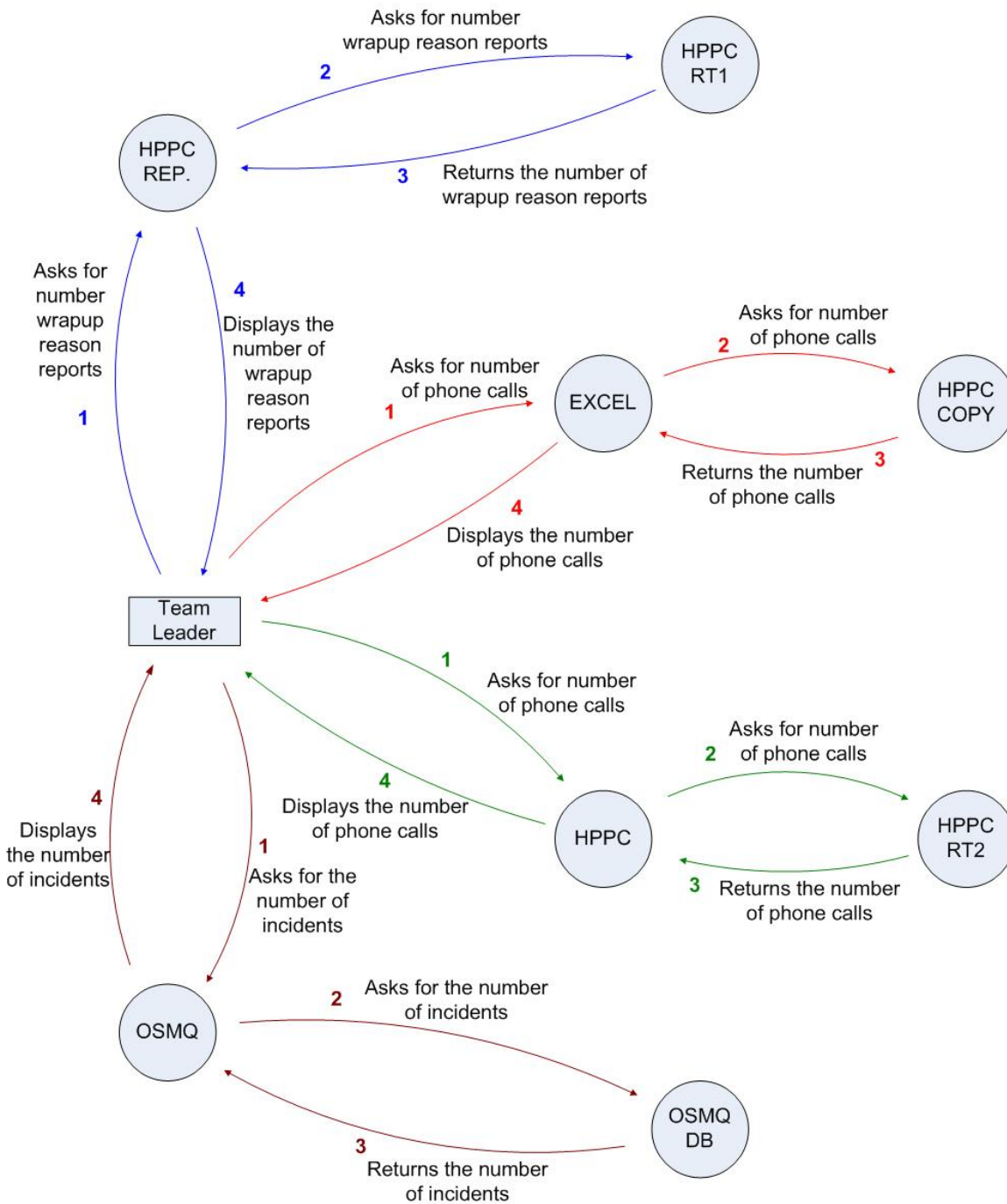


- The two Team Leaders are getting the number of calls from the HPPC Real Time database
- The Escalation Manager is getting the response time from a copy of the HPPC database
- Non of the Team Leaders are taking regard for the Wrapup Reason Reports

### Assumptions

- All of them should take regard of the Wrapup Reasons Reports
- Response times could be collected from both HPPC and Excel.

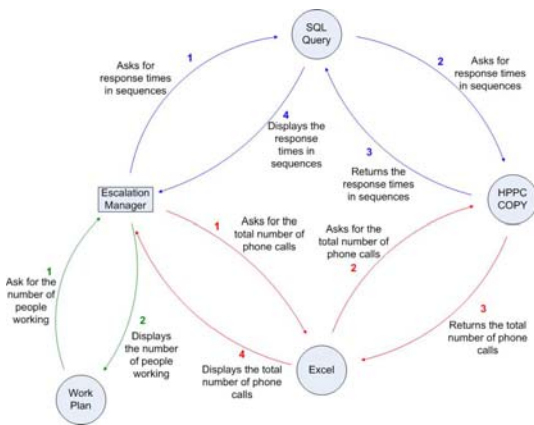
Consolidation Model:



- HPPC RT1 and HPPC RT2 WEB2 are the same process but divided in the figure to make it more understandable

## 6. How do you measure the workload and plan how many that is going to work?

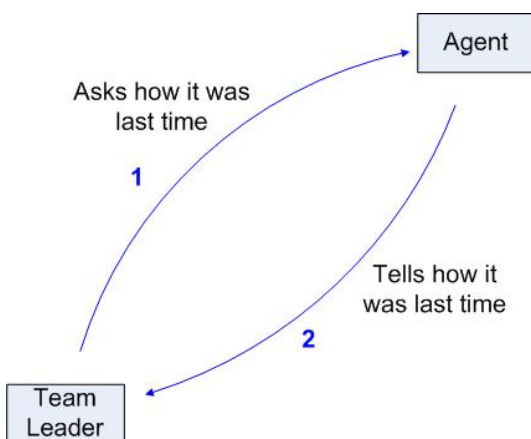
Bjørn Henrik Pedersen (Escal. Mng.)    Raymond Wiik (Team Leader)



Routines

Anders Refsdal (Team Leader)

Summary



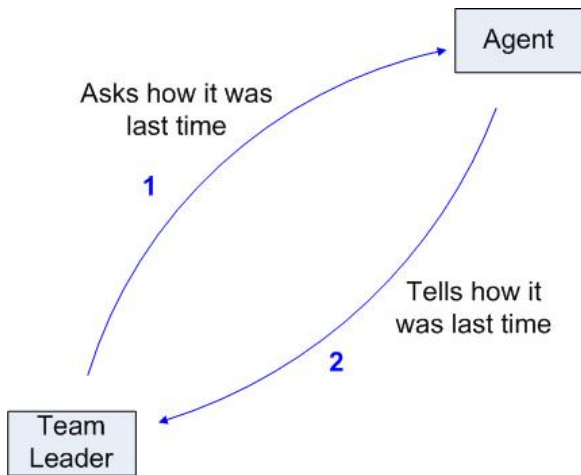
- The Escalation Manager uses an Excel macro to get the number of phone calls and gets response time sequences from HPPC by SQL Query Enterprise and compare this with the Work Plan
- Both Team Leaders are doing this manually
  - o One talks with agents
  - o The other one has routines

### Assumptions

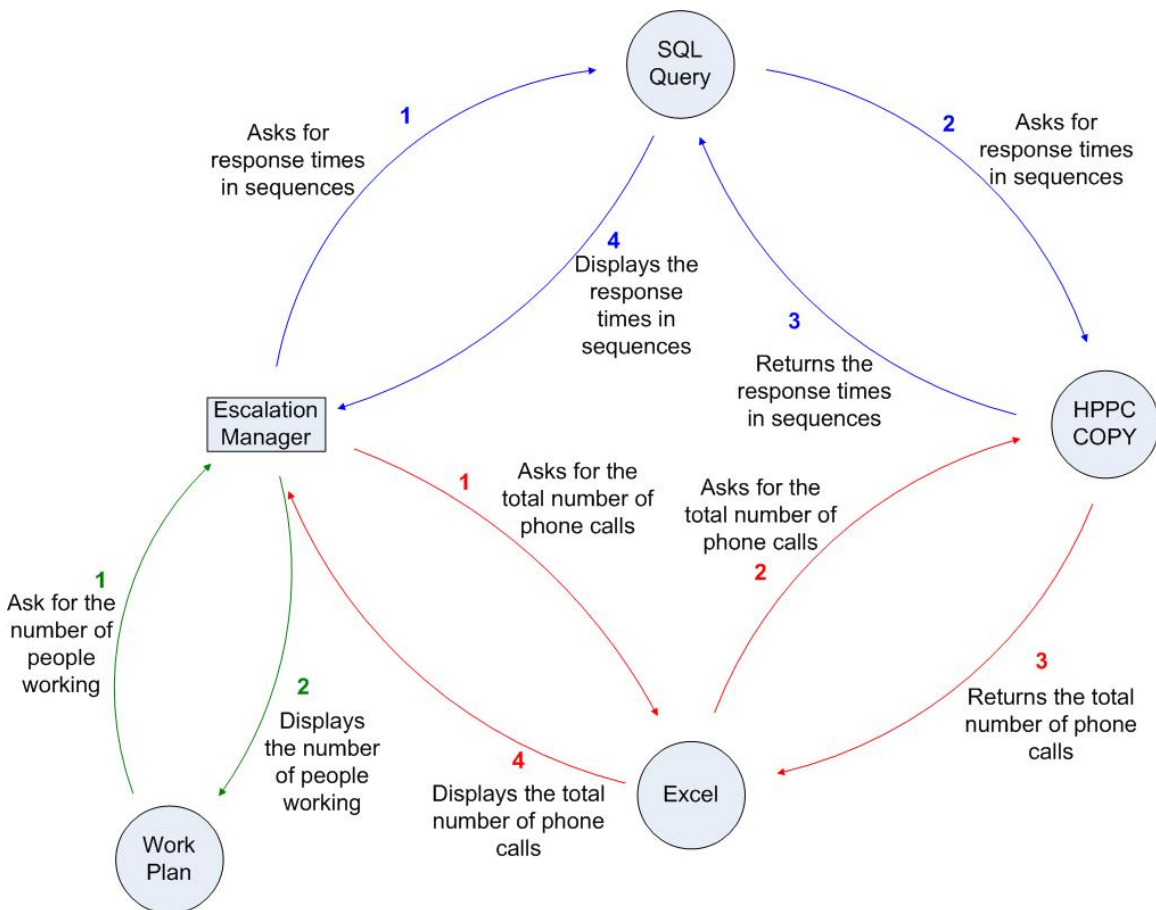
- The two Team Leaders doesn't have rights to use the SQL Query Enterprise to get response time sequences
- Both Team Leaders should talk with the people who worked the last time if there is a special situation.

### Consolidation Models:

#### Team Leaders:



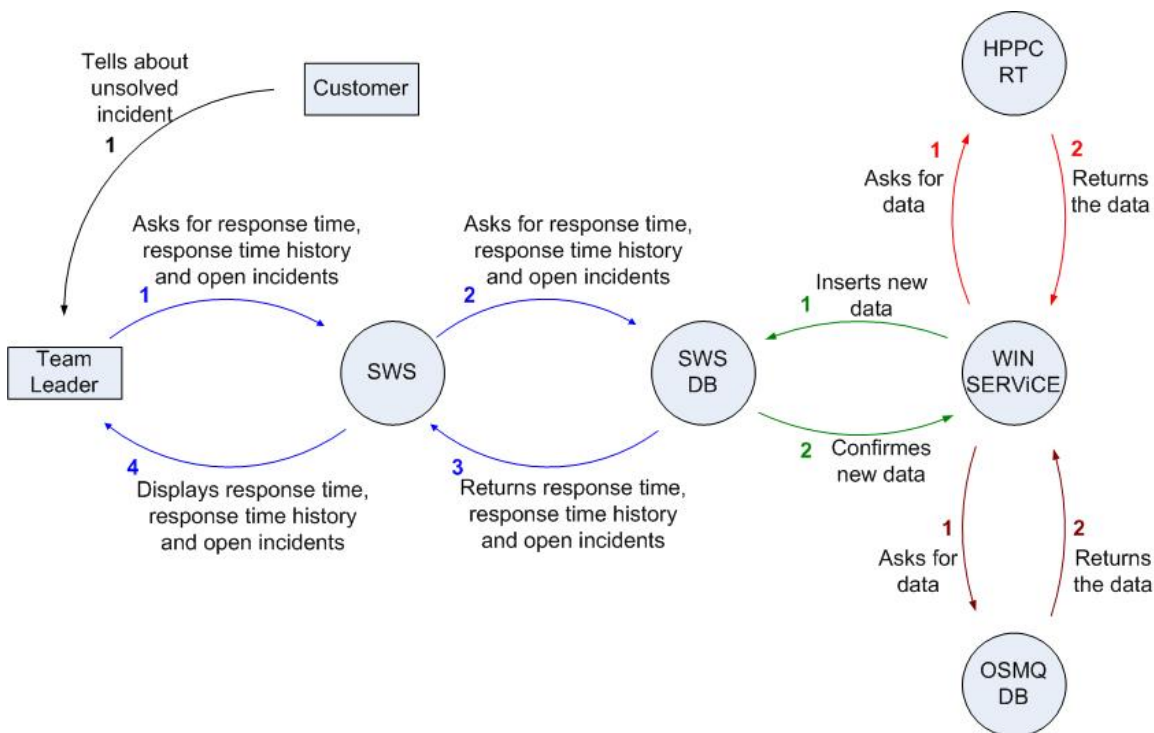
#### Escalation Manager:



## Appendix F: Work Redesign

The Snapshot Website has been implemented to suggest better ways to work.

### 1. How is it possible to discover that you are not within target of an agreed SLA?



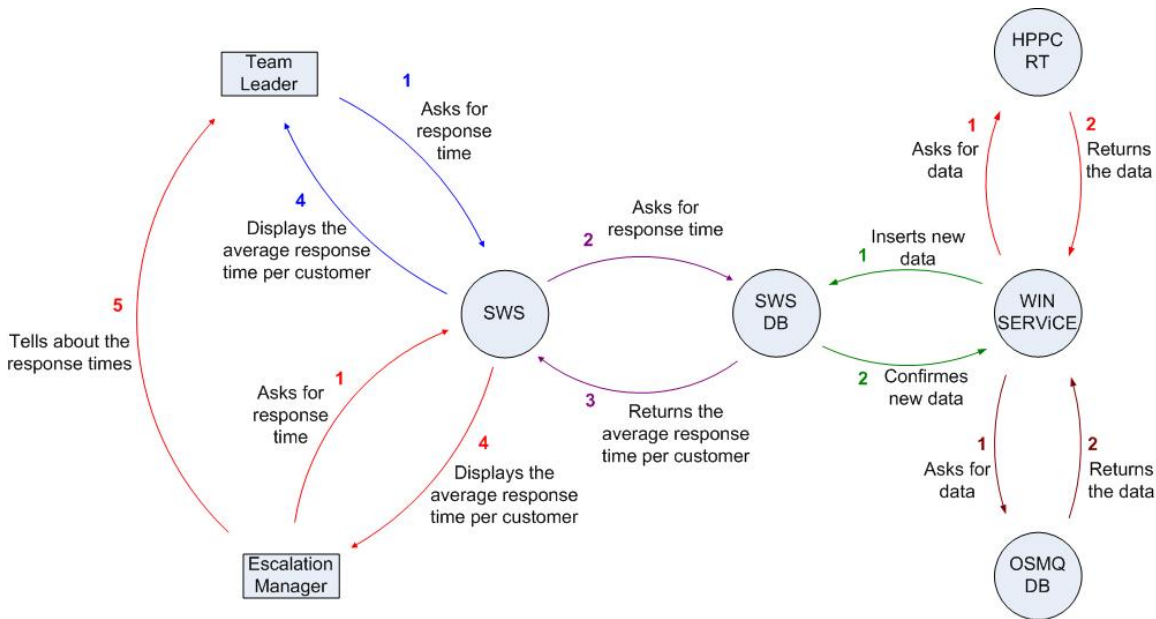
#### Changes:

- The Team Leader and the Escalation Manager will only use SWS to get information
- SWS will update data from the databases at certain intervals and give data from the last update when a Team Leader or Escalation Manager asks
  - o There are no triggering from the Team Leader or Escalation Manager to make SWS access the Databases

#### Advantages:

- The databases which in the daily use have a large amount of traffic will get less traffic since they are only accessed at certain intervals.
- The Team Leader will only have to deal with one GUI

## 2. How do you monitor the response time?



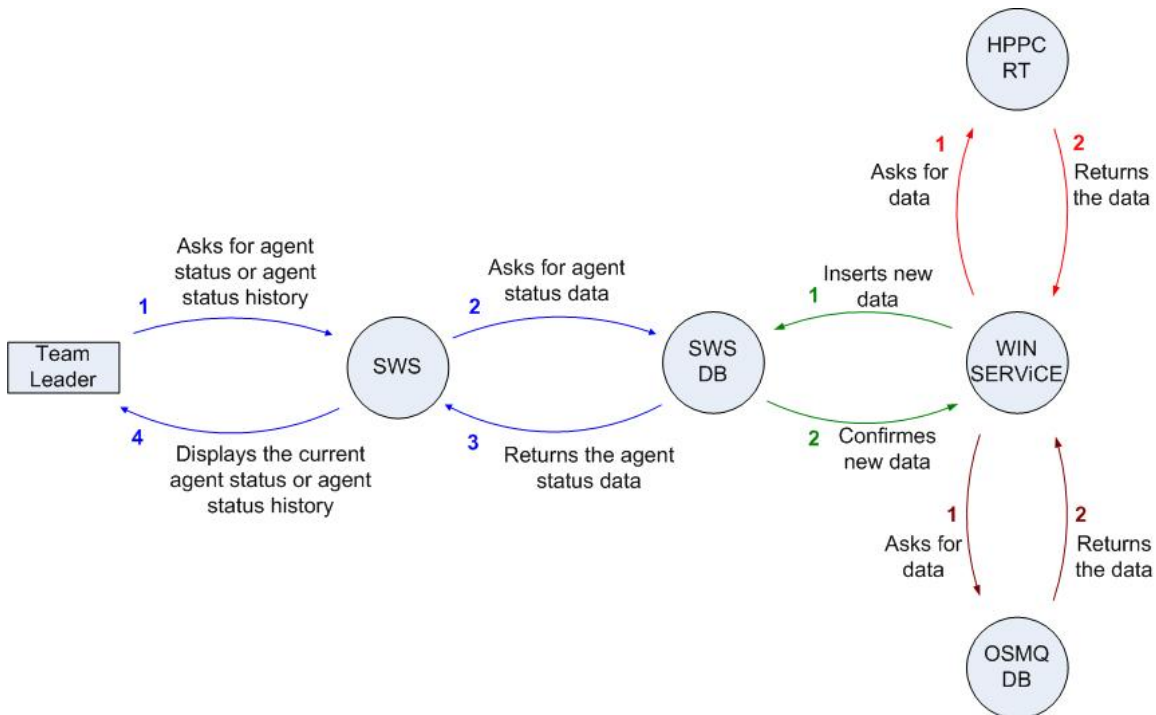
### Changes:

- The Team Leader and the Escalation Manager will only use SWS to get information
- SWS will update data from the databases at certain intervals and give data from the last update when a Team Leader or Escalation Manager asks
  - o There are no triggering from the Team Leader or Escalation Manager to make SWS access the Databases

### Advantages:

- The databases which in the daily use have a large amount of traffic will get less traffic since they are only accessed at certain intervals

### 3. How do you monitor the agents?



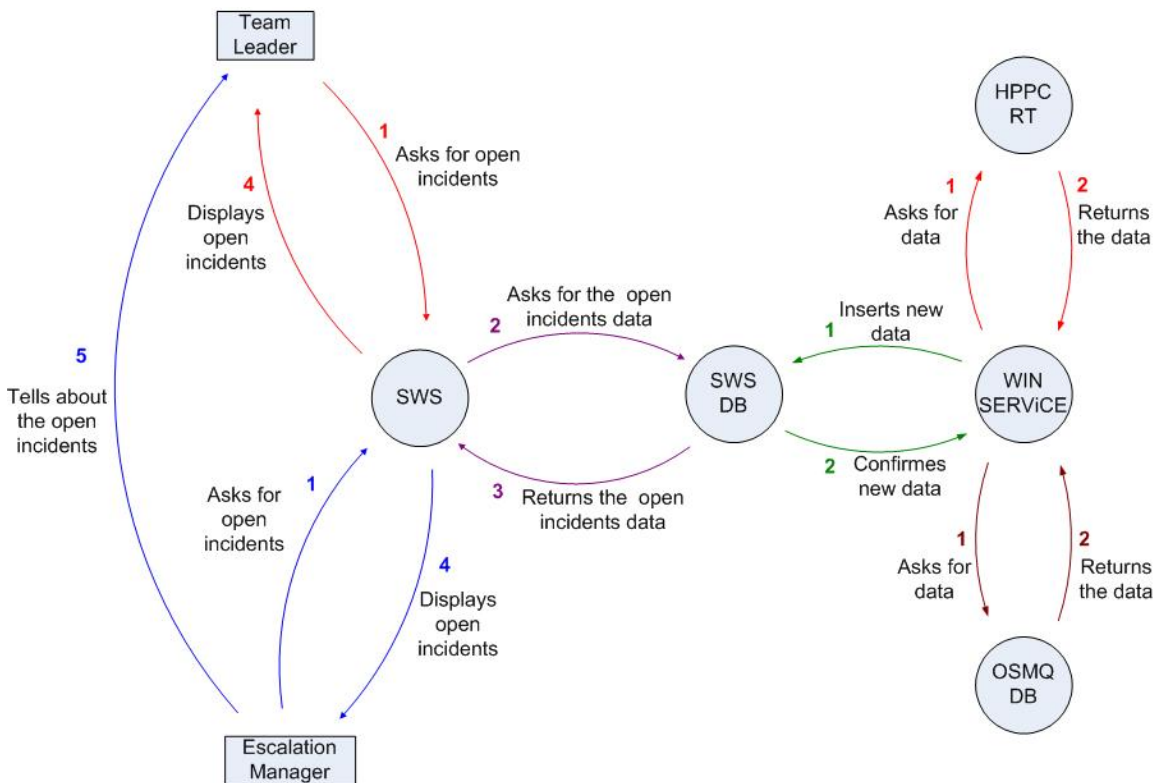
#### Changes:

- The Team Leader and the Escalation Manager will only use SWS to get the current agent status or the agent history.
- SWS will update data from the databases at certain intervals and give data from the last update when a Team Leader or Escalation Manager asks
  - o There are no triggering from the Team Leader or Escalation Manager to make SWS access the Databases

#### Advantages:

- The databases which in the daily use have a large amount of traffic will get less traffic since they are only accessed at certain intervals
- The Team Leader or Escalation Manager will only have to deal with one GUI

#### 4. How do you monitor open incidents?



#### Changes:

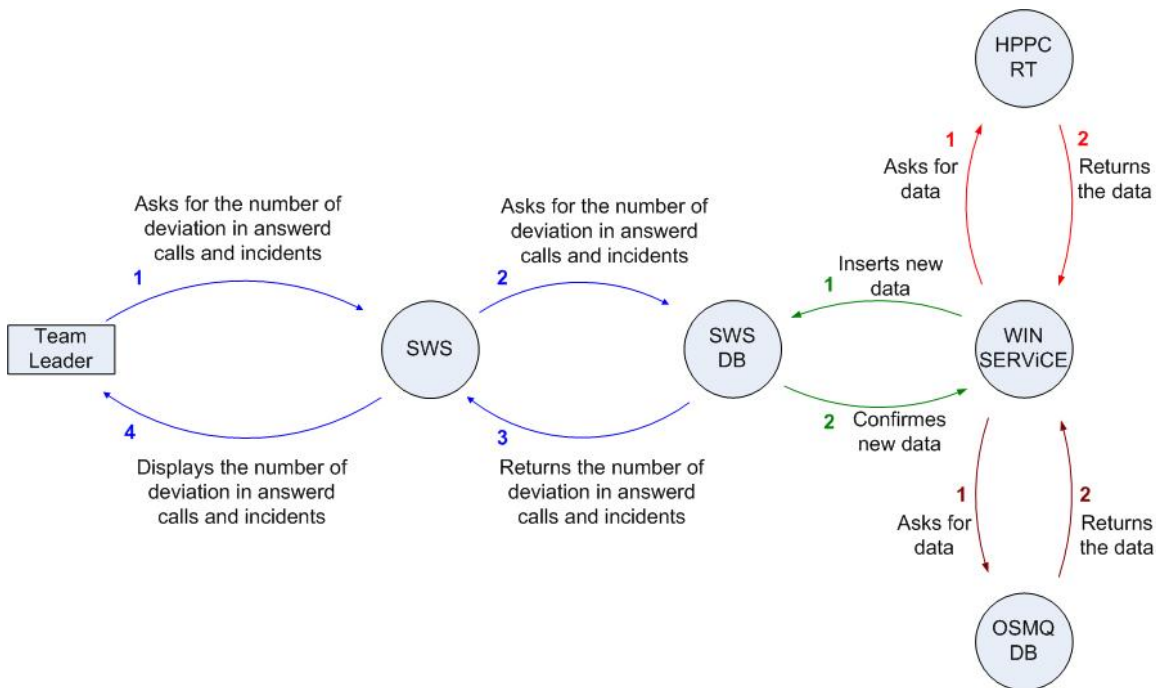
- The Team Leader and the Escalation Manager will only use SWS to get the open incidents
- SWS will update data from the databases at certain intervals and give data from the last update when a Team Leader or Escalation Manager asks
  - o There are no triggering from the Team Leader or Escalation Manager to make SWS access the Databases

#### Advantages:

- The databases which in the daily use have a large amount of traffic will get less traffic since they are only accessed at certain intervals
- The Team Leader or Escalation Manager will only have to deal with one GUI



## 5. How do you find deviations between the number of OSMQ incidents and the number of phone calls?



### Comments:

- The number of phone calls can be collected from both the HPPC RT and the HPPC COPY. It is preferred that the data is collected from the HPPC COPY to save traffic load for the HPPC RT. But the data is transferred from HPPC RT to HPPC COPY once every night and if the data hasn't been transferred yet the HPPC RT must be used.
- It could still be that the databases must be accessed if a Team Leader or Escalation Manager asks for a date not stored in the local SWS database.

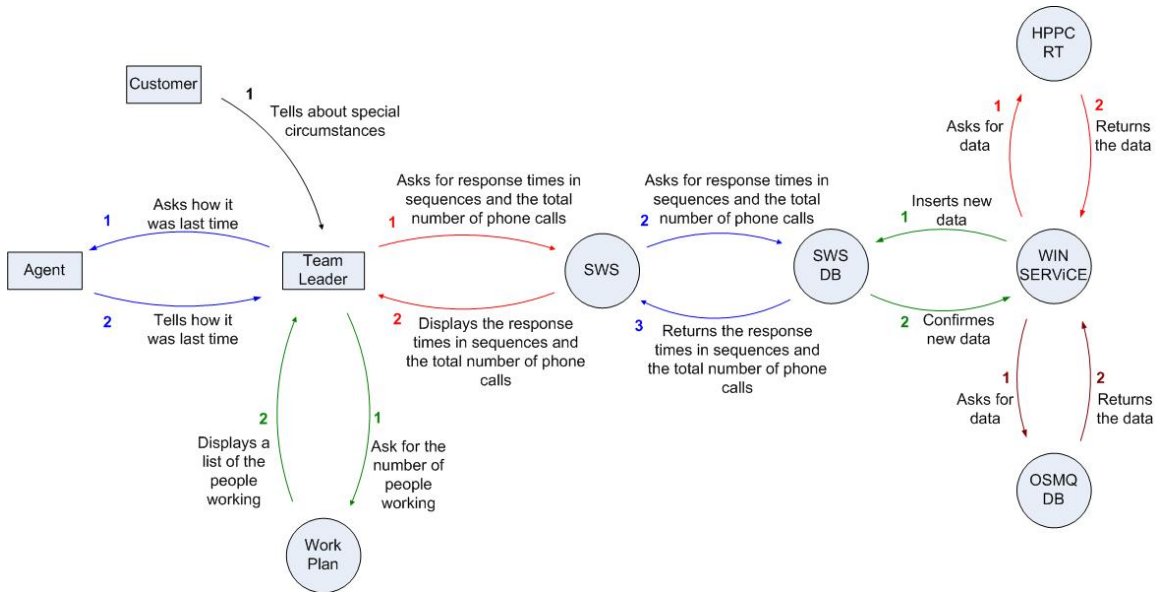
### Changes:

- The Team Leader and the Escalation Manager will only use SWS to get the deviations in the number of calls and the number of incidents.

### Advantages:

- No manual calculation is needed
- The Team Leader or Escalation Manager will only have to deal with one GUI

### 6. How do you measure the workload and plan how many that is going to work?



**Comments:**

- The Work Plan is an Excel document

**Changes:**

- The Team Leaders are able to get the response time in sequences which they didn't have access to before
- 

**Advantages:**

- It is possible for the Team Leaders and the Escalation Managers to get the response time in sequences and they can easily compare this with the work plan.
- The work load could be calculated with only a little bit of manual work

## Appendix G: User Environment Design

The system is divided in three sub systems (illustrated in the floor plan):

### 1 The User system

The user System is an open system where everyone has access. It consists of three modules; Menu, Main, and Login.

If the user wants to get to the two other subsystems a login is requested. The password and username will determine if the user will have access to the two sub systems, only one of them or no access at all.

### 2 The Management system

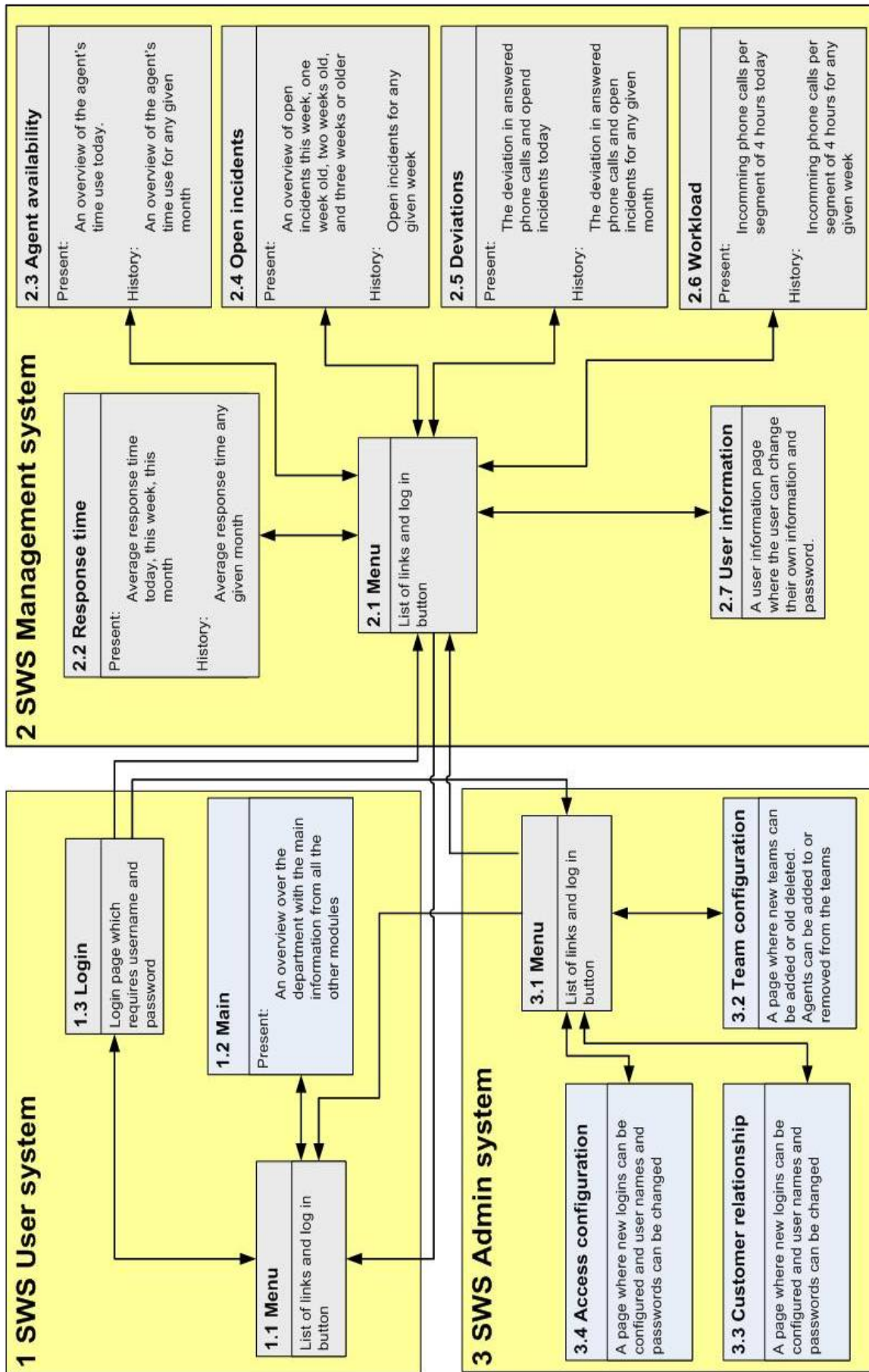
The Management system is typically meant for Team Leaders, not the regular agent. The system consists of 7 modules; Menu, Response time, Agent availability, Open Incidents, Deviations, Workload, and User information.

The user which logged in through the User system will have the possibility to use the menu from the User system when he is logged into the Management system. If the user will be able to enter the Admin system he must be authorized to this also through the login the User system. If he is authorized here it is possible to go to the admin system from the Menu.

### 3 The Admin system

The Admin system is typically meant for Escalation Managers which are controlling the whole department. It consists of 4 modules; Menu, Team configuration, Customer relationship, and Access configuration.

When a user is logged in through the User system and the user has access to the Admin system the user will also have access to the Management system and can move around in the system through the Menu modules.



## Appendix H: Mockup and test with customers

### Summary of Mockup and Test with Customer

**Main:**

OK!

**Login:**

Looks OK, but think of technology to use for login, Active Directory or an own login system.

**Response time:**

It is very good that the response time for each team is displayed and that it is possible to see the response times for each customer within each team.

If possible make a option to switch from graphs to tables for easier export of data to other programs.

**Agent availability:**

OK!

**Open incidents:**

Change the shown information to: Open incidents this week, older than two days, older than a week, older than two weeks, older than three weeks, four weeks or older, and the history where you can select a from date.

**Deviations:**

Make a calendar for start date and a calendar for end date and show the deviation history between those dates.

**Workload:**

If possible the time periods of which the workload is calculated over could be flexible so that the users of the admin system can change them.

**Access configuration:**

When adding a new user to the Management system there should be possible to choose which models the user will have access to view.

**Customer relationship:**

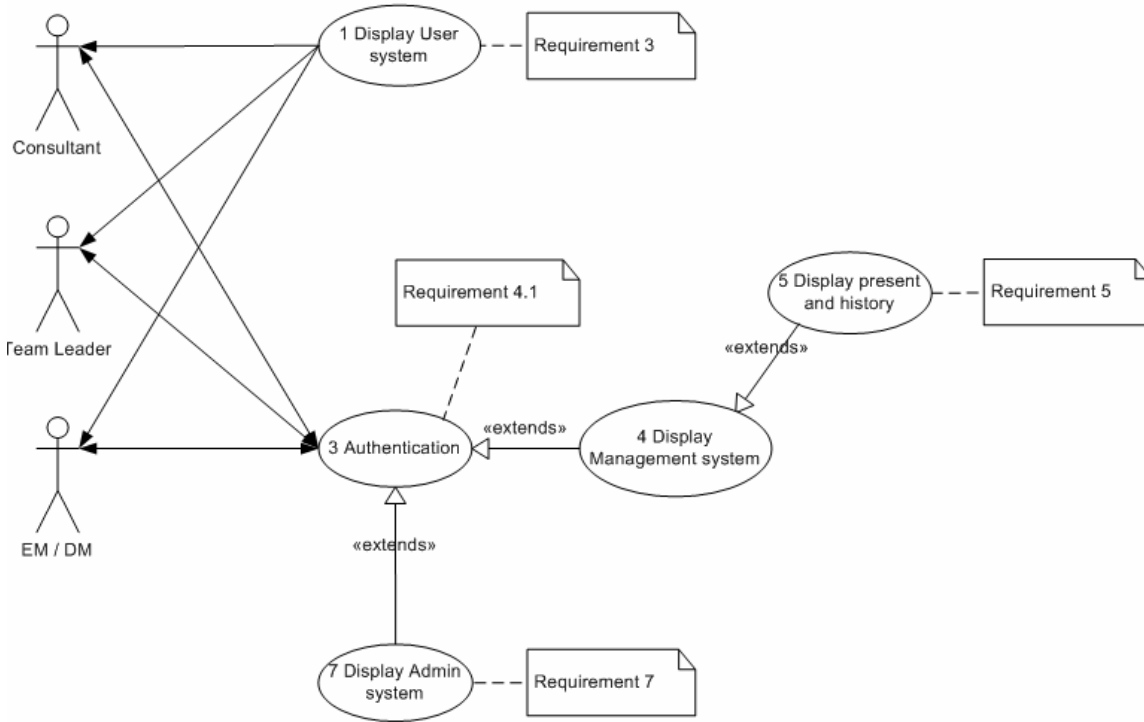
OK!

**Team configuration:**

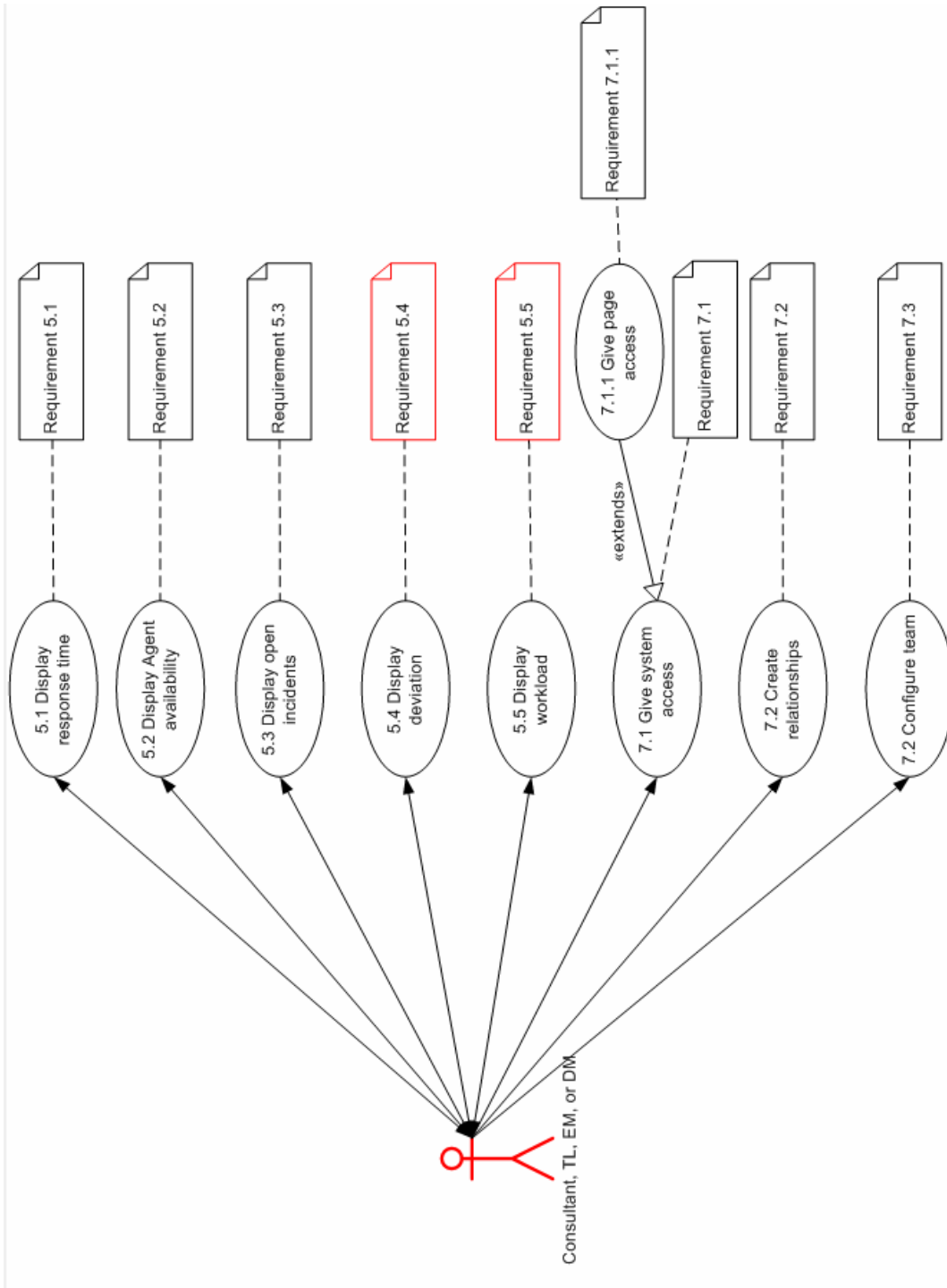
Team configuration will be needed and will have the possibility to add new teams, delete teams, add an agent to a team, or remove an agent from a team

# Appendix I: Diagrams from the development

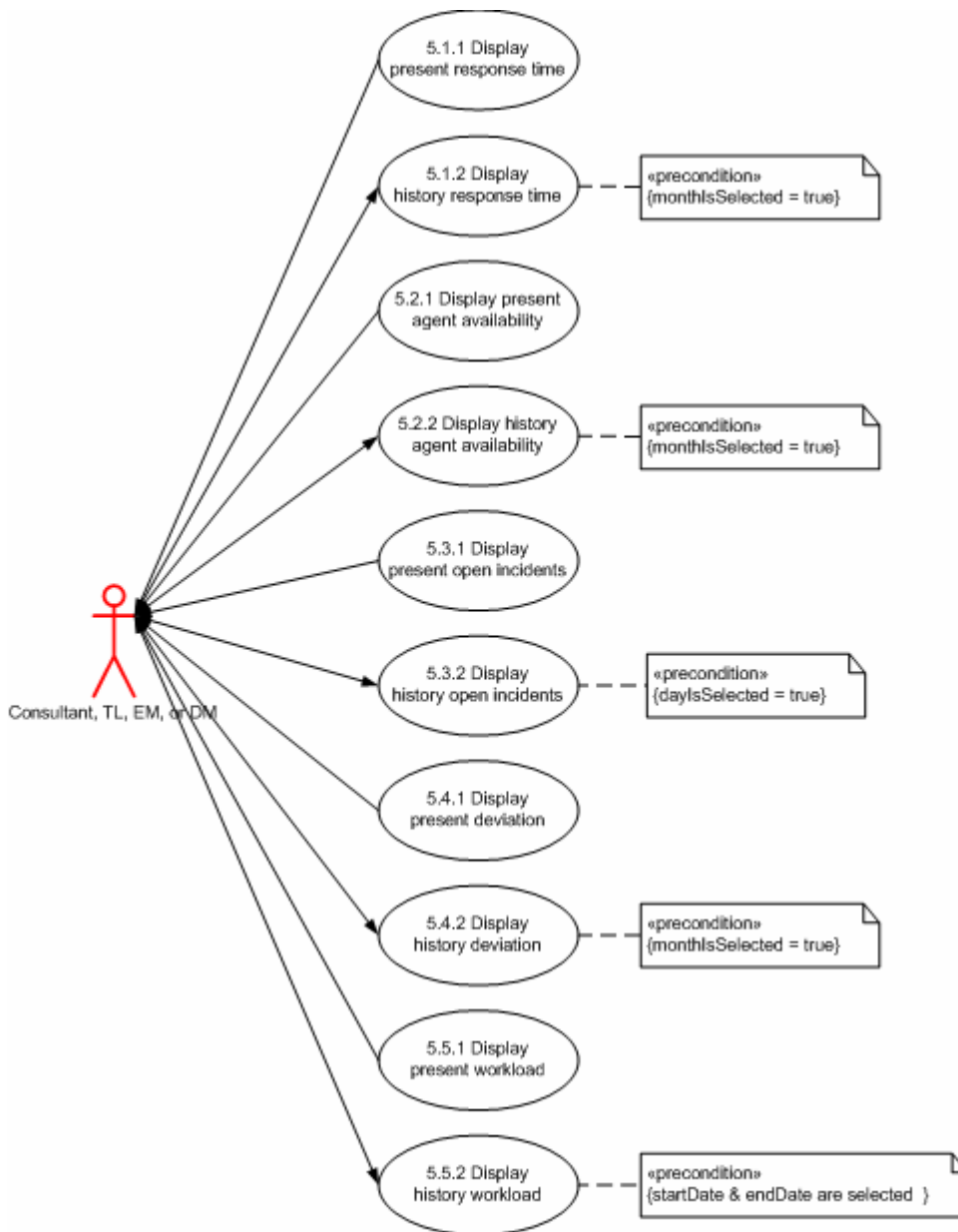
## I.1 Use Case 0



## I.2 Use Case 1:

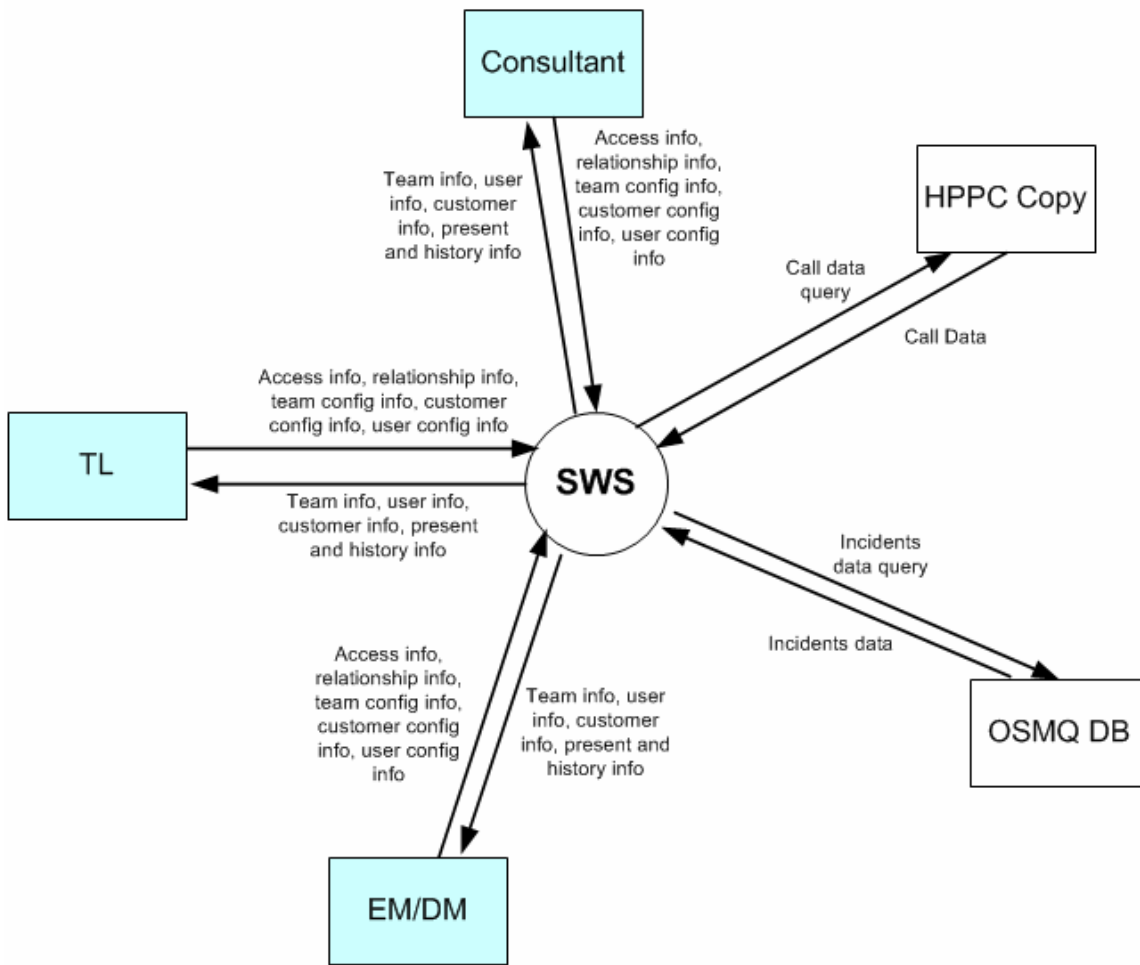


### I.3 Use Case 2:

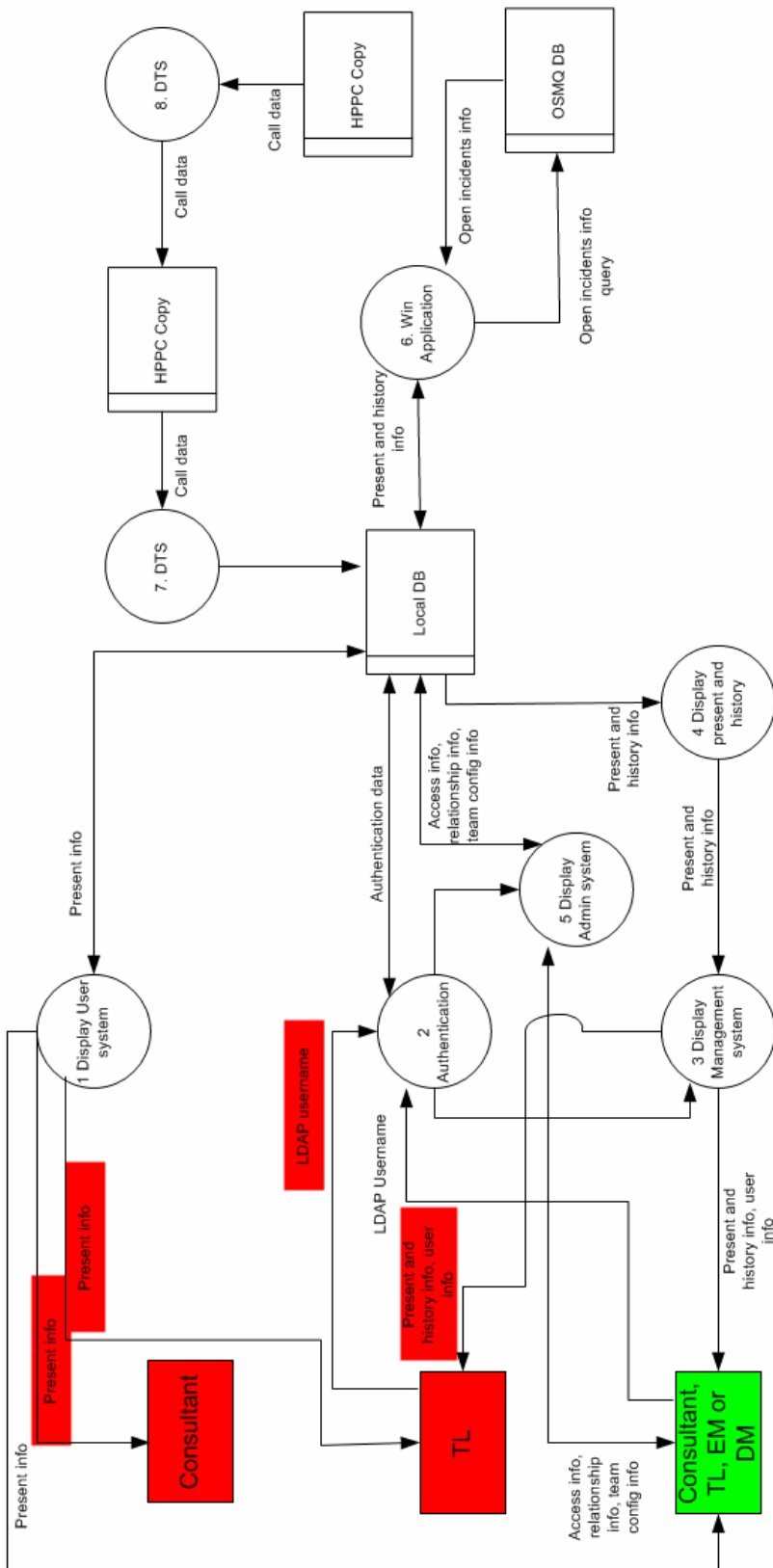




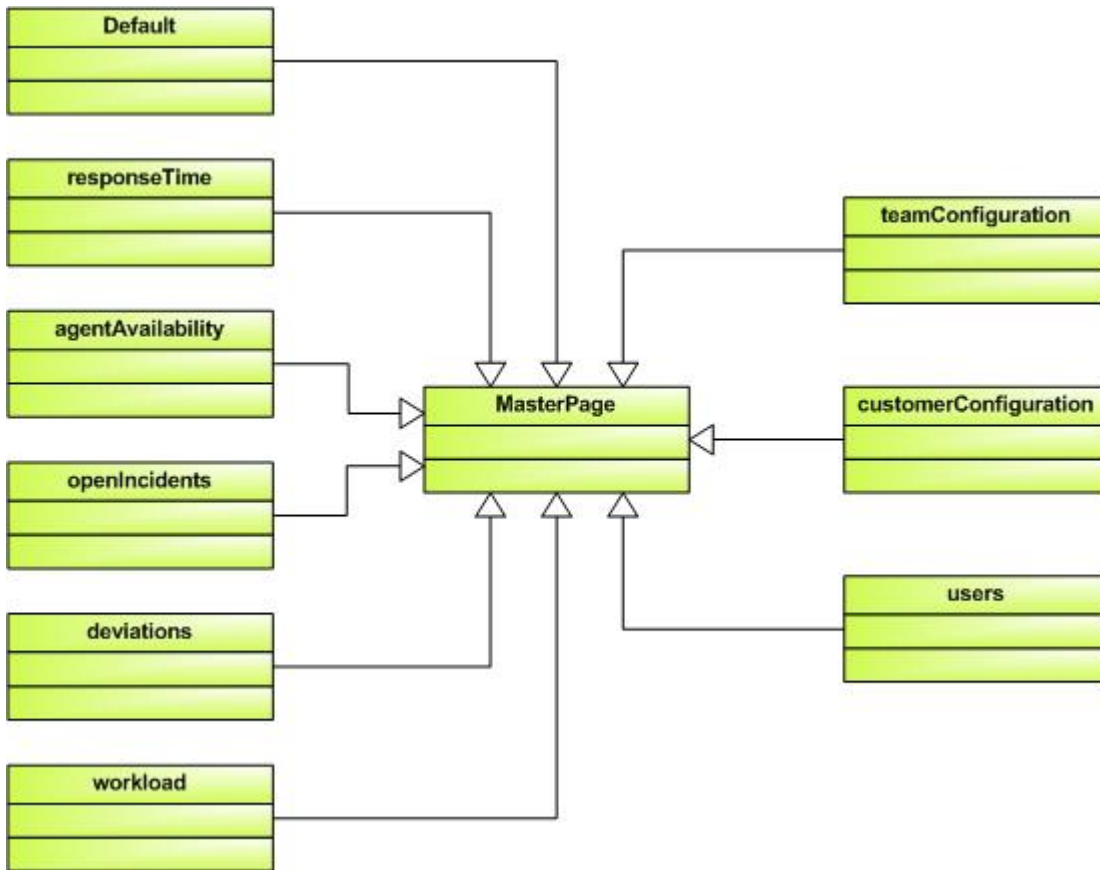
### I.4 DFD 0:



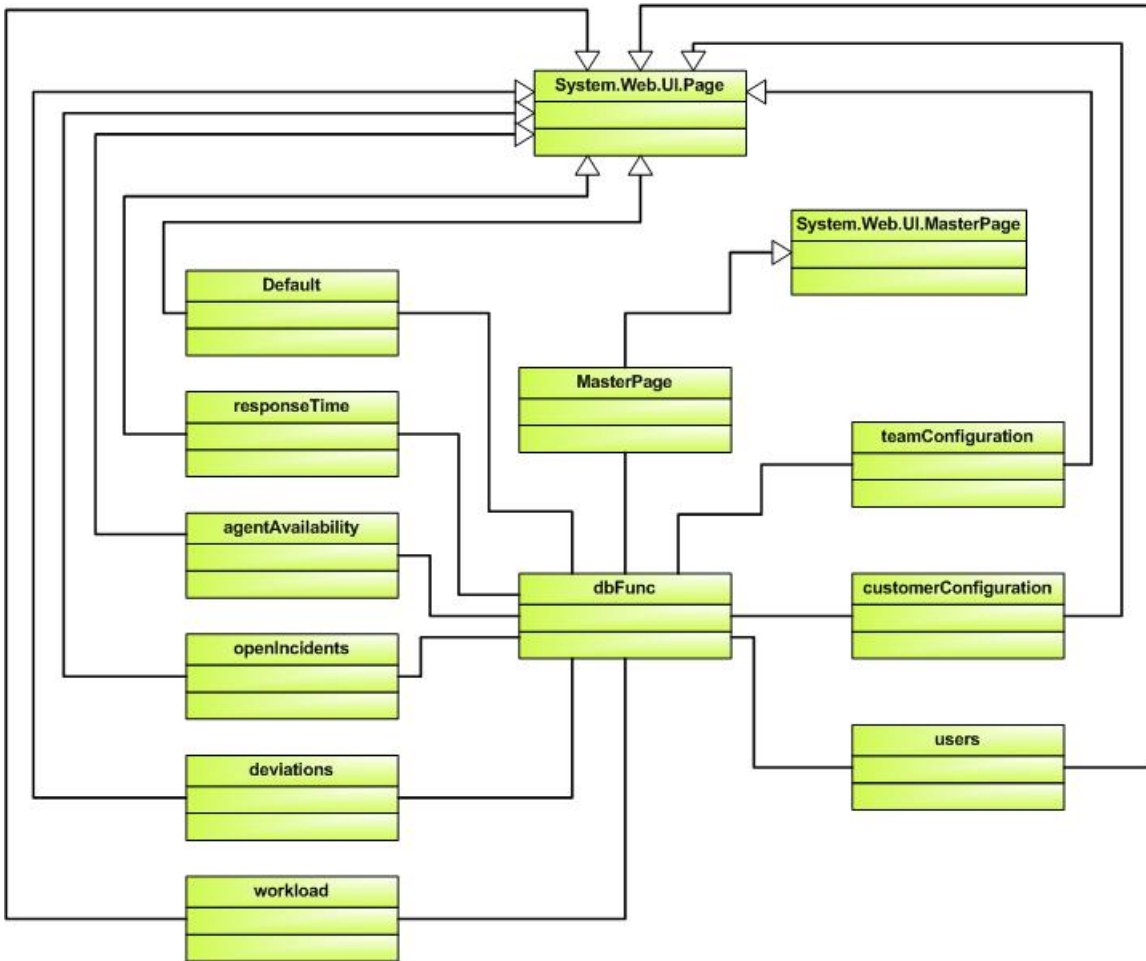
I.5 DFD 1:



### I.6 Class diagram - ASPX



### I.7 Class diagram – C#



**Each class with methods:**

workload
-dbData : dbFunc -date2 : string #Page_Load(in sender : object, in e : EventArgs) -createWorkload(in date : string, in date2 : string, in history : bool) : Table -generateHeadLinesUpper(in date : string, in date2 : string, in history : bool) : TableRow -generateHeadLinesLower(in history : bool) : TableRow -generateTeam(in teamName : string) : TableCell -generateWorkload(in teamID : int, in date : string, in date2 : string, in history : bool) : TableRow -createRow(in name : string, in seven : int, in eleven : int, in fifteen : int, in nineteen : int, in twentythree : int, in history : bool, in colored : bool, in numDays : int) : TableRow -createHistory() #c_dateSelector_SelectionChanged(in sender : object, in e : EventArgs) #c_dateSelector2_SelectionChanged(in sender : object, in e : EventArgs) #c_dateSelector_VisibleMonthChanged(in sender : object, in e : MonthChangedEventArgs)

agentAvailability
#Page_Load(in sender : object, in e : EventArgs) -createAgentAvailability(in date : string, in history : bool) : Table -generateHeadLinesUpper(in date : string, in history : bool) : TableRow -generateHeadLinesLower(in history : bool) : TableRow -generateTeam(in teamName : string) : TableCell -generateAgents(in dsAgents : DataSet, in numAgents : int, in date : string, in history : bool) : TableCell -generateCurrentStatus(in currentStateID : int, in currentStateName : string) : TableCell -createHistory() #c_dateSelector_SelectionChanged(in sender : object, in e : EventArgs) #c_dateSelector_VisibleMonthChanged(in sender : object, in e : MonthChangedEventArgs)

customerConfiguration
-dbData : dbFunc #Page_Load(in sender : object, in e : EventArgs) -createTeams(in date : string) -createAddCustomers(in date : string) -createRemoveCustomers(in date : string, in teamID : int) -fixCheckBoxes() #rbl_RemoveTeams_SelectedIndexChanged(in sender : object, in e : EventArgs) #b_removeFromTeam_Click(in sender : object, in e : EventArgs) #b_addToTeam_Click(in sender : object, in e : EventArgs) #cb_delete1_CheckedChanged(in sender : object, in ea : EventArgs)

dbFunc
-dbSnapshot : Database
+dbFunc() +getTeams(in date : string) : DataSet +getAllTeams(in date : string) : DataSet +getTeamsThisMonth(in date : string) : DataSet +getTeamsBetweenDates(in date1 : string, in date2 : string) : DataSet +getNumCust(in teamID : int, in date : string) : int +getNumAgents(in teamID : int, in date : string) : int +getNumAgentsThisMonth(in teamID : int, in date : string) : int +getAvgTeamResponseTime(in teamID : int, in date : string) : DataSet +getAvgCustResponseTime(in customerID : int, in date : string) : DataSet +getCustomersDay(in teamID : int, in date : string) : DataSet +getCustomersWeek(in teamID : int, in date : string) : DataSet +getCustomersMonth(in teamID : int, in date : string) : DataSet +getAgents(in teamID : int, in date : string) : DataSet +getAgentsThisMonth(in teamID : int, in date : string) : DataSet +getNumAgentsInState(in teamID : int, in date : string) : DataSet +getNumAgentsLoggedIn(in agentInState : DataSet) : int +getNumOpenIncidentsForTeam(in teamID : int, in date : string) : int +getOpenIncidentsForTeam(in teamID : int, in date : string) : DataSet +getDeviationDataForTeam(in teamID : int, in date : string) : DataSet +getDeviationDataForTeamHistory(in teamID : int, in date : string) : DataSet +getDeviationDataForAgent(in agentID : int, in date : string) : DataSet +getDeviationDataForAgentHistory(in agentID : int, in date : string) : DataSet +getInCallData(in teamID : int, in date : string) : DataSet +getAgentAvailabilityData(in agentID : int, in date : string) : DataSet +getAgentAvailabilityDataHistory(in agentID : int, in date : string) : DataSet +getInCallDataHourly(in date : string, in teamID : int) : DataSet +getAnsweredCallDataHourly(in teamID : int, in date : string) : DataSet +getAnsweredCallDataHourlyHistory(in teamID : int, in date1 : string, in date2 : string) : DataSet +getAbandonCallDataHourly(in teamID : int, in date : string) : DataSet +getAbandonCallDataHourlyHistory(in teamID : int, in date1 : string, in date2 : string) : DataSet +getAgentsLoggedInDataHourly(in teamID : int, in date : string) : DataSet +getAgentsLoggedInDataHourlyHistory(in teamID : int, in date1 : string, in date2 : string) : DataSet +getAgentsNotInTeam(in date : string) : DataSet +deleteTeam(in date : string, in teamID : int) : int +validate(in un : string) : DataSet +createTeam(in date : string, in teamName : string, in telephoneTeam : int) : int +addAgentToTeam(in date : string, in teamID : int, in agentID : int) : int +addCustomerToTeam(in date : string, in teamID : int, in customerID : int, in custRelStarted : string) : int +removeAgentFromTeam(in date : string, in teamID : int, in agentID : int) : int +getCustomersNotInTeam(in date : string) : DataSet +getCustomersInAllTeams(in date : string) : DataSet +getCustomersInTeam(in date : string, in teamID : int) : DataSet +removeCustomerFromTeam(in date : string, in teamID : int, in customerID : int) : int +getUsers() : DataSet +getModules() : DataSet +getUsersModules(in userID : int) : DataSet +deleteUsersModules(in userID : int) : int +deleteUser(in userID : int) : int +addUserModule(in userID : int, in moduleID : int) : int +addUser(in userName : string, in fullName : string) : int +checkIfUsersAdded(in userName : string) : int +getUserID(in userName : string) : int +testDatabase(in date : string) : string

Default
-dbData : dbFunc -date : string -dsTeam : DataSet
#Page_Load(in sender : object, in e : EventArgs) -CreateMain() -generateHeadLinesUpper(in date : string, in history : bool) : TableRow -generateHeadLinesLower(in history : bool) : TableRow -generateTeam(in teamName : string) : TableCell -generateCustomers(in dsCustomers : DataSet, in numCust : int, in avgTeamResponseTime : int) : TableCell -generateAgentState(in teamID : int, in numCust : int, in agentInState : DataSet) : TableCell -generateOpenIncidents(in teamID : int) : TableCell -generateDeviations(in teamID : int, in numCust : int) : TableCell -generateWorkload(in teamID : int, in dsAgentInState : DataSet) : TableCell

<b>deviations</b>
-dbData : dbFunc
#Page_Load(in sender : object, in e : EventArgs) -createDeviaton(in date : string, in history : bool) : Table -generateHeadLinesUpper(in date : string, in history : bool) : TableRow -generateHeadLinesLower(in history : bool) : TableRow -generateTeam(in teamName : string) : TableCell -generateAgents(in dsAgents : DataSet, in numAgents : int, in date : string, in history : bool, in teamID : int) : TableCell -createHistory() #c_dateSelector_SelectionChanged(in sender : object, in e : EventArgs) #c_dateSelector_VisibleMonthChanged(in sender : object, in e : MonthChangedEventArgs)

<b>MasterPage</b>
-dbData : dbFunc -sep : char -sep2 : char
#Page_Load(in sender : object, in e : EventArgs) -fixLinkAppearance(in pageName : string) -showButtons(in dsPages : DataSet) -removeAllButtons() -isViewAble(in dsPages : DataSet, in pagename : string) : bool -getCurrentUser() : string

<b>openIncidents</b>
-dbData : dbFunc
#Page_Load(in sender : object, in e : EventArgs) -createOpenIncidents(in date : string, in history : bool) -addDataToSC(in scData : SeriesCollection, in openIncidents : int, in teamName : string, in i : int) : SeriesCollection -getOpenIncidentValues(in dsOpenIncidents : DataSet, in param : string) : int -createHistory() -setUpCharts() #c_dateSelector_SelectionChanged(in sender : object, in e : EventArgs) #c_dateSelector_VisibleMonthChanged(in sender : object, in e : MonthChangedEventArgs)

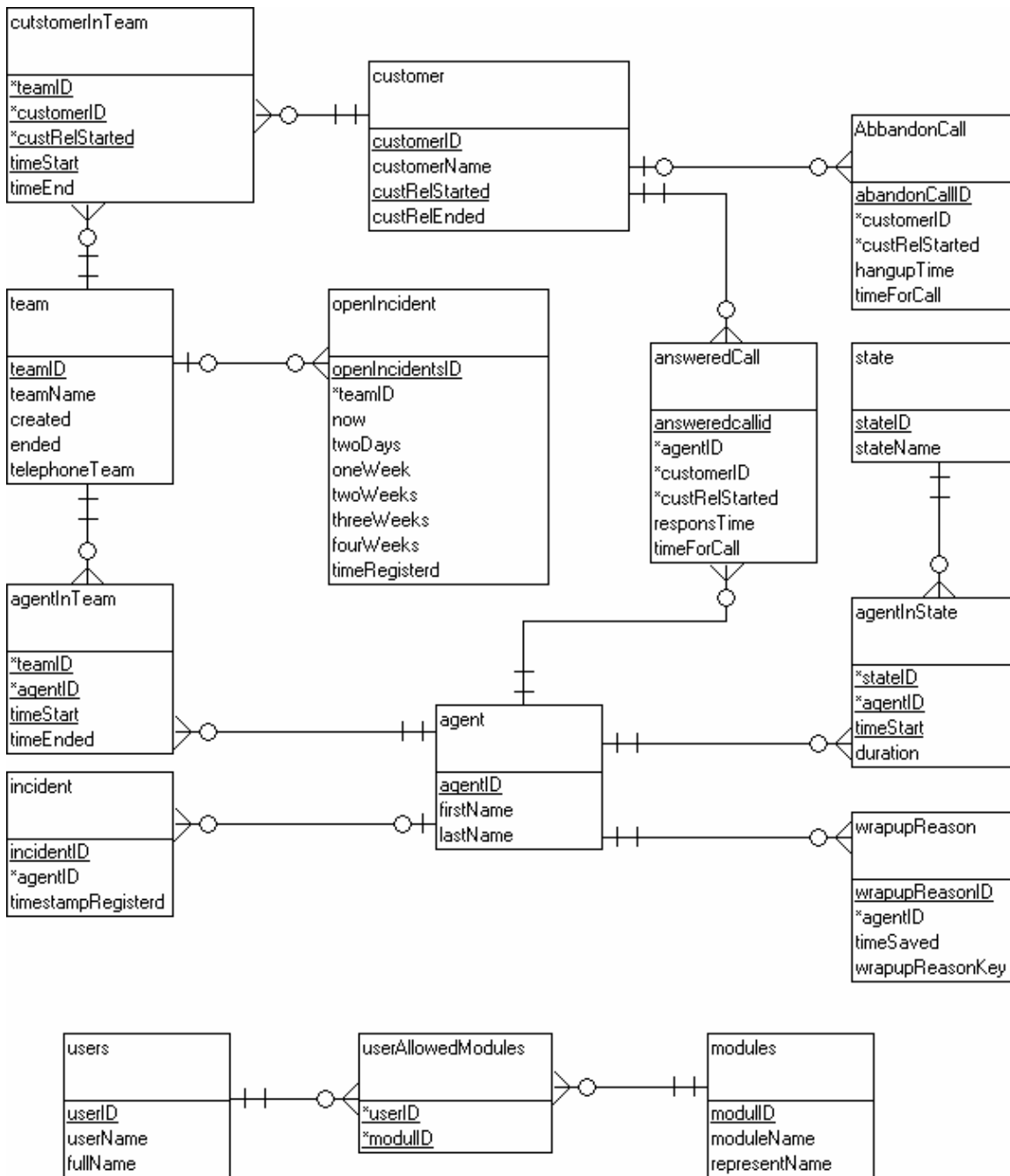
<b>responseTime</b>
-localDate : string -dbData : dbFunc
#Page_Load(in sender : object, in ea : EventArgs) -createResponseTimeDep(in date : string, in dsTeams : DataSet) -createResponseTimeDepNow(in date : string, in dsTeams : DataSet) -createResponseTimeDepWeek(in date : string, in dsTeams : DataSet) -createResponseTimeDepMonth(in date : string, in dsTeams : DataSet) -createDepartmentHistory() -getAvgResponseTimeDep(in date : string, in dsTeams : DataSet, in periode : string) -createResponseTimeTeam(in teamID : int, in teamName : string, in date : string) -createResponseTimeTeamNow(in teamID : int, in teamName : string, in date : string) -createResponseTimeTeamWeek(in teamID : int, in teamName : string, in date : string) -createResponseTimeTeamMonth() -createTeamHistory() -getAvgResponseTimeTeam(in date : string, in dsCustomers : DataSet, in periode : string) -generateRadioButtons(in dsTeams : DataSet) -generateRadioButtonsHistory() #rb_teamsNow_SelectedIndexChanged(in sender : object, in eArg : EventArgs) #rb_teamsHistory_SelectedIndexChanged(in sender : object, in eArg : EventArgs) #c_dateSelector_SelectionChanged(in sender : object, in e : EventArgs) #c_dateSelector_VisibleMonthChanged(in sender : object, in e : MonthChangedEventArgs)

<b>teamConfiguration</b>
-dbData : dbFunc
#Page_Load(in sender : object, in e : EventArgs) -createTeams(in date : string) -createAddAgents(in date : string) -createRemoveAgents(in date : string, in teamID : int) -fixCheckBoxesDelete() -fixCheckBoxesRemove() #b_addToTeam_Click(in sender : object, in e : EventArgs) #b_removeFromTeam_Click(in sender : object, in e : EventArgs) #b_createTeam_Click(in sender : object, in e : EventArgs) #b_deleteTeam_Click(in sender : object, in e : EventArgs) #rbl_RemoveTeams_SelectedIndexChanged(in sender : object, in e : EventArgs) #cb_delete1_CheckedChanged(in sender : object, in ea : EventArgs) #cb_removeCust1_CheckedChanged(in sender : object, in ea : EventArgs)

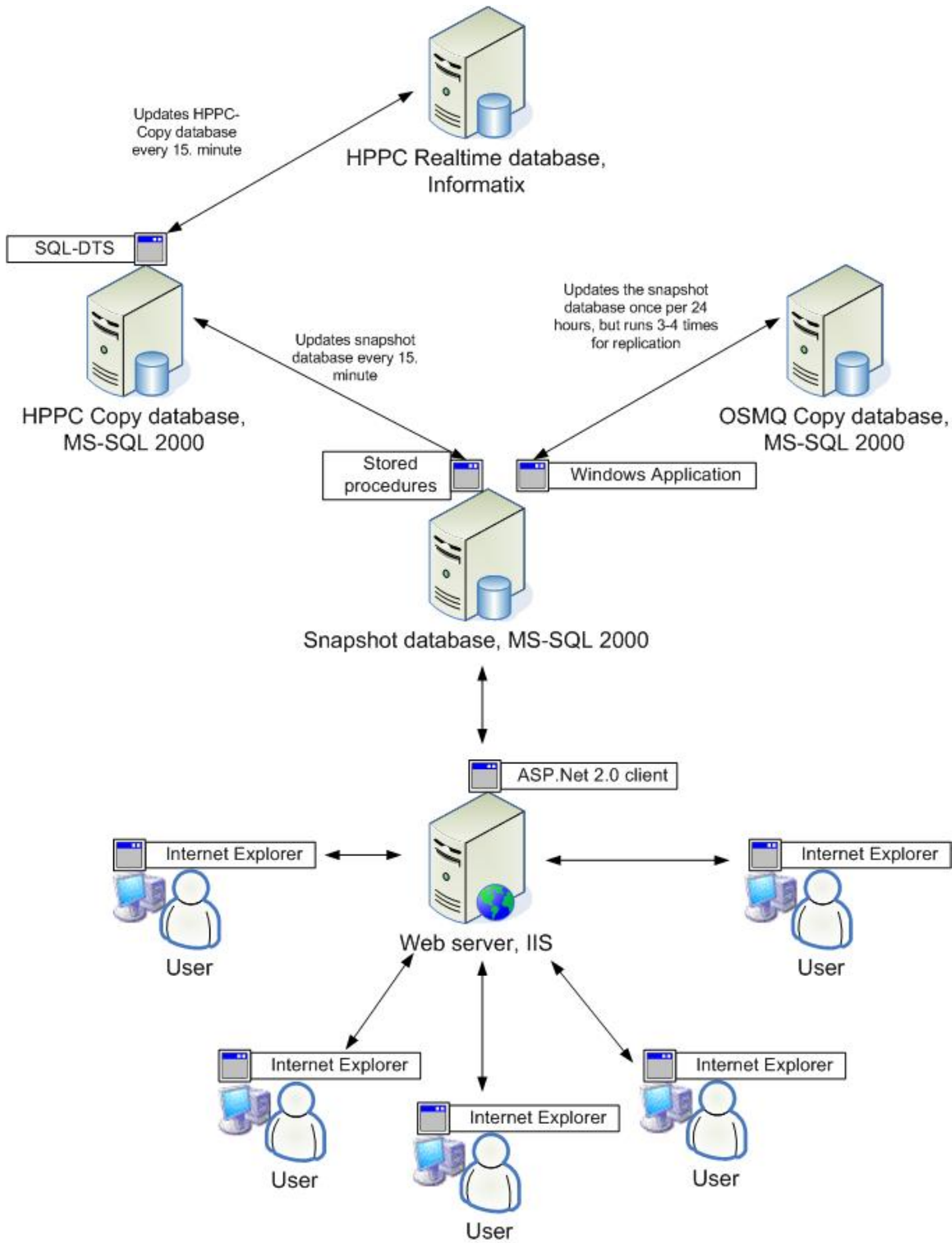
<b>users</b>
-dbData : dbFunc
#Page_Load(in sender : object, in e : EventArgs) -createModules() -createPossibleUsers() -createTestPossibleUsers() : ArrayList -createCurrentUsers() -fixCheckBoxes() -getAllUsers() : ArrayList #cb_delete1_CheckedChanged(in sender : object, in ea : EventArgs) #ddl_currentUsers_SelectedIndexChanged(in sender : object, in ea : EventArgs) #ddl_possibleUsers_SelectedIndexChanged(in sender : object, in ea : EventArgs) #b_alterUser_Click(in sender : object, in ea : EventArgs) #b_addUser_Click(in sender : object, in ea : EventArgs) #b_deleteUser_Click(in sender : object, in ea : EventArgs)



### I.8 Entity Relationship diagram



### I.9 System Overview



## Appendix J: Detailed Development Description

### J.1 Class diagram

We were both new at ASP.net development, and started with finding examples for ASP.net class diagrams. It was not easy to find any good examples on ASP.net class diagrams. All of those we could were only showing code classes that where not a part of the aspx pages.

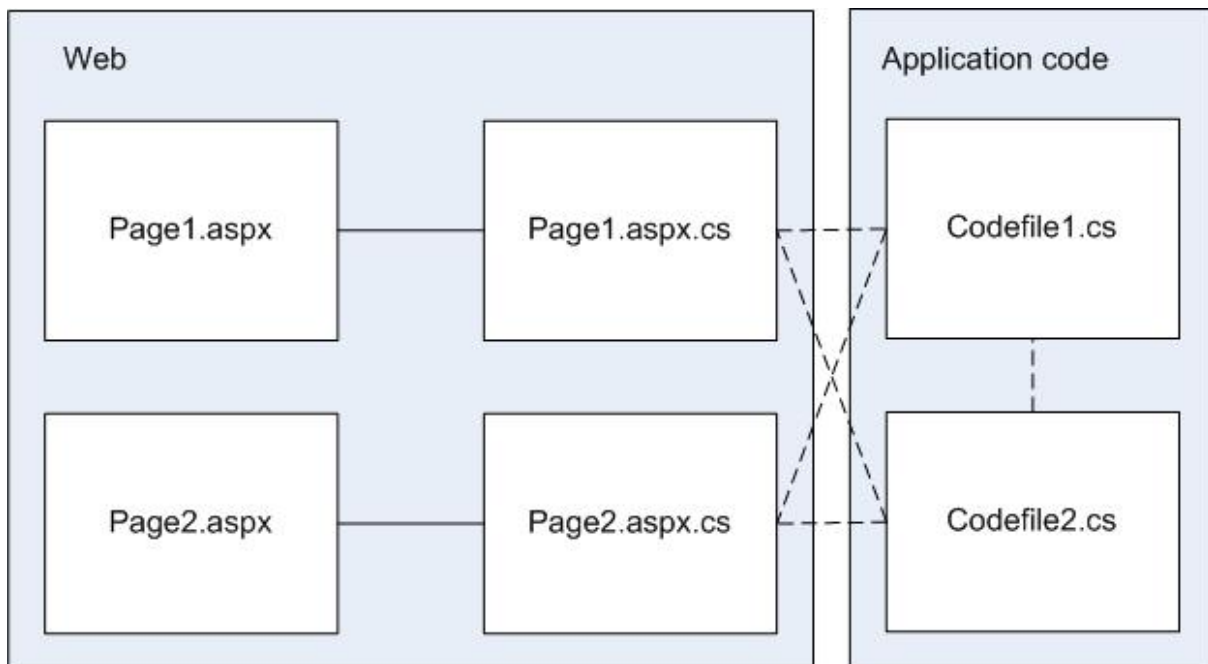


Figure 1, ASP.net code file communication (C# example).

As we can see in figure 1 there are files containing C# code (files with last name of .cs) in the web section and in the application code section. The files containing C# code in the web section are directly involved with the information on the website. The files in the application code section are communicating with the web section code files, but they do not have any direct connection to controls on the website.

The example class diagrams we found did only contain the files (classes) in the application section. Our application is a type of application where most of the logic is in the aspx.cs files in the web section. The application section had very few files, only three when we started the project. But still, we created a class diagram with only three files.

We changed the class diagram several times during the development and the three classes in the application section was reduced to one. The class diagram made even less sense than in the beginning of the development.

We found out that a class diagram with all code files (classes), even those from the web section, would give us much more meaning.

We created our own type of class diagram containing all the code files, and the connections between them See appendix I.

This diagram became pretty large, so we removed all attributes and methods and create a smaller diagram for each class showing all attributes and methods.

We did also create a class diagram for the ASP.net files showing the relationship between them and the master page

## **J.2 Web application**

The web application was the biggest developing task.

We did not have any experience with C# or ASP.net before we started on this thesis. We created several small test applications with an objective to test and learn how different features worked before we used it in the web application.

We started to develop the main page, which was the only page in the user system. It was developed as a single website only containing one information table. When the main page was finished we started the first page in the management system, and so on, until all pages in the management system was finished.

We had several pages with content, but link between them. It was time to create the masterpage. The masterpage contains html layout (where thing are placed on the page), such as menu (for linking the pages together), and some logic for authentication which made sure that it was impossible to access a page which the user wasn't authorized for and that only the authorized pages was shown in the menu.

When the master page was finished we started to add each of previous developed user and management pages with a menu item and authorization data to the master page, building the website.

Together with the masterpage, we did also create a login page with a username and password prompt.

All authorized users gets a user name and password which from the page administrator. In the initial state of the website, before the user logged in, there are only two possible page choice in the menu; main, and log in. When the user is logging in, the menu changes and the pages that user was authorized for are shown in the menu.

We had discussed this matter around passwords with them during the user requirement specification. But after talking with the SBS database expert we were told that they had a rule against saving password and username other places than in the Active Directory. We had to redo and change our login system.

The new system was going to use the Active Directory to retrieve usernames and authorize on them. This meant that there is no need to log in on the website to se the pages you are allowed to see. If the user is logged in on the computer with username and password, the user will automatically see the pages that he is authorized for in the menu when entering the website.

This was implemented and developed in two steps because of the access to the Active Directory. We developed the system which authorized on username on our own lap tops, and the system for retrieving the username from the Active Directory on a SBS computer. We did not have access to get the username from LDAP at SBS from our laptops were we developed everything else.

The page was now working as intended, with our test usernames. But there was no possibility to add users, add or delete a team, add an agent to a team, or add a customer to a team, without writing an SQL query. To make it possible for the regular user to do these actions we had to create the admin system.

We spent some time on finding the best controls to use in the admin system. Boxes where items were moved between them, radio buttons, or three view. We ended up with radio buttons, which was something we agreed on with SBS. One of the reasons for using radio buttons was that they were used in the management system.

There were three admin pages, and were created simultaneously, since they were using the same type of logic. We did not use the same method as before, with creation in a single page, instead they were created directly into the masterpage system.

### **J.3 GUI**

When we developed the masterpage, it was also focus on the website's GUI. All HTML and placement of objects was done in the masterpage.

Colors, sizes, borders, link styles, and all other graphical elements were done in a cascading style sheet.

The website was designed after the style guides from Siemens. The style guide has rules for colors, fonts, logos, table sizes and so on.

The menu we chose, page tabs, was not a part of the Siemens style guide. We used fonts and colors from the style guide, but the presentation of the menu was decided together with the SBS web designer. She gave us a lot good and constructive of feed back, and even helped us out with the testing.

### **J.4 Windows application**

A Windows Service was supposed to be used, in the original system design, for collecting data from the HPPC real time and the OSMq copy databases and store this at the snapshot database.

We stated to develop the Windows Service against a backup copy of the OSMq database on our laptops. It worked with our local copy of the OSMq database, but when we tried it at the SBS network there was some access problems. This was sorted out pretty quick by the SBS database expert. The Windows Service had to be registered with the system and all logging was done to the event application log.

We had bigger problems with the HPPC database. We were not allowed to have any stored procedures at the server, but having all the SQL queries in the Windows Service would not create any big problems. The problems came when we tried to get access to the HPPC database. The Informix database did not support OLE DB, and was not open for ODBC connections from all machines. If they opened ODBC for us, we could be hold responsible any crash or other incidents that could occur. If the HPPC system was offline for a short amount of time, this could cost SBS a lot of money.

When we were about to deploy the Window Service out the server, we were told that Windows Services weren't very popular at SBS. They wanted us to create a GUI less Windows application which could be started as a scheduled program instead. The GUI less Windows application is much easier to control, and launching times are controlled by the system admin, not by the application itself, as in a Windows Service. Since we had the most of the data retrieval code ready, this wasn't any big job. The porting was finished in less than one day.

One of the changes we had to do was to create a new logging feature where we didn't use the Event application log which our Windows Service did but instead time stamp errors and write them to a text file.

The windows application was a bit easier to deploy as well, no need to register it with the host computer which needs to be done with a Windows Service.

## J.5 Database, stored procedures and DTS

The first version of the database was created soon after the Snapshot ER diagram was proclaimed OK by the SBS database expert.

The database was written in SQL as a creation script which could be executed every time we wanted a fresh instance of the database.

The database was filled with test information for four regular work days, which was used during the development of the rest of the system.

[20] This would make it easy to reuse code, and it would let them see what kind of data we used in the database.

We created stored procedures for all data retrieval and to insert data on the Snapshot database and for all data retrieval from the OSMq database.

The stored procedures were created while we programmed when we needed them, after the extreme programming mentality. There was also made changes to them while the programming was going on. If two procedures were doing almost the same job, they were joined and used as one if possible.

There was a lot of fixing up on the stored procedures when they were deployed and tested on real data for the first time. There was linguistics problems and some date errors, but nothing that couldn't be fixed in reasonable time

To retrieve data from the HPPC real time database, as we mentioned earlier, created some problems. But we were lucky and the SBS database expert helped us. He had a DTS [21] job going for retrieving data from the real time database and to the copy database once a day. He changed this DTS so that it would run every 15<sup>th</sup> minute and included the data we needed to the Snapshot database.

Together we planed to dump data from the HPPC copy database, into temp tables in our database, using the same DTS as he used for retrieving from the HPPC real time database.

But once again we were lucky, our database and the HPPC copy database was placed on the same database server. That meant that we could use stored procedures to retrieve data from the HPPC copy database and insert the data directly into to our database without using temp tables. This was much quicker, and there was no need to worry about double storage of data.

## J.6 Bug fixing

During the whole development we spent time on bug fixing. We had a philosophy for bugs which was to fix them right away. If they are “saved for later” it is very easy to forget them, and the bug will still be there when deploying.

We found it easier to stop the current work and fix bugs, than keep on working and go back to fix bugs later.

We found a lot of bugs and problems during the evaluation of the system. Most of the problems were of a character that was fixed quickly.

We did also discover two major database problems for SBS.

When we tried to deploy the Windows application we discovered over 1700 duplicated listings in their OSMq copy database. The duplication had been going on for a while and occurred when they failed retrieved data from the OSMq and had to be started over. Some data was already retrieved, but it was fetched again.

We had to rewrite our queries to the database to be resistant to duplicates, since SBS didn't have time to fix this right away.

We did also discover duplicated listing in the HPPC copy database. This was no actual error, but it was not described in the HPPC manual. There could be more duplicates to, but they were hard to find in older data where they didn't have all the data fields we needed to the

Snapshot database. The SBS database expert took care of this and removed duplicates so it was possible for us to use the database.



## Appendix K File Lists

System parts	Lines	Number of files
WebApplication	5575	25
Windows Application	475	5
Stored Procedures	5107	78
<b>Sum</b>	<b>11157</b>	<b>108</b>

### K.1 Web Application

Website class, code, and style files					
FileID	FileName	Description	Location	Size (lines)	Finished
1001	dbFunc.cs	The interface between the snapshot database and the web application. All functions that are using the snap shot databases are collected in this class.	~/App_Code/	857	02.05.06
1002	default.css	The Cascading Style Sheet that are holding all appearance information to the website. Everything except the calender appearance is defined in this file.	~/App_Themes/Default	488	02.05.06
1003	agentAvailability.aspx	The file that holds all HTML/ASP information for the "Agent Availability" page. Mostly presenting data in real time and history in tables.	~/	44	02.05.06
1004	agentAvailability.aspx.cs	The code file which generates data to the agentAvailability.aspx file. All information used is retrieved trough dbFunc.cs. The data is generated into tables which are displayed on the web.	~/	408	02.05.06
1005	customerConfiguration.aspx	The file that holds all HTML/ASP information for the "Customer conf." page. The users can add or delete customers in team by using radiobuttons and buttons.	~/	89	02.05.06
1006	customerConfiguration.aspx.cs	The code file which generates data to the customerConfiguration.aspx file. All information used is retrieved or sent trough dbFunc.cs. Customers are added or removed from a Team.	~/	212	02.05.06
1007	Default.aspx	The file that holds all HTML/ASP information for the "Main" page. This is also the page the user automaticly is forwarded to when going in to the site for the first time.	~/	14	02.05.06
1008	Default.aspx.cs	The code file which generates data to the Default.aspx file. All information used is retrieved or sent trough dbFunc.cs and treated in this file before presented to web in one table.	~/	509	02.05.06

1009	deviations.aspx	The file that holds all HTML/ASP information for the "Deviations" page. The data is presented in two tables, one realtime and one for history.	~/	44	02.05.06
1010	deviations.aspx.cs	The code file which generates data to the deviations.aspx file. All information used is retrived or sent trough dbFunc.cs and treated in this file before presented to web in a table.	~/	404	02.05.06
1011	MasterPage.master	The main HTML/ASP content page. The whole frame for the page is created here. All the other .aspx files uses this as a framework	~/	53	02.05.06
1012	MasterPage.master.cs	The code file which contains all page logic. All user access and verification is done here.. All information used is retrived or sent trough dbFunc.cs and treated in this file before presented to web	~/	205	02.05.06
1013	openIncidents.aspx	The file that holds all HTML/ASP information for the "Open Incidents" page. All information used is retrived or sent trough dbFunc.cs and treated in this file before presented to web. Information is presented in dotNetCharting graphs	~/	134	02.05.06
1014	openIncidents.aspx.cs	The code file which generates data to the openIncidents.aspx file. All information used is retrived or sent trough dbFunc.cs and treated in this file before presented to web	~/	283	02.05.06
1015	responseTime.aspx	The file that holds all HTML/ASP information for the "Response Time" page. Displays a graph for the department with all the teams response times, and graphs for a selected team with response times.	~/	120	02.05.06
1016	responseTime.aspx.cs	The code file which generates data to the responseTime.aspx file. All information used is retrived or sent trough dbFunc.cs and treated in this file before presented to web. Information is presented in dotNetCharting graphs.	~/	423	02.05.06
1017	teamConfiguration.aspx	The file that holds all HTML/ASP information for the "Team/Agent conf" page. Where the user can add or remove agents to or from a team, and create or delete a team.	~/	137	02.05.06
1018	TeamConfiguration.aspx.cs	The code file which generates data to the customerConfiguration.aspx file. All information used is retrived or sent trough dbFunc.cs.	~/	252	02.05.06
1019	users.aspx	The file that holds all HTML/ASP information for the "Users conf" page. The "superior" users can add or remove other users access to the system.	~/	63	02.05.06

1020	users.aspx.cs	The code file which generates data to the users.aspx file. All information used is retrieved or sent through dbFunc.cs.	~/	297	02.05.06
1021	workload.aspx	The file that holds all HTML/ASP information for the "Users conf" page. The number of incoming calls, answered calls and agents logged in in sequences.	~/	79	02.05.06
1022	workload.aspx.cs	The code file which generates data to the workload.aspx file. All information used is retrieved or sent through dbFunc.cs.	~/	387	02.05.06
1022		Sum		5502	lines

## K.2 Windows Application

Windows application code and config files				
FileID	FileName	Description	Size (lines)	Finished
2001	OSMqReader.Desginer.cs	Create the GUI, and make sure it is not visible	43	03.05.06
2002	OSMqReader.cs	Updating the snapshot database with data from the OSMqExtract database. Runs 3-4 times every day, but data will only be transferred once if nothing special happens	204	03.05.06
2003	dbFunc.cs	Holding all functions that interacts with the databases.	195	05.05.06
2004	Program.cs	The main method that are executed at startup	18	03.05.06
2005	App.config	Holds connectionstrings to the databases.	15	03.05.06
2005		SUM	475	

### K.3 Database

Stored Procedures for the Snapshot Database				
ProcID	Stored Procedure Name	Description	Size (lines)	Finished
3001	SNAPSHOT_addAgentToTeam.sql	Add an agenet to a team	27	03.05.06
3002	SNAPSHOT_addCustomerToTeam.sql	Adds a customer to a team	27	03.05.06
3003	SNAPSHOT_addUser.sql	Adds a user	26	03.05.06
3004	SNAPSHOT_addUserModule.sql	Adds a user module	25	03.05.06
3005	SNAPSHOT_checkIfUserIsAdded.sql	Returns 1 if user already is added to the database and 0 if it is a new user	35	03.05.06
3006	SNAPSHOT_createTeam.sql	Inserts Data into a Team	28	03.05.06
3007	SNAPSHOT_deleteAgentsInTeam.sql	Adds an End.date to all agents in a team	28	03.05.06
3008	SNAPSHOT_deleteCustomersInTeam.sql	Adds an End.date to all customer in a team	27	03.05.06
3009	SNAPSHOT_deleteTeam.sql	Adds an End date to the team	28	03.05.06
3010	SNAPSHOT_deleteUser.sql	Deletes an users	27	03.05.06
3011	SNAPSHOT_deteUsersModules.sql	Deletes a users modules	26	03.05.06
3012	SNAPSHOT_getAbandonedCallDataForTeamHourly.sql	Returns Abandoned calls in sequences of 4 hours, arg_date is sent in as yyyy-mm-dd 00:00:00	63	03.05.06
3013	SNAPSHOT_getAbandonedCallDataForTeamHourlyHistory.sql	Returns Abandoned calls in sequences of 4 hours between two dates	72	03.05.06
3014	SNAPSHOT_getAgentAvailabilityData.sql	Returns Agent Availability data	112	03.05.06
3015	SNAPSHOT_getAgentAvailabilityDataHistory.sql	Returns Agent Availability data over a month	55	03.05.06
3016	SNAPSHOT_getAgents.sql	Gets all agents	21	03.05.06
3017	SNAPSHOT_getAgentsInTeam.sql	Returns the agents in a team at a given date	30	03.05.06
3018	SNAPSHOT_getAgentsInTeamThisMonth.sql	Returns the agents in a team that month	28	03.05.06
3019	SNAPSHOT_getAgentsLoggedInHourly.sql	Returns the number of agents logged in in sequences of 4 hours	62	03.05.06
3020	SNAPSHOT_getAgentsLoggedInHourlyHistory.sql	Returns agents logged in in sequences of 4 hours between two dates	74	03.05.06
3021	SNAPSHOT_getAgentsNotInTeam.sql	Returns a set of agents which are not member of any team	30	03.05.06
3022	SNAPSHOT_getAnsweredCallDataForTeamHourly.sql	Returns Answered calls in sequences of 4 hours	53	03.05.06
3023	SNAPSHOT_getAnsweredCallDataForTeamHourlyHistory.sql	Returns agents logged in in sequences of 4 hours between two dates	69	03.05.06
3024	SNAPSHOT_getAvgResponseTime.sql	gets the average responsetime for a customer	41	03.05.06
3025	SNAPSHOT_getAvgTeamResponseTime.sql	Returns the average response time for a team for a given date, that week, and that month	65	03.05.06
3026	SNAPSHOT_getCustomersByTeamID.sql	Returns customerid and customername by team id	90	03.05.06
3027	SNAPSHOT_getCustomersByTeamIDDay.sql	Returns customerid and customername by team id	36	03.05.06

3028	SNAPSHOT_getCustomersByTeamIDMonth.sql	Returns customerid and customername by team id that has been a member that month	35	03.05.06
3029	SNAPSHOT_getCustomersByTeamIDWeek.sql	Returns customerid and customername by team id that has been a member that week	35	03.05.06
3030	SNAPSHOT_getCustomersInAllTeams.sql	Returns a set of customers which are member of any team	34	03.05.06
3031	SNAPSHOT_getCustomersInTeam.sql	Get customers in a team right now	30	03.05.06
3032	SNAPSHOT_getCustomersNotInTeam.sql	Returns a set of customers which are not member of any team	34	03.05.06
3033	SNAPSHOT_getDeviationDataForAgent.sql	returns the number of calls, number of wrapupreasons and the number of registered incidents for an agent at a date	50	03.05.06
3034	SNAPSHOT_getDeviationDataForAgentHistory.sql	returns the number of calls, number of wrapupreasons and the number of registered incidents for an agent at a month	46	03.05.06
3035	SNAPSHOT_getDeviationDataForTeam.sql	returns the number of calls, number of wrapupreasons and the number of registered incidents	48	03.05.06
3036	SNAPSHOT_getDeviationDataForTeamHistory.sql	returns the number of calls, number of wrapupreasons and the number of registered incidents for a month	51	03.05.06
3037	SNAPSHOT_getInCallDataForTeam.sql	Returns incomming Call data for a team	38	03.05.06
3038	SNAPSHOT_getInCallDataForTeamHourly.sql	Returns incomming Call data for a team in a periode for 2 hours	38	03.05.06
3039	SNAPSHOT_getIncommingCallDataForTeam.sql	Returns incomming call data for a team for one day	38	03.05.06
3040	SNAPSHOT_getMaxIncidentDate.sql	Returns the latest date any incidents are registered in the database	23	03.05.06
3041	SNAPSHOT_getMaxOpenIncidentTeamDate.sql	Returns the latest date any open incidents are registered in the database on a team	31	03.05.06
3042	SNAPSHOT_getModules.sql	Gets all modules in the database	23	03.05.06
3043	SNAPSHOT_getNumAgentsForTeam.sql	Returns the number of agents in a team at a given date	27	03.05.06
3044	SNAPSHOT_getNumAgentsForTeamThisMonth.sql	Returns the number of agents in a team the month of a given date	29	03.05.06
3045	SNAPSHOT_getNumAgentsInSateForTeam.sql	Returns the number of agents currently in a state With 15 minutes lag	73	03.05.06
3046	SNAPSHOT_getNumCustByTeamID.sql	Returns the number of customer for a team where the teamID is supplied	30	03.05.06
3047	SNAPSHOT_getOpenIncidentsByTeamIDAndDate.sql	returns open incidents for a given team at a given date	26	03.05.06
3048	SNAPSHOT_getTeams.sql	Returns the name of all teams which are in the database not takeing any notice if it is a phoneteam	30	03.05.06

3049	SNAPSHOT_getTelephoneTeams.sql	gets all teams that are telephoneteams	27	03.05.06
3050	SNAPSHOT_getTelephoneTeamsBetweenDates.sql	gets all teams that are telephoneteams between two dates	31	03.05.06
3051	SNAPSHOT_getTelephoneTeamsThisMonth.sql	gets all teams that are telephoneteams this month	28	03.05.06
3052	SNAPSHOT_getUserID.sql	Gets a users module	26	03.05.06
3053	SNAPSHOT_getUsers.sql	Gets all users in the database	24	03.05.06
3054	SNAPSHOT_getUsersModules.sql	Returns all modules allowed for an user	31	03.05.06
3055	SNAPSHOT_getUsersModulesByUsername.sql	Returns all modules allowed for an user	31	03.05.06
3056	SNAPSHOT_insertIncidents.sql	Inserts incidents with Id, wich agent that performed and the timestamp when the incident was registerd.	38	03.05.06
3057	SNAPSHOT_insertOpenIncidents.sql	A function which inserts	52	03.05.06
3058	SNAPSHOT_removeAgentFromTeam.sql	Adds an end date to an agent in a team	34	03.05.06
3059	SNAPSHOT_removeCustomerFromTeam.sql	Adds an end date to a customer in a team	32	03.05.06
3060	HPPC_updateAbandonCall.sql	updates the snapshot tables with data from the HPPC abandoncall table	51	03.05.06
3061	HPPC_updateAgentInState.sql	Updates the agentInState data in the snapshot database with information from HPPC_answeredcalls and HPPC_agentstatereason	75	03.05.06
3062	HPPC_updateAgents.sql	Inserts new agents in the snapshot database	36	03.05.06
3063	HPPC_updateAnsweredCall.sql	Updates and inserts answered call data in the snapshot database with data from the HPPC Calldata table	51	03.05.06
3064	HPPC_updateCustomers.sql	Update customers in the snapshot database with info from the HPPC calltype table	53	03.05.06
3065	HPPC_updateData.sql	Starts all the other update proceduers for updating the snapshot tables with data from the HPPC tables	36	03.05.06
3066	HPPC_updateWrapupReason.sql	Updates the snapshot wrapup reason table with data from the HPPC wrapupReason table	50	03.05.06
3067	CREATE_createUserTables.sql	Creates the user tables in the snapshot database	61	03.05.06
3068	CREATE_createSnapshotTables.sql	Creates the Snapshots tables in the snapshot database	310	03.05.06
3069	CREATE_createHPPCTables.sql	creates the HPPC tables in the snapshot database	224	03.05.06
3070	CREATE_update.sql	Creates a table with only one field for updating with at datetime for the last update.	30	09.05.06
3071	TESTDATA_HPPCData.sql	Fills upp the HPPC-dump tables with data	44	03.05.06
3072	TESTDATA_snapshotData.sql	Fills up the snapshot tables with testdata	1493	03.05.06
3073	TESTDATA_userData.sql	Filles up the user tables with test data	72	03.05.06

3074	initate_snapshot	Initiates the snapshot database with fake data. All values are 0, and the in argument is used as all dates	55	08.05.06
3075	grant_rights	Changes the exec rights on all stored procedures in the snapshot database to be allowed for the SnapShotWriter	25	08.05.06
3075		SUM	4994	

The "SNAPSHOT\_" prefix is removed, but used here to make it easier to see where the file belong

Stored Procedures for the OSMQ copyDatabase				
ProclD	Stored Procedure Name	Description	Size (lines)	Finished
3101	OSMQ_getOpenIncidents.sql *	Gets open incident for a spesific date for a spesific team	57	03.05.06
3102	OSMQ_getIncidents.sql	Gets all incidents from a given date and to today	29	03.05.06
3102		SUM	86	

The "OSMQ\_" prefix is removed, but used here to make it easier to see where the file belong  
\* created by Frank Lyder Hassel

Triggers for the Snapshot copyDatabase				
ProclD	Stored Procedure Name	Description	Size (lines)	Finished
3201	TRIGGER_startUpdate.sql	When a record is inserted into the updated table, the trigger starts the HPPC_update procedure	27	09.05.06
3201		SUM	27	

The "TRIGGER\_" prefix is removed, but used here to make it easier to see where the file belong

## Appendix L: Evaluation – Usability inspection

### L.1 Heuristic evaluation – Hand out

#### System overview

##### 1. Objectives

The system intends to monitor service performance within department 6211-CIC at Siemens Business Services.

The system has three modules: User system, Management system, and Admin system.

*The User system*, represented by the main-page, displays present information on response time, agent availability, incidents, deviation, and workload. This page intends to give a quick and easy to understand information on the current status for each team and the department as a whole. The User system is accessible to all users.

*The Management system*, represented by the pages for response time, agent availability, incidents, deviation, and workload, displays present and historic information. The Management system is only accessible to the team leaders (TL), the escalation manager (EM), and the department manager (DM).

*The Admin system* is only accessible to the EM and the DM.

##### 2. Intended Audiences

The users of the system are highly skilled users of computer systems.

There are three kinds of users: The agents, the team leaders (TL), and the escalation manager (EM) / department manager (DM). The users are divided based on their access-level.



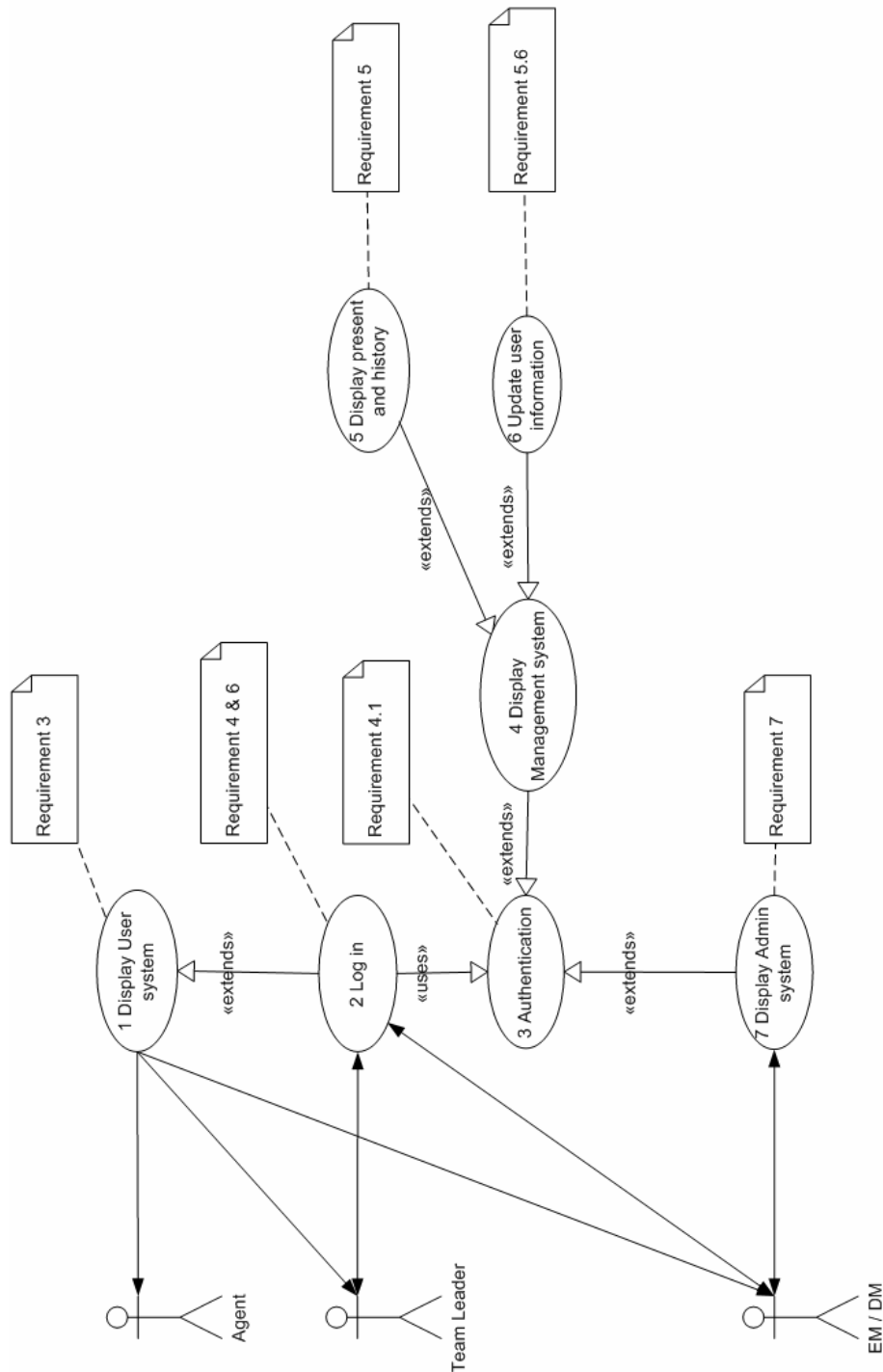


Figure 1: Use case to understand the roles of the different users

**Expected Usage Patterns / Goals of the system**

In this section we present the main goals of the system. You as an evaluator can use these goals to explore the system. When you are ready, please evaluate the system and find usability problems based on the ten heuristics stated on the checklist given to you (attachment A). Additional usability problems or comments that you want to add can be written down on attachment B.

When evaluating the Management system, please note that test data are mainly available on the 20<sup>th</sup> of February and it’s respective week.

A list of goals of the system was handed out to each evaluator during the Heuristic evaluation. The evaluators could then use these goals as a help to explore the system.

*User system goals:*

- I. A user should be able to get a quick and easy-to-understand view of present and historic data from the other pages (response time, agent availability, incidents, deviation, and workload). The information is to be displayed in one single main page.
- II. A user should be able to view the response time for each customer and average response time for each team. If the response time is more than 30 sec. the user should be notified.
- III. A user should be able to see how many agents that are in the available-, unavailable-, work-, and talk state for each team.
- IV. A user should be able to see how many open incidents registered on each team on the current date.
- V. A user should be able to view the deviation on each team on the current date.
- VI. A user should be able to view the workload for each team on the current date.

*Management system goals:*

- I. A user can only access the Management system after they have logged into the Siemens's intranet. Only team leaders, the escalation manager and the department manager will be given access to the Management system. An agent should not be able to access the Management system.

Note: This feature has not been implemented yet!

- II. The system should only display the pages that correspond to a user's access level.

Note: This feature has not been implemented yet!

- III. The Management system includes the pages for response time, agent availability, incidents, deviation, and workload. Each of these pages will give the user detailed information, both present and historic.
- IV. The Management system will have a page for response time which presents the average response time today, this week, and this month for the department. A user can select a team to view the teams' response time for each customer. A team leader or EM / DM should also be able to select an earlier month and see the response time for that month.
- V. A TL/EM/DM should be able to see how much time each agent spends on "talk time", "work time", "available time", and "unavailable time". It should also be possible to select an earlier month and see the time spent that month.
- VI. A TL/EM/DM should be able to see in the Open Incidents page how many open incidents department 6211-CIC and each team has in their backlog this week, Older

than 2 days, older than a week, older than 2 weeks, older than 3 weeks, and 4 weeks or older. The user should have the opportunity to select an earlier week and see the open incidents that week.

- VII. The Management system has a page for Deviations which will display the number of answered phone calls, the number of wrap up reason reports, the number of registered incidents, and show the deviation between answered calls and registered incidents this day. A user should be able to select a month to see the deviations in that month.
- VIII. The Management system has a page for workload which will display the number of incoming phone calls, answered phone calls, and the number of agents logged in. The information will be displayed in periods of 4 hours throughout the day. A user should be able to select an earlier week and see the workload that week.

## L.2: Heuristic evaluation checklist [17]

The evaluators were asked to evaluate the system and find usability problems based on the ten heuristics stated on the heuristic evaluation checklist given to them. The checklist was provided to assist the evaluators of finding usability problems. Additional usability problems or comments were written down on a separate piece of paper.

Evaluator : \_\_\_\_\_

Observer : \_\_\_\_\_

Date:

## Heuristic Evaluation - A System Checklist

This is a system checklist to help you find usability problems in relation to the ten heuristics given by Jakob Nielsen. We have also provided you with general examples on each heuristic.

If you agree with the question check YES, if you disagree check NO. If you don't feel the question applies to the system check N/A.

### 1. Visibility of System Status

**The system should always keep user informed about what is going on, through appropriate feedback within reasonable time.**

For example, once a user clicks the 'Submit' button on an order form, within a few seconds they'll require feedback that tells them their order has been received. This feedback might appear in the form of a separate page, or popup, which also contains a 'back to site' link indicating where the user can go next.

- "Where am I?" and
- "Where can I go next?".

#	Review Checklist	Yes	No	N/A
1.1	Does every display begin with a title or header that describes screen contents? ? (Mao: Er det en overskrift eller en forklaring øverst for hver	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	side?)	
<b>Comments:</b>		
1.2	Is there a consistent icon design scheme and stylistic treatment across the system? ? (Mao: Er ikoner, menyer, tabeller, etc konsistente i systemet?)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
1.3	Is a single, selected icon clearly visible when surrounded by unselected icons?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
1.8	Is there some form of system feedback for every operator action? (Mao: Gir systemet respons på om en handling er utført korrekt?)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
1.1 0	Is there visual feedback in menus or dialog boxes about which choices are selectable?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
1.1 4	Is the current status of an icon clearly indicated? ? (F.eks: Kan du se om valgt meny-item er valgt?)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

#	Review Checklist	Yes	No	N/A
1.1	If there are observable delays (greater than fifteen seconds) in the	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6	system's response time, is the user kept informed of the system's progress?	
<b>Comments:</b>		
1.2 5	Is the menu-naming terminology consistent with the user's task domain? (Dvs., matcher meny-navnene med terminologien brukt på Siemens?)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
1.2 9	If users must navigate between multiple screens, does the system use context labels, menu maps, and place markers as navigational aids? (eksempel: breadcrumbs)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

## 2. Match Between System and the Real World

**The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.**

For example, a real world concept applied on the Web is the shopping cart. On many sites, you click once to select an item (the equivalent of picking it up off the shelf in a real store), click again to "add to basket" (or place it in your trolley) and then a third time to confirm your intention to buy (or move to the checkout).

#	Review Checklist	Yes	No	N/A
2.1	Are icons concrete and familiar?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
2.2	Are menu choices ordered in the most logical way, given the user, the	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	item names, and the task variables?	
<b>Comments:</b>		
2.5	If shape is used as a visual cue, does it match cultural conventions? (Matcher grafikken med Siemens' spesifikasjoner?)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
2.6	Do the selected colors correspond to common expectations about color codes?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
2.9	On data entry screens, are tasks described in terminology familiar to users?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
2.1 2	Do menu choices fit logically into categories that have readily understood meanings?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

### 3. User Control and Freedom

---

**Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.**

An example of a control element might be a "home" button that appears on every page. It's a simple way to let users feel in control of the system - they know they can "go home" (or opt-out) at any stage in the process.

#	Review Checklist	Yes	No	N/A
3.4	When a user's task is complete, does the system wait for a signal from the user before processing? (f.eks bruk av dialogboks som bruker kan klikke "OK" for å gå videre)			<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>				
3.1 2	If menu lists are long (more than seven items), can users select an item either by moving the cursor or by typing a mnemonic code? (eks på mnemonic code: ctr+a)			<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>				
3.1 4	Are menus broad (many items on a menu) rather than deep (many menu levels)?			<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>				
3.2 1	Can users easily reverse their actions? (eksempel ved bruk av ctr+z eller en angrefunkjson)			<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>				
3.2 3	Can users set their own system, session, file, and screen defaults? (for eksempel bestemme hvordan farger skal vises på skjermbildet)			<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>				

#### 4. Consistency and Standards

---



**Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.**

For example, if the user wants to return to the main page then the label on the link should be "Home" or "Homepage", rather than some obscure reference.

#	Review Checklist	Yes	No	N/A
4.1	Have industry or company formatting standards been followed consistently in all screens within a system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b> Not possible for the evaluators at HIA to know.				
4.2	Has a heavy use of all uppercase letters on a screen been avoided?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
4.4	Are integers right-justified and real numbers decimal-aligned?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
4.5	Are icons labeled? (f.eks: pop-up text ved mouse-over)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
4.6	Are there no more than twelve to twenty icon types?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
4.8	Does each window have a title?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				

4.9	Are vertical and horizontal scrolling possible in each window?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.1 2	Are menu choice lists presented vertically?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.1 4	Are menu titles either centered or left-justified?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.1 9	Are field labels / radio buttons consistent from one data entry screen to another?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.2 0	Are fields and labels left-justified for alpha lists and right-justified for numeric lists?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.2 2	Are attention-getting techniques used with care?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.2 3	Intensity: two levels only	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

4.2 4	Size: up to four sizes	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.2 5	Font: up to three	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.2 7	Color: up to four (additional colors for occasional use only)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.2 9	Are attention-getting techniques used only for exceptional conditions or for time-dependent information?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.3 0	Are there no more than four to seven colors, and are they far apart along the visible spectrum?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.3 1	Is a legend provided if color codes are numerous or not obvious in meaning? (legend = forklaring /tegnforklaring)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.3	Have pairings of high-chroma, spectrally extreme colors been avoided?	<input type="radio"/> <input type="radio"/> <input type="radio"/>

2		
<b>Comments:</b>		
4.3 3	Are saturated blues avoided for text or other small, thin line symbols?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.3 6	Are system objects named consistently across all prompts in the system?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
4.3 9	Are menu choice names consistent, both within each menu and across the system, in grammatical style and terminology?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

## 5. Help Users Recognize, Diagnose, and Recover From Errors

**Error messages should be expressed in plain language (NO CODES), precisely indicate the problem, and constructively suggest a solution.**

For example, if a form is completed incorrectly, the error message should alert the user to this, identify which fields will need to be refilled, and perhaps highlight those fields when the user returns to complete the form after they dismiss the error message.

#	Review Checklist	Yes	No	N/A
---	------------------	-----	----	-----

5.1	Is sound used to signal an error?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.7	Are error messages grammatically correct?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
5.8	Do error messages avoid the use of exclamation points?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
5.10	Do error messages avoid an anthropomorphic tone?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
5.12	Do messages place users in control of the system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
5.16	Do error messages inform the user of the error's severity?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
5.17	Do error messages suggest the cause of the problem?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
5.18	Do error messages provide appropriate semantic information?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
5.19	Do error messages provide appropriate syntactic information?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

<b>Comments:</b>		
5.20	Do error messages indicate what action the user needs to take to correct the error?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

## 6. Error Prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

#	Review Checklist	Yes	No	N/A
6.3	Is the menu choice name on a higher-level menu used as the menu title of the lower-level menu?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
6.4	Are menu choices logical, distinctive, and mutually exclusive?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
6.6	If the system displays multiple windows, is navigation between windows simple and visible?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
6.1 1	Does the system prevent users from making errors whenever possible?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

<b>Comments:</b>		
6.1 2	Does the system warn users if they are about to make a potentially serious error? (f.eks ved å slette en bruker)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

## 7. Recognition Rather Than Recall

**Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.**

For example, if you create a Website with a lot of submenus, then use a system that will let the users know what section they are in at all times. You could do this by leaving a breadcrumb trail, or maybe applying a color scheme that differentiates the various sections.

#	Review Checklist	Yes	No	N/A
7.5	Are prompts, cues, and messages placed where the eye is likely to be looking on the screen?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
7.6	Have prompts been formatted using white space, justification, and visual cues for easy scanning?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b> Table format not chosen? Appropriate?				
7.7	Do text areas have "breathing space" around them?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				

7.8	Is there an obvious visual distinction made between "choose one" menu and "choose many" menus?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.1 1	Is white space used to create symmetry and lead the eye in the appropriate direction?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.1 2	Have items been grouped into logical zones, and have headings been used to distinguish between zones?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.1 3	Are zones no more than twelve to fourteen characters wide and six to seven lines high?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.1 4	Have zones been separated by spaces, lines, color, letters, bold titles, rules lines, or shaded areas?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.1 9	Is reverse video or color highlighting used to get the user's attention?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.2	Are borders used to identify meaningful groups?	<input type="radio"/> <input type="radio"/> <input type="radio"/>



2		
<b>Comments:</b>		
7.2 3	Has the same color been used to group related elements?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.2 4	Is color coding consistent throughout the system?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.2 5	Is color used in conjunction with some other redundant cue?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.2 6	Is there good color and brightness contrast between image and background colors?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.3 3	Are inactive menu items grayed out or omitted?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.3 4	Are there menu selection defaults?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

7.3 7	Are there salient visual cues to identify the active window? (salient = fremtredende)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
7.3 8	Are function keys / radio buttons arranged in logical groups?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

## 8. Flexibility and efficiency of use

**Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the “average” user (e.g., physical or cognitive ability, culture, language, etc.).**

For example, Take Amazon. They save the personal information that's provided by customers upon purchase. Then, each time the customer makes another purchase, they can retrieve their information with a single click. In this way, Amazon provides customers with a way to avoid filling in an extensive form each time they buy a product at the store.

#	Review Checklist	Yes	No	N/A
8.3	Can users define their own synonyms for commands? (for eksempel: lage egne snarveier)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
8.9	If menu lists are short (seven items or fewer), can users select an item by moving the cursor?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

<b>Comments:</b>		
8.1 1	If the system uses a pointing device, do users have the option of either clicking on fields or using a keyboard shortcut? (som f.eks bruk av tab)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
8.1 4	On menus, do users have the option of either clicking directly on a menu item or using a keyboard shortcut? (for eksempel Ctrl + P for printing)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

## 9. Aesthetic and Minimalist Design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

#	Review Checklist	Yes	No	N/A
9.1	Is only (and all) information essential to decision making displayed on the screen?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
9.2	Are all icons in a set visually and conceptually distinct?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				
9.3	Have large objects, bold lines, and simple areas been used to distinguish icons? (distinguish = framheve)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

<b>Comments:</b>		
9.4	Does each icon stand out from its background?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
9.5	If the system uses a standard GUI interface where menu sequence has already been specified, do menus adhere to the specification whenever possible? (Mao: holder menyen seg fast gjennom de ulike skjermbildene)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
9.6	Are meaningful groups of items separated by white space?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
9.7	Does each data entry screen have a short, simple, clear, distinctive title?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
9.8	Are field labels brief, familiar, and descriptive? (Er f.eks tekstfelt forklaringer kjent for brukeren?)	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		
9.1 1	Are menu titles brief, yet long enough to communicate?	<input type="radio"/> <input type="radio"/> <input type="radio"/>
<b>Comments:</b>		

## 10. Help and Documentation

---

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

#	Review Checklist	Yes	No	N/A
10.2	Are there any on-line instructions available?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Comments:</b>				

### L.3: Results from the inspection

Nr	Problem description	Inspector			Rating	Comment
		1	2	3		
<b>Category 1: Menu/navigation</b>						
1.1	Not all of the pages have a main header. A header which is explanatory and eye catching can help the user to navigate.	X		X	3	All pages now have one main header. The size on the main header is kept. Sub headers are made smaller to separate them from the main headers.
1.2	The menu doesn't stand out. Provide more space between the menu tabs and the logo-header.	X		X	4	The empty space above the menus has been increased from half the size of the menu to the same size as the menu.
<b>Category 2: System response</b>						
2.1	There are no tool tips when performing mouse-over on the web items.	X			2	Tool tips are added on the menu and all calendars.
2.2	The system presents incomprehensible error messages to the user	X	X		4	Corrected
2.3	The user can not click on error messages to get complementary explanation.	X			2	The error messages are written out in full context if they are to any meaning to the user. If there is an unknown error the users will just get notified that an error has occurred.

2.4	System response-/error messages marked in red can be difficult to notice.			X	4	The response-/error messages are now boldfaced and have now a larger font-size.
2.5	No sound for critical feedback has been implemented	X	X	X	1	Alarm sounds signaling breach on SLA requirements have been discussed, but will not be implemented.
2.6	Error messages don't suggest the cause of the problem	X			4	Corrected.
2.7	Error messages don't indicate what actions the user needs to take to correct the error.	X			4	Corrected
<b>Category 3: Presentation of data/information</b>						
3.1	There are too many strong colors on the main page	X	X	X	4	The web designer at Siemens helped us selecting new colors. The colors are replaced with lighter colors (except the red color which is signaling an alarm).
3.2	The pages for Open incidents and Response time should have a horizontal dividing line between the graphs.	X			0	Horizontal dividing lines made the page confusing. As a remark it is important to remember that the pages would look a bit different when published. The "adds" will disappear when the pages is displayed at the domain where it is licensed to.
3.3	There are no explanation for the colors used on the graph	X	X		0	Correction is not needed. The colours are intuitive.

3.4	The placing of the calendar on each page can be hard to find for the user. On the Response time page and the Open incidents page the user has to scroll to find the calendar.	X			3	There is added a bookmark for [HISTORY] below the headline on all pages which contains history.
3.5	The use of black font on red background is hard for the user to see.			X	4	All black text on red items is changed to white and bolded out.
3.6	The pink color used on the Main page to mark the number of unavailable agents is too sharp.			X	3	A more pinkish soft color was provided by the Siemens Web designer.
3.7	Header icons are not left-justified with one another. The "Snapshot Website" icon label is not on the same vertical alignment as the "Performance Monitoring System" icon label.			X	2	Corrected.
3.8	According to Siemens' specifications, the color of the background lines in the tables should be changed from orange to grey/green.			X	3	The color on the background lines are changed from orange to a grey/ green color according to Siemens Style Guides.
3.9	The size of the Siemens logo should be smaller			X	2	Corrected.
3.10	The content on the "Team and Agent Configuration" page should be left justified			X	2	All Content on "Team and Agent Configuration" has now been left justified.
3.11	The vertical dividing line between "Add agent to team" and "Remove agent from team" on the "Team and agent configuration" page should be drawn further down			X	3	The vertical dividing line between "Add Agent To Team" and "Remove Agent From Team" on the "Team and Agent Configuration" page has been made longer. The dividing line is now "covering" the whole section.



3.12	Items have not been grouped into logical zones, and headings have not been used to distinguish between zones	X			4	Corrected. The data on the "Team and Agent Configuration" page have been grouped and headings for each group have been provided.
<b>Category 4: Consistency</b>						
4.1	Inconsistent tables on the different pages. There seems to be no table structure.	X		X	4	All tables have gotten the same colors. Every other row is colored in all tables where there are more than two rows.
4.2	Inconsistent use of headers. There should be a main header on the "Team and Agent Configuration" page. The four sub headers should have a smaller font.		X	X	4	There has been added one main header "Team and Agent Configuration". The sub headers have been decreased in size.
<b>Category 5: Other problems</b>						
5.1	No online help function is provided. One inspector at HIA feels such a function can be helpful. The inspector at Siemens does not think an online help function is necessary as long as the user knows who to contact if the user are in question of what to do. A mailto link to the webmaster accessible on each page is suggested to be added.	X	X	X	2	A mail link to webmaster is added. There has also been added some more explaining text on each of the pages.
5.2	Personalization of the pages does not exist.	X		X	1	Personalization in the sense of each module (there are three modules: user system, management system and admin system) have been discussed. However, the inspector at Siemens doesn't think this is

						necessary.
5.3	According to Siemens' guide lines, change the footer from SBS to Siemens Business Services.			X	0	The text in the footer now states "Siemens Business Services".

## Appendix M: Evaluation - User test 1

### M.1 Task Scenarios user test 1

1. Anta at du har nettopp kommet på jobb. Du starter systemet, og ser umiddelbart at blant annet Esthetique og Skatteetaten har en gjennomsnittlig responstid over avtalt SLA. Hvordan ser du dette?
2. Du ønsker å få en oversikt over gjennomsnittlig responstid for team 1 – team 5. Hva er gjennomsnittlig responstid i dag for team 1, team 2, team 3, team 4 og team 5?
3. Du ønsker videre å finne ut hvor stor gjennomsnittlig responstid Team 3 har brukt på Utdanningsetaten, Thorn NO og Skatteetaten denne uken.
4. Du er nå nyskjerrig på hvor mye tid (i %) agenten ved navn Rich Nash har brukt i dag på hvert av de følgende tilstandene: Available? Unavailable? Work? Talk?
5. Du ønsker nå å finne ut raskt hvor mange agenter i Team 3 som er tilgjengelige (i Available tilstanden) akkurat nå
6. Anta at du befinner deg i nåtid, og ønsker å finne ut hvor mange åpne saker (open incidents) Team 3 hadde 22. februar 2006.
7. Anta nå at du er team leder for Team 4 og ønsker å planlegge hvor mange du vil ha på jobb 24. februar neste år. Du velger derfor å finne ut hvor stor arbeidsmengde (workload) Team 4 har hatt 24. februar 2006 mellom kl 0500 – 0900.
8. Anta at dere har fått en ny ansatt, Anders Larsen. Legg til / registrer den nye agenten. Den nye agenten skal kun ha tilgang til "Response time" og "Agent availability".
9. Du skal nå opprette et nytt team, Team 7. Legg til to ledige agenter i det nye teamet. Du skal så legge til kunder til det nye teamet. Legg til Statoil, Acta og Narvesen som kunder for Team 7. Acta skal allikevel ikke tilhøre Team 7, du må dermed fjerne Acta fra Team 7.
10. Du får beskjed om at Team 7 skal fjernes. Slett Team 7, dens kunder og agent releasjoner.

**M.2: Results user test 1**

Nr	Problem description	User			Rating	Comment
		1	2	3		
<b>Category 1: Menu/navigation</b>						
1.1	Two of the users don't use the Main page to get a quick view of present information	X		X	4	Main has gotten a new head line and a description under that headline. This will explain to unfamiliar users what the page is displaying. The size of the "Main" tab has also been increased.
1.2	<p>On task two, user 2 and 3 spends respectively 68 and 90 seconds to find average response time current date for team 1 – team 5 as they both use the Response time page to solve the problem. User 1 saves time by using the Main page to solve the problem.</p> <p>Two of the users spend much time on task 5, 2min 38sec and 55 sec. Both users are going back and forth through the different pages (not the Main page) instead of going straight to the Main page. In comparison user 1 spends only 4 seconds to solve the task.</p>		X	X	4	Same correction as problem 1.1
1.3	One user spends 3 minutes and 40 seconds on task 6. The user has problems with finding the right page. When the user finds the right page, the user doesn't discover the "arrow buttons" on the calendar to switch between the months.	X			3	The text on the calendar has been increased, the same has the arrows. The arrows are changed into buttons which are easier to notice and probably easier associated with clicking.
<b>Category 2: System response</b>						
2.1	Feedback/response messages from the system were hard to discover.		X	X	3	The error/feedback field has been enlarged and the font has been bolded out.

<b>Category 3: Presentation of data/information</b>						
3.1	One user spends 1 minute to understand that one has to select a certain date on the calendar to view, for instance, response time that month.			X	3	The text used in the calendars has been bolded out and sat to be bullet points. See also problem 5.1.
3.2	One user spends 90 seconds to find average response time current date for team 1 – team 5 on task nr 2. The user doesn't see the numbers on the graph right away.			X	3	This is fixed, the numbers on the graphs has been increase and bolded out.
3.3	All of the users had problems with distinguishing the "Team and Agent configuration" page and the "Users" page. It seems it is not intuitive for the users where the different functions can be found between the three pages ("Team and agent configuration", "Customer configuration", and "Users") of the Admin system.	X	X	X	4	There has also been added some more explanation text.
3.4	When deleting a team and its customer- and agent relations, two of the users did not find the section for deleting a team <u>and its relations</u> right away.		X	X	4	There has been added explanation text below the head line. Each of the "sub head lines" has been decreased in size. This will make the page divided up and each part will be easier to find.
<b>Category 4: Consistency</b>						
<b>Category 5: Other problems</b>						
5.1	Two of the users are used to select a period on the calendar in the current phone system by using the shift-key. When the users are trying to execute the same function in the SWS a new window pops up with the "The page can not be displayed" error message.	X		X	1	The text was changed on all pages. The same text, or almost the same text, was chosen to make it easier for the user to recognize the calendar function.  The time period which the calendar selected

						<p>data for was written in capitals to make it easier to distinguish the calendars which gave data for a MONTH and the calendars which gave data for a DAY or a WEEK.</p> <p>There is no easy way to disable the shift-select option. Hence, this will not be changed</p>
<b>Category 6: Input</b>						
6.1	All of the users were startled over the drop-down-menu used to add a new user on the Users page.	X	X	X	3	There has been added text above the drop down menus, describing their functionality.
6.2	Deleting a customer from a team does not work	X	X	X	4	Deleting customer is fixed. It worked all the time, but a condition in a store procedure updating the screen was wrong. That is fixed.
6.3	A user has to select the team checkbox every time the user adds a customer to that team. For example, if a user adds a customer to team 7 the user has to select team 7 again in order to add a new customer to team 7.			X	3	Corrected. This function has been implemented.

### M.3 Time spent on user test 1:

Task	Time used on each task (seconds)			
	Test user			Total
	1	2	3	
1	113	50	34	197
2	64	68	90	222
3	47	55	46	148
4	76	78	240	394
5	4	158	55	217
6	220	42	45	307
7	68	67	54	189
8	74	83	72	229
9	67	99	122	288
10	9	55	90	154
<b>Total</b>	<b>742</b>	<b>755</b>	<b>848</b>	<b>2345</b>

### M.4 Number of errors performed in user test 1:

Task	Nr of errors performed *			
	Test user			Total
	1	2	3	
1	1(1)		1(1)	2(2)
2		1(1)	1(1)	2(2)
3				
4				
5		1(1)	1(1)	2(2)
6	3(3)			3(3)
7	1(1)			1(1)
8	1(1)		1(1)	2(2)
9				
10		1 (1)	3(3)	4(4)
<b>Total</b>	<b>6(6)</b>	<b>3(3)</b>	<b>7(7)</b>	<b>16(16)</b>

\* Parentheses indicates the number of recoverable errors given the right information



**M.5 Number of system errors identified by the user in user test 1:**

Task	Nr of system errors identified*			
	Test user			Total
	1	2	3	
1				
2				
3				
4				
5				
6	1(0)			1(0)
7				
8				
9		1(0)	2(1)	3(1)
10				
<b>Total</b>	<b>1(0)</b>	<b>1(0)</b>	<b>2(1)</b>	<b>4(1)</b>

### M.6 Number of times users express clear joy/frustration in user test 1:

Task	Nr of times user expresses clear joy/frustration			
	Test user			Total
	1	2	3	
1				
2			1/0	1/0
3	1/0			1/0
4	1/0			1/0
5				
6	0/2			0/2
7		1/0		1/0
8			0/1	0/1
9			0/2	0/2
10	1/0			1/0
<b>Total</b>	<b>3/2</b>	<b>1/0</b>	<b>1/3</b>	<b>5/5</b>

### M.7 Semantic differential scale user test 1

#### User 1:

<b>Pleasant</b>	1	2	3a	4	5	6	7	<b>Irritating</b>
<b>Complete</b>	1	2	3	4	5	6	7	<b>Incomplete</b>
<b>Simple</b>	1	2	3	4	5	6	7	<b>Complicated</b>
<b>Fast to use</b>	1	2	3	4	5	6	7	<b>Slow to use</b>
<b>Safe</b>	1	2	3	4	5	6	7	<b>Unsafe</b>

a. The user may have been influenced by the errors user 1 made on task 6

Median: 3.4

#### User 2:

<b>Pleasant</b>	1	2	3	4	5	6	7	<b>Irritating</b>
<b>Complete</b>	1	2	3	4	5	6	7	<b>Incomplete</b>
<b>Simple</b>	1	2	3	4	5	6	7	<b>Complicated</b>
<b>Fast to use</b>	1	2	3	4	5	6	7	<b>Slow to use</b>
<b>Safe</b>	1	2	3	4	5	6a	7	<b>Unsafe</b>

- a. Delete function does not work

*Median: 4.0*

### User 3:

<b>Pleasant</b>	1	2	3	4a	5	6	7	<b>Irritating</b>
<b>Complete</b>	1	2	3	4	5	6	7	<b>Incomplete</b>
<b>Simple</b>	1	2	3	4b	5	6	7	<b>Complicated</b>
<b>Fast to use</b>	1	2	3	4	5	6	7	<b>Slow to use</b>
<b>Safe</b>	1	2	3	4	5	6	7c	<b>Unsafe</b>

- a. The user may have been influenced by the user's misunderstanding with task 4  
 b. The user commented that the calendar function, the need to repeat the selection of team 7 on task 9, and the confirmation of deleting a customer could be a little hard to understand.  
 c. Delete function does not work

*Median: 5.0*

## Appendix N: Evaluation - User test 2

### N.1 Task scenarios user test 2

1. Anta at du har nettopp kommet på jobb. Du starter systemet, og ser umiddelbart at blant annet Esthetique og Skatteetaten har en gjennomsnittlig responstid over avtalt SLA. Hvordan ser du dette?
2. Du ønsker å få en oversikt over gjennomsnittlig responstid for team 1 – team 5. Hva er gjennomsnittlig responstid i dag for team 1, team 2, team 3, team 4 og team 5?
3. Du ønsker videre å finne ut hvor stor gjennomsnittlig responstid Team 3 har brukt på Utdanningsetaten, Thorn NO og Skatteetaten denne uken.
4. Du er nå nyskjerrig på hvor mye tid (i %) agenten ved navn Rich Nash har brukt i dag på hvert av de følgende tilstandene: Available? Unavailable? Work? Talk?
5. Du ønsker nå å finne ut raskt hvor mange agenter i Team 3 som er tilgjengelige (i Available tilstanden) akkurat nå
6. Anta at du befinner deg i nåtid, og ønsker å finne ut hvor mange åpne saker (open incidents) Team 3 hadde 22. februar 2006.
7. Anta nå at du er team leder for Team 4 og ønsker å planlegge hvor mange du vil ha på jobb 24. februar neste år. Du velger derfor å finne ut hvor stor arbeidsmengde (workload) Team 4 har hatt 24. februar 2006 mellom kl 0500 – 0900.
8. Anta at dere har fått en ny ansatt(brukeren ligger allerede inne i systemet til Siemens, men må legges til i denne web siden), Anders Larsen. Legg til / registrer den nye brukeren i Snapshot web siden. Den nye brukeren skal kun ha tilgang til "Response time" og "Agent availability".
9. Du skal nå opprette et nytt team, Team 7. Legg til to ledige agenter i det nye teamet. Du skal så legge til kunder til det nye teamet. Legg til Statoil, Acta og Narvesen som kunder for Team 7. Acta skal allikevel ikke tilhøre Team 7, du må dermed fjerne Acta fra Team 7.
10. Du får beskjed om at Team 7 skal fjernes. Slett Team 7, dens kunder og agent releasjoner.

## N.2 Results user test 2

Nr	Problem description	User				Rating	Comment
		1	2	3	4		
<b>Category 1: Menu/navigation</b>							
1.1	One user does not use main on task 1 and spends 37 seconds on the task. Instead the user solves the task by finding the right answer by using the "Response time" page. The three other users use main to solve the task and spends respectively 4, 14 and 16 seconds. User 2 spends 1 minute and 55 seconds on task 5 as the user does not go to the "Main" page right away to solve the task. In comparison, the others users spend 6, 8, and 22 seconds on this task		X	X		0	Three out of 4 users solved task 1 and task 5 satisfactorily, the average time spent on this task improved from 66 seconds to 18 second, and changes has been made since user test 1 to make the "Main" page more evident. Hence, no changes are needed.
<b>Category 3: Presentation of data/information</b>							
3.1	One user discovers that present information on the department as a whole is missing on the main page.				X	3	This will take too much time to correct at this stage of the project. This problem will be put on the further development list.
3.2	User 3 spends 49 seconds on task 3 as he has some problem of reading the numbers (in black fonts) on the graph on the "Response time" page. User 2 appears to have some difficulties reading the numbers (in black fonts on the graph (blue background) on the "Open incidents" page, but solves task 6 quickly.		X	X		2	Changes has been done after user test 1 to change the font color to white, however this change did not improve the visibility of the numbers as the graphs are in 3D. No changes will be made.

3.3	One user finds the "Delete team" checkbox and the "Delete team" button confusing as both items has the same text.		X			4	Text has been changed on all "Delete Check boxes" for illustration that they are there for confirmation reasons and not function as a tool for removal.
<b>Category 4: Consistency</b>							
4.1	The history table size on the "Agent availability" page does not have the same size as the present table, making some of the text in the history table cover two rows.	X				3	The table size has been changed so that data only will be on one row.
<b>Category 5: Other problems</b>							
5.1	User 4 discovers that a user can not select a period on the calendar to get information within this period, a user can only get history information for a specific week or month.				X	4	There has been added a new calendar on the page for "Workload". There are now on calendar representing the "From Date" and another calendar representing a "To Date". The user will get the workload between the two dates selected when using the two calendars.
5.2	One user finds it misleading when clicking on a certain date on the calendar, information for that week or that month is displayed.				X	4	Corrected. See problem nr 5.1.

### N.3 Time spent on user test 2:

Task	Time used on each task (seconds)				
	Test user				Total
	1	2	3	4	
1	4	14	37	16	71
2	23	37	44	64	168
3	41	16	49	33	139
4	25	32	44	24	125
5	8	115	6	22	151
6	25	27	27	29	108
7	26	48	45	44	163
8	24	31	38	55	148
9	64	72	72	134	342
10	12	30	7	24	73
<b>Total</b>	<b>252</b>	<b>422</b>	<b>369</b>	<b>445</b>	<b>1488</b>

### N.4 Number of errors performed in user test 2:

Task	Nr of errors performed *				
	Test user				Total
	1	2	3	4	
1			1(1)		1(1)
2		1(1)			1(1)
3					
4					
5		1(1)			1(1)
6					
7					
8					
9				1(1)	1(1)
10					
<b>Total</b>	<b>0(0)</b>	<b>2(2)</b>	<b>1(1)</b>	<b>1(1)</b>	<b>4(4)</b>



### N.5 Number of system errors identified by the test users in user test 2:

Task	Nr of system errors identified*				
	Test user				Total
	1	2	3	4	
1					
2				1(0)	1(0)
3					
4					
5					
6					
7				1(0)	1(0)
8					
9					
10					
<b>Total</b>	<b>0(0)</b>	<b>0(0)</b>	<b>0(0)</b>	<b>2(0)</b>	<b>2(0)</b>

## N.6 Number of times users expresses clear joy/frustration in user test 2:

Task	Nr of times user expresses clear joy/frustration				
	Test user				Total
	1	2	3	4	
1	1/0			2/0	3/0
2			1/0		1/0
3					
4				3/0	3/0
5	2/0	0/1		1/0	3/1
6	1/0			4/0	5/0
7	1/0			0/1	1/1
8	1/0				1/0
9				0/1	0/1
10	1/0	1/0			2/0
<b>Total</b>	<b>7/0</b>	<b>1/1</b>	<b>1/0</b>	<b>10/2</b>	<b>19/3</b>

## N.7 Semantic differential scale user test 2

### User 1

<b>Pleasant</b>	<b>1</b>	2	3	4	5	6	7	<b>Irritating</b>
<b>Complete</b>	1	<b>2<sup>a</sup></b>	3	4	5	6	7	<b>Incomplete</b>
<b>Simple</b>	1	2	<b>3</b>	4	5	6	7	<b>Complicated</b>
<b>Fast to use</b>	1	<b>2</b>	3	4	5	6	7	<b>Slow to use</b>
<b>Safe</b>	<b>1</b>	2	3	4	5	6	7	<b>Unsafe</b>

a. Almost complete, but the user finds one minor usability problem

Median: 1.8

### User 2

<b>Pleasing</b>	1	2	3	4	5	6	7	<b>Irritating</b>
<b>Complete</b>	1	2	3	4	5	6	7	<b>Incomplete</b>
<b>Simple</b>	1	2	3a	4	5	6	7	<b>Complicated</b>
<b>Fast to use</b>	1	2	3	4b	5	6	7	<b>Slow to use</b>
<b>Safe</b>	1	2	3	4	5	6	7	<b>Unsafe</b>

a. The user finds the "delete team" checkbox and button confusing.

b. The system appears to be slow; however the SWS prototype runs on a slow laptop.

*Median: 2.4*


### User 3

<b>Pleasing</b>	1	2	3	4	5	6	7	<b>Irritating</b>
<b>Complete</b>	1	2	3	4	5	6	7	<b>Incomplete</b>
<b>Simple</b>	1	2	3	4	5	6	7	<b>Complicated</b>
<b>Fast to use</b>	1	2	3	4	5	6	7	<b>Slow to use</b>
<b>Safe</b>	1	2	3	4	5	6	7	<b>Unsafe</b>

*Median: 1.8*

# Appendix O: Screenshots

## O.1 Main



**A Snapshot of Department 6211**

This is an overview department 6211's performance so far today. It could be up to 15 minutes delay on the data displayed here.

Status at: 2006-02-20 14:46:50

Team	Response Time	Agents	Open Incidents	Deviations	Workload
<b>Team 1</b>	customer 1	14	Available: 2	201	Answered calls: 23 - Wrapup reasons: 4 - Incidents: 10 = <b>Deviations</b> : 9
	customer 2	28	Unavailable: 0		
	customer 3	39	Work: 0		
	customer 4	22	Talk: 3		
	<b>Avg. Response time</b>	27			
<b>Team 2</b>	customer 5	34	Available: 0	202	Answered calls: 17 - Wrapup reasons: 1 - Incidents: 8 = <b>Deviations</b> : 8
	customer 6	42	Unavailable: 1		
	customer 7	30	Work: 0		
			Talk: 3		
	<b>Avg. Response time</b>	36			
<b>Team 3</b>	customer 8	33	Available: 1	205	Answered calls: 21 - Wrapup reasons: 4 - Incidents: 8 = <b>Deviations</b> : 9
	customer 9	34	Unavailable: 1		
	customer 10	52	Work: 1		
			Talk: 1		
	<b>Avg. Response time</b>	39			
<b>Team 4</b>	customer 11	28	Available: 2	204	Answered calls: 36 - Wrapup reasons: 1 - Incidents: 8 = <b>Deviations</b> : 27
	customer 12	36	Unavailable: 0		
	customer 13	32	Work: 0		
			Talk: 2		
	<b>Avg. Response time</b>	32			
<b>Team 5</b>	<b>Avg. Response time</b>	0	Available: 0	200	Answered calls: 0 - Wrapup reasons: 0 - Incidents: 0 = <b>Deviations</b> : 0
			Unavailable: 0		
			Work: 0		
			Talk: 0		

Copyright 2006, Siemens Business Services, Department 6211-CIC.  
Webmaster

## O.2 Response Time

**Snapshot Website**

Department 6211-CIC
Performance Monitoring System
SIEMENS

Main
Response Time
Agent Availability
Open Incidents
Deviations
Workload
Team/Agent Conf
Customer Conf
User Conf

### Response Time

Response Times for Department 6211-CIC today, this week and this Month.

Response Times for each Team today, this week and this Month.

**Department Today**

Team	Avg. Response Time (Seconds)
Team 1	27
Team 2	36
Team 3	39
Team 4	32
Team 5	30

**Team 1, To day**

customer	Avg. Response Time (Seconds)
customer 1	14
customer 2	28
customer 3	39
customer 4	22

**Department This Week**

Team	Avg. Response Time (Seconds)
Team 1	29
Team 2	34
Team 3	33
Team 4	30
Team 5	30

**Team 1, This Week**

customer	Avg. Response Time (Seconds)
customer 1	24
customer 2	33
customer 3	36
customer 4	24

**Department This Month**

Team	Avg. Response Time (Seconds)
Team 1	29
Team 2	34
Team 3	33
Team 4	30
Team 5	30

**Team 1, This Month**

customer	Avg. Response Time (Seconds)
customer 1	24
customer 2	33
customer 3	36
customer 4	24

### Response Time History

To get Response Time data for a spectral MONTH:

- Find the wanted month with the calendar month selection option
- Select ANY date within that month
- The history data for that day will then appear
- It is not possible to use the shift key to select more than one date!

Prev	februar 2006												Next	
	30	31	1	2	3	4	5							
1	6	7	8	9	10	11	12							
2	13	14	15	16	17	18	19							
3	20	21	22	23	24	25	26							
4	27	28	1	2	3	4	5							
5	6	7	8	9	10	11	12							

**Department februar 2006**


Team	Avg. Response Time (Seconds)
Team 1	29
Team 2	34
Team 3	33
Team 4	30
Team 5	30

**Team 2, februar 2006**

customer	Avg. Response Time (Seconds)
customer 5	32
customer 6	41
customer 7	30


Copyright 2006, Siemens Business Services, Department 6211-CIC. WebApp.j2ee

### O.3 Agent Availability



## Snapshot Website

Siemens Business Services



Department 6211-CIC
Performance Monitoring System

Main
Response Time
Agent Availability
Open Incidents
Deviations
Workload
Team/Agent Conf
Customer Conf
User Conf

#### Agent Availability

[HISTORY]

Status at: 2006-02-20 14:46:50

Team	Agent Name	Current State	Seconds In Current State	Total Calls Handled	Time spent in each state in Percent (%)			
					Available	Unavailable	Work	Talk
Team 1	Aki-Petteri Berg	Talk	115	6	34,29	0	0	65,71
	Alexander Ovechkin	Talk	115	3	34,29	0	0	65,71
	Saku Koivo	Avaiable	115	5	100	0	0	0
	Peter Forsberg	Talk	115	5	0	0	0	100
	Milan Hejduk	Avaiable	115	4	65,71	0	34,29	0
Team 2	Bryan McCabe	Talk	115	6	34,29	0	0	65,71
	Martin St-Louis	Unavaivable	115	3	0	0	100	0
	Martin Straka	Talk	115	0	0	0	34,29	65,71
	Jussi Jokinen	Talk	115	8	34,29	0	0	65,71
Team 3	Jarkku Ruutu	Work	115	3	0	0	65,71	34,29
	Vincent Lecavalier	Unavaivable	115	9	100	0	0	0
	Rick Nash	Talk	115	4	34,29	0	0	65,71
	Jaromir Jagr	Avaiable	115	5	100	0	0	0
Team 4	Ilya Kovalchuk	Talk	115	12	34,29	0	0	65,71
	Alexi Yashin	Avaiable	115	5	65,71	0	34,29	0
	Joe Sakic	Talk	115	9	0	0	34,29	65,71
	Pete Thornton	Avaiable	115	10	65,71	0	34,29	0
Team 5			0	0	0	0	0	0

#### Agent Availability History

To get Agent Availability data for a special MONTH:  
 - Find the wanted month with the calenders month selection option  
 - Select ANY date within that month  
 - The history data for that day will then appare  
 - It is not possible to use the shift key to select more then one date!

Prev februar 2006 Next

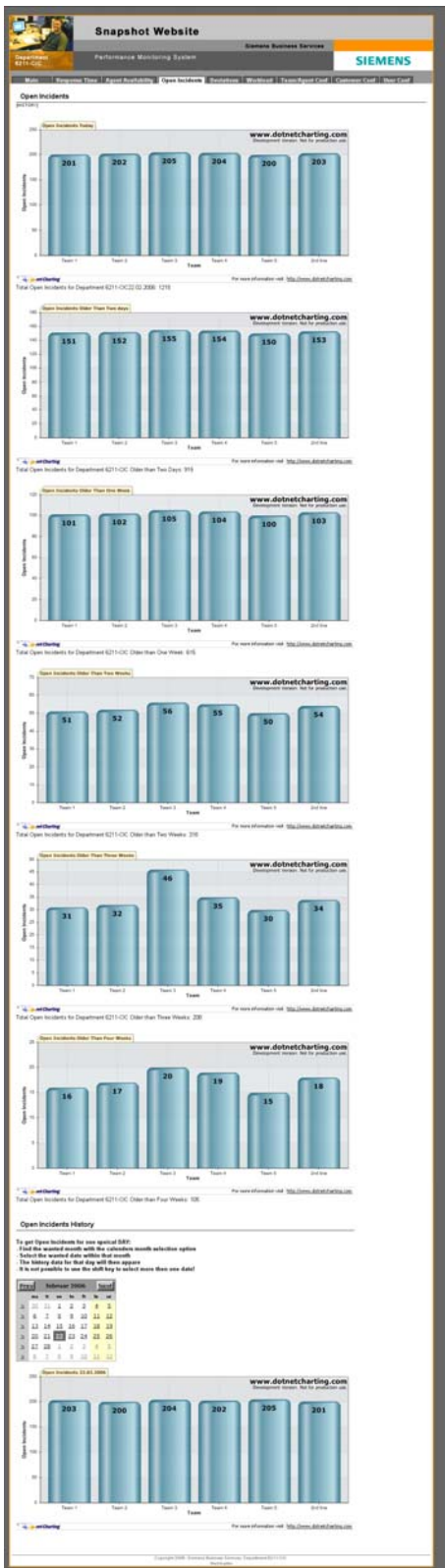
ma	ti	on	to	fr	lo	so
≥ 30	31	1	2	3	4	5
≥ 6	7	8	9	10	11	12
≥ 13	14	15	16	17	18	19
≥ 20	21	22	23	24	25	26
≥ 27	28	1	2	3	4	5
≥ 6	7	8	9	10	11	12

Resume for: februar 2006


Team	Agent Name	Total Calls Handled	Time spent in each state in Percent (%)			
			Available	Unavailable	Work	Talk
Team 1	Aki-Petteri Berg	15	34,29	0	0	65,71
	Alexander Ovechkin	12	34,29	0	0	65,71
	Saku Koivo	14	100	0	0	0
	Peter Forsberg	14	0	0	0	100
	Milan Hejduk	13	65,71	0	34,29	0
Team 2	Bryan McCabe	15	34,29	0	0	65,71
	Martin St-Louis	12	0	0	100	0
	Martin Straka	0	0	0	34,29	65,71
	Jussi Jokinen	17	34,29	0	0	65,71
Team 3	Jarkku Ruutu	12	0	0	65,71	34,29
	Vincent Lecavalier	18	100	0	0	0
	Rick Nash	13	34,29	0	0	65,71
	Jaromir Jagr	14	100	0	0	0
Team 4	Ilya Kovalchuk	21	34,29	0	0	65,71
	Alexi Yashin	14	65,71	0	34,29	0
	Joe Sakic	18	0	0	34,29	65,71
	Pete Thornton	19	65,71	0	34,29	0
Team 5		0	0	0	0	0

Copyright 2006, Siemens Business Services, Department 6211-CIC.  
Webmaster

# O.4 Open Incidents




# 0.5 Deviations



## Snapshot Website

Siemens Business Services

Performance Monitoring System



Department 6211-CIC
Main
Response Time
Agent Availability
Open Incidents
Deviations
Workload
Team/Agent Conf
Customer Conf
User Conf

---

### Deviations

[HISTORY]

Status at: 2006-02-20 14:56:50

Team	Agent Name	Answered phone calls	Incidents	Wrap up reasons	Deviations	Deviations in %
Team 1	Aki-Petteri Berg	5	2	1	2	40
	Alexander Ovechkin	3	0	1	2	66,67
	Saku Koivu	3	1	1	1	33,33
	Peter Forsberg	3	1	1	1	33,33
	Milan Hejduk	2	0	1	1	50
<b>Total for team</b>		<b>23</b>	<b>4</b>	<b>10</b>	<b>9</b>	<b>39,13</b>
Team 2	Bryan McCabe	4	1	1	2	50
	Martin St-Louis	3	0	1	2	66,67
	Martin Straka	0	0	1	-1	0
	Jussi Jokinen	6	0	1	5	83,33
<b>Total for team</b>		<b>17</b>	<b>1</b>	<b>8</b>	<b>8</b>	<b>47,06</b>
Team 3	Jarkku Ruutu	1	0	1	0	0
	Vincent Lecavalier	7	1	1	5	71,43
	Rick Nash	2	2	1	-1	0
	Jaromir Jagr	3	1	1	1	33,33
<b>Total for team</b>		<b>21</b>	<b>4</b>	<b>8</b>	<b>9</b>	<b>42,86</b>
Team 4	Ilya Kovalchuk	10	1	1	8	80
	Alexi Yashin	3	0	1	2	66,67
	Joe Sakic	7	0	1	6	85,71
	Pete Thornton	7	0	1	6	85,71
<b>Total for team</b>		<b>36</b>	<b>1</b>	<b>8</b>	<b>27</b>	<b>75</b>
Team 5		0	0	0	0	0

---

### Deviation History

To get Deviation data for a special MONTH:  
 - Find the wanted month with the calendars month selection option  
 - Select ANY date within that month  
 - The history data for that day will then appear  
 - It is not possible to use the shift key to select more then one date!

Prev
februar 2006
Next

	ma	ti	on	to	fr	le	ss
≥	30	31	1	2	3	4	5
≥	6	7	8	9	10	11	12
≥	13	14	15	16	17	18	19
≥	20	21	22	23	24	25	26
≥	27	28	1	2	3	4	5
≥	6	7	8	9	10	11	12


Resume for: februar 2006

Team	Agent Name	Answered phone calls	Incidents	Wrap up reasons	Deviations	Deviations in %
Team 1	Aki-Petteri Berg	15	5	8	2	13,33
	Alexander Ovechkin	12	0	8	4	33,33
	Saku Koivu	14	4	8	2	14,29
	Peter Forsberg	14	2	8	4	28,57
	Milan Hejduk	13	1	8	4	30,77
<b>Total for team</b>		<b>68</b>	<b>12</b>	<b>40</b>	<b>16</b>	<b>23,53</b>
Team 2	Bryan McCabe	15	1	8	6	40
	Martin St-Louis	12	0	8	4	33,33
	Martin Straka	0	2	8	-10	0
	Jussi Jokinen	17	0	8	9	52,94
<b>Total for team</b>		<b>44</b>	<b>3</b>	<b>32</b>	<b>9</b>	<b>20,45</b>
Team 3	Jarkku Ruutu	12	1	8	3	25
	Vincent Lecavalier	18	2	8	8	44,44
	Rick Nash	13	7	8	-2	0
	Jaromir Jagr	14	5	8	1	7,14
<b>Total for team</b>		<b>57</b>	<b>15</b>	<b>32</b>	<b>10</b>	<b>17,54</b>
Team 4	Ilya Kovalchuk	21	1	8	12	57,14
	Alexi Yashin	14	2	8	4	28,57
	Joe Sakic	18	2	8	8	44,44
	Pete Thornton	19	2	8	9	47,37
<b>Total for team</b>		<b>72</b>	<b>7</b>	<b>32</b>	<b>33</b>	<b>45,83</b>
Team 5		0	0	0	0	0

Copyright 2006, Siemens Business Services, Department 6211-CIC.  
Webmaster




## 0.6 Workload



### Snapshot Website

Siemens Business Services



Department 6211-CIC
Performance Monitoring System

Main
Response Time
Agent Availability
Open Incidents
Deviations
Workload
Team/Agent Conf
Customer Conf
User Conf

#### Workload

[HISTORY]

Status at: 2006-02-20 00:00:00

Team	Type	05:00-09:00	09:00-13:00	13:00-17:00	17:00-21:00	21:00-01:00
Team 1	Incomming Calls	6	11	12	2	0
	Answered Calls	2	7	12	2	0
	Agents Logged In	0	0	5	0	0
Team 2	Incomming Calls	2	13	7	3	2
	Answered Calls	2	8	6	1	0
	Agents Logged In	0	0	4	0	0
Team 3	Incomming Calls	0	13	12	3	0
	Answered Calls	0	9	10	2	0
	Agents Logged In	0	0	4	0	0
Team 4	Incomming Calls	2	21	21	2	1
	Answered Calls	2	17	16	1	0
	Agents Logged In	0	0	4	0	0
Team 5	Incomming Calls	0	0	0	0	0
	Answered Calls	0	0	0	0	0
	Agents Logged In	0	0	0	0	0

#### Workload History

To get Response Time data for a special WEEK:  
 - Find the month which contains the wanted week with the calenders month selection option  
 - Select ANY date within that the wanted week  
 - The history data for that day will then appare  
 - It is not possible to use the shift key to select more then one date!

From Date

Prev
februar 2006
Next

	ma	ti	on	to	fr	lø	ss
>	30	31	1	2	3	4	5
>	6	7	8	9	10	11	12
>	13	14	15	16	17	18	19
>	20	21	22	23	24	25	26
>	27	28	1	2	3	4	5
>	6	7	8	9	10	11	12

To Date

Prev
februar 2006
Next


	ma	ti	on	to	fr	lø	ss
>	30	31	1	2	3	4	5
>	6	7	8	9	10	11	12
>	13	14	15	16	17	18	19
>	20	21	22	23	24	25	26
>	27	28	1	2	3	4	5
>	6	7	8	9	10	11	12

Resume for: 19.02.2006 to 21.02.2006

Team	Type	05:00-09:00	09:00-13:00	13:00-17:00	17:00-21:00	21:00-01:00
Team 1	Avg.Incomming Calls	3,33	9,33	15	2	0
	Avg.Answered Calls	2	8	9,33	2	0
	Avg.Agents Logged In	0	0	1,67	0	0
Team 2	Avg.Incomming Calls	2,67	7,67	9	2,67	0,67
	Avg.Answered Calls	2,33	6	4,67	1,67	0
	Avg.Agents Logged In	0	0	1,33	0	0
Team 3	Avg.Incomming Calls	3	6,67	9	2,33	0
	Avg.Answered Calls	2,67	5,33	7,33	2	0
	Avg.Agents Logged In	0	0	1,33	0	0
Team 4	Avg.Incomming Calls	3,67	11,33	12,67	2,33	0,33
	Avg.Answered Calls	3,67	10	9	2	0
	Avg.Agents Logged In	0	0	1,33	0	0
Team 5	Avg.Incomming Calls	0	0	0	0	0
	Avg.Answered Calls	0	0	0	0	0
	Avg.Agents Logged In	0	0	0	0	0

Copyright 2006, Siemens Business Services, Department 6211-CIC.  
Webmaster


## O.7 Team/Agent Conf.



### Snapshot Website

Siemens Business Services

Department 6211-CIC Performance Monitoring System



Main
Response Time
Agent Availability
Open Incidents
Deviations
Workload
Team/Agent Conf
Customer Conf
User Conf

---

#### Team And Agent Configuration

Delete or Create a Team and Add or Remove Agents from a Team

##### Create Team

Team Name:

Is telephone team

##### Delete Team

Team 1

Yes, Delete Team.

##### Add Agent To Team

Available Agents:	Available Teams:
<input checked="" type="radio"/> HenrikBorge	<input checked="" type="radio"/> Team 1
<input type="radio"/> GöranHermansson	<input type="radio"/> Team 2
<input type="radio"/> BjarteBjønnes	<input type="radio"/> Team 3
	<input type="radio"/> Team 4
	<input type="radio"/> Team 5
	<input type="radio"/> 2nd line

Select an agent and a team and press the button.


##### Remove Agent From Team

Teams:	Agents in the Team:
<input type="radio"/> Team 1	<input checked="" type="radio"/> Bryan McCabe
<input checked="" type="radio"/> Team 2	<input type="radio"/> Martin St-Louis
<input type="radio"/> Team 3	<input type="radio"/> Martin Straka
<input type="radio"/> Team 4	<input type="radio"/> Jussi Jokinen
<input type="radio"/> Team 5	
<input type="radio"/> 2nd line	

Yes, Remove Agent

Copyright 2006, Siemens Business Services, Department 6211-CIC.  
Webmaster


## O.10 Customer Conf.



### Snapshot Website

Siemens Business Services

Department 6211-CIC Performance Monitoring System



Main
Response Time
Agent Availability
Open Incidents
Deviations
Workload
Team/Agent Conf
Customer Conf
User Conf

---

#### Customer Configuration

Add or Remove Customers to or from a team. Customers are to be chosen by the customers currently registered in the HPPC telephone system.

##### Add Customer To Team

Available Customers:	Available Teams:
<input checked="" type="radio"/> customer 14 (Not in team)	<input checked="" type="radio"/> Team 1
<input type="radio"/> customer 15 (Not in team)	<input type="radio"/> Team 2
<input type="radio"/> customer 16 (Not in team)	<input type="radio"/> Team 3
<input type="radio"/> customer 17 (Not in team)	<input type="radio"/> Team 4
<input type="radio"/> customer 1	<input type="radio"/> Team 5
<input type="radio"/> customer 10	<input type="radio"/> 2nd line
<input type="radio"/> customer 11	
<input type="radio"/> customer 12	
<input type="radio"/> customer 13	
<input type="radio"/> customer 2	
<input type="radio"/> customer 3	
<input type="radio"/> customer 4	
<input type="radio"/> customer 5	
<input type="radio"/> customer 6	
<input type="radio"/> customer 7	
<input type="radio"/> customer 8	
<input type="radio"/> customer 9	

Select a customer and a team and press "Add To Team".

##### Remove Customer From Team

Teams:	Customers in the Team:
<input checked="" type="radio"/> Team 1	<input checked="" type="radio"/> customer 1
<input type="radio"/> Team 2	<input type="radio"/> customer 2
<input type="radio"/> Team 3	<input type="radio"/> customer 3
<input type="radio"/> Team 4	<input type="radio"/> customer 4
<input type="radio"/> Team 5	
<input type="radio"/> 2nd line	

Yes, Remove Customer

Copyright 2006, Siemens Business Services, Department 6211-CIC.  
Webmaster

## O.11 User Conf.

The screenshot shows a web application interface for 'Snapshot Website' under 'Siemens Business Services'. The page is titled 'User Configuration' and is part of a 'Performance Monitoring System'. A navigation menu includes 'Main', 'Response Time', 'Agent Availability', 'Open Incidents', 'Deviations', 'Workload', 'Team/Agent Conf', 'Customer Conf', and 'User Conf'. The 'User Configuration' section contains two main panels: 'Add a new User' and 'Alter or remove a user'. The 'Add a new User' panel includes a dropdown for 'New User' (currently showing 'testuser6, Jan Hansen'), a list of pages to grant access to (all unchecked), and an 'Add User' button. The 'Alter or remove a user' panel includes a dropdown for 'Current users' (currently showing 'testuser1, Thomas Andersen'), a list of pages to allow or disallow (all checked), a 'Yes, Delete this user' checkbox, and 'Alter user' and 'Delete user' buttons. A footer contains copyright information: 'Copyright 2006, Siemens Business Services, Department 6211-CIC. Webmaster'.

**Snapshot Website**

Siemens Business Services

Department 6211-CIC Performance Monitoring System

SIEMENS

Main Response Time Agent Availability Open Incidents Deviations Workload Team/Agent Conf Customer Conf **User Conf**

### User Configuration

Add or remove customers to the system and give or change their rights. A user does not need to login to the Snapshot website. Permissions are given by the users windows (LDAP) login. Therefore are only users currently in the LDAP directory free to add.

#### Add a new User

Select a user from the Siemens ORS user base (LDAP logins) and determine what the new user is allowed to use. Then add the user

New User:

Check pages which the user will have access to:

- Response Time
- Agent Availability
- Open Incidents
- Deviations
- Work Load
- Team Configuration
- Customer Configuration
- Users

#### Alter or remove a user

Brows the current added users in the system, change their permissions, or delete them.

Current users:

Allowed Pages (check for allowed, uncheck for disallow):

- Response Time
- Agent Availability
- Open Incidents
- Deviations
- Work Load
- Team Configuration
- Customer Configuration
- Users

Yes, Delete this user

Copyright 2006, Siemens Business Services, Department 6211-CIC. Webmaster

## Appendix P: Testimonial from SBS

# SIEMENS

Oslo, 24.5.2006

**Attest for:  
Ferdigstillelse av prosjektarbeid relatert til hovedfagsoppgave, vår 2006.**

Trond J. Undrum og Thomas Andersen har i perioden januar 2006 - mai 2006 arbeidet med deres hovedfagsoppgave ved Siemens Business Services, Linderud.

Arbeidet på Linderud har bestått i å utvikle en dynamisk webside, som automatisk henter ut diverse data for videre analyse og tiltak. Disse dataene er ment primært å brukes av ansatte, ledere og kundekontakter i avdeling Customer Interaction Center.

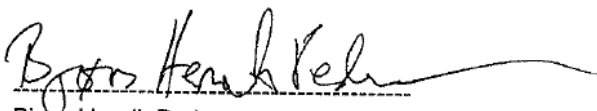
Avdeling Customer Interaction Center jobber primært med Incident håndtering etter ITIL (IT Infrastructure Library) modellen. Dette betyr at alle typer henvendelser / hendelser løses på bruker -og servernivå løpende.

Applikasjonens funksjonalitet er i tråd med de krav Siemens Business Services hadde før utviklingen startet. Blant annet dekkes følgende funksjonalitet:

- Antall åpne hendelser i saksbehandlingssystemet.
- Responstid på telefon
- Avvik mellom besvarte samtaler og registrerte saker
- Arbeidsmengde

Begge kandidatene har i hele perioden vist stor interesse innenfor fagområdet systemutvikling, og har således klart å knytte faglige interesser sammen med Siemens Business Services sine interesser på en utmerket måte. Kandidatene har tatt i bruk Contextual Design som designmetode for å komme frem til ønsket resultat. Metoden baserer seg på en kundeorientert tilnærming til systemutvikling. Metoden har vist seg svært inspirerende og faglig innholdsrik gjennom hele prosessen.

I den perioden hvor hovedfagsoppgaven har pågått, har begge kandidatene hele tiden jobbet målrettet og strukturert frem til overlevering. Arbeidsfordelingen dem i mellom har i hele perioden vært klart definert. Det endelige resultatet har vist seg å være brukervennlig, funksjonelt og robust med tanke på fremtidig gjenbruk.



Bjørn Henrik Pedersen  
Cand. Scient. i Systemarbeid  
Siemens Business Services

## **Appendix Q: User Manual**



# **Snapshot User Manual**

A guide to use, understanding, and roll out for Snapshot Website version 1.0  
For Siemens Business Services, May 2006

**Thomas Andersen & Trond J. Undrum**  
Master Thesis, Høgskolen I Agder (Agder University College) 2006

## About this guide

This guide is created as a beginner's guide and a reference to the snapshot system.

It introduces each of the content pages in the Snapshot website, and explains extended use on each of them.

Configuration of the snap shot website is explained a bit closer with a step by step walkthrough.

The roll out section gives a very short description of what's needed to deploy the system at a new location.

All real names and all real customers are removed from this guide to protect Siemens Business Services interests. Because of this are the screen shots from a test server which are not within the dotnetcharting graph license.

There could be some differences between the pictures in this guide and the pictures found on the version in use on the Siemens intranet.

## Contents:

About this guide .....	223
Contents: .....	224
Figures:.....	225
1. Use of the real time and history data.....	226
1. Use of the real time and history data.....	226
1.1 Main.....	226
1.2 Response Time .....	227
1.2.1 See response times for a specific team.....	228
1.3 Agent Availability .....	229
1.4 Open Incidents .....	230
1.5 Deviations.....	231
1.6 Workload .....	232
1.6.1 Create Workload History.....	233
1.7 Display history .....	234
2. Configuration of teams, agents in team, customers in team, and users .....	235
2.1 Team And Agent Configuration.....	235
2.1.1 Create A New Team .....	236
2.1.2 Delete Team .....	236
2.1.3 Add An Agent To A Team.....	237
2.1.4 Remove An Agent From A Team.....	238
2.2 Customer Configuration .....	239
2.2.1 Add A Customer To A Team.....	240
2.2.2 Remove A Customer From A Team .....	241
2.3 User Configuration .....	242
2.3.1 Add a new user to the system .....	243
2.3.2 Alter A User Already In The System .....	244
2.3.3 Delete a user from the system .....	245
3. Roll out the system .....	246
3.1 The Main System .....	246
3.1.1 The Web Application.....	246
3.1.2 The Database .....	247
3.2 Data updating.....	247



## Figures:

Figure: UM.1.1: The main page.....	226
Figure: UM.1.2: Response time explanation .....	227
Figure: UM.1.3: Response time, select team radio buttons .....	228
Figure: UM.1.4: Response time, select team radio buttons .....	228
Figure: UM.1.5: Agent Availability .....	229
Figure: UM.1.6: Open Incidents.....	230
Figure: UM.1.7: Deviations .....	231
Figure: UM.1.8: Workload.....	232
Figure: UM.1.9: Workload calendars.....	233
Figure: UM.1.10: Calendar .....	234
Figure: UM.2.1: Agent and team configuration.....	235
Figure: UM.2.2: Create a new Team .....	236
Figure: UM.2.3: Delete Team .....	236
Figure: UM.2.4: Add an agent to a team .....	237
Figure: UM.2.5: Remove an agent from a team .....	238
Figure: UM.2.6: Customer configuration. ....	239
Figure: UM.2.7: Add a Customer to a team.....	240
Figure: UM.2.8: Remove a customer from a team. ....	241
Figure: UM.2.9: User configuration.....	242
Figure: UM.2.10: User configuration.....	243
Figure: UM.2.11: Alter a user already in the system .....	244
Figure: UM.2.12: Delete a user from the system.....	245
Figure: UM.3.1: System overview for the main parts .....	246
Figure: UM.3.2: System overview, data update .....	247

# 1. Use of the real time and history data

We will go through each of the content pages, explain shortly and show that content page.

## 1.1 Main

The main page displays a snapshot of the department's current performance for each team (each point is marked with the same number in figure UM.1.1).

1. The team name is colored green or red depending on the average response time for that team. An average response time over thirty means red and under thirty means green.
2. Response times per customer in each team and the average team response time
3. The number of agents in each state for each team
4. Open Incidents (from yesterday, the update will only go once a day) for each team
5. Deviations in registered incidents and answered calls for the team
6. Workload in incoming calls answered calls and the number of agents currently logged in for the team

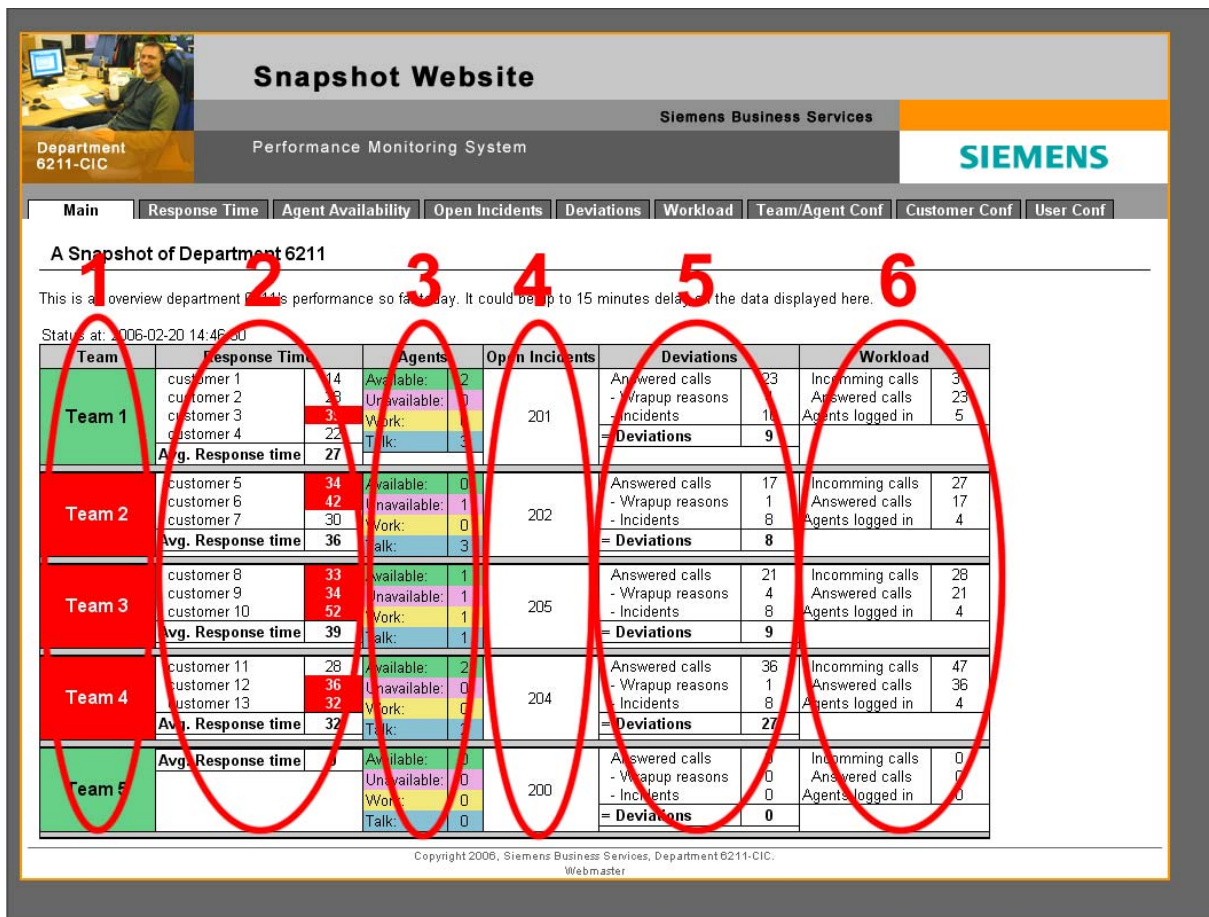


Figure: UM.1.1: The main page

## 1.2 Response Time

The response time page displays the response time as a summary for each team or for each of the customers in one team

From figure UM.1.2:

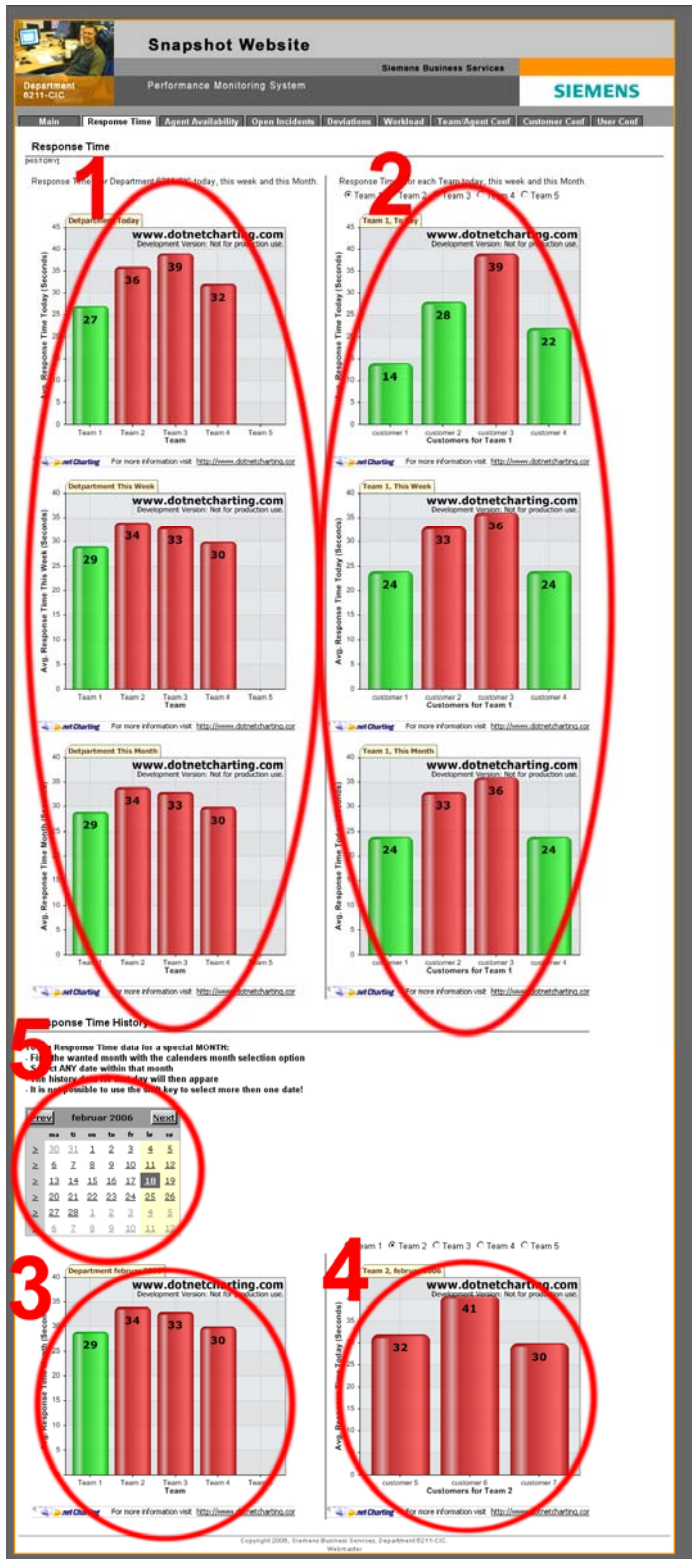


Figure: UM.1.2: Response time explanation

1. A summary for each team today, this week and this month on the three first graphs on the left

2. A summary for each customer in a team today, this week and this month on the three first graphs on the right For selecting teams, see chapter 1.2.1.

3. The lowest graph on the left displays history for the department, see chapter 1.7, on how to display history. (not displayed if history is not created)

4. The lowest graph on the right displays history for the team, see chapter 1.7, on how to display history and chapter 1.2.1 on how to select a team. (not displayed if history is not created)

5. Just below the three graphs on the left there is a calendar. This is used for creating history. For more information see chapter 1.7, on how to display history.

### 1.2.1 See response times for a specific team

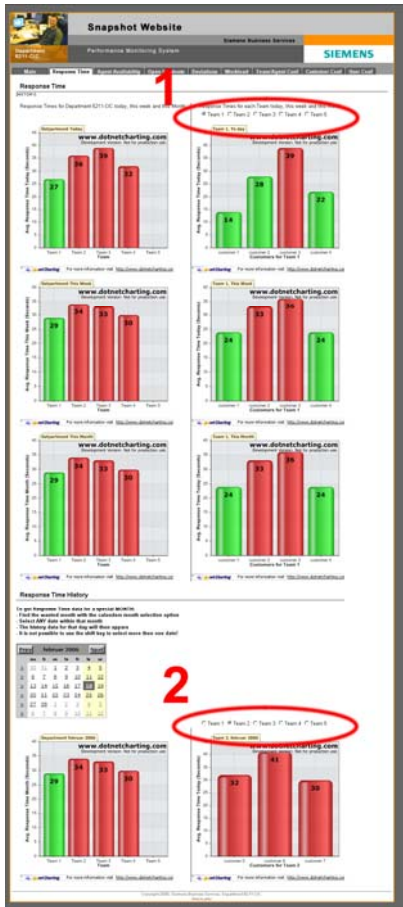


Figure: UM.1.3: Response time, select team radio buttons

When response time page is opened there are three graphs on the left and three graphs on the right.

The three graphs on the right does by default display the response time for the customers in the first team in the database.

To change which team that are displayed in the three graphs, use the radio buttons marked with 1 on figure UM.1.3.

The radio buttons will have a team name, as in figure UM.1.4, and by selecting the radio button in front of the team name on the wanted team, the response times for that team will be displayed in the three graphs (marked with 2 on figure UM.1.2) on the right.

The radio buttons marked with 2 on figure UM.1.3 will appear when history is created, and they will work in the same way as the once marked with one, but they will only affect the graph marked as 4 on figure UM.1.2.



Figure: UM.1.4: Response time, select team radio buttons

### 1.3 Agent Availability

1. The real time Agent Availability data is displayed in a table. Each team is presented with all agents, the current state, time in that state, number of calls handled, and a percentage of time in each of the different states throughout the day.
2. The history data for Agent Availability is located on the bottom and only displayed when history is created (see chapter 1.7 on how to create history). The history table will not contain the field for current state, and time in that state.
3. Below the real time table there is a calendar. This is used for creating history. For more information see chapter 1.7, on how to display history.

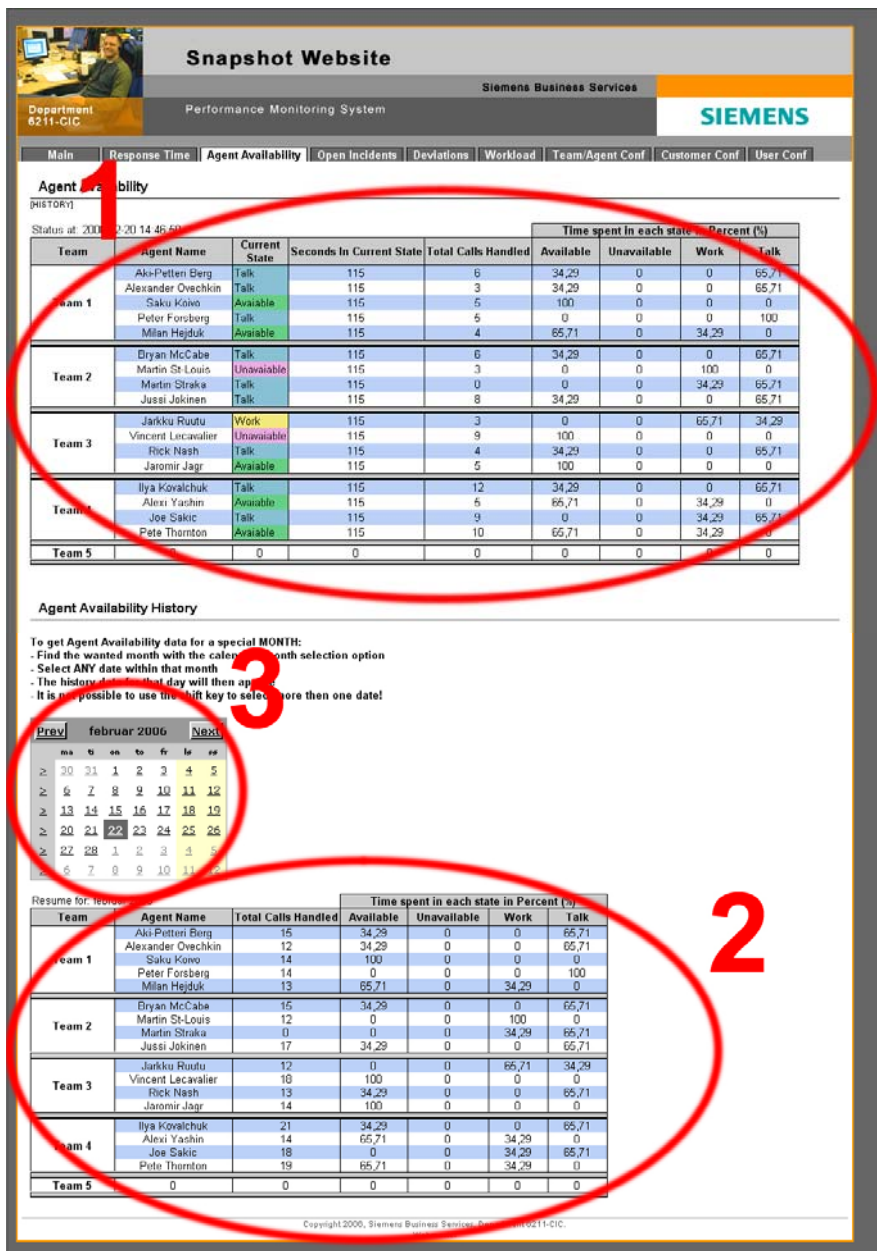


Figure: UM.1.5: Agent Availability

### 1.4 Open Incidents

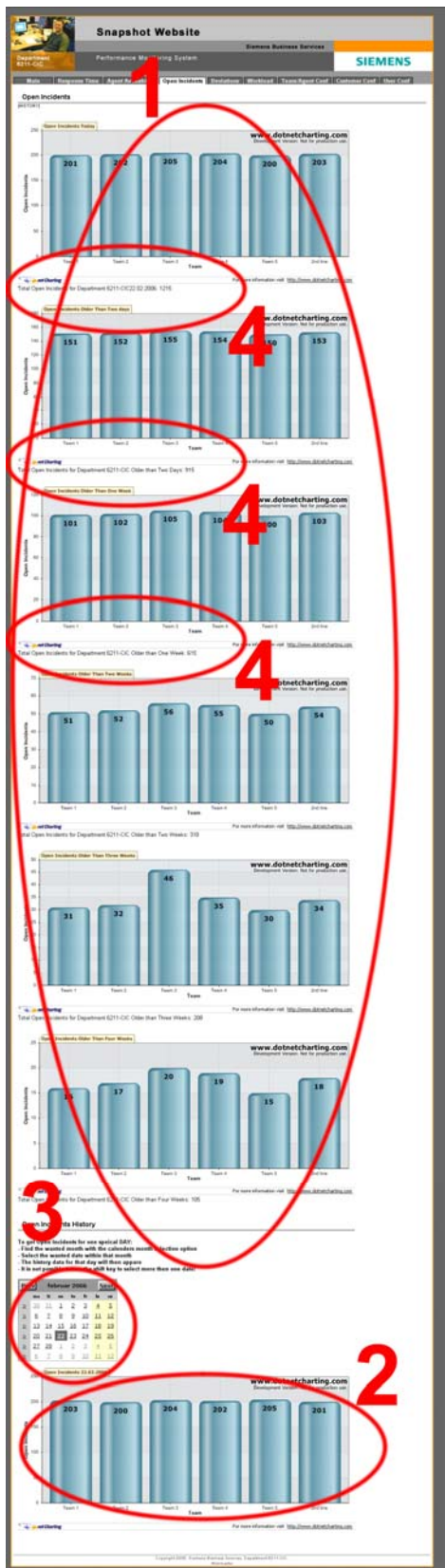


Figure: UM.1.6: Open Incidents

1. The Open Incident real time data is displayed in several graphs, where each graph is containing information for each of the teams. There are graphs for open incidents right now, open incidents older than two days, open incidents older than one week, open incidents older than two weeks, open incidents older than three weeks, and open incidents older than four weeks.
2. The Open incident history data is presented in one single graph containing open incidents for all teams on a certain date, and only displayed when history is created (see chapter 1.7 on how to create history).
3. Below all the real time data graphs there is a calendar. This is used for creating history. For more information see chapter 1.7, on how to display history.
4. Underneath each of the graphs under point 1 there is a sentence summarizing the whole department and saying how many open incidents the whole department's got. The sentence is to be found under every graph even though only three is marked.

### 1.5 Deviations

1. The real time Deviations data is displayed in a table. Each team is presented with all agents, the number of answered calls, the number of registered incidents, the number of registered wrap up reason reports and the deviation between the registered incidents and wrap up reason reports against the answered phone calls in number and percentage throughout the day.
2. The history data for Deviations is located on the bottom and only displayed when history is created (see chapter 1.7 on how to create history).
3. Below the real time table there is a calendar. This is used for creating history. For more information see chapter 1.7, on how to display history.

**Snapshot Website**  
Siemens Business Services  
Department 6211-CIC Performance Monitoring System

Navigation: Main | Response Time | Agent Availability | Open Incidents | **Deviations** | Workload | Team/Agent Conf | Customer Conf | User Conf

#### Deviations

Status at 2006-02-20 14:56:50

Team	Agent Name	Answered phone calls	Incidents	Wrap up reasons	Deviations	Deviations in %
Team 1	Alexander Ovechkin	5	2	1	2	40
	Saku Koiwo	3	1	1	1	33,33
	Peter Forsberg	3	1	1	1	33,33
	Milan Hejdjuk	2	0	1	1	50
	<b>Total for team</b>	<b>23</b>	<b>4</b>	<b>10</b>	<b>9</b>	<b>39,13</b>
Team 2	Bryan McCabe	4	1	1	2	50
	Martin St-Leous	3	0	1	2	66,67
	Martin Straka	0	0	1	-1	0
	Jussi Jokinen	8	0	1	5	83,33
<b>Total for team</b>	<b>17</b>	<b>1</b>	<b>8</b>	<b>8</b>	<b>47,06</b>	
Team 3	Jarkku Runtu	1	0	1	0	0
	Vincent Lecavalier	7	1	1	5	71,43
	Rick Nash	2	2	1	-1	0
	Jaromir Jagr	3	1	1	1	33,33
	<b>Total for team</b>	<b>21</b>	<b>4</b>	<b>8</b>	<b>9</b>	<b>42,86</b>
Team 4	Ilya Kovalchuk	10	1	1	8	80
	Alexi Yashin	3	0	1	2	66,67
	Joe Sakic	7	0	1	6	85,71
	Pete Thornton	7	0	1	6	85,71
	<b>Total for team</b>	<b>36</b>	<b>1</b>	<b>8</b>	<b>27</b>	<b>75</b>
Team 5		0	0	0	0	0

#### Deviation History

To get deviation data for a special MONTH:  
 - Find wanted month with the calendar's month selection option  
 - Select any date within that month  
 - Any data for that day will then appear  
 - If it is not possible to see the shift key to select more than one date!

Calendar: februar 2006

Team	Agent name	Answered phone calls	Incidents	Wrap up reasons	Deviations	Deviations in %
Team 1	Petteri Berg	15	5	8	2	13,33
	Alexander Ovechkin	12	0	8	4	33,33
	Saku Koiwo	14	4	8	2	14,29
	Peter Forsberg	14	2	8	4	28,57
	Milan Hejdjuk	13	1	8	4	30,77
<b>Total for team</b>	<b>68</b>	<b>12</b>	<b>40</b>	<b>16</b>	<b>23,53</b>	
Team 2	Bryan McCabe	15	1	8	6	40
	Martin St-Leous	12	0	8	4	33,33
	Martin Straka	0	2	8	-10	0
	Jussi Jokinen	17	0	8	9	52,94
<b>Total for team</b>	<b>44</b>	<b>3</b>	<b>32</b>	<b>9</b>	<b>20,45</b>	
Team 3	Jarkku Runtu	12	1	8	3	25
	Vincent Lecavalier	10	2	8	8	44,44
	Rick Nash	13	7	8	-2	0
	Jaromir Jagr	14	5	8	1	7,14
	<b>Total for team</b>	<b>57</b>	<b>15</b>	<b>32</b>	<b>10</b>	<b>17,54</b>
Team 4	Ilya Kovalchuk	21	1	8	12	57,14
	Alexi Yashin	14	2	8	4	28,57
	Joe Sakic	18	2	8	8	44,44
	Pete Thornton	19	2	8	9	47,37
	<b>Total for team</b>	<b>72</b>	<b>7</b>	<b>32</b>	<b>33</b>	<b>45,83</b>
Team 5		0	0	0	0	0

Figure: UM.1.7: Deviations

### 1.6 Workload

1. The real time Workload data is displayed in a table. Each team is presented with, the number of incoming calls, the number of answered calls, and the number of agents logged in time four hour sequences throughout the day.
2. The history data for Workload is located on the bottom and only displayed when history is created (see chapter 1.6.1 on how to create workload history).
- 3 & 4. Below the real time table there are two calendars. They are used for creating history. For more information see chapter 1.6.1, on how to display history.

**Snapshot Website**  
Siemens Business Services  
Department 6211-CIC Performance Monitoring System

**Workload**

Team	Type	05:00-09:00	09:00-13:00	13:00-17:00	17:00-21:00	21:00-01:00
Team 1	Incoming Calls	6	11	12	2	0
	Answered Calls	2	7	12	2	0
	Agents Logged In	0	0	5	0	0
Team 2	Incoming Calls	2	13	7	3	2
	Answered Calls	2	8	6	1	0
	Agents Logged In	0	0	4	0	0
Team 3	Incoming Calls	0	13	12	3	0
	Answered Calls	0	9	10	2	0
	Agents Logged In	0	0	4	0	0
Team 4	Incoming Calls	2	21	21	2	1
	Answered Calls	2	17	16	1	0
	Agents Logged In	0	0	4	0	0
Team 5	Incoming Calls	0	0	0	0	0
	Answered Calls	0	0	0	0	0
	Agents Logged In	0	0	0	0	0

**Workload History**

To get Response Time data for a special WEEK:  
 - Find a month which contains the wanted week with the calendars month selection option  
 - Select a date within that the wanted week  
 - The history data for that day will then appear  
 - It is not possible to use the shift key to select multiple dates!

From Date:  To Date:

Team	Type	05:00-09:00	09:00-13:00	13:00-17:00	17:00-21:00	21:00-01:00
Team 1	Avg. Incoming Calls	3,33	9,33	15	2	0
	Avg. Answered Calls	2	8	9,33	2	0
	Avg. Agents Logged In	0	0	1,67	0	0
Team 2	Avg. Incoming Calls	2,67	7,67	9	2,67	0,67
	Avg. Answered Calls	2,33	6	4,67	1,67	0
	Avg. Agents Logged In	0	0	1,33	0	0
Team 3	Avg. Incoming Calls	3	6,67	9	2,33	0
	Avg. Answered Calls	2,67	5,33	7,33	2	0
	Avg. Agents Logged In	0	0	1,33	0	0
Team 4	Avg. Incoming Calls	3,67	11,33	12,67	2,33	0,33
	Avg. Answered Calls	3,67	10	9	2	0
	Avg. Agents Logged In	0	0	1,33	0	0
Team 5	Avg. Incoming Calls	0	0	0	0	0
	Avg. Answered Calls	0	0	0	0	0
	Avg. Agents Logged In	0	0	0	0	0

Figure: UM.1.8: Workload



### 1.6.1 Create Workload History

The Workload history is different than the other history creations, but it is quit fast and simple to create.

There are two calendars, marked with number 3 and 4 in figure UM.1.8.

From Date								To Date																	
Prev		februar 2006						Next		Prev		februar 2006						Next							
	ma	ti	on	to	fr	lø	sø		ma	ti	on	to	fr	lø	sø		ma	ti	on	to	fr	lø	sø		
≥	30	31	1	2	3	4	5		≥	30	31	1	2	3	4	5		≥	30	31	1	2	3	4	5
≥	6	7	8	9	10	11	12		≥	6	7	8	9	10	11	12		≥	6	7	8	9	10	11	12
≥	13	14	15	16	17	18	19		≥	13	14	15	16	17	18	19		≥	13	14	15	16	17	18	19
≥	20	21	22	23	24	25	26		≥	20	21	22	23	24	25	26		≥	20	21	22	23	24	25	26
≥	27	28	1	2	3	4	5		≥	27	28	1	2	3	4	5		≥	27	28	1	2	3	4	5
≥	6	7	8	9	10	11	12		≥	6	7	8	9	10	11	12		≥	6	7	8	9	10	11	12

Figure: UM.1.9: Workload calendars

The first of the two calendars is the “from date”, the second is the “to date”.

If From = To, the workload for that day will be created

If From < To, the workload between the two dates will be created

If From > To, there will be an error and no history will be created.

## 1.7 Display history

On the page for Response Time, Agent Availability, Open Incidents, and Deviations there could be found a calendar as in figure UM.1.10.

<u>Prev</u>	februar 2006							<u>Next</u>
	ma	ti	on	to	fr	lø	ss	
≥	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	
≥	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	
≥	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	
≥	<u>20</u>	<b><u>21</u></b>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	
≥	<u>27</u>	<u>28</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	
≥	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	

Figure: UM.1.10: Calendar

The Calendar will create history data for the page it is placed on if it used correctly.

To switch between months in the calendar, juts press the Prev (previous month) or the Next (Next month) buttons.

To create history for Open Incidents use the Prev and Next button to find the wanted month and select the wanted date in that month by clicking on the date in the calendar. The open incidents for that day will then appear in the history graph.

To create history for Response Time, Agent Availability, and Deviations use the Prev and Next button to find the wanted month and select any date within that month by clicking on the date number. All dates within that month will present the same history data in the history table or graph.

## 2. Configuration of teams, agents in team, customers in team, and users

This chapter will describe how to configure teams, agents in team, customers in teams, and users.

### 2.1 Team And Agent Configuration

1. Create a team, for more information see chapter 2.1.1
2. Delete a team, for more information see chapter 2.1.2
3. Add an agent to a team, for more information see chapter 2.1.3
4. Remove an agent from a team, for more information see chapter 2.1.4

Figure: UM.2.1: Agent and team configuration.

### 2.1.1 Create A New Team

The create team section is marked with 1 at figure UM.2.1.

Figure: UM.2.2: Create a new Team

To create a team:

1. Start by filling out the team name, example: Team 1
2. Check if the team is a telephone team or not. If the team is not using HPPC to answer calls it's not a telephone team.
3. Click the "Create Team" button and the team will be created.

### 2.1.2 Delete Team

The delete team section is marked with 2 at figure UM.2.1.

Figure: UM.2.3: Delete Team

To delete a team:

1. Select wanted team in the combo box
2. Check the "Yes, Delete Team" check box and the "Confirm Delete" check box will appear.
3. Check the "Confirm Delete" check box.
4. Click the "Delete Team" button, and the team will be deleted.

### 2.1.3 Add An Agent To A Team

The add agent to team section is marked with 3 at figure UM.2.1.

**Add Agent To Team**

---

<b>Available Agents:</b> <ul style="list-style-type: none"><li><input checked="" type="radio"/> HenrikBorge</li><li><input type="radio"/> GöranHermansson</li><li><input type="radio"/> BjarteBjønnes</li></ul>	<b>Available Teams:</b> <ul style="list-style-type: none"><li><input checked="" type="radio"/> Team 1</li><li><input type="radio"/> Team 2</li><li><input type="radio"/> Team 3</li><li><input type="radio"/> Team 4</li><li><input type="radio"/> Team 5</li><li><input type="radio"/> 2nd line</li></ul>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Select an agent and a team and press the button.

Figure: UM.2.4: Add an agent to a team

To add an agent to a team:

1. Select an available agent with the radio button. If the agent is already in a team he must be removed first, see chapter 2.1.4.
2. Select an available team with the radio button. If the team does not exist, it must be created first, see chapter 2.1.1.
3. Click the “Add To Team” button, and the agent is added to the team.

### 2.1.4 Remove An Agent From A Team

The remove agent from team section is marked with 4 at figure UM.2.1.

The figure consists of two side-by-side screenshots of a web interface titled "Remove Agent From Team".

**Left Screenshot:**

- Teams:** A list of radio buttons for Team 1, Team 2, Team 3, Team 4, Team 5, and 2nd line. A red circle labeled '1' encloses this list.
- Agents in the Team:** A list of radio buttons for Aki-Petteri Berg, Alexander Ovechkin, Saku Koivo, Peter Forsberg, and Milan Hejduk. A red circle labeled '2' encloses this list.
- Confirmation:** A checkbox labeled "Yes, Remove Agent" with a red circle labeled '3' around it.
- Button:** A button labeled "Remove Agent From Team".

**Right Screenshot:**

- Teams:** A list of radio buttons for Team 1, Team 2, Team 3, Team 4, Team 5, and 2nd line.
- Agents in the Team:** A list of radio buttons for Aki-Petteri Berg, Alexander Ovechkin, Saku Koivo, Peter Forsberg, and Milan Hejduk.
- Confirmation:** A checked checkbox labeled "Yes, Remove Agent" and an unchecked checkbox labeled "Confirm Removal" with a red circle labeled '5' around it.
- Button:** A button labeled "Remove Agent From Team" with a red circle labeled '4' around it.

Figure: UM.2.5: Remove an agent from a team

To remove an agent from a team:

1. Select which team the agent is in with the radio buttons. When the team is selected, the list of agents in that team will appear
2. Select the agent in the team to be removed with the radio buttons.
3. Check the "Yes, Remove Agent" check box and the "Confirm Removal" check box will appear.
4. Check the "Confirm Removal" check box.

Click the "Remove Agent From Team" button, and the agent will be removed from the team.

## 2.2 Customer Configuration

1. Add a customer to a team, for more information see chapter 2.2.1
2. Remove a customer from a team, for more information see chapter 2.2.2

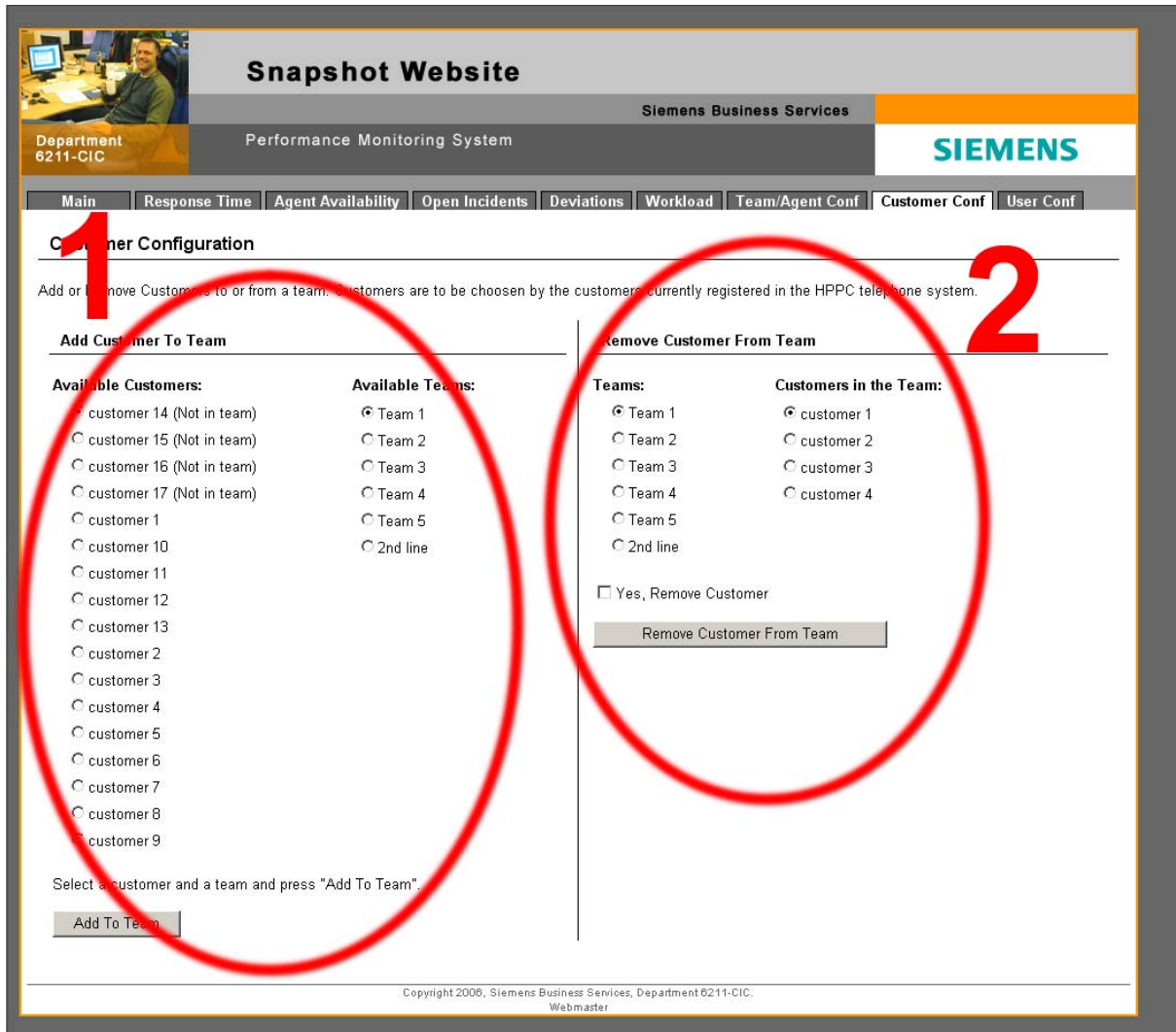


Figure: UM.2.6: Customer configuration.

### 2.2.1 Add A Customer To A Team

The add customer to team section is marked with 1 at figure UM.2.6.

To add a customer to a team:

- 1 Select a customer from the list of customers currently in the HPPC database with the radio buttons. If the customer is not in the list, make sure it is added in HPPC (this could take 15 minutes from the add until it appears on the snapshot website).
- 2 Select which team the customer is to be added to. If the team does not exist, it must be created first, see chapter 2.1.1.
- 3 Click the “Add To Team” button, and the customer will be added to the team.

Figure: UM.2.7: Add a Customer to a team.



## 2.2.2 Remove A Customer From A Team

The remove customer from team section is marked with 2 at figure UM.2.6.

The figure consists of two side-by-side screenshots of a web form titled "Remove Customer From Team".

**Left Screenshot:**

- 1** (Red circle): Points to the "Teams:" section containing radio buttons for Team 1, Team 2 (selected), Team 3, Team 4, Team 5, and 2nd line.
- 2** (Red circle): Points to the "Customers in the Team:" section containing radio buttons for customer 5 (selected), customer 6, customer 7, customer 14, and customer 15.
- 3** (Red circle): Points to the checkbox labeled "Yes, Remove Customer".
- At the bottom is a button labeled "Remove Customer From Team".

**Right Screenshot:**

- The "Yes, Remove Customer" checkbox is now checked.
- 4** (Red circle): Points to a newly appeared checkbox labeled "Confirm Removal".
- 5** (Red circle): Points to the "Remove Customer From Team" button.

Figure: UM.2.8: Remove a customer from a team.

To remove a customer from a team:

1. Select which team the customer is currently in with the radio buttons.
2. Select the agent within that team.
3. Check the "Yes, Remove Customer" check box and the "Confirm Removal" check box will appear.
4. Check the "Confirm Removal" check box.
5. Click the "Remove Customer From Team" button, and the customer will be removed from the team

## 2.3 User Configuration

1. Add a new user to the system, for more information see chapter 2.3.1
2. Alter a user already in the system, for more information see chapter 2.3.2
3. Delete a user from the system, for more information see chapter 2.3.3

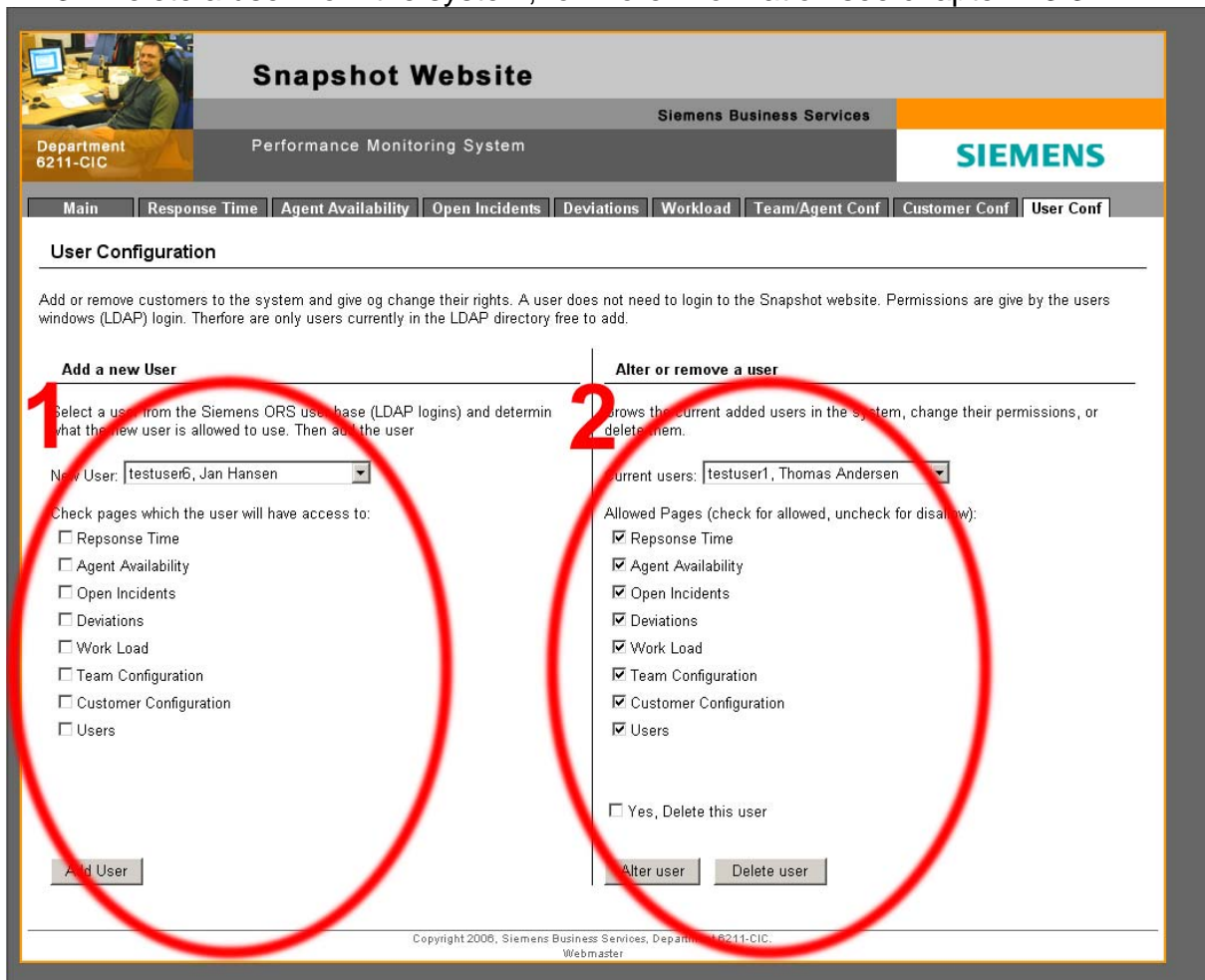


Figure: UM.2.9: User configuration.

### 2.3.1 Add a new user to the system

The add a new user section is marked with 1 at figure UM.2.9.

**Add a new User**

---

Select a user from the Siemens ORS user base (LDAP logins) and determine what the new user is allowed to use. Then add the user

New User:  **1**

Check pages which the user will have access to:

- Response Time
- Agent Availability
- Open Incidents
- Deviations **2**
- Work Load
- Team Configuration
- Customer Configuration
- Users

**3**

Figure: UM.2.10: User configuration.

Add a new user to the system:

1. Select a user from the Active Directory from the drop down list. All users currently registered in the Active Directory will appear in the drop down list and can be added. If the user doesn't exist in the drop down list, check with the administrators of the Active Directory.
2. Use the check boxes to select which pages the user will be allowed to see. All checked pages will be pages the user is allowed to see.
3. Click the "Add User" button, and the user will be added to the system.

### 2.3.2 Alter A User Already In The System

The add agent to team section is marked with 2 at figure UM.2.9.

**Alter or remove a user**

Browse the current added users in the system, change their permissions, or delete them.

Current users: testuser1, Thomas Andersen

Allowed Pages (check for allowed, uncheck for disallow):

- Response Time
- Agent Availability
- Open Incidents
- Deviations
- Work Load
- Team Configuration
- Customer Configuration
- Users

Yes, Delete this user

Alter user Delete user

Figure: UM.2.11: Alter a user already in the system

Alter a user which already is in the system:

1. Select a user already in the system from the drop down list.
2. Check or uncheck the check boxes to alter which pages the user is allowed to see. All checked pages are allowed to see.
3. Click the "Alter User" button, and the user will be altered in the system.

### 2.3.3 Delete a user from the system

The add agent to team section is marked with 2 at figure UM.2.9.

The figure displays two screenshots of the 'Alter or remove a user' interface. Both screenshots show a dropdown menu for 'Current users' containing 'testuser1, Thomas Andersen'. Below this, there is a list of 'Allowed Pages' with checkboxes for 'Reponse Time', 'Agent Availability', 'Open Incidents', 'Deviations', 'Work Load', 'Team Configuration', 'Customer Configuration', and 'Users'. In the left screenshot, the 'Yes, Delete this user' checkbox is checked, and the 'Delete user' button is visible. In the right screenshot, the 'Confirm delete' checkbox is checked, and the 'Delete user' button is visible. Red circles and numbers (1, 2, 3, 4) highlight the dropdown menu, the 'Yes, Delete this user' checkbox, the 'Confirm delete' checkbox, and the 'Delete user' button respectively.

Figure: UM.2.12: Delete a user from the system.

Delete a user from the system:

1. Select a user already in the system from the drop down list.
2. Check the “Yes, Delete this user” check box, and the “Confirm delete” check box will appear.
3. Check the “Confirm delete” check box.
4. Click the “Delete user” button, and the user will be deleted from the system (the user will still be in the Active Directory)

### 3. Roll out the system

The user system consisting of the local database and the web application can be deployed at other locations than just Department 6211-CIC at Linderud in Oslo.

#### 3.1 The Main System

The two main parts of the system, the web application and the database, needs:

- A web server which supports ASP.net 2.0
- A database server which supports stored procedures

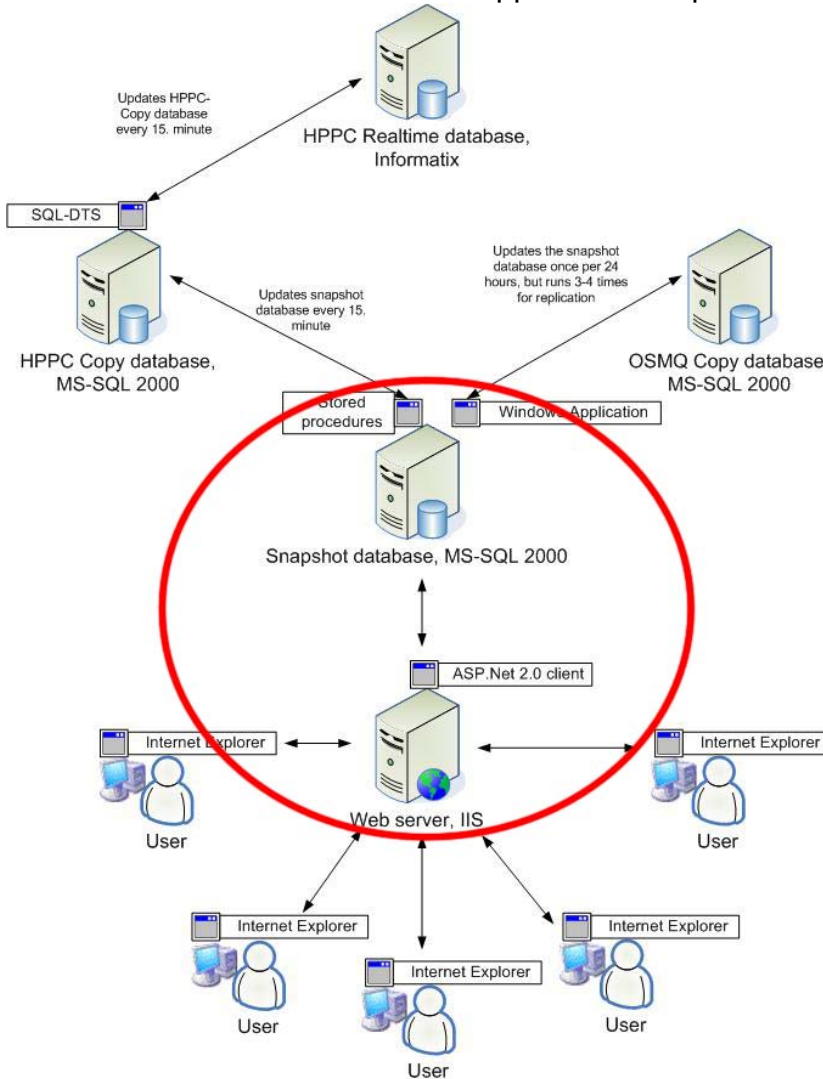


Figure: UM.3.1: System overview for the main parts

The web server is working independent without any connections to the application or system which fetches data from the underlying databases.

##### 3.1.1 The Web Application

The web application can be moved to a new web server only by copying the folder containing the application to a web directory on the new web server.

The web application has a database.config file where the connection string for the database to be used. When the folder is moved and the database.config file is updated, the web application will communicate with the database and it will be ready for use.

### 3.1.2 The Database

The database needs to be created on a database server manually.

All tables and constraints can be created with the CREATE\_snapshot stored procedure.

The database must have all stored procedures except the OSMQ\_ stored procedures installed which provides the web application with data.

## 3.2 Data updating

Data must be filled into the snapshot database. How this is done is not important, but it is important that the right data is imported.

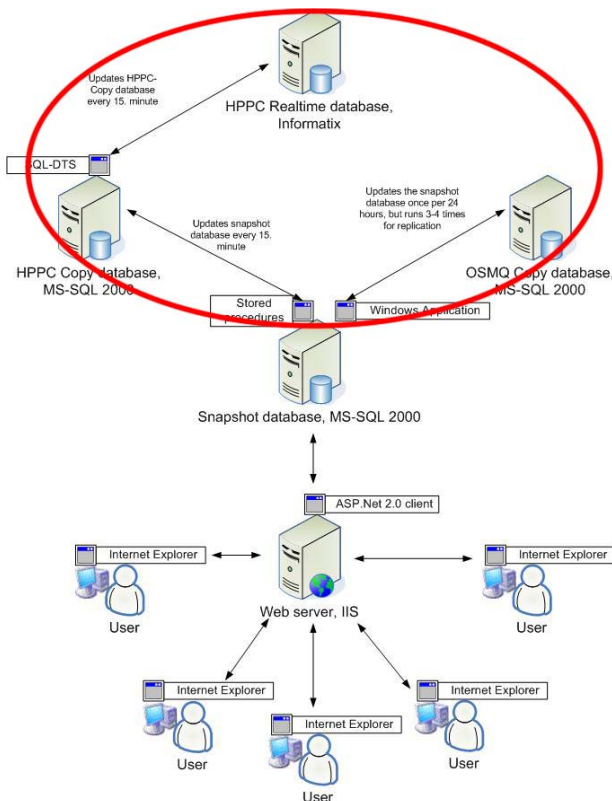


Figure: UM.3.2: System overview, data update

The data that must be imported:

- Incidents registered in the system used for incident registering
- Open incidents (incidents which hasn't been solved yet)
- Agents answering calls and registering incidents
- Agents in state (talking, working, available, or unavailable)
- Customers calling in
- Answered call data (response time, from which customer, call time, and the agent answering)
- Abandon call data(abandoned time (time from call start until it was hung up), from which customer, call time)

- Wrap up reason reports (calls which are reminders from customers, personal calls, or something else which are not registered as an incoming phone call)

Today HPPC\_ stored procedures through a DTS and OSMQ\_ stored procedures with the Windows application are responsible for updating the data.

The HPPC stored procedures are copying data from the HPPC Copy database at a given interval, and are started by the DTS job which updates the HPPC Copy database with information from the HPPC real time database.

The Windows application runs several times a day, but this is only to ensure that all data is retrieved from the OSMq Copy database. It is started as a scheduled program and is using stored procedures at the OSMq copy database to retrieve data, and stored procedures at the Snapshot database to insert data.