

# Visualising and scaling information in a flight operations schedule

Authors: Birger Hedberg-Olsson & Omar Khan (LTH)

Supervisors: Mattias Wallergård (EAT, LTH) & Vadim Feldman (Flygprestanda AB)

Examiner of master thesis: Joakim Eriksson (EAT, LTH)

## Preface

This master thesis was carried out at the Division of Ergonomics and Aerosol Technology, Department of Design Science, Lund University in collaboration with Flygprestanda AB in Malmö.

We (the authors) both study Information and Communication Engineering Technologies at LTH, where we specialise in Design and Usability. Our main interest is working with the user interface and the user perception. This includes user studies, usability studies, documentation and of course coding. In the future we wish to use our skills and experiences to simply make life easier for people

We would like to thank Vadim Feldman (supervisor at Flygprestanda AB) and Mattias Wallergård (supervisor at LTH). We also want to take the opportunity to thank each other, and our families and friends for their support and love.

Birger Hedberg-Olsson & Omar Khan

## Abstract

Flygprestanda AB supplies IT services for airline operations, management and control. The system delivered to facilitate this is the Flight Operation Control System or FOCS. This thesis addresses the Schedule View service included in the FOCS system. The Schedule View is a tool where the user gets an overview of aircrafts and planned flights. In addition aircrafts and flights can be added and modified.

It is crucial that the usability is not hampered by the number of aircrafts and their respective flights. Since Flygprestanda's customers vary in size, from large airlines to single aircraft customers (business jet), the Schedule View should be able to fulfil all needs.

With an iterative work process we will locate flaws in the user interface, and design and implement improvements. To ensure quality of our implemented solutions an evaluation phase will be included in the iterative work process. The results of our evaluation activities will be presented as a framework for service implementation in this domain.

## Sammanfattning

Flygprestanda AB levererar IT-tjänster för flygindustrin. Bland dessa ingår produktion FOCS som används för att schemalägga och hantera flighter. Denna avhandling behandlar tjänsten 'Schedule View' som är en del av FOCS. Schedule View är det verktyg operatörer använder för att få en överblick över planerade flygningar i realtid. Utöver detta kan operatören med detta verktyg göra förändringar i den planerade verksamheten.

Det är viktigt att användbarheten inte påverkas negativt av antalet flygplan och deras flighter i systemet. Eftersom Flygprestandas kundes varierar i storlek, från stora flygbolag till affärsflyg, är det nödvändigt att Schedule View anpassar sig till att lösa uppgiften.

Med en iterativ arbetsprocess ämnar vi att finna svagheter i användargränssnittet, samt utforma och implementera förbättringar. För att säkerställa kvalitén på våra implementerade lösningar ingår en utvärderingsfas i det iterativa arbetet. Resultatet kommer att presenteras i form av ett ramverk för utveckling av tjänster för flygindustrin.

**Key words:** interaction design, usability evaluation, usability design, scalability, user interface

# List of Contents

1	1	Introduction.....	8
1.1	1.1	Structure of report.....	8
1.2	1.2	Background.....	9
1.2.1	1.2.1	Flygprestanda.....	9
1.2.2	1.2.2	Scope.....	9
1.3	1.3	Issues.....	9
1.4	1.4	Thesis purposes.....	9
1.4.1	1.4.1	First purpose.....	10
1.4.2	1.4.2	Second purpose.....	10
1.4.3	1.4.3	Third purpose.....	10
1.5	1.5	Goals.....	10
1.6	1.6	Resources.....	10
1.7	1.7	Delimitations.....	10
1.8	1.8	Structure of Master Thesis.....	11
2	2	Theoretical framework.....	12
2.1	2.1	Usability.....	12
2.1.1	2.1.1	Definitions.....	12
2.1.2	2.1.2	Usability principles.....	13
2.2	2.2	User centred design.....	13
2.3	2.3	Data gathering techniques.....	15
2.3.1	2.3.1	Interview.....	15
2.3.2	2.3.2	Observations.....	15
2.3.3	2.3.3	Questionnaire survey.....	16
2.3.4	2.3.4	Brainstorm.....	17
2.4	2.4	Evaluation and test methods.....	17
2.4.1	2.4.1	Heuristics evaluation.....	17
2.4.2	2.4.2	DECIDE-framework.....	18
2.4.3	2.4.3	Usability testing.....	19
2.5	2.5	Requirements engineering.....	20
2.6	2.6	Pre-study activities.....	20
2.6.1	2.6.1	SWOT analysis.....	21
2.6.2	2.6.2	Stakeholder analysis.....	21
2.6.3	2.6.3	Work breakdown structure.....	22
3	3	Specification & design.....	23
3.1	3.1	Planning.....	23
3.2	3.2	Research.....	23
3.2.1	3.2.1	Publications Database.....	23
3.2.2	3.2.2	State of the art.....	25
3.2.2.1	3.2.2.1	Current Schedule View.....	25
3.2.2.2	3.2.2.2	Domain research.....	25
3.2.2.3	3.2.2.3	Similar systems.....	25
3.2.2.4	3.2.2.4	Competitors.....	25
3.3	3.3	Pre-study.....	27
3.3.1	3.3.1	SWOT analysis.....	27

3.3.2	3.3.2 Stakeholder analysis.....	27
3.3.3	3.3.3 Interview.....	28
3.3.3.1	3.3.3.1 First part of interview.....	29
3.3.3.2	3.3.3.2 Second part of interview.....	29
3.4	3.4 Requirements specification.....	30
3.5	3.5 Work breakdown structure.....	31
4	4 Usability design & evaluation.....	32
4.1	4.1 Iteration 1.....	33
4.1.1	4.1.1 Brainstorm.....	33
4.1.2	4.1.2 Heuristics evaluation.....	33
4.1.3	4.1.3 Interview.....	35
4.1.4	4.1.4 Observation Husarrest.....	35
4.2	4.2 Iteration 2.....	38
4.2.1	4.2.1 Assessment test – Test person .....	38
4.3	4.3 Iteration 3.....	40
4.3.1	4.3.1 Feature review.....	40
4.4	4.4 Iteration 4.....	41
4.4.1	4.4.1 Validation test.....	41
4.4.2	4.4.2 Questionnaire Survey.....	41
5	5 Implementation.....	43
5.1	5.1 Code implementation.....	43
5.2	5.2 Verification.....	43
6	6 Results.....	44
6.1	6.1 Results from iterative work.....	44
6.1.1	6.1.1 Mock-ups.....	44
6.1.2	6.1.2 Prototypes.....	48
6.1.3	6.1.3 Implementations.....	53
6.1.4	6.1.4 Concept / ideas.....	53
6.2	6.2 Results from SWOT analysis.....	54
6.3	6.3 Results from stakeholder analysis.....	56
6.4	6.4 Results from interview.....	58
6.5	6.5 Results from Work Breakdown Structure.....	58
6.6	6.6 Results from Observation.....	59
6.7	6.7 Results from Assessment test.....	62
6.8	6.8 Results from Validation test.....	63
7	7 Discussion.....	63
7.1	7.1 Project initiation.....	63
7.2	7.2 Results.....	63
7.2.1	7.2.1 Iteration 1.....	63
7.2.2	7.2.2 Iteration 2.....	64
7.2.3	7.2.3 Iteration 3.....	64
7.2.4	7.2.4 Iteration 4.....	65
7.3	7.3 Project environment.....	65
7.4	7.4 Usability.....	66
7.5	7.5 User centred design.....	66
8	8 Conclusions.....	67
9	9 References.....	68

9.1	9.1 Factual references.....	68
9.2	9.2 Inspirational publications.....	69
10	10 Appendix.....	70
10.1	10.1 Framework used when conducting Observation.....	70
10.2	10.2 Assessment test.....	72
10.2.1	10.2.1 Test plan.....	72
10.2.2	10.2.2 Scenario list.....	73
10.2.3	10.2.3 Test monitor guidelines.....	74
10.3	10.3 Validation test.....	76
10.3.1	10.3.1 Scenario list.....	76
10.3.2	10.3.2 Test monitor guidelines.....	76
10.4	10.4 QUIS Questionnaire with modifications made.....	78
10.5	10.5 Internal review.....	82
10.6	10.6 Publications database.....	83
10.7	10.7 Results from interview.....	84
10.8	10.8 Results from Assessment test.....	85
10.9	10.9 Results from Validation test.....	90

## Dictionary

Aircraft flight manual (AFM)	A document containing the information and instruction required to operate the aircraft in a safe manner.
ADEP	ICAO code that determines what airport a flight departs from.
ADES	ICAO code that determines the destination airport for a flight.
Cancel (a flight)	The action of cancelling a pre-planned flight
ETA	A flights' estimated time of arrival
ETD	A flights' estimated time of departure
Flight	A journey made by an air-plane, that has a place of departure and arrival, a departure- and arrival time and a route.
Flight schedule (view/service/component)	A component in the FOCS-system that allows the user to build a schedule that handles flights and data related to them.
Flight operations control system (FOCS)	The system for which the scheduling component is develop and will be integrated into.
Initiator	The project customer, our main contact and supervisor at Flygprestanda
Slot time	CTOT calculated take-off time, an allotted time-frame within which the air-planes' take-off has to take place
STA	The planned arrival time of a flight
STD	The planned departure time of a flight
Tail	A specific aeroplane.
Pairing	Assigning crew to a specific flight.
Re-schedule (a flight)	The act of giving a planned flight a new time table
Route (flight)	The way or path an aeroplane negotiates to get from its place of departure to its destination.

# 1 Introduction

## 1.1 Structure of report

The report of the master thesis is structured in the following sections:

### *1. Introduction*

The presentation of the academic and commercial issues that constitutes the foundation of the master thesis.

### *2. Theoretical framework*

The academic areas the authors draw their knowledge from as well as the descriptions of the tools and frameworks used in activities concerning the master thesis.

### *3. Specification & design phase*

The initial project activities performed at Flygprestanda and activities to gather information for requirements specification.

### *4. Usability design & evaluation*

The activities that were performed in order to ensure the usability of Schedule View with focus on the issues and purposes described in section 1.

### *5. Implementation*

Describing the process of transferring concepts and prototypes into the FOCS Schedule View software.

### *6. Results*

The collection of results that was gathered from activities conducted throughout this thesis. This is used as an internal reference in this report.

### *7. Discussion*

A general discussion section concerning areas of interest and issues encountered while working with this thesis.

### *8. Conclusions*

A presentation of the conclusions regarding the purposes of this thesis.

### *9. References*

### *10. Appendix*

## 1.2 Background

In this section we introduced Flygprestanda AB, to the scope of the flight operations, and by extension the schedule view, project.

### 1.2.1 Flygprestanda

Flygprestanda AB produce software and databases for hundreds of airline companies all over the world. Their services includes both software development and technical-engineering calculations with focus on delivering all information needed to perform a commercial flight from one place to another.

### 1.2.2 Scope

The scope of this project is to investigate and implement improvements for a scheduling component for use in flight planning operations. The component will be used by an airline operator for both long and short term planning, delay recovery and maintenance scheduling. The schedule will be integrated with the existing flight planing system and serve a central role in the daily operations for an airline. It is crucial that the component is easy to use and allows a user to solve the required tasks with minimum effort.

## 1.3 Issues

At an early stage in the project we sat down with the initiator to discuss the service and what tasks and phases the project will comprise of. We discussed what the key elements in a successful schedule service rely on and what issues relate to them. Together with the initiator we specified the following list of potential issues:

- In what way should a schedule visualize data/information?
- Which information should be visualized at the same time?
- How should the service scale the visualization, without impairing the usability?
- How to design the desired functionality so that all user types can complete their required scheduling tasks in a satisfactory fashion?
- The service should minimize the load on the user's short term memory.
- Shorter time spent/less actions per task.
- What functionality is required of the service?

## 1.4 Thesis purposes

From an academic point of view, just implementing the schedule component for Flygprestanda would not be sufficient for a master thesis. Instead, we wanted to generalize the task and focus on usability and interaction design issues that emerge. We specified three purposes using the issues above. The first purpose relates to schedules and similar services in general and the two following concentrates more on the schedule component developed for Flygprestanda.

### 1.4.1 First purpose

Since a schedule at times visualizes a lot of information simultaneously, it is crucial that the interface does not confuse the user in its representation. The first purpose is a simple question, but answering it requires solving issues which we will discuss later in the thesis:

- **In what way should a schedule visualize information?**

### 1.4.2 Second purpose

The first purpose concentrates on how information should be visualized. But what happens when the amount of information grows? Will more information have an impact on the visualization? Since Flygprestanda has different customer types with different needs (in this case the customers vary in size i.e. amount of tails and flights) the schedule needs to be able to scale, and fit different needs:

- **How should the service scale the visualization, without impairing the usability?**

### 1.4.3 Third purpose

The first and second purposes covers the visualization of the service, but not the interaction between the users and the service and the issues that can occur:

- **What are the main issues regarding usability when interacting with the service?**

## 1.5 Goals

Our initiator at Flygprestanda presented their overall goals for this project:

- Investigation of all involved stakeholders and requirements
- Investigation and specification of the interaction design for a scheduling component
- Implementation of a component supporting scheduling operations

By fulfilling Flygprestanda's goals stated above and also answering the three purposes for academic value, our master thesis would be complete.

## 1.6 Resources

The project was carried out at Flygprestanda AB in Malmö. We were provided with workstations, computers and all software necessary for the project. We got assistance from Flygprestanda personnel such as the manager (initiator), system architect, programmers, sales department and the graphics team. Our supervisor from the Division of Ergonomics and Aerosol Technology was Mattias Wallergård who helped us with issues of academic nature and with this report.

## 1.7 Delimitations

The main focus of this master thesis as a part of the FOCS project is designing the interaction of the Schedule View service of FOCS with focus on the thesis purposes. Implementation and verification also play a role, but is not the priority of the thesis.

## 1.8 Structure of Master Thesis

### *1. Start-up & Investigation*

We started the project by meeting with the initiator at Flygprestanda to discuss the content and goals of the project. Together we identified which problems we were going to address and solve. With this information at hand we specified the purpose of the project. Finally we identified potential risks and issues that could occur during the project.

### *2. Specification and design*

This phase began with a pre-study covering the managerial aspects of the project as well as resulting in the completion of the project requirements specification. Following the pre-study we began the iterative design process resulting in concepts and prototypes to be implemented.

### *3. Implementation*

The implementation phase introduced our current design solutions into the Schedule View software. This was also carried out in an iterative manner where each iterations was followed by a validation of current interaction design.

### *4. Validation*

In order to validate the work we used different tools to asses the accuracy of our implementation. This validation became the foundation for the next iteration of design.

## 2 Theoretical framework

The following sections describe theories that were used as a basis for the decisions concerning the interaction design on the Schedule View we have made during the course of the project. The initial section discuss theories regarding the concept of usability. Following are description of usability tools used in this project. Lastly, there is a section describing requirements engineering, a field the authors felt useful and techniques that have been used as required by the Flygprestanda project development profile.

### 2.1 Usability

Usability is a term often used during product development. Grasping what usability really constitutes can sometimes be difficult due to the situational nature of usability. There is no format, standard or framework that will result in an arbitrary product being usable.

The usability of a product depends on the user, the environment, in what situation it is being used and the task at hand. An interface that works and is considered usable in a certain scenario may not be at all usable in another environment, or with another type of user.

The situational nature of usability combined with the need for an approach to the subject in an engineering fashion has resulted in forming of general principles and terms, defined by leading academics in the field of interaction design. This has been done in order to quantify the different traits of usability and to be able to create tools that ensures usability when interacting with the developed artefact.

A way to concretize and measure usability during a development project is to pick one of these definitions and use it as a basis when specifying, designing and validating the product.

The following subsections present some of the most widely used definitions, in short, including the REAL model that was used as the basis during this project. The field of usability is vast and the aspects of the definitions and their often contradictory terms and principles is of huge interest to the authors and we encourage the reader to immerse themselves in the subject.

#### 2.1.1 Definitions

##### **ISO 9242-11 standard**

The ISO standard for usability defines it as: “The 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.”[1]

##### **Definition by Jakob Nielsen**

Nielsen defines usability as an attribute that assesses how easy user interfaces are to use. He defines usability using five “quality attributes”[2]:

- Learnability – How easy is it for users to accomplish basic tasks the first time they encounter the design?
- Efficiency – Once users have learned the design, how quickly can they perform tasks?

- Memorability – When users return to the design after a period of not using it, how easily can they re-establish proficiency?
- Errors – How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction – How pleasant is it to use the design?

### **REAL model by Jonas Löwgren**

Jonas Löwgren describes usability with the so called REAL model[3], which stand for Relevance, Efficiency, Attitude and Learnability:

- Relevance – How well the system answers to users needs.
- Efficiency – How efficiently users can perform their tasks.
- Attitude – The user's subjective feelings towards the system.
- Learnability – How easy the system is to learn and how easy it is to remember how to use the system.

### **2.1.2 Usability principles**

This section describes design principles stated by Donald Norman[4]:

- Visibility – If a function is visible it will be easier to use than if it is not visible or hard to spot.
- Feedback – Inform the user what actions are being performed, the consequences of actions, what results has been accomplished and what state the system is in.
- Mapping – Describes the relationship between objects. For example, if you turn the steering wheel in a car clockwise the car will turn right.
- Constraints – Limiting actions that can be performed in the current state of the system
- Affordance – An object in the system should be self explanatory in two matters, how to use it and what it achieves. For example a button should invite the user to click it.
- Consistency – A consistent system is one that follow rules. For example, an action performed in two different states of the system should result in the same way.

## **2.2 User centred design**

In User Centred Design (UCD) the users are involved during all stages of the project. In an early stage user studies will be held to locate needs and limitations of the users. During the design phase and the implementation phase user tests are held to validate and confirm ideas and implemented functionality. Lastly surveys are held to verify delivered solutions and to be used as a basis when continuing improving the product.[5]

An example of activities that can be included in UCD[6]:

### **Analysis Phase**

- Meet with key stakeholders to set vision
- Include usability tasks in the project plan
- Assemble a multidisciplinary team to ensure complete expertise
- Develop usability goals and objectives
- Conduct field studies
- Look at competitive products
- Create user profiles
- Develop a task analysis
- Document user scenarios
- Document user performance requirements

### **Design Phase**

- Begin to brainstorm design concepts and metaphors
- Develop screen flow and navigation model
- Perform a walk-through of design concepts
- Begin design with paper and pencil
- Create low-fidelity prototypes
- Conduct usability testing on low-fidelity prototypes
- Create high-fidelity detailed design
- Do usability testing again
- Document standards and guidelines
- Create a design specification

### **Implementation Phase**

- Do ongoing heuristic evaluations
- Work closely with delivery team as design is implemented
- Conduct usability testing as soon as possible

### **Deployment Phase**

- Use surveys to get user feedback
- Conduct field studies to get info about actual use
- Check objectives using usability testing

## 2.3 Data gathering techniques

The following sections describes techniques used gathering data during the project.

### 2.3.1 Interview

The interview is a very useful tool in interaction design. It is very versatile, easy to facilitate and can be used separately or as part of another method for debriefing or introductory purposes. Depending on the structure one chooses for the interview, the interview lets you approach interaction and usability issues of different variety and at different stages of the design development.

The structure of the interview depends on the type of data the interviewer is gathering and in what stage of the project the interview takes place. In the early stages of development, an unstructured approach to the interview that resembles a topic related conversation is appropriate. Later on when for example performing evaluation and validation, a structured approach with predetermined questions with corresponding sets of answers is favourable.

The unstructured interview is a very costly approach since it produces a large amount of data that needs analysis. At the opposite end the structured interview is more or less an oral questionnaire and lacks the exploratory qualities and in-depth analysis of the unstructured approach.

We use the cross section of the approaches described above, the semi-structured interview. This approach combines the structure and control with exploration and analysis.

#### **Semi-structured interview**

This technique consists of a set of questions in the form of topics that the interviewer wants to discuss. This question set is maintained so that other interviews can be performed in the same fashion. The idea is to discuss the specified topics and ask follow-up questions regarding the contents of the discussion. By doing this the semi-structured interview is intended to exhaust the topics of all vital information.

[7][8][9]

### 2.3.2 Observations

Users can have a hard time both describing how they go about solving the task of their day-to-day work and describing in detail what the tasks entail. User observations help designers gather vital data regarding the user's goals, which context tasks are performed, and to some extent insight into the users work domain. Depending on the orientation of work, observations can be made both early on, to gain knowledge regarding the user and issues they encounter in their work, and as part of evaluation testing of the properties of prototype design.

Direct observation in the field entails observation of the user in their normal work environment engaged with tasks of a day-to-day nature. The field observations fill the gaps left by questionnaires and interviews made in the early stages of interaction design. They add detail to what activities the users partake in and why, as well as shed light on issues both known but also issues that the user might not be aware of because prolonged use of a tool or service has accustomed the user to the inferior design.

To help deal with the complexity and frequently changing nature of the events and tasks observed, a framework of what is to be observed is helpful. This framework assists the observer in keeping in mind what is to be observed in order to complete the goals of the observation. For example:

- Space – What is the physical space like, how is the layout of the work environment?
- Actors – What are the relevant details regarding the users that are observed?
- Activities – What are the actors doing and why?
- Objects – What physical objects are present in the actors work environment?
- Acts – What are specific individual actions?
- Events – Is what you observe part of a special event?
- Time – What is the sequence of events?
- Goals – What are the actors trying to accomplish?
- Feelings – What is the mood of the group and of individuals?

The field observation demands the observer to establish the degree of his or her participation. At one end of the spectrum the observer is totally detached from the environment, the task at hand and the actors. The observer is characterized as an *outsider* or *passive observer*. One can however argue how detached the passive observer is since he or she is infant in the field and therefore must have some impact on events. Nevertheless a passive observer takes no deliberate actions during the observation.

At the other end of the spectrum is the fully immersed observer. The *insider* or *participant observer* strives to be a full member of the study group and actively take action to solve the task at hand with them. This role is inherently difficult since the observer needs to separate between the two roles of observer and participant, to engage and observe at the same time is arguably a hard feat.

The topic of a ethnographic approach to field observations will not be covered. The approach is not used since its features revolve around the participatory role of the observer within a study group. [7][8]

### 2.3.3 Questionnaire survey

Generally a questionnaire and a structured interview can both be used to gather the same data. The benefit of the questionnaire is that performing a survey with one can be done in a much larger scale. Furthermore the questionnaire allows for less hassling of the user, who might not have the motivation for the interview situation but can accept the more anonymous format of filling out the questionnaire. The questionnaire is also less costly since users and interviewers do not need to meet in person which eliminates the need to facilitate a locale and the expenses of travel. The questionnaire is also much more easy to scale, both in performing the survey and collection of data, than an interview and is therefore appropriate if you have a large number of user that you can distribute to.

The trade off for the low cost of performing a survey using a questionnaire is the difficult task of designing consistent and unambiguous questions. This is important because no one will be at hand if the user is uncertain of the wording of questions. In this case the answers given will suffer in validity and worse might not be answered at all.[7][8]

The following general design guidelines are provided by [7] as a means to minimize the issues of performing a survey with a questionnaire:

- Mind the sequencing of questions. The impact of a question can be influenced by their ordering in the questionnaire.
- Consider the target group. Are there groupings within the target group that require different

questionnaire formats.

- Provide clear instructions on how to complete the questionnaire.
- Balance between white-space and the need for a compact format must be achieved. A long questionnaire deters the participant to complete it.

### 2.3.4 Brainstorm

The brainstorm technique is not a usability specific tool and we used it during both the specification and design phase as well as when working with usability and evaluation.

This technique is meant to generate ideas and concept in a free format where all additions are welcome without critique. The brainstorm will undoubtedly result in infeasible concepts and ideas, but if these are discussed and analysed a feasible alternative is often found or become the seed for other ideas.

The final result of the brainstorm can be used for the elicitation of requirements, generating new features or concepts, or as an instrument to address issues all through the project development.

## 2.4 Evaluation and test methods

### 2.4.1 Heuristics evaluation

This is a technique used to inspect the usability of an artefact by means of experts that, guided by a set of usability principles, evaluate the design. The features and different aspects of the artefact are compared to the principles and analysis and where it is necessary change suggestions are made. One set of principles are as follows [7]:

- *Visibility of system status*  
The system shall, with appropriate feedback, keep the user informed of its current state and what processes are active.
- *Match between system and the real world*  
The system shall code content with suitable syntax, familiar to the user from the real world. Information displayed shall follow logical and natural convention and appear in logical and natural order.
- *User control and freedom*  
The system shall prevent the user from choosing an unwanted action, and in the case of this event enable the user to undo the action and return to a previous state without unnecessary dialogue.
- *Consistency and standards*  
The system shall follow platform standards. There shall be no unambiguous syntax or icons, the user shall never question what a feature or action entails.
- *Error prevention*  
Eliminate error-prone conditions, add short and clear confirmation dialogues, if needed, before committing an action and hide features that for the time being are not accessible.
- *Recognize rather than recall*

Minimize user memory load by intuitive and visible actions, objects and options. User shall not act as memory in-between dialogue windows.

- *Flexibility and efficiency of use*  
The system shall cater to both novice and experienced users. Allow for tailoring of frequently used actions.
- *Aesthetic and minimalist design*  
Dialogue shall not contain information that is irrelevant or rarely used.
- *Help users recognize, diagnose and recover from errors*  
Plain language error messages, that precisely indicate the problem and suggests a solution.
- *Help and documentation*  
Although a manual should not be needed in order to use the system, instructions could prove necessary. The information in this documentation shall be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

The evaluator shall go through the different aspects and elements of the system interface regarding the different heuristics and cross-examine and compare them to one another. It is also suggested that category-specific heuristics be developed, and used as a complement, since the ones described above can be too general for some products.

The number of evaluators to be used in order to ensure the majority of usability issues are found is a topic of debate. It is argued that 3-5 expert evaluators will find 75% of all usability issues. Employing multiple experts can be very costly, however the logistics concerning the technique are usually very simple since the technique entails no special equipment or environment (this is naturally dependant on what kind of artefact is to be evaluated). [7]

## 2.4.2 DECIDE-framework

Different evaluation activities have different strengths and weaknesses depending on the state of a project or product development. The DECIDE-framework provides clarity when specifying the scope of an evaluation and choosing which evaluation method should be used.

The framework consists of the following steps:

1. *Determine the goals*  
Specify the high-level goals of the evaluation in order to specify scope and identifying viable evaluation methods
2. *Explore the questions*  
Identify the type of questions that the evaluation should aim to answer. Specify the format of these questions and, if it is necessary, break them down into subsets of questions.
3. *Choose the evaluation approach and methods*  
By making cross references between your issues and needs with the strengths and weaknesses of the evaluation methods the DECIDE-framework facilitates the choice of the correct method or methods.
4. *Identify the practical issues*  
Identify issues, and specify actions to reduce their impact, concerning: the user, test facilities and equipment, schedule and budget constraints.

5. *Decide how to deal with the ethical issues*

If the evaluation imply ethical issues, such as individual privacy, actions must be specified to handle them.

6. *Evaluate, analyse, interpret and present the data*

The interpretation, analysis and evaluation of the resulting data must be made in regard of the factors concerning reliability, validity and biases.

The first step of specifying goals should always be the initial step of the DECIDE evaluation process. The following steps can, and often should be, performed in an iterative fashion and without a specific order.[7]

### 2.4.3 Usability testing

In order to address design issues of a user interface one can perform usability testing. The technique of usability testing is user centred and consists of placing a test subject in a monitored environment, instruct them to perform certain tasks with tools you have supplied them with and collecting measurements regarding the test subjects performance. These collected measurements are then interpreted and analysed with the objective to improve the interaction between user and tool.

Usability testing can be done throughout the development process but serve different purposes depending at what stage they are performed. Generally there are four types of tests which are here described in short:

- *Exploratory test*

An early form for exploring and coming to terms with strengths and flaws with design prototypes. Different approaches to object representation, communication of information, structuring of the interface and other subject are explored and weighted against each other. This type of testing often make use of a walk-through (a “guided tour” of all functions and interactions that comprise on interface) with low-fidelity prototypes depicting different design proposals.

- *Assessment test*

This test is used following up on the findings from exploratory testing. The purpose being on assessing the accuracy of implemented features to the design concepts as well as an assessment of the efficiency of these concepts. Instead of walk-through, this type of testing rely on the user completing tasks and measurable data being collected.

- *Validation test*

The objective of a validation, or verification, test is to compare the design to some sort of usability standard or benchmark before product launch. These standards are usually defined early in the development process and originate from for example previous usability testing, user surveys or heuristic work. The validation test is also task oriented with the addition that interaction between test subject and test monitor is minimal to none. Quantitative data is collected with observations as well as underlying reasons for substandard performance by form of debriefing interview or questionnaire.

- *Comparison test*

The comparison test is made in conjunction with another type of test. This type of testing entails two different design alternatives are tested side-by-side and their strengths and flaws are compared by the accumulated quantitative and qualitative data. This type of testing can be

performed throughout the project at all stages since it is an more an addition to the other test types than a usability test of its own.

Usability testing entails much preparation on the monitors part regarding the test environment and materials, developing tasks and scenarios, debriefing the test subject and analysing the obtained data. This section will not discuss these extensive areas of the field. We do, however, encourage the reader to on their own delve deeper into the subject if she feels inclined. The sections *4.2.1 Assessment test* and *4.4.1 Validation test* cover, in more detail, the test methodologies that were used during this project, how the tests were designed and performed as well as the resulting conclusions and design changes. [7][8]

## 2.5 Requirements engineering

The processes and activities concerning requirements engineering can be organized in three general phases:

1. *Elicitation and analysis*  
Finding, structuring and specifying requirements.
2. *Validation*  
Working with the customer to ensure the specified requirements match their demands.
3. *Verification*  
Controlling that the product is fulfilling the requirements

The first phase of elicitation and analysis produce the requirements specification document. This document is the framework of the product development and will contain all goals, concepts and features the product shall encompass.

The requirements are divided into two types:

- *Functional requirements*  
These requirements specify, at different levels, the functions of the product and their inner and outer computation, recording and transmission of data. They are often specified using textual descriptions, data diagrams, with conceptual models or with task descriptions.
- *Quality- or Non-functional requirements*  
The quality requirements specify in what manner the functions shall perform. They are often divided between the areas of performance, usability and maintenance.

Apart from these types of requirements there are in some cases also specifications of managerial requirements that handle issues of financial and legislative nature regarding the contract with the customer and also within the project development.

Validation and verification of requirements is not covered in this thesis and the theories regarding these areas of software requirements engineering is therefore omitted.

The activities of elicitation, styles of structuring and specifying elicited requirements are numerous and we let the reader decide if they want to further their knowledge in the areas.[9]

## 2.6 Pre-study activities

The following section describes the theories behind the activities performed during the pre-study phase.

They are performed as common practise at Flygprestanda AB when initiating a project.

### 2.6.1 SWOT analysis

SWOT analysis (strengths, weaknesses, opportunities, threats) is a technique used at an early stage in a business project as part of the planning. The goal is to better understand strengths and weaknesses of the project, and also for identifying opportunities and potential risks you could face. [10]

A SWOT analysis can be conducted by following these steps:

- For each of the four keywords (strengths, weaknesses, opportunities, threats), attributes concerning the project is identified.
- The following table is used to cross reference attributes associated to the different keywords:

	<b>Strengths</b>	<b>Weaknesses</b>
<b>Opportunities</b>	Advantages	Temptations
<b>Threats</b>	Defend against	Vulnerability

[11]

- One or several conclusions is drawn and documented for each attribute cross referenced with another. Activities to address issues and business opportunities are specified and when possible every conclusion should have a specified action and owner.
- A prioritization is carried out to determine which of the conclusions are the most important. [12]

### 2.6.2 Stakeholder analysis

Flygprestanda conducts stakeholder analyses as a part of their project pre-studies. This is done in order to identify the actors that affect or are affected by the project. The level of their influence, involvement and willingness of contribution is asserted. The analysis will define the communication channels that need be established in-between stakeholders and responsibility for maintaining the communication is assigned.

The stakeholder analysis is also a step to make sure that Flygprestanda are aware of all the risks that are associated to negative stakeholders that have an adverse effect on the project.[9]

#### **Stakeholder identification for requirement elicitation**

The identification of stakeholders is crucial during requirements engineering. It is essential to ensure that all the stakeholders different needs, attitudes and interests are taken into account when eliciting requirements. The stakeholders that contribute to the project are especially important since they assume their contribution will result in a return. This assumed return must be delivered and in turn be specified as requirements.

For each stakeholder we would seek to answer the following questions:

- Who are the stakeholders?
- What goals do they see for the system?
- Why would they like to contribute?

- What risks and costs do they see?
- What kind of solutions, suppliers, and resources do they see?

[9]

### **2.6.3 Work breakdown structure**

A work breakdown structure (WBS) is a tool used to hierarchically specify and logically organize the scope in a project. It is represented in a tree structure with a number of hierarchic levels (often three levels). The goals of the project is divided into tasks and sub-tasks (components) and each component get a specified position in the tree. A tree structure in this manner specifies the relationship between all components which is useful in the planning stage of the project because it gives a good overview over the entire project. When all sub-tasks for a specific component have been completed, the component (parent) is considered complete. By conducting a WBS you can break down a large project into smaller, more manageable tasks. [13]

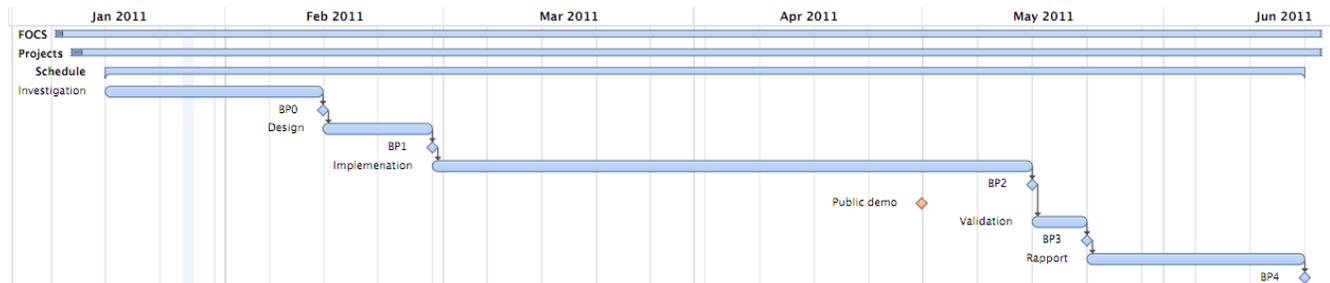
An essential rule in WBS is the 100% rule. It states that each level in the WBS should represent 100% of the work needed to complete the project. This ensures that nothing is neglected in fulfilling the project goal or a sub-goal in the project. [14]

## 3 Specification & design

### 3.1 Planning

We initiated the specification and design phase with a meeting with our supervisor at Flygprestanda we planned and estimated the time for the projects different activities.

The project is structured into five activities, as shown in Illustration 1:



*Illustration 1: Gantt-scheme presenting project phases*

Although the Gantt-scheme gives the appearance of a linear waterfall styled approach to this project the design, implementation and validation activities were, as discussed earlier, performed iteratively. The main purpose of the scheme is to give an estimation on how the time shall be divided between the different activities during the project.

### 3.2 Research

The planning was followed by different forms of research to obtain knowledge of the domain and areas in close and somewhat distant relation to the domain.

#### 3.2.1 Publications Database

##### Introduction

In order to gain a deeper understanding of our field of work and its current state of the art, we conducted research that aimed to construct a database of different articles, literature, web pages and notes from conferences and seminars.

This database would be used through out the different stages of the project. It was used in the early stages as a knowledge base for our queries and for inspiration when we designed the features of the service. We were also able to verify or deny the assumptions and hypothesis we made along the way with references from other studies. Finally the database would help us with the validation of our findings and conclusions.

##### Method

To acquire the material needed to create an acceptable database we needed to define in which areas we would search for information and how we would limit the search.

We argued that this could be solved by outlining a number of terms vital to the project, and combine them with each other and then use the combination of terms as search queries.

We arbitrarily excluded some combinations that we felt were of no use. Likewise we saw that some of the search queries resulted in too numerous publications in a wide area of subjects, and parsing through the search result would have been far too taxing.

To counteract this, we made another search iteration where we added another search term i.e a combination of three terms from our outline were used as search queries. This resulted in fewer hits than when the two-term combination was used as search queries, but the result was narrowed down to more suitable areas and subjects of interest.

The resulting publications from the searches made were scrutinized in the following fashion:

1. We found the title intriguing.
2. The abstract gave the impression that this was a publication suitable for our database.
3. A skimming of different sections of the publication.
4. Adding or discarding the publication for the first draft/version/edition of the database, taking note of which query was responsible for a database entry.
5. A thorough reading of the articles in the first draft of the database, noting sections of interest.
6. Discarding publications from the first draft, creating the final version of the database.

Apart from adding the resulting artefacts to our database from this structured search, we added different publications we came across by direct references from our mentor or that were referenced to in publications in our database. These publications were scrutinized in the same fashion as mentioned above.

### **Delimitations**

One can argue about the impact on the flaws of our research that, we for example only used one search engine and made arbitrary decisions on legitimacy and importance of publications.

During the initial phase we tried different search engines. We found that there were a lot of redundant hits, with the use of both different engines and search queries, this created redundant work. In order to minimize this taxation on our work load we chose to stick with the LibHub engine which we felt gave the most suitable hits and that covered the most databases.

We were discouraged by our supervisor to construct a method to discern legitimacy and importance for this project, he argued/stated that this is more suited for when one is conducting research for a paper at a higher academic level.

### **Results**

We present the result of our research (the first draft of the database) number, the outlined terms used, how the terms were combined into queries and what exclusions were made in section *10.6 Publications database*.

The final draft of the database is not included in this document but is included, in full, in the reference section.

### 3.2.2 State of the art

This section describes the state of the current Schedule View, its competitors, similar systems and our domain research.

#### 3.2.2.1 Current Schedule View

When we started this project there was already a version of the service up and running. An evaluation of the Schedule View as it was before we started our project is found in section 4.1.2 *Heuristic evaluation*.

#### 3.2.2.2 Domain research

Activities performed during the pre-study combined with studying publications concerning the domain gave us the base knowledge needed to perform this project. Apart from the inspirational value of this work we were also exposed to the issues and constraints in the field of flight operations.[23][24][25]

#### 3.2.2.3 Similar systems

We were interested in finding other systems or software that resembles the Schedule View or that is used in a similar manner as the Schedule View. Since our software is designed to solve a very specific task it does not have a lot of users compared to other software, which made UCD harder to conduct. Our software is used only in a business environment, we have no users that use the Schedule View privately. We came up with the idea to locate software that is partly or entirely structured or designed in a similar way as the Schedule View.

During the project we came across programs or program types that were useful to us as inspiration for concepts and solutions. From these we focused on the following:

- Logic Pro – A professional music application.
- Hipmunk – Search engine for finding flights and hotels.[15]
- Inspirational publications of similar systems.[26][27][28]

#### 3.2.2.4 Competitors

We know of a few competitors that offer software that solves the same tasks as the Schedule View does. However, we have not had any opportunity to use any of them. We have looked at screen-shots from the PDC Airline-suite promotional documentation, shown in *Illustration 2* and *Illustration 3*, and used this as inspirational material.

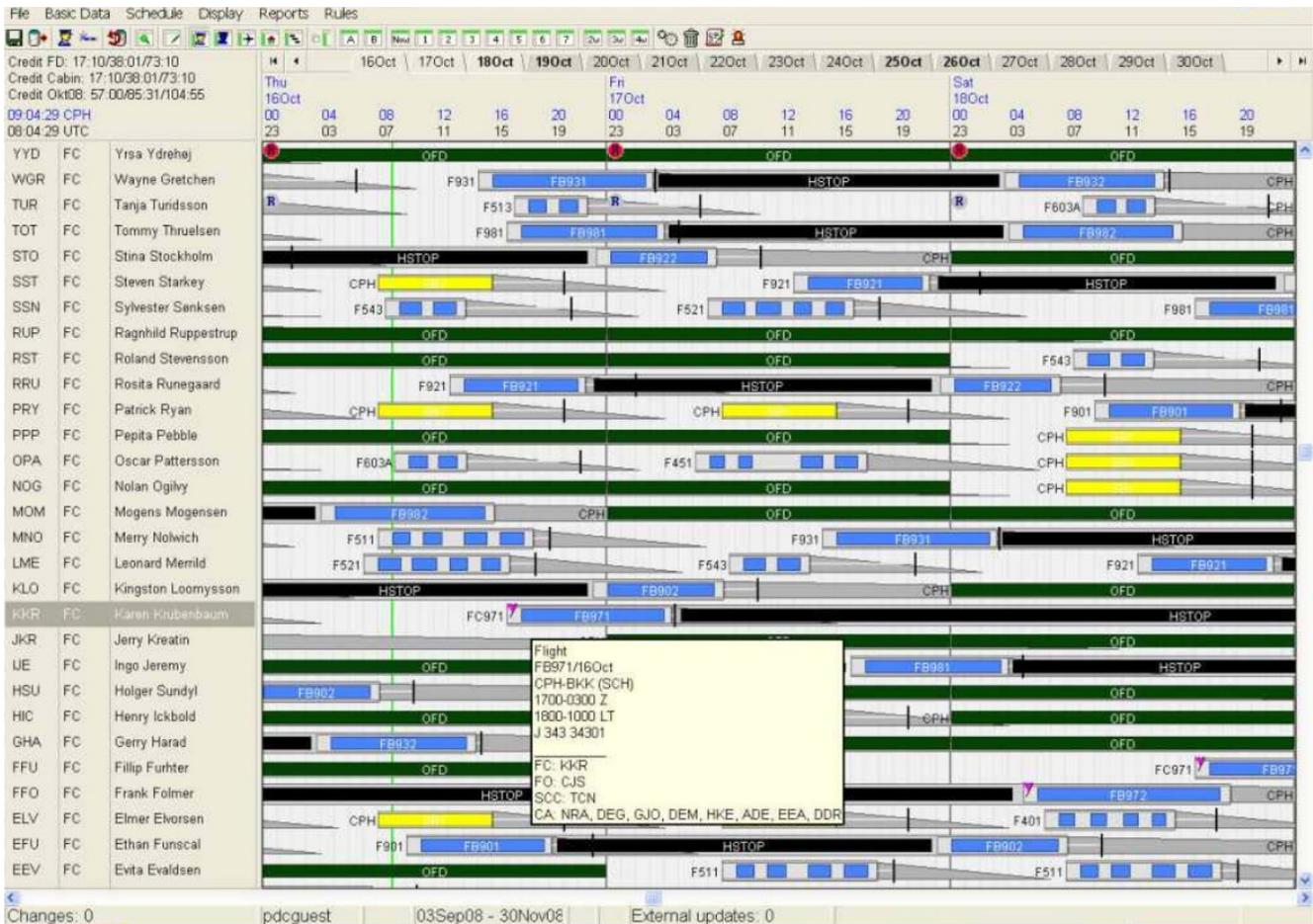


Illustration 2: The PDC Crew client application

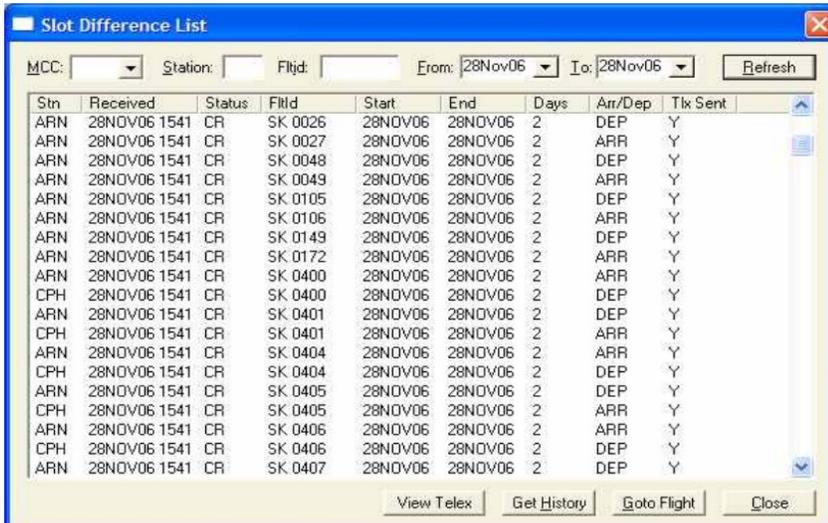


Illustration 3: The PDC Slot handler application

### 3.3 Pre-study

The sections of chapter 3.3 describe how the pre-study activities were implemented, and in what manner and capacity the results were used, during this project.

#### 3.3.1 SWOT analysis

Our purpose of conducting a SWOT analysis was mainly to identify potential threats and risks that we had not already thought of. We conducted the analysis together with the initiator who had more experience in the field. The analysis found four conclusions to be crucial as shown in Figure 1:

ID#	Conclusions	Actions	Handled by
D11	A well specified and user centred interaction design will shorten the implementation and verification phases.	Take time working with interaction design. Do not make assumptions. Verify quality of our interaction design.	Project group
D12	The fact that we are late to market should not influence the quality of the interaction design since this is important.	Assign appropriate amount of resources to the design phase. Do not rush the interaction design.	Project group Initiator
T1	New development without the end user in mind can result in undesired functionality.	Implementation should originate from requirements.	Project group
T3	We need to keep in mind that we have end users with different experience and background.	Verify the design with different user types.	Project group

Figure 1: Excerpt from SWOT-Analysis

The analysis did not reveal any new risks or threats, but instead it confirmed the importance of good usability and interaction design.

An action list for all the high priority conclusions can be found in section 6.2 *Results from SWOT analysis*.

#### 3.3.2 Stakeholder analysis

As a part of the project pre-study we performed a stakeholder identification and analysis along with the project initiator. The motive of this was threefold:

1. As a part of Flygprestanda's business administration procedure
2. In order to construct a tool for requirements elicitation
3. To identify the FOCS schedule view services' end-users for interface design purposes

The stakeholder analysis was conducted following the framework specified in section 2.6.2 *Stakeholder analysis*. The result of the stakeholder analysis can be found in section 6.3 *Results from stakeholder analysis*.

The following stakeholders are identified in this project, their placement show the extent of

communication, red represents more communication and white represents less communication. The stakeholders placement also show the amount of influence we would like them to have, as shown in Figure 2. Blue colour indicates where we believe the stakeholders to be at the moment:

	Wish to influence	Don't wish to influence
<b>Should influence</b>	LTH Authorities Sales (end user)	Authorities Competitors Customer prospects
	FOCS Testers Graphics Sales (flygp)	Schedulers (end user) Dispatchers (end user) Sales (flygp) Project Group Initiator
<b>Should not influence</b>		Sponsor Competitors

Figure 2: Chart depicting stakeholder analysis

### 3.3.3 Interview

In this section we describe the interview we performed with project stakeholders as a part of the pre-study. You will find the section regarding the Interview as an usability design activity in section 4.1.3 *Interview*.

This interview was conducted in order to obtain knowledge regarding the present activities in the domain, in our case the flight operations domain, and what problems exist. We also had hopes that the interview could provide information relating to possible solutions to issues and conflicts within the domain. The information gathered would serve as a source when eliciting and specifying requirements and by extension when designing the user interface. We followed the framework for a semi-structured interview that is described in section 2.3.1 *Interview*. The results of the interview and the structure of the interview is found, in its raw format, in section 6.4 *Results from interview*.

### Conclusions

The conclusions, and resulting elicitation and specification of requirements, that the interview generated are specified in the requirements specification document in the form of traceability of requirements. The conclusions made from the interview with a usability- and interaction design viewpoint is discussed in section 4.1.3 *Interview*. Due to the sensitive nature of the requirements specification it is not included in this report. Instead we present what information gained during the interview was deemed vital for our project and general areas of interest.

### 3.3.3.1 First part of interview

#### **Daily work and critical tasks**

The most frequently occurring tasks described, concerning the users and the schedule service, are those concerning flight planning, scheduling of tails and pairing. The most crucial of these tasks relate to when flights are delayed, the need for overview when the number of flights is great and shortening the procedure of changing work shift.

#### **Present issues and stressing conditions**

The critical issues in the overall domain concerning this project is the lack of an all encompassing system for flight planning operations. When a task is divided into sub-tasks the operator has to rely on several systems to complete it. This implies that the operator must act as an interface between different systems. These routines are unique from operator to operator, they are a burden and take time and energy to learn and perfect. Due to these circumstances, the experienced operators are supported by the current systems, but during times of stress the systems strain their performance.

#### **Ideas, solutions and viable functionality**

The solutions and the desired functionality that were mentioned are:

- A role based scheduling view where the interface's different components are customized depending on the user type.
- The service should be susceptible to changes made in rules and legislation that relate to air traffic control and especially that the service present these limitations implicitly.
- A test area- or “playground” scheduling view where one can test new routes in order to explore new business ventures. Display full analysis with costs and other useful data.
- Slot time management functionality.
- Attach flight logs and other documents related to flights to flight objects in scheduling view.
- The customer/user has described a desire to perform documentation, work done with flight manuals and similar tasks, digitally instead of with the traditional paper and pen model.

#### **Conflicts**

The following conflicts exist within the customers domain and could affect the project

- The customer is on good terms with Flygprestanda's competitors.
- Can FOCS, and thereby the schedule view service, easily integrate and interact with the customers current systems? Will FOCS be able to extract information from databases belonging to for example flight control authorities?
- Routines and work procedures lack standardization. Which leads to each flight planning operator having a more or less unique approaches to the same tasks.
- Introducing a new software in the organization is costly

This concludes the results obtained from the first part of the interview.

### 3.3.3.2 Second part of interview

The second part of the interview revolved around the sales department's point of view on the project. We discussed how they are marketing FOCS and which services the customers are showing interest in

and which are difficult to show the value of. It was noted that good user experience and usability is important to get the customer to show interest and to use as motivation when changing their work routines. It was also mentioned that the users desires the work flow to be as close to automation as possible, and that this has been described as an important factor when evaluating new systems and services.

### 3.4 Requirements specification

Creating the requirements specification was made as a part of the Schedule View project Pre-study. It functioned as an assurance in-between the project stakeholders that their needs and requirements were represented as functionality in the schedule service.

We made an early distinction in-between the functional- and non-functional (quality) requirements, stating our intentions of only specifying functional requirements in the requirement specification. We would leave the assurance of quality requirement for the activities of the usability engineering phase of the project. In layman's terms we would assure that the correct functions were included in Schedule View by means of requirements engineering and assure the design of the features was correct by implementing usability engineering.

The work with the requirements specification followed the theories described in section 2.5 *Requirements engineering* as well as the practises found beneficial from previous projects both from the authors courses and practices of Flygprestanda. A large set of initial requirements were elicited from our activities. These requirements were first sorted after requirement level:

- Goal level
- Domain level
- Product level
- Design level

The requirements were also sorted by their type as one of the following:

- Managerial requirement
- Functional requirement
- Data requirement

After the sorting of requirements they were specified using different formats e.g. data dictionary, function lists and task descriptions in the requirements specification document.

As the requirements grew we worked with versions of the specification document. This entailed that requirements were not redesigned but discarded in place for of a new requirement with a superior specification. Discarded requirements were kept as well as the reasons for discarding them, this was to maintain traceability in the document. The specified requirements were given unique identifiers as well as a tag noting which activity they had been elicited from, this was also a means to ensure high traceability.

Unfortunately we cannot include the requirements specification in this report due to its sensitive contents. However, we want to highlight its importance in product development. The requirements specification worked as a basis when specifying the WBS.

### **3.5 Work breakdown structure**

The WBS was performed by the project group together with the initiator following the framework described in section 2.6.3 *Work breakdown structure*.

The specified requirements were divided into hierarchic sections with regards to which feature of Schedule View they would be implemented into. These sections would then be translated into actions by our supervisor, as demanded by the WBS, the completion and verification of these actions are described in section 5 *Implementation*.

The final construction of requirements into hierarchic feature sections can be found in section 6.5 *Result from work breakdown structure*.

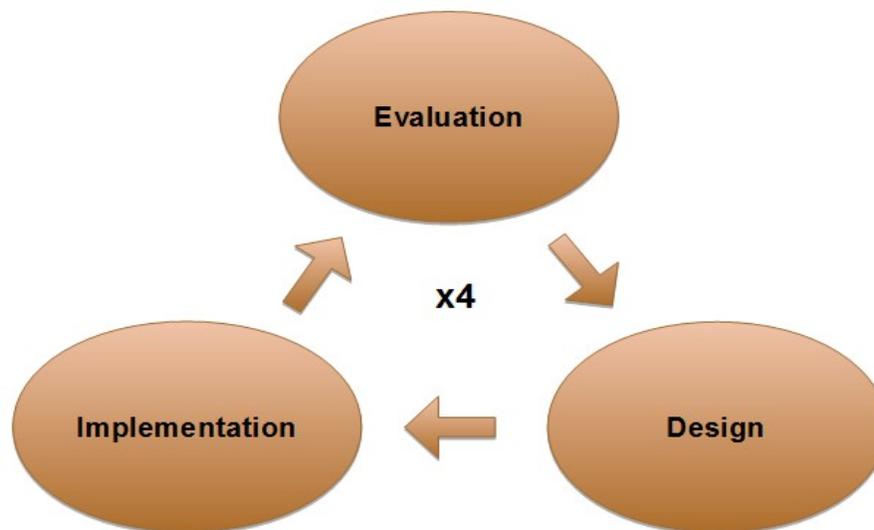
## 4 Usability design & evaluation

The design process in this project consisted of four iterations. This chapter describes our work and results during each of the iterations. The first iteration includes an evaluation of an early version of the Schedule View (developed by Flygprestanda). The following three iterations evaluated changes made by us concerning the user interface and the way users interact with the software. Iteration 3 was conducted internally, meaning no external personal participated in this iteration.

Results from each evaluation phase was used during the design phase in the upcoming iteration.

Each of the iterations consisted of three phases:

- Evaluation – identify issues
- Design – specify solutions and create prototypes
- Implementation – translate prototypes to Java code



*Illustration 4: Phases in each of the iterations*

### Precondition

As previously mentioned we had difficulties finding users to participate in our tests and studies, mainly due to lack of time and heavy work loads of Flygprestanda's customers. Instead we had to improvise, to be able to conduct product development that would keep the user in focus. An early wish (or requirement) from the initiator was that a user should be able to manage the software without hassle as long as he or she possesses knowledge of the domain and has experience of working with computers. Of course this requirement is not optimal since it is not measurable, but it did give us an early understanding who the users are, and what kind of background they have.

Flygprestanda as a company possesses great experience and knowledge about the flight industry and flight operations. Even if we did not have any opportunities to meet any of the customers (users), people inside Flygprestanda have experience working with them. Using different methods we were able to get to know our users and user needs through colleagues working at Flygprestanda.

One of the employees at Flygprestanda had previously worked as a crew scheduler (similar tasks as flight scheduler), which is one of the user groups to the Schedule View. We saw the opportunity, and used her in two evaluation tests in iteration 3 and 4. We realized that even if she had a similar background as our “real users” we can never be certain that findings from tests conducted with her would fully agree with findings found if we could conduct a test with one of Flygprestanda's customers.

## 4.1 Iteration 1

Our aim in the first iteration was to initiate the usability process of the project by evaluating the old version of the Schedule View and begin the design of the new version. Since we were new to the FOCS project we wanted to get a feel for the software and everyone involved (see section 3.3.2 *Stakeholder analysis*). The project group started off with a brainstorm session together with the initiator to learn more about the domain and the purpose of the Schedule View.

In iteration 1, in addition to evaluating the old version of the Schedule View, we conducted an interview, an observation, and additional brainstorm sessions.

### 4.1.1 Brainstorm

Each iteration was initiated with one or several brainstorm sessions. The goal was to solve problems and to come up with and discuss new ideas. This was achieved by discussing results from previous iterations and to plan what needs to be done in the upcoming iteration. Naturally the first part was skipped in iteration 1.

The project group was always present during brainstorm sessions and sometimes other Flygprestanda personnel (e.g the initiator or the graphics team) was invited to participate. The reason being that we wanted to utilize the different expertise of Flygprestanda staff.

We did not want to miss a potential solution or idea because of the session being held in a manner that made participants feel uncomfortable. Therefore brainstorm sessions were held in a relaxed and safe environment to maximize efficiency and outcome. We used a whiteboard where we wrote every topic and idea that was to be addressed.

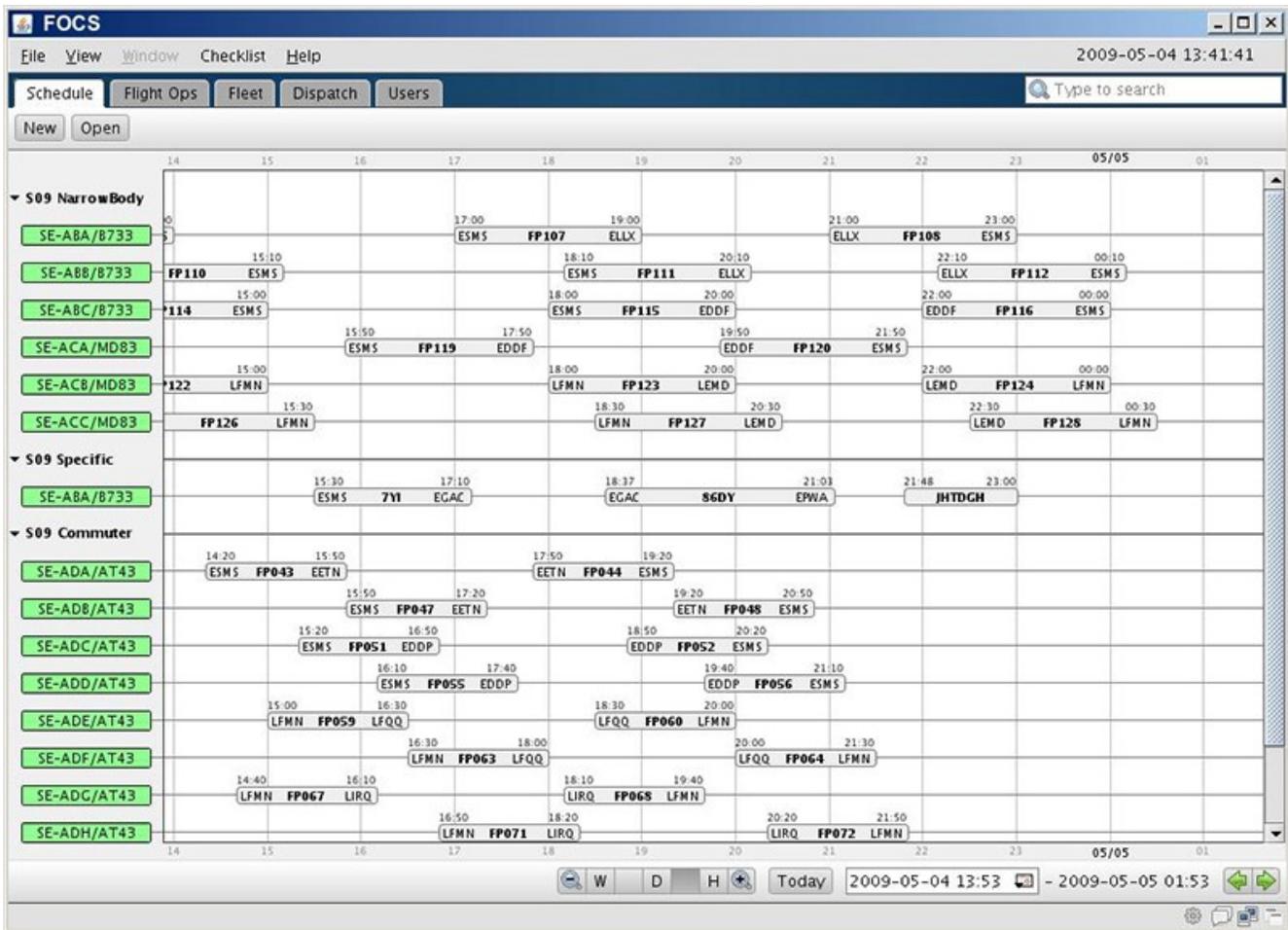
In this stage we had no verification that results originating from a brainstorm session (such as a solution to an existing problem or an entirely new idea) would really work if it was implemented. Therefore we used other tools, for example heuristic evaluations, to verify or validate these results.

We held at least one brainstorm session together with the graphics team in each iteration. These sessions processed current issues and ideas as well as mock-ups and prototypes. A close co-operation with the graphics team was crucial to develop a usable and aesthetically pleasing service.

Each iteration was concluded with a brainstorm session where we decided what to do next, which prototypes and ideas to implement. This tool was used throughout all of the iterations, and we will not include it in each iteration section below since the utilisation of the tool did not vary between iterations.

### 4.1.2 Heuristics evaluation

Each iteration included a heuristics evaluation. In the initial iteration we evaluated the old Schedule View, and in the following three we evaluated results from the previous iteration. Our goal with this activity was to highlight the weaknesses of the interface and feature interaction design.



Schedule View as of 2009-05-05

We used all the principles described in section 2.4.1 *Heuristic evaluating*, and focused on the principles that applied to visualising information, amount of feedback and the consistency of the interface. We chose to work with this focus so that issues and solutions that were found during this evaluation would relate to our purposes. Issues that were not closely related to our purposes, but still vital to ensure usability, were forwarded to Flygprestanda. This ensured that all issues would be addressed and it also helped us with limiting our workload.

The last heuristic concerning documentation and help is not considered in this project. The service at this state does not include a manual and it is the goal to design a service that need no instructions to operate. Furthermore, designing a manual is a project in itself and would be too time-consuming to include in this thesis.

The evaluation was carried out by the project group together with the initiator, to ensure finding the majority of usability issues. As with the brainstorming sessions, we used this evaluation tool in all of the iterations. We will not include it in the following iteration sections below since the utilisation of the tool did not vary between iterations.

### 4.1.3 Interview

In this section we describe the outcome of the stakeholder interview from an usability and interaction viewpoint, and how it impacted the design of Schedule View.

The interview had answers to questions that needed to be validated, partly since the information was second hand and not directly from the user but also since solutions provided by the users might not be optimal or need re-designing.

We did this by first evaluating their suggestions heuristically and comparing them to data from our theoretical framework. Those suggestions that proved interesting, we inserted into our prototypes and/or the requirements specification.

The interview had a broader scope than just the Schedule View, namely the entirety FOCS system. This resulted in needed to further analyse if and how the answers and statements concerned the Schedule View. The results of the interview can be found in section 6.4 Results from interview.

The following areas were introduced into an early design or prototype:

- The importance of cross-component functionality was emphasized a number of times and resulted in adding information and notifications from other services of FOCS into prototypes of schedule view components
- The wish to implement a role-based design depending on the type of user was added as a requirement (but later stricken from the specification and re-worked as the edit/information mode concept)
- The users have explained that they desire the scheduling of flights, as well as assigning pilot and crew, finding routes and creating documents, to be automated with “a click of a button”. Neither the users' upper management, stakeholders at Flygprestanda or the project group regard this as a preferable solution, due manly to legislative and security reasons regarding the domain. We choose to interpret the wish for automation to mean the design should be as straight forward as possible and managing features to solve tasks should require as few steps/actions as possible. This was considered both in prototypes and in the requirement document.
- Lastly, the stakeholders encouraged the project group to think of solutions and inspiration from outside the domain. The flight operations field is somewhat dated and in some instances still use solutions dating from the late 1960s. This notion will serve as guidance when brainstorming for areas of interest regarding field observation and other research.

### 4.1.4 Observation *Husarrest*

#### **Introduction**

Due to the fact that observing the end user of the schedule service proved impossible, we determined that we needed to do an observation of work carried out in a similar fashion as to that of the flight operations personnel, with similar tasks aided by software with a similar interface.

We argued that in order to design new functionality, and improve the existing, we needed to understand the manner of conduct associated with solving problems with the aid of a schedule interface. An interface with a schedule design that have the same issues of scaling when handling multiple objects and the visualization of information regarding these objects.

Furthermore, we wanted to explore the different functions of a interface that was designed to solve a

different task but in a similar fashion. We reasoned that there was a high chance that the interface would include functions that could be implemented in our schedule service or inspire us to design functions that we otherwise would not have thought of.

After a brainstorm session we concluded that the interface of software used when digitally creating music could possibly have similarities in its design to that of our schedule service, and after studying examples of different software designed for DJ's we felt that this was the right interface to compare with.

For the observation we contacted the local artists of *Husarrest* and they agreed to us conducting an observation on them while working. Husarrest, a DJ trio, consists of Oscar, Truls and David and they perform in local, national and international venues. When Husarrest compose their own material they use the commercial off-the-shelf software *Logic Pro*.

### **Logic Pro**

Logic Pro is a digital audio workstation for the Mac OS X platform and at this time the most current version is Logic Pro 9. The software was created by the developer *Emagic*, that was bought by Apple in 2002, and was originally developed for the Atari platform. With the rise of the personal computer the software was given more functionality, especially regarding audio processing, and became available on the Mac OS and Windows platforms, which gained popularity for the software. Logic Pro enables the user to create music using features such as: software instruments, synthesizers, audio effects, that include distortion, equalization filters and delays, and recorded samples among others.

### **Method**

Prior to conducting the observation we compiled an aid scheme consisting of areas of interest regarding the interface and the users operations, and general guidelines described in section 2.3.2 *Observation*. In section 10.1 *Framework* used when conducting observation, you will find the aid scheme we designed and used during the observation.

The notes that were taken during the observation can be found in raw form in section 6.6 *Results from observation*.

During our observation we aimed at being passive observers and only engage in questioning the study group when an action or event needed clarification. We felt that this approach was more scientific in combination with our lack of knowledge of the environment, tasks and tools which we felt would only infuriate the experienced study group.

### **Similarities between Schedule View and Logic Pro**

The following section aims to explain the main reasons that we came to choose the Logic Pro audio workstation as the platform for our user observation.

Logic Pro dedicates the interfaces' left hand side to a column or panel for audio channels that run parallel to one another and their respective samples are depicted as segments in a large main work area called *Arrange*. The right hand panel was, during our observation, configured to display directories of recorded and imported samples.

The bottom part of the interface consists of an area that allows the user to edit an audio sample, add effects and other options divided into different tabs. There are also buttons in the bottom part to play, pause, stop, etc all or selected tracks and a window displaying information regarding a selected audio sample, such as length, sample frequency, beats per minute etc.

Furthermore, there are options that prompt different pop-up windows that allow the user to modify

tracks, record sounds, interface with attached hardware and instruments. These were not included in the observation and will not be addressed further.

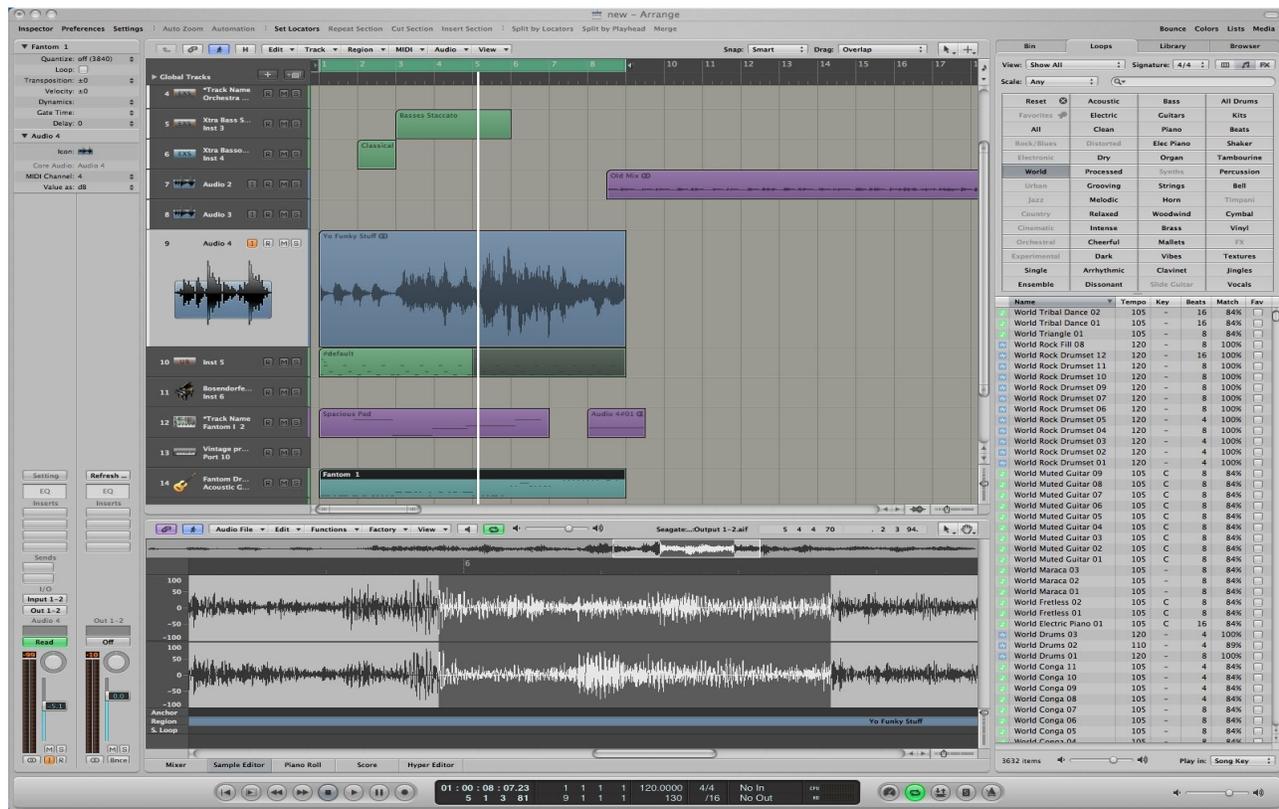


Illustration 5: Logic Pro interface, copy right Apple Inc

The two largest similarities between Logic Pro and FOCS Schedule View is Logic Pro's representation of audio channels and audio samples in the arrange compared with Schedule View's representation of aircraft and their flights in the schedule area. The observations made concerning these areas was the focus for drawing vital conclusion that would have significance to designing the interaction of Schedule View.

## Conclusions

With visual and auditive feedback Logic Pro helps the user organize a large number of samples in the *Arrange*. Additional information is displayed by highlighting current areas of interest and when searching for and importing material, playback of the sample directory.

Logic Pro uses common operations like copy, paste and drag and drop for managing samples. A few of these will enhance the user experience in the Schedule View, while some of them are less relevant.

While displaying wanted information is of great importance, one should be careful not too display too much at the same time as this can confuse the user. An idea to be analysed is to map the level of detail depending on the level of zoom the interface is displaying.

Since we observed Husarrests' work flow as very dynamic with a lot of trial and error, the software must be forgiving. Undo actions and cancel actions that allow the user to revert to a previous state, in combination with a general design that minimizes user mistakes will help in obtaining a forgiving

interface.

Relevant and frequently used functionality should always be easily accessed, for example using appropriate menus. Another way to obtain this is by making the interface customizable, however we do not consider it an appropriate approach in the Schedule View. The users of the Schedule View work in shifts at terminals where individual customization would cause confusion.

As previously concluded, zoom functions is frequently used and is a major part of the work process, this is one function that is fundamental when operating the Schedule View. The zoom function is used in a similar fashion in Logic Pro which makes our observation regarding this valuable in designing the Schedule View. We were given examples of inherited flaws of Logic Pro's zoom function equally or more valuable to that of our observations.

Apart from the debriefing of Logic Pro we also conducted a unstructured evaluation of the old FOCS interface, and noted their comments and ideas on improvements which are also found in section 6.6 *Result from observation*.

## 4.2 Iteration 2

Findings from the evaluation activities in the first iteration indicated that features in the service were hard to use and that the service sometimes seemed inconsistent. This led to re-designing areas and features of the service. Low level mock-ups were produced that was refined with the help of Flygprestanda's graphics department.

Together with the project initiator we prioritised which features should be implemented during the first iteration, and then we entered the implementation phase.

As we initialized our second iteration, we concluded that we needed to assess the accuracy and efficiency of the first implementations we made from the prototypes of the first iteration. The tool we chose for this was the *assessment test*.

### 4.2.1 Assessment test – Test person

We followed the framework and guidelines described in section 2.4.3 *Usability testing*, as well as using our experience drawn from completing numerous usability and interaction evaluation courses at the EAT department of LTH and Lund University Cognitive Science department combined with publications and literature on the subject [16][7][8].

This choice was made using the DECIDE-framework also described in section 2.4.2 *DECIDE-framework*.

The goals of the assessment test were the following:

- Provide insight in efficiency and accuracy of current design implementations as well as their usability
- Provide a standard for future usability testing and evaluation
- Highlight issues and opportunities regarding interaction

The questions used to obtain our goals were fashioned as scenarios. These test scenarios were constructed as a part of a test plan designed to function as the structure of the assessment test.

Other than keeping our test subject anonymous, and assuring the test subject (TS) that the results of the

test would in no way affect TS's position at Flygprestanda, we found no ethical issues to address.

Practical issues were also addressed in the test plan.

### **Test plan**

The general format of the test was as follows:

1. *Perform a pilot test*

In order to find issues and flaws in the scenarios or tasks, we argued that the best solution to this was to perform a pilot test. The conclusions drawn from this type of test should focus on the test itself, its structure, how well formed and unambiguous tasks are described as well as the performance of the test monitors (TM's).

2. *Introduction and walk through*

Since TS has no knowledge regarding either FOCS or Schedule View we would conduct a brief introduction to the services (both Schedule View and FOCS as a whole) and what tasks it is meant to help the user solve. We would also use this opportunity for other logistical aspects concerning the test.

3. *Conducting the test scenarios*

With the use of a handful of scenarios consisting of different tasks we would instruct what problems we wanted TS to solve. We would, as TM's, take the passive test monitoring role and only intervene when necessary.

4. *Debriefing*

After conducting the test we would debrief TS using a semi-structured interview as script to complement the test results with qualitative data and explore reasons to "good and bad performance" during the assessment test.

The test plan and the guidelines for the semi-structured interview used during the debriefing can be found in section *10.2 Assessment test*.

### **Pilot test**

During the pilot test we found some minor flaws in both the test environment, regarding some missing objects, and the task descriptions had some ambiguous task descriptions. These flaws were fixed during the pilot test and would not be an issue during the actual test.

A solution regarding browsing or cycling/scrolling through a selected aircraft's next/previous flight with "<>"-buttons associated with the aircraft label. The schedule area will centre on/jump to the next/previous flight of the selected aircraft.

We found it hard to get accurate time performance readings from the sub-tasks. The pilot tester kept pausing and describing his intentions, which we had instructed him to. This led to us having to arbitrarily pause the timing when the test pilot was occupied with describing intent and problems, and then resuming timing when the test pilot went back to completing the task.

This led us to conclude that we should have a short discussion at the end of each task instead of during the completion of the sub-tasks. This also allowed us to get a more distinct "Last impression".

Finally the pilot testing gave us opportunity to refresh our skills in test conduct and gave an estimate on how much time it would take to perform the test, which was longer than expected.

The results gathered from the assessment test is found in section *6.7 Results from assessment test*.

## 4.3 Iteration 3

At this stage of the project a large number of design proposals, mock-ups, prototypes and implemented solutions had been accumulated from the pre-study, the first and the second development iteration.

We concluded that we needed an extensive internal evaluation phase during the third iteration where the accumulated artefacts of our previous work would be reviewed. We would use this to delimit the project by dividing the artefacts into action groups to allow us to focus on critical issues. The feature review was made in parallel with this iteration's heuristic evaluation and the process is described in the following section.

### 4.3.1 Feature review

We started of this activity with compiling all the accumulated ideas, concepts, mock-ups and implemented features from the previous iterations and project phases. The status of these artefacts were evaluated using the results gained from the previously described activities of this project, in combination with this iteration's round of heuristic evaluation.

Depending on the given status, an action framework was specified for each artefact. The last step of the review was the grouping of similar artefacts depending on their specified action. The following five action groups were defined:

1. *Iteration 4*

This group contains the features that have all been implemented and released, some have also undergone previous evaluation. They are all focus features in the planned evaluation activities in iteration 4 of the project.

2. *Concepts*

Most of the features in the *Concepts* group are in a conceptual state. Their status is that they, for the most part, have not been implemented and are still being researched or the evaluation of the initial implementation have proven them to be suboptimal and in need of redesign.

3. *Low priority*

The features in the *Low priority* group entail that they have functionality which stakeholders have expressed as desired but not vital. These features have been released with their basic implementation and their design and performance have proven adequate during evaluation. They will remain outside the project focus unless future activities result in the need to bring them in.

4. *Outside project scope*

In order to delimit the project, features with the status still at an early concept or with low level mock-up were assigned this group. Flygprestanda will follow up with work on these features in future versions of FOCS depending on the results of research and evaluation activities concerning them.

5. *Cancelled or replaced*

Features in this group have either been replaced by another or cancelled due to poor performance in previous activities and evaluations.

The result of the feature review is found in section *10.5 Internal review*.

## 4.4 Iteration 4

This iteration presents the final activities of the project, made to ensure the usability of our design. Iteration 4 contains a validation test, an internally held heuristic evaluation and finally a survey held at one of Flygprestanda's customers.

The fourth and last iteration is where our project ends. Resulting mock-ups, prototypes, feature and interface design originating from the fourth iteration has not been completely implemented, however it has been delivered to Flygprestanda for them to use in future iterations and implementations.

### 4.4.1 Validation test

Using the DECIDE framework we concluded that a validation test on the schedule view's current interface design would be suitable.

As with the assessment test, we used the theoretical frame work in combination with our experience with usability testing and the findings and quantitative data collected from the assessment test.

The goals of the validation test were the following:

- Assert validity of new feature and interface design
- Assure equal or superior performance of interaction regarding benchmarks set during the assessment test
- Highlight issues and opportunities regarding interaction

We fashioned a test plan that included practical and ethical issues as well as the format of the test and the scenarios for the TS to perform.

The validation test followed the same structure as with the assessment test with introduction and walk-through followed by the TS performing test scenarios, and ending with debriefing of TS.

The validation test plan and TM guidelines are found in section *10.3 Validation test*. The results of the validation test are found in section *6.8 Results from validation test*.

### 4.4.2 Questionnaire Survey

The questionnaire constructed in this project follow the QUIS format.[17] The Questionnaire for User Interaction Satisfaction is one of the most widely used formats for evaluating interfaces[7]. We complement this format with a few open ended questions that are sequenced in the beginning of the questionnaire when the participants attention and interest is at its peak. The open ended questions are kept short, to the point and clearly worded to minimize misunderstanding. The QUIS format is presented in section *10.4 QUIS questionnaire* with modifications made were we also mark the questions modified to better suit our survey.

The QUIS questionnaire allows the user to indicate their feelings toward the system in a formal and structured way that is void of all technical terms. The resulting analysis of the survey supplies Flygprestanda with empiric qualitative data concerning their product. With this they can prioritize areas of design that the user finds lacking, or if need be, they can conduct further usability evaluation or survey the users in more detail regarding problem areas. The format of the questionnaire will furthermore function as a standard for Flygprestanda's future interaction and usability evaluations, both concerning the FOCS service and other ventures.

The results of the questionnaire are yet to be submitted by Flygprestanda's customer. These results, and their analysis, will not be included in this report and will be left as base for the future development of the interaction and usability of FOCS.

## 5 Implementation

### 5.1 Code implementation

The implementation was done using Java programming language using a development tool called *Eclipse*. Integrated with Eclipse we used the subversion tool *SVN* and a software called *Maven* handling class dependencies. This enabled us to work on feature prototypes for Schedule View and upon completion upload our implementation into the FOCS source directories.

Implementation was specified by action tickets assigned to us through Flygprestanda's managerial tool *Scope*. The ticket consists of a description of what implementation is demanded, where the demand originates from, in our case the WBS, and the deadline for completion.

Flygprestanda use a predecessor to the agile development format RAD with three week release iterations which we tried to synch our implementations with to the best of our programming abilities.

The source code, or any other code, implemented as a part of this thesis is property of Flygprestanda AB and will not be included into this report.

### 5.2 Verification

Aiding us in the work with verification of our implementation was Flygprestanda's software testers. Upon completing a ticket and checking it as solved, they would perform their test suits and report any issues, bugs or unwanted behaviour on our codes part.

The testers submitted bug reports to Scope, and they were addressed during Flygprestanda software development departments weekly meeting where the rest of the software team could comment or suggest a solution.

## 6 Results

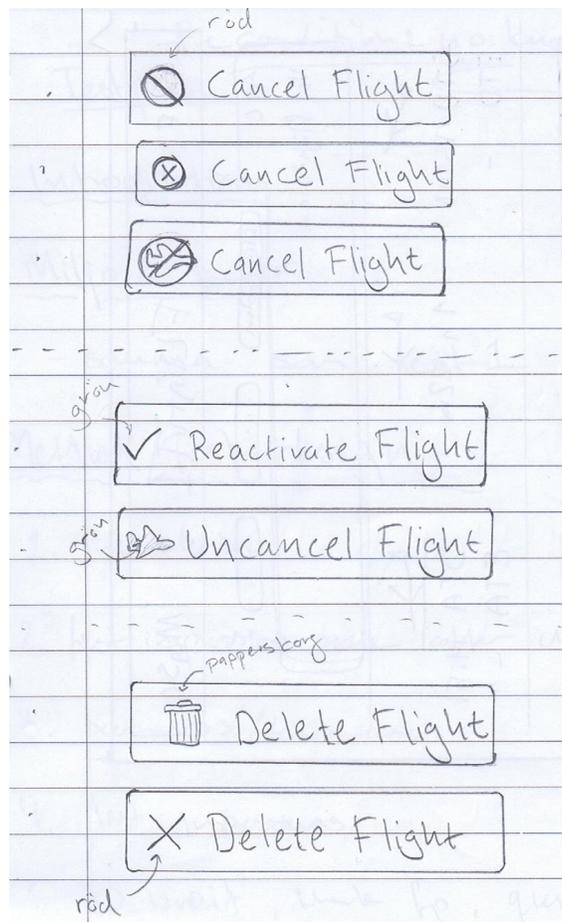
This chapter will present results found throughout the entire project. We will first present results that are directly linked to the Schedule View such as prototypes, mock-ups, ideas and concepts. Thereafter we will continue to describe detailed results from each activity in the project.

### 6.1 Results from iterative work

Results from the four iterations will be presented in this section. It contains ideas, concepts, mock-ups, prototypes and implemented functionality. The mock-ups and prototypes below have either been implemented or discarded. In some cases they have been put on hold due to the need of further investigation before taking an action.

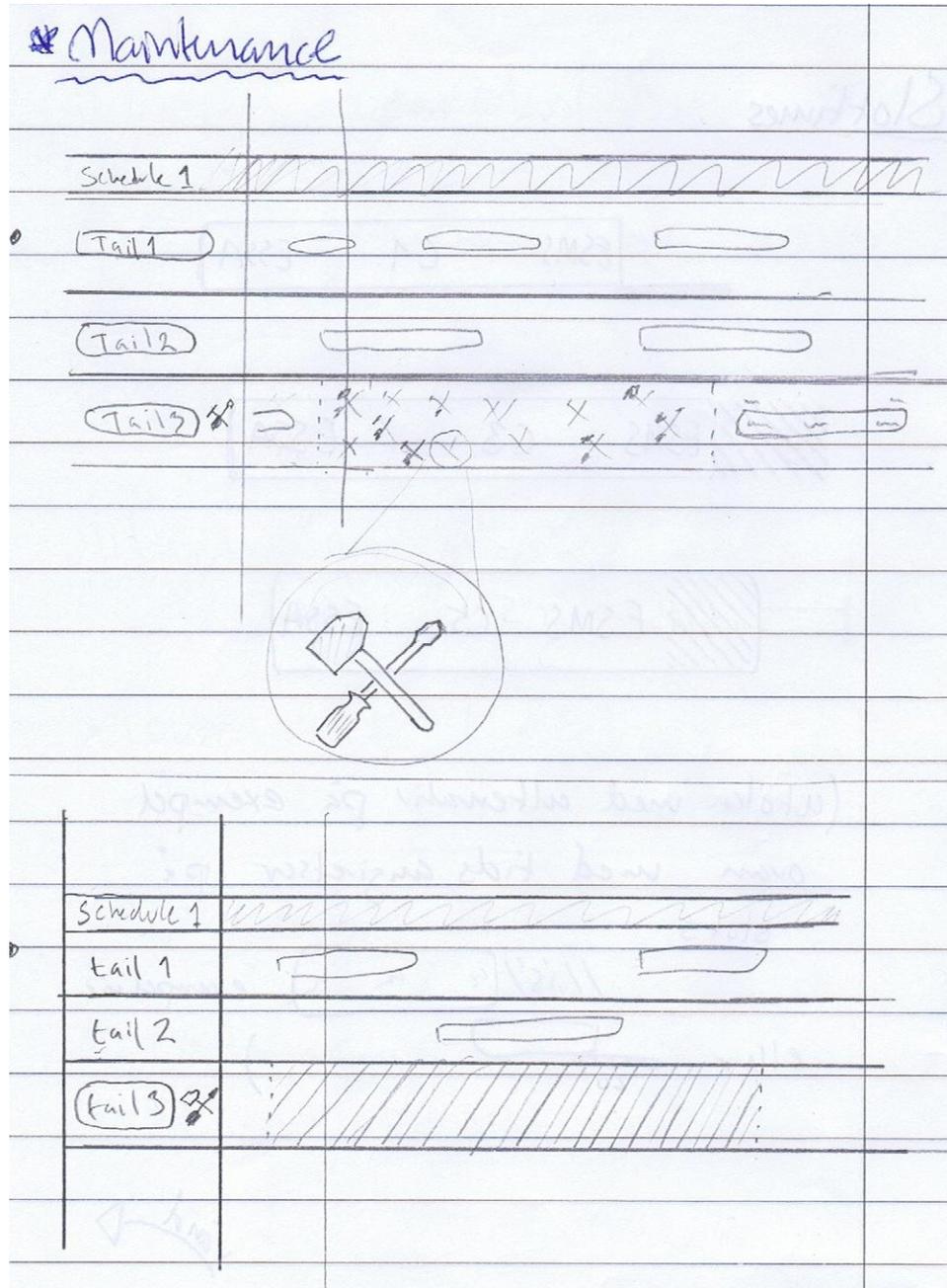
#### 6.1.1 Mock-ups

- *Illustration 6* shows a mock-up of feature button design and iconography.



*Illustration 6: Mock-up of button design.*

- *Illustration 7* shows a mock-up over the maintenance concept, where an aircraft can be temporarily unavailable due to maintenance or lease. The aircraft will still be visible in the schedule but no flights can be scheduled during the time the aircraft is set to maintenance or lease.



*Illustration 7: Mock-up of maintenance concept*

- Illustration 8 shows a mock-up of the panel showed when a flight is clicked. At the top of the panel the flight object is displayed containing the flight name and departure/arrival airports (given in ICAO-codes). Underneath lays a few tabs containing relevant information for the flight. At the bottom the times (departure time, in-air time, arrival time) for the flight are presented followed by option buttons.

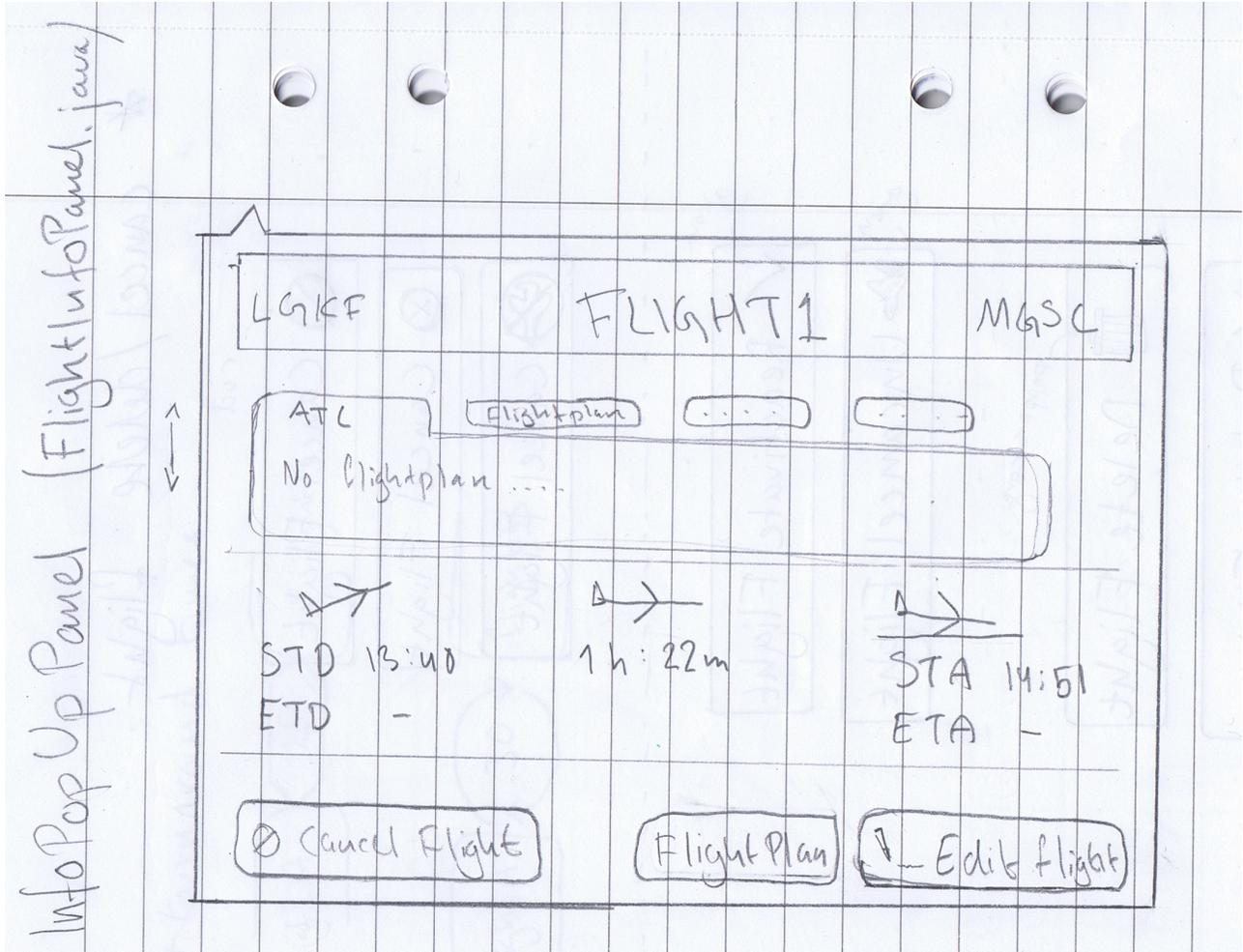
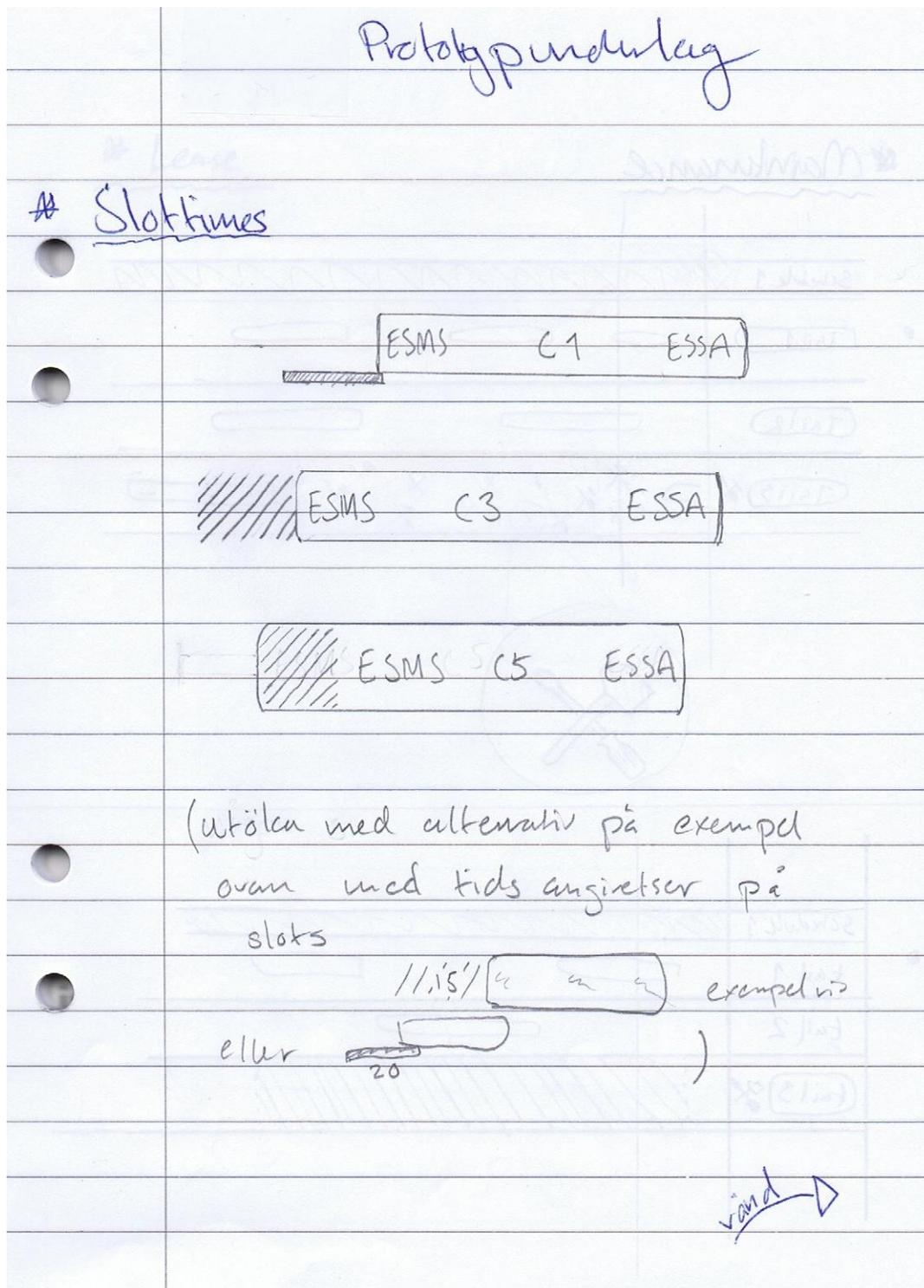


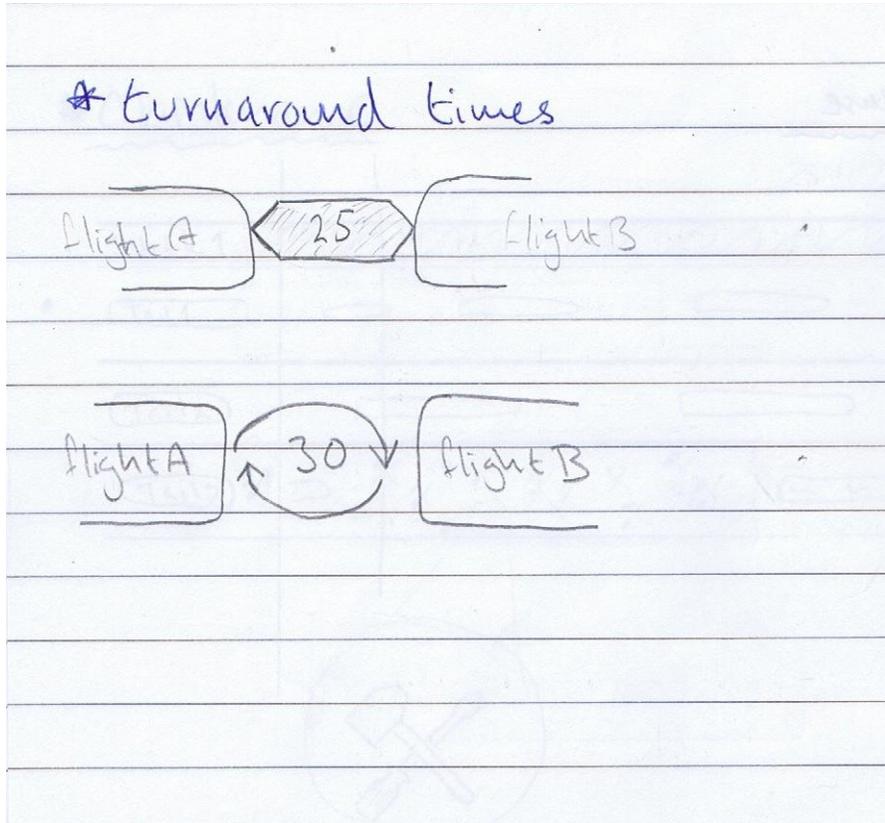
Illustration 8: Mock-up of flight info panel

- *Illustration 9* shows a mock-up of the slot time concept. Slot time is the time window assigned to a flight when the aircraft is allowed to depart.



*Illustration 9: Mock-up of slot time concept*

- *Illustration 10* shows a mock-up of the turnaround times concept. Turnaround time is the time an aircraft must remain parked at the gate before departing again.



*Illustration 10: Mock-up of the turnaround time concept*

### 6.1.2 Prototypes

- *Illustration 11* shows a prototype of the drop down panel shown when a flight is clicked in edit mode.

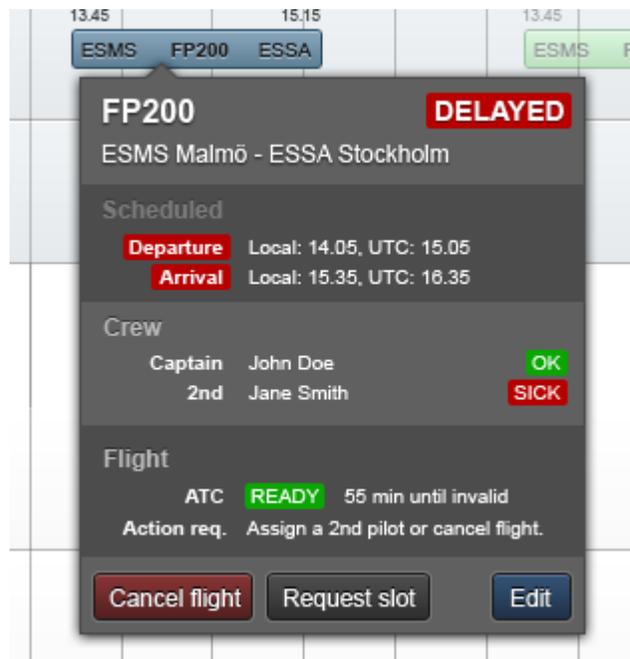
*Illustration 11: Flight edit panel*

- *Illustration 12* shows a prototype of the presentation when a flight is delayed. The concept slot times is also displayed here (the blue line underneath the flight).



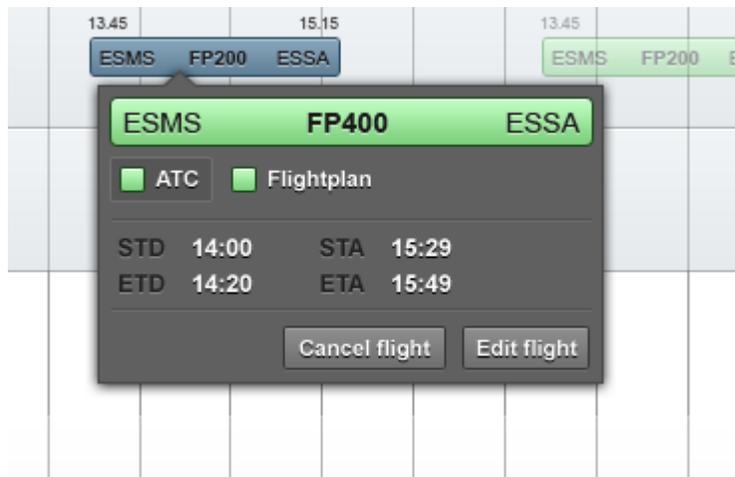
*Illustration 12: Delayed flight and slot times.*

- *Illustration 13* shows an early prototype of the drop down panel shown when a flight is clicked in info mode.



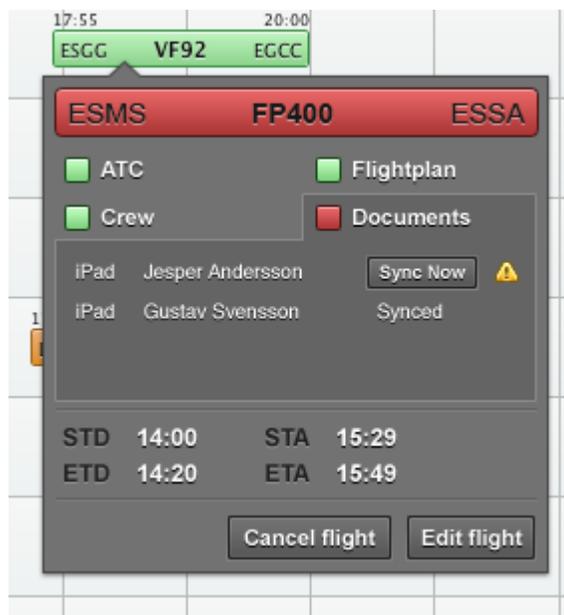
*Illustration 13: Early prototype of flight info panel*

- *Illustration 14* shows a prototype of the panel shown when a flight is clicked in info mode (this is the prototype we used when we implemented this functionality).



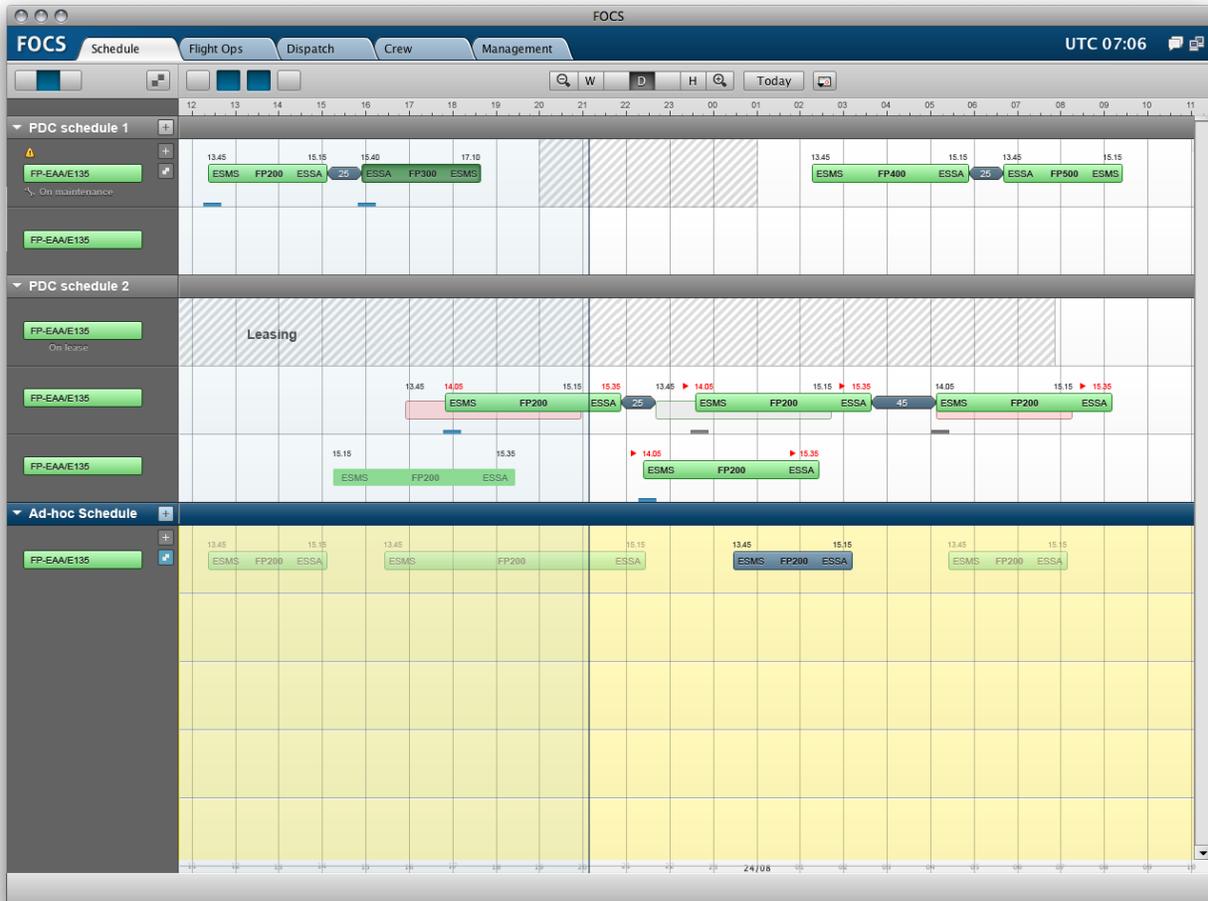
*Illustration 14: Flight info panel*

- *Illustration 15* shows a prototype of the drop down panel shown when a flight is clicked in info mode.



*Illustration 15: Flight info panel*

- *Illustration 16* shows a prototype of the playground concept if it was implemented in the Schedule View. The idea with the playground concept was to let the user plan and try new routes without disrupting the live schedule. The playground area in question is the bright yellow section of the schedule area.



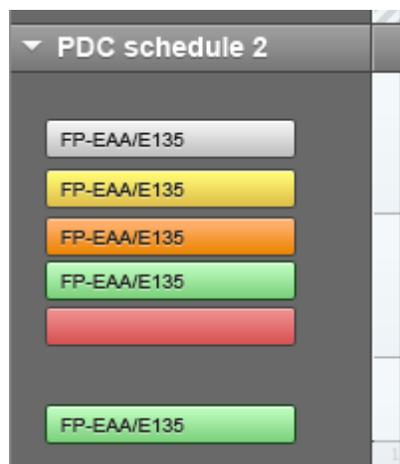
*Illustration 16: Playground concept*

- *Illustration 17* shows a prototype of the turnaround times concept, the slot times concept and the maintenance/lease concept.



*Illustration 17: Slot times, turnaround times and maintenance/lease*

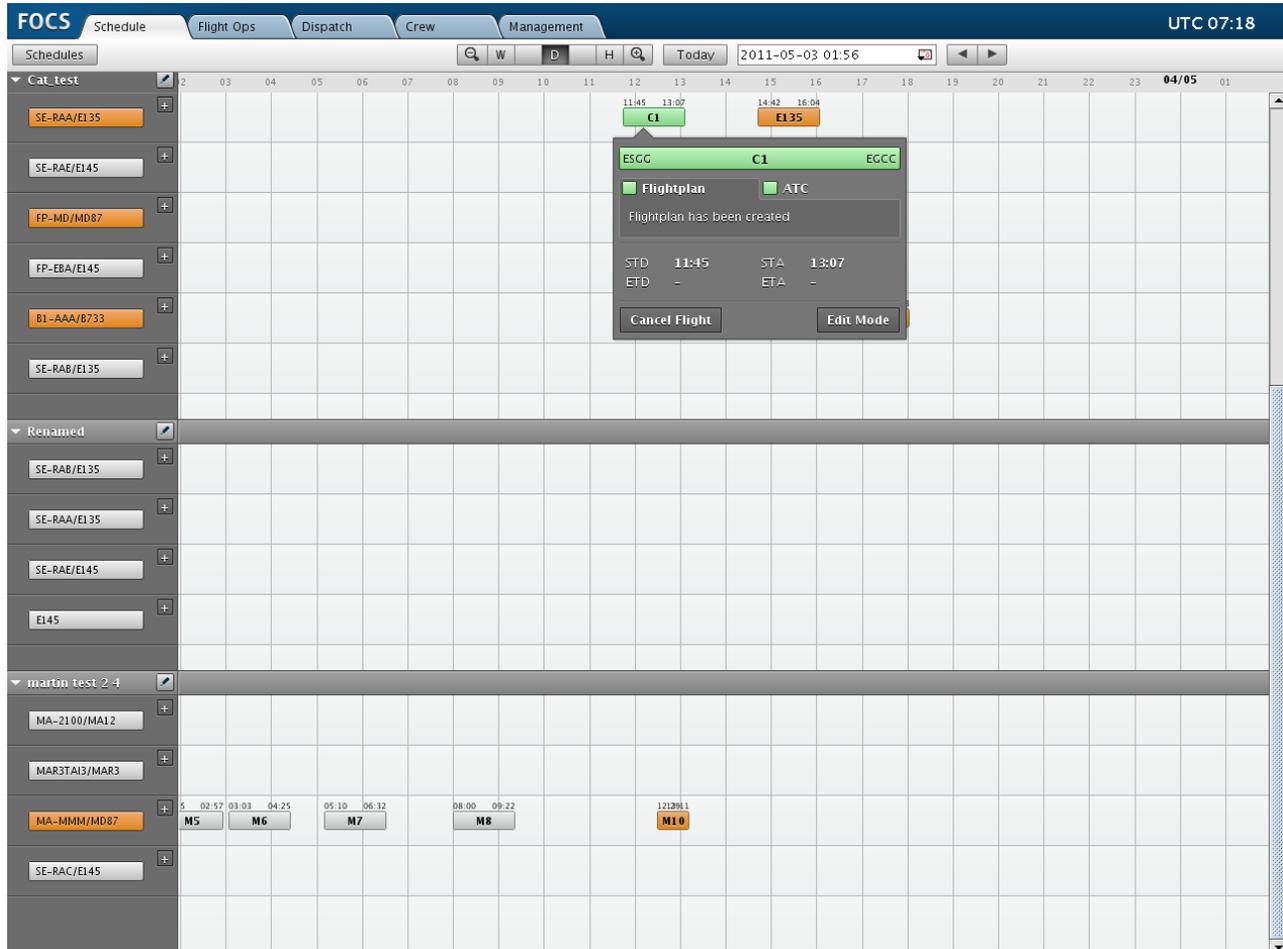
- *Illustration 18* shows a prototype of different colours for the aircraft object.



*Illustration 18: Aircraft color prototype*

### 6.1.3 Implementations

- *Illustration 19* shows the implemented version of the Schedule View as of iteration four.



*Illustration 19: Schedule View implemented*

### 6.1.4 Concept / ideas

The following are results from the development iterations that went no further than a conceptual design. They are future ventures for Flygprestanda and will not be discussed due to their sensitive nature to the business of Flygprestanda.

- Hub/airport view
- Feature guides/wizards
- Role based schedule area
- Facilitate trip planning
- Crew handling

## 6.2 Results from SWOT analysis

A complete result from the SWOT analysis is not presented in this report since it contains company sensitive material. Below is an action list for high priority conclusions of the SWOT analysis:

ID#	Conclusions	Actions	Handled by
A8	Graphics are already standardized, focus on interaction design and functionality.	Use graphics department as a resource.	Project group
A11	Competitors services lack in usability and UX.	Use as sales point when marketing service.	Sales
D1	Delimitations in the project is important to prevent the workload exceeding the project deadlines.	Write a project plan. Define milestones. Monitor the work at hand continuously, compare with project plan. Rewrite project plan if necessary	Project group Initiator
D2	Not exceeding the time frame of the project should in some sense be a priority compared to complete or extra functionality. This prioritizing is done to ensure more initial market shares.	Plan action time table Implement functionality in order of importance. Follow delimitations and time table in project plan.	Project group Initiator Sales
D8	If it is discovered that the graphical design needs changes, it should be known that this might have a big impact on the time and budget.	Use graphics department as resource when evaluating graphical design, and if redesign of graphics are needed.	Project group Initiator Graphics
D11	A well specified and user centred interaction design will shorten the implementation and verification phases.	Take time working with interaction design. Do not make assumptions. Verify quality of our interaction design.	Project group
D12	The fact that we are late to market should not influence the quality of the interaction design since this is important.	Assign appropriate amount of resources to the design phase. Do not rush the interaction design.	Project group Initiator
V4	Due to lack of experience, each development phase will require a start-up where the project group familiarizes with the task at hand.	Consider this conclusion when planning. Initiator should support the project group with necessary resources	Project group Initiator
V6	The size of the project and the limited resources/time will affect the level of completeness	Follow project plan and stay within delimitations.	Project group Initiator
T1	New development without the end user in mind can result in undesired	Implementation should originate from requirements.	Project group

	functionality.		
T3	We need to keep in mind that we have end users with different experience and background.	Verify the design with different user types.	Project group
T7	Tempting to implement a stand alone service with no re-usability.	Verify which components are re-usable Use implementation methods that enable re-usability	Project group

Figure 3: High priority conclusions of the SWOT analysis

**Action list for priority conclusions:**

ID#	Conclusions	Actions	Handled by
A12	Additional resources in interaction design is a valuable sales point	Inform customer prospects via the website.	Sales
A13	Work done regarding interaction design in the scheduling view can be reused as standards when implementing other views.	Document verified interaction design for future use.	Project group
D7	Since much of the graphical design is already standardized, the project group should not focus in these areas.	Leave graphical design to the graphics.	Project Group Graphics
V1	Since we do not really know the end user the specification and design phase could exceed it's time frame.	Do not rush the design phase.	Project group
V3	Important to know who possesses the knowledge regarding the end users/customers.		Management
V9	Important to know who possesses the knowledge regarding the code base of FOCS.		Management
T8	Implementation should focus on prioritized services and functionality.	Follow WSB.	Project group
T12	The new development should build on the existing code base, and improve it where necessary.	Improve old code base.	Project group
T16	Staff with knowledge regarding competitors can be included in schedule view project.		Management

Figure 4: Action list for priority conclusions

## 6.3 Results from stakeholder analysis

### 1. Project group

To deliver this service, our first step was to elicit and specify the requirements of the project stakeholders. If requirements are specified or interpreted incorrectly it can lead to unwanted or incorrect functionality in the system. This can in turn lead to prolonged validation and verification, that will require further reiterations of design and implementation. Delays in the project can result in insufficient implemented functionality or a deadline that is pushed back. These setbacks can be very costly.

The project group will work with a combination of requirement engineering and usability engineering to obtain a successful service design. The implementation of the service will follow Flygprestandas 3 week iteration process standard and will be validated and verified after these iterations.

### 2. Initiator

The initiator has similar goals as the project group. He has ordered the development of a service intended for a specified task, and he expects to receive one accordingly. The initiator intends to support the project group during the project by supplying resources and by maintaining constant communication with, and in-between, the project group and other stakeholders. The initiator will be of great use in the project since he possess domain knowledge regarding the aviation industry.

The risks linked to the project that concern the initiator are those connected to the state of the market the service is intended for. If expectations and research regarding the market are inaccurate, this could impact the sales of FOCS which the service is a part of.

### 3. Sponsor

The sponsor wants the project to be profitable, it should result in a product that sells as expected or better. Another of the sponsor's goals in the project is to see that the experience and know-how, regarding software development projects, within the company increases.

The sponsor sees the same risks as the initiator but with a wider company scope.

### 4. Dispatchers/Schedulers (end users)

The developed service must be an improvement to the system already in use by the operator. The work done by the operator shall, with the help of the new service, be less cumbersome and take less time than with the previous service used for the same tasks. Apart from the service being more efficient, the operator wants the user experience to be pleasant. The look and feel of the service is important to the operator, it must invite him to use the service regularly and not look to other ways to accomplish the tasks at hand.

The different types of operators, that will use the developed service, can supply the project with vital information regarding their day-to-day work, the problems they face and what critical task the service must enable them to complete. They will also enable the project group to verify and validate that their design of the service is good by participating in different interaction testing.

The operators work in a fast paced and sometimes very stressful environment, and the operators need to focus on their tasks. A new service can be a risk when if it inadvertently steals attention or needs too much attention when handling it. Introducing a new software service can be a burden to the operators because they need to learn to conduct their operations in a new fashion.

### *5. Sales (end user)*

The sales department, on the customer side, wishes to be able to use the product in an exploratory fashion. They will want to use the product to investigate new business ventures for their company by comparing the cost of a new route and customer demands.

The sales department can participate in the same fashion as the dispatchers/schedulers/end-users and see the same risks.

### *6. FOCS*

The FOCS team wants the schedule view to become an integrated part of the application, this is a step towards FOCS becoming a complete system. They will support the project group with their knowledge and experience regarding FOCS functionality whenever it is needed.

The FOCS team have concerns with the development of the scheduling view. There is a risk that the implementation will differ from FOCS system standards and therefore will be hard to integrate in the system. They also fear that their support to the project group can add to much weight to their workload.

### *7. Sales department at Flygprestanda*

The sales department wants the project to result in a service that is interesting for the product they will sell to the intended customer, and that this will maintain or improve the relations between the customer and Flygprestanda. The sales department would also like that the service will interest other customer prospects with similar needs.

The largest risk for the sales department is that the service developed will be less competitive than intended. Another concern is that customer prospects will feel that the service is not optimal for their needs.

The Flygprestanda sales department are very knowledgeable regarding what functionality the customer has expressed is required for their different tasks. This information will be very valuable during the project.

### *8. Testers*

The new service that is going to be implemented as an addition to FOCS will be tested to ensure that the code is free from errors and bugs. The goal is to add functionality without impairing the overall performance of FOCSs or the other services it encompasses.

The testers will run tests, on a functional level, during the development and contribute in the form of finding faults and bugs in the developed code.

Should the implementation of the service contain large amount of bad code, testing the service could be costly in resources and time.

### *9. Graphics*

The main goal for the graphics team is so the system follows the graphical profile of the company, users should recognize the style of Flygprestanda's software. They are also interested in optimizing the user experience.

The graphics team must control the design and implementation regarding graphical profile. If the service design differs from other Flygprestanda systems there is a risk of losing product familiarity in the service.

## *10. LTH*

The LTH supervisor sees this service development as an enabling part of the master thesis the project group is writing parallel to the project. He will help the project group with issues of an academic nature, which will help the project group focus on the design and implementation of the service.

The supervisor sees the risk that the project group will focus entirely on the service and not complete the other parts of the thesis that are required.

## *11. Authorities*

The developed service must operate within the bounds of the rules and regulations that are enforced by the authorities. Therefore it is important for them to communicate what is required of the service for it to be approved, and used in flight scheduling operations.

The influence and resources they can provide the project with will be in the likes of documentation regarding regulations in the fields the system will operate.

## *12. Competitors*

Flygprestanda's competitors would like to see what features and functionality our service will include. They will then incorporate the most successful of these in their new releases. They wish for the implementation to be sub-par to their current services that are on the market, but inspire future solutions.

Inadvertently, the competitors current systems will inspire the project with which functionality is needed, how to design service functions, what design is flawed and how to improve it.

## *13. Customer prospects*

The customer prospects of Flygprestanda want flight planning operations-software with a scheduling view service that enables them to: supports their operators, improve the quality of the work done by operators and generally facilitates the work done by their operators to a larger extent than their current systems. They also want tools that require less amount of resources to introduce in the company environment and when training new operators.

Introduction of new software in a company is a large risk and cost factor for the customer prospects since this involves gaining support for the change throughout the organization, training the affected employees and generating new routines. These actions are varying in difficulty depending on how fixed the organization is in its old routines and notions regarding their work.

## **6.4 Results from interview**

Results from interview in its raw form (in Swedish) is found in *Appendix 10.7*.

## **6.5 Results from Work Breakdown Structure**

*Illustration 20* shows results from work breakdown structure.

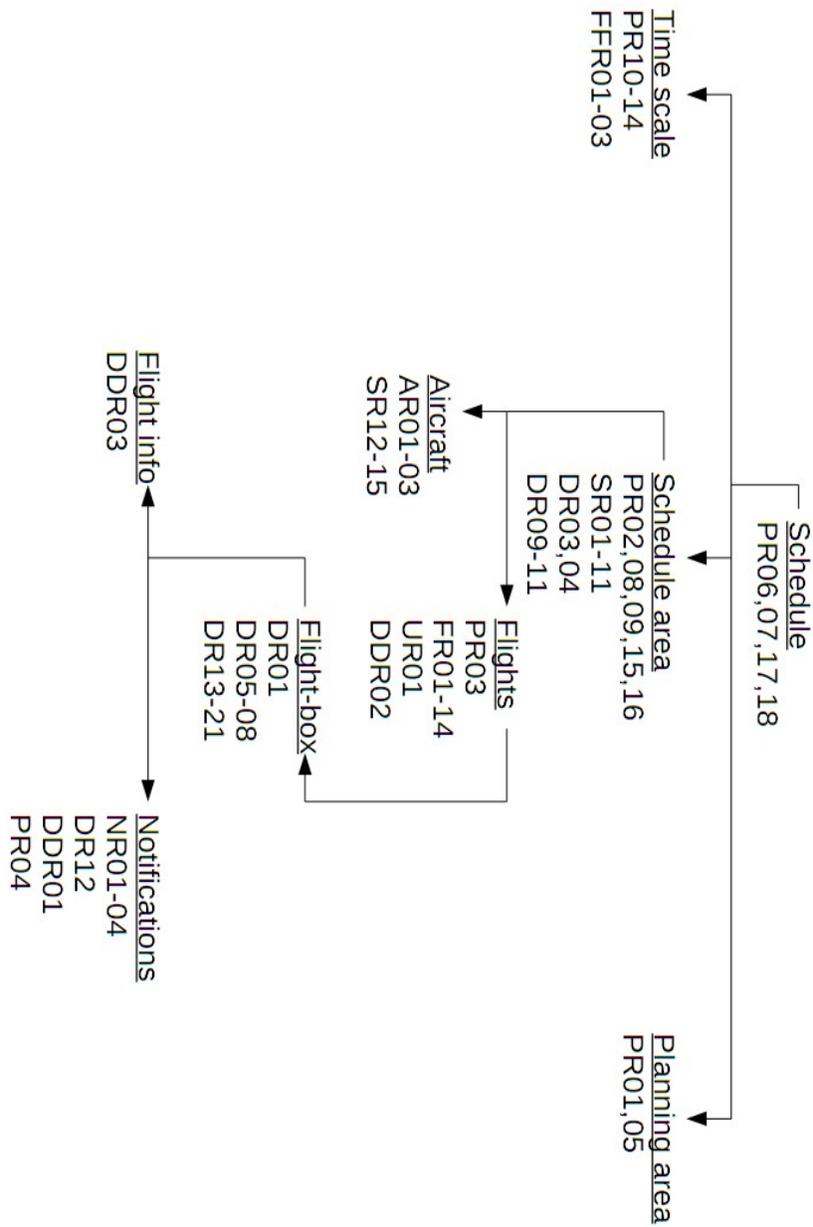


Illustration 20: Work breakdown structure

## 6.6 Results from Observation

### Observations made on David and Oscar while using Logic Pro

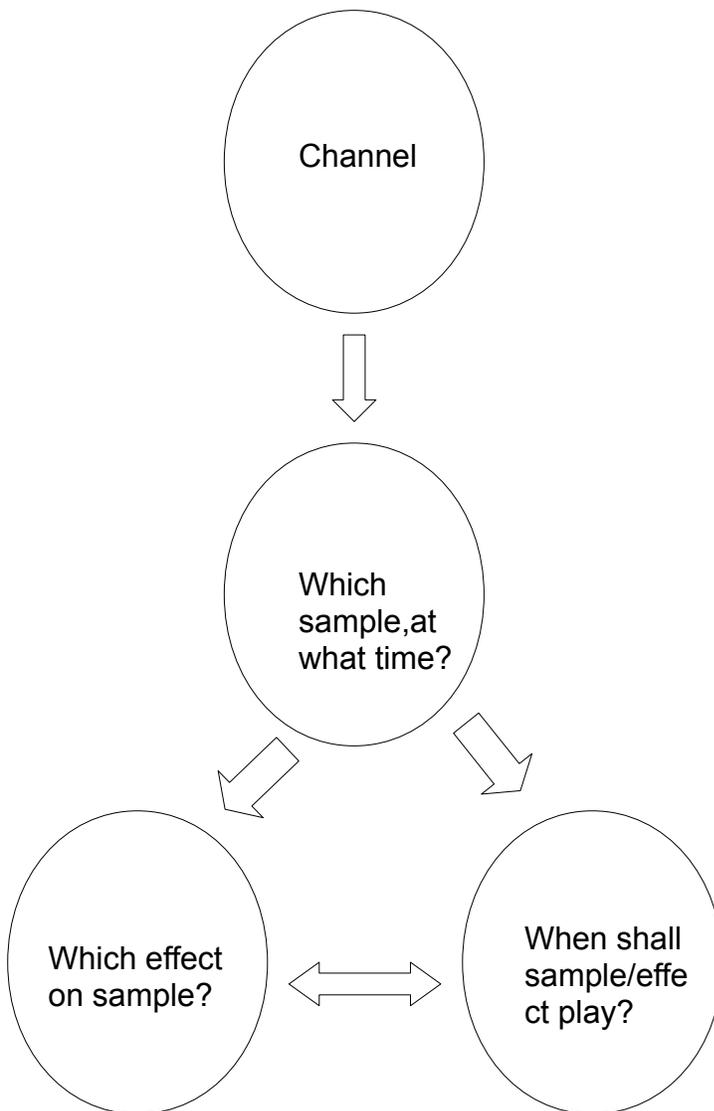
- **Cross-hair:** When an object is moved or dragged into the Arrange a white cross is displayed behind the object. This makes it easier to see where the object will land when the mouse is

released.

- **Expand individual channel:** In Logic Pro when you highlight a channel (tail) the whole row gets highlighted. The row also grows in height, and more information about the channel is displayed. The remaining channels stay the same way as before.
- **Preview playback:** When scrolling through/hovering over the samples in the sample database (Browser), that is located in the right hand column, Logic Pro starts playing that sample. It stops playing when you move the cursor away from the sample.
- **Colour code.** The user is able to colour code samples in the Arrange in order to differentiate the samples from one another.
- A general work area (Arrange) display all the samples that currently make up the tune, situated below the arrange is a work area for individual samples (Sample Editor). It allows the user to edit samples...
- Logic Pro has a column, situated on the left hand side of the Arrange, that manages Sound channels. Each Channel have associated buttons, mapped on their right hand side, to them that allows the user to; freeze, mute, record music to and/or play sound output from just that channel.
- **Drag'n'Drop:** Objects in the Arrange can be dragged and dropped within the Arrange, in-between different Channels.
- **Copy paste:** The user is able to use copy and paste functions on samples in the Arrange.
- The user is able to drag and drop samples from the Browser to the Arrange.
- **Drag'n'Drop Area:** The user is able to select an area in the Arrange in a standard fashion by holding down the mouse, dragging it over the screen which creates a rectangle and letting go of the mouse button. This enables the user to edit multiple samples simultaneously or drag and drop multiple samples between different channels and from different time slots. (more or less change X and Y factors for an area selected by the user)
- **Expand/decrease:** “Pulling” (pressing down the mouse button when the cursor is at the end/beginning of sample, and dragging the cursor) at the beginning and end of a sample is done to define start- and stop time.
- **Present crucial information:** When dragging a sample from the Browser, Oscar is not made aware that the sample is more than 30 minutes long. This causes aggravation and a break in the work momentum, Oscar removes the sample and then drags the same sample to the Sample Editor and exclaims: “Ahh, it was 30 minutes long” (Åh, den var 30 minuter lång).
- **Present wanted information:** David and Oscar is unsure how many beats per minute (BPM) a sample that they are editing has, they search and find this information within 10 seconds but it still disrupts their work flow.
- **Appropriate detail grade:** An observed pattern of David and Oscars work flow is that work done while in a zoomed-out view to obtain an approximate rough build. After this is done they zoom in at different time zones to tighten the samples and give them a more definite time frame for when to start playing and stop (if the samples are being looped). This is a hierarchical- or level based work flow observed.
- **Forgiving / undo:** An observation made on the general work flow of David and Oscar is that

their work is very dynamic. There is a lot of trial and error being performed, a sample is added to a Channel, it is played along with all or some other Channels, at different time slots and during these processes a consensus is made and it is judged whether the new sample is contributing to the song or shall be discarded. A function that facilitates this dynamic work flow are the buttons associated with each channel.

- **Fast understandable presentation:** In order to find an appropriate sample, a lot of parsing through Library and scrolling in the Browser is done.
- An approximate low level work flow process map for David and Oscar is described/detailed as shown in *illustration 21*:



*Illustration 21: Low level work flow process map*

## Debrief:

- Both David and Oscar agree that a large work area (Arrange) is vital for the success of Logic Pro.
- Short keys for frequently used functions and tools is requested from David and Oscar, they have customized the interface to some extent but customizable short keys/short keys are something they feel is missing in Logic Pro.
- Important according to the users is appropriate right-click menus. Relevant and often-used functions should be easy to access, for example by right clicking or short keys.
- The zoom function is very important in Oscar and David's "work pattern" as described above, and they are not content with Logic Pro's implementation of it.
- The zoom tool. Mastering the zoom tool is crucial when working with the software. In Logic Pro there are three different ways of using the zoom:
  - Dragging the mouse up and down while clicking and holding the time line.
  - Using keyboard short keys.
  - Dragging the horizontal scrollbar.

Using the third option does not focus the zoom on the time line, instead it scrolls at the same time as it zooms. The users thought this was very odd and it did not feel natural at all for them. The users preferred to use the first or the second option. They liked the first option, but they did not think the functionality was obvious, they considered it a tool for more experienced users.

## Old FOCS evaluation

- David and Oscar feel that situating flight objects on top of individual time axis/time lines clutters the interface and irritates the eye. Each flight should be given a whole row to fill as Samples do in Logic Pro's Channels.
- Departure- and arrival times should not be visible at all/all the time on/near the flight objects. Instead a hover-function or something similar is implied/suggested.
- Information regarding flights should be presented within the confines of the flight object. David explains: "Keep everything within the boxes, including the time" (Håll allt inom boxarna, tiden också).

## What modifications have they made on Logic-pro regarding interaction?

David and Oscar use customized short keys for different *screen sets* which entails switching in-between different views depending on which type modifications they want to make on a sample or channel. This gives them a quick way to access different features of the program.

## 6.7 Results from Assessment test

Results from assessment test in its raw form is found in *Appendix 10.8*.

## 6.8 Results from Validation test

Results from validation test in its raw form is found in *Appendix 10.9*.

# 7 Discussion

This section comprise of general discussion regarding the entire project. We will discuss the project purposes, methods used, results produced and also issues that has occurred during the project and the writing of this report.

## 7.1 Project initiation

The goal with this project from Flygprestanda's point of view was to receive a service that performs the required task with minimum effort for the users. The Schedule View should be able to adapt to customers that vary in size and have different needs. This was a concern as we stated our first two purposes:

- In what way should a schedule visualize information?
- How should the service scale the visualization, without impairing the usability?

The two initial purposes covers the visualization of the service but not the interaction between the user and the system. How should we design the system to minimize issues that can occur? What issues can a user have using the system? How should we handle an issue when it occurs? To be able to answer these questions we stated a third purpose:

- What are the main issues regarding usability when interacting with the service?

Purpose three initially stated “In what way should users interact with the service and its functionality?”. We felt that this purpose was hard to address which led to it being reworked into the current statement. We argue that the original statement was too loosely worded and lacked impact. When we reworked purpose 3 we used an approach that resembles that of specifying requirements. The specification of a requirement must enable testability i.e. quantifiable results, this is now possible when addressing purpose 3 as a requirement but was not in its original wording.

## 7.2 Results

In this section we will discuss the iterative work process and significant occurrences during the iterative work. We will address our efforts to enhance the user experience, and also discuss the difficulties encountered and other important findings.

The iterative nature of the work done in the design, implementation and validation phases have resulted in a likewise iterative nature of conclusions gradually growing in clarity and impact as the development iterations progressed.

### 7.2.1 Iteration 1

Since iteration 1 consisted of a re-work of parts of the system (including a new colour scheme) changes made in this stage are more noticeable. Before the re-work menus were most often accessed by right-clicking. In some cases it was not self explained which objects contained menus and which did not. The solution we found for this was to make a button of every click-able object. A button invites users to

click it and reacts to being clicked upon (in our case it looks like it has been pressed down). In addition to this the look and feel of the menus was improved as well with regard to structure and implicit naming of options and actions.

In addition to re-working the menus and buttons in the service, we also, together with the graphics team, developed a new colour scheme. When doing this we also had to take Flygprestanda's graphical profile into account.

Iteration 1 included other graphic fixes that could appear small when you describe them, but they impact the user interface quite a lot. For example we merged the top and bottom bar, so that the actual work area could be enlarged. Another example is that we put every flight object in between two vertical lines instead of on top of one.

We introduced a new element that we called *Playground*. The idea with the Playground was to give the users an area where they could plan ahead without their changes being saved at once. When the user felt satisfied with changes made they could be transferred to the actual schedule. The Playground concept got replaced later in the project.

At this stage we also worked with the concepts regarding the Schedule View level of detail. After data gathering and evaluation of prototypes, a framework was constructed. In order to delimit micro stress and change blindness, less textual and iconic information will be displayed when the user has zoomed out.[18][19] Furthermore, the severity of an incomplete task regarding a flight in combination with the proximity of the time line will colour code the flight object with a corresponding colour, as is shown in *illustration 16* found in section 6.1 *Results* from iterative work.

### 7.2.2 Iteration 2

The most significant activity in iteration 2 was the assessment test. The results from the test showed us which parts of the system did not work as well as intended. A critical issue that came to our attention was the lack of an *undo* action. It was discovered in the pilot test and later confirmed during the assessment test. We found that the service was not forgiving enough if the user managed to perform an unwanted action. On several occasions where the test subject performed an unwanted action she was unable to continue due to not knowing what had happened, and there was no way to undo the last action. We started to design solutions for this issue which would be implemented later on.[20]

Apart from the critical issue discovered a few minor design flaws were also found. Fixing these led to a more efficient service and a more pleasant experience while using it.

### 7.2.3 Iteration 3

We felt that the Playground was a nice feature but we could not really motivate the need of it. Still we wanted the user to be able to plan new flights and get live feedback without actually changing the state of the schedule. We discarded the Playground and replaced it with a new concept. In the new concept every schedule can be viewed in info mode or edit mode. When in info mode flights and aircrafts cannot be added or modified, its purpose is solely to present information. Entering edit mode makes it possible to modify and add aircrafts and flights. An idea for future implementation is to make different users have different access levels.

Panels appearing when clicking a flight also received a rework in this iteration. Together with the graphics team we designed a new prototype which we implemented. We suggest solutions such as the

info/edit expansion window. This approach will have much less of a disorientating effects than that of the pop up window.[21]

A new way of handling flights (in edit mode) was introduced in this iteration. To create a new flight you simply click the position in the schedule where you want it to start. A panel appears where needed information can be inserted. To move a flight you simply drag-and-drop it (still in edit mode). By simplifying the handling of flights the efficiency of the system increases. The user can perform tasks more rapidly, with fewer actions, which makes the overall feel of the system smoother.

#### 7.2.4 Iteration 4

The purpose of this iteration is similar to the purpose of iteration 2. A validation test was held to verify implemented functionality. As we had hoped for, no critical issues were found at this stage. A critical issue found this late in the project could heavily delay the release of the upcoming version of the system. It could imply the risk of faulty design or implementations which is both time consuming and expensive. Instead the validation test helped us, as in iteration 2, locate a few design and interaction flaws that hampered the overall feel and flow of the service.

An important finding of the validation test concerned the design of info- and edit panels. The test subject stated that the edit panel made her feel uneasy when she approached it. The design, she stated, felt unfinished and “not real”. She carried on explaining that she much rather consult the info panel when solving a problem. The edit panel, shown in *illustration 11* found in section 6.1 *Results from iterative work*, had not received our proposed standardized design. This standard incorporates the domain standards in combination with our concept of mapping the associated times in accordance to the natural representation of the departing location, the aircraft being in flight and the arrival at the destination. Furthermore all objects that concern a flight shall be colour coordinated as shown with the colouring of *tail objects*, their upcoming *flight objects* and that flight objects *info/edit expanding window illustration 14* and *16* found in section 6.1 *Results from iterative work*. We had proposed that this type of design should be consistent throughout the interface as a vital part of solving usability issues and improve user experience.[20]

Our opinion in this matter was very much criticised by key stakeholders before conducting the validation test. It gave us great comfort and assurance that the info panel, as shown in *illustration 14* found in section 6.1 *Results from iterative work*, with our proposed design out performed the design of the edit panel. The validation test leads us to believe in our proposed use of the consistent use of the standard representation of information in FOCS Schedule View.

The questionnaire was sent to one of Flygprestanda's customers to investigate their attitude towards the changes to the service interface. In what aspects were they satisfied, and what needed improvements? Unfortunately we did not receive any responses to the questionnaire with the explanation that they had no time to answer it. As stated earlier the survey is an important factor when delivering and introducing a new software. How can you develop a service designed for a specific type of user when you have limited or no access to the user in question?

### 7.3 Project environment

Throughout this project the authors have thrived in an environment where help with mock-ups and prototype work, implementing code, testing that code, and great overall supervision always was at hand. In this environment there has also been contradictions and disagreements, which is not

necessarily something negative. On the contrary, constructive arguments for and against concepts and ideas have contributed greatly to the outcome of this thesis. It is however always hard to argue with the graphics department when the line of argument is “this looks better” or “that's not as good looking as this”. When the system architect tells you that the users have grown accustomed to interaction that you have tested with poor performance, and changing it “would only make them mad”, how do you proceed?

## 7.4 Usability

The authors, and studies in the field of aesthetics, do agree with the graphics department with the argument that great aesthetics improve the perceived usability of an interface but it should not trump a design that has superior usability interaction.[22]

Likewise the authors agree that changing standardized interaction that has been implemented for an extensive period of time is a tricky business. However, we perceive good organizational skills in combination with new improved interaction to have far greater advantages than using a standard that “has always been”. Introducing a new product or interface design is just as much a managerial activity as it is an activity for the tech-department, were creating understanding for the change and showing its benefits to all stakeholders is key. This can be done by for example a time of running both system designs in parallel. A human being will always oppose change until she is made aware of the benefits.

This stated we are aware of the conservative nature of Flygprestanda's customers and the flight operations domain in general, it is taxing to bring about change with hesitant or even reluctant response.

## 7.5 User centred design

A serious limitation of this thesis is the distance from the end user. We have addressed this issue earlier but feel it needs a final mention. We initially requested to perform usability testing with the customers users, our supervisor would ask but told us not to get our hopes up. The request was rejected on the basis that their work in shifts would not allow for any operators to be freed up from work, an observation would also be out of the question.

This was disheartening, but we still had hope. In the early phase of the project we got indications from different sources [interview][initiator][article] that the domain is conservative and change is few and far between, instead we should take inspiration elsewhere. The data gathering activities performed, such as interviews, brainstorming and the field observation, provided the project with useful concepts and solutions that were incorporated into the interaction design of FOCS. It is the authors shared opinion that the activities during this project conducted have had a user centred approach, although not academically optimal it has been the most optimal in the practical sense that it has been realistic and at par with the trade of the flight operations industry.

This stated we feel that some stress testing, a format we are not that accustomed to, would have been great to perform. This will also be suggested as a future usability venture for Flygprestanda when presenting this thesis to them.

## 8 Conclusions

The conclusions we have made with regards to the format of information in the flight operations schedule are as follows:

- Textual information shall be presented in domain standardized syntax with ICAO codes and other standard abbreviations in combination with the proposed concept of standardized flight object design.
- Information that need to capture the users attention will colour code all objects associated to the information.

In order to allow for the scaling of the interface with multiple schedules, tails and flights the following conclusions were made:

- A higher level of zoom must entail less information of textual and iconic nature being presented and vice versa.
- The time line will dictate when the colour coding of schedule objects will occur.

The last purpose address the main usability issues of the schedule interface. Our work have generated the following conclusion in this regard:

- Avoid a multi window approach such as pop-up and dialogue windows.
- Focus efforts on the contradictory issue of aesthetics and minimalism with the need for information visualization at closer levels of zoom.

By following these recommendations you have a sound foundation when designing the interaction with a scheduling tool and in a sense any tool with interaction revolving around visualizing and scaling large quantities of information.

## 9 References

### 9.1 Factual references

- [1] ISO 9241-11: Guidance on Usability (1998), [www.iso.org](http://www.iso.org), 25/5 2011
- [2] Jakob Nielsen's website, <http://www.useit.com/alertbox/20030825.html>, 25/5 2011
- [3] Jonas Löwgren (1993), Human Computer Interaction, Studentlitteratur, Lund, Sweden.
- [4] Donald A. Norman (2002), The Design of Everyday Things, The Perseus Books Group, New York, USA.
- [5] Jan Gulliksen et al. (1999), User Centred Design in Practise – Problems and Possibilities, KTH, Stockholm, Sweden.
- [6] Usability Professionals' Association, [http://www.upassoc.org/usability\\_resources/about\\_usability/what\\_is\\_ucd.html](http://www.upassoc.org/usability_resources/about_usability/what_is_ucd.html), 25/5 2011
- [7] Helen Sharp et al. (2007) Interaction design -Beyond human-computer interaction, John Wiley and sons, Ltd, Chichester, England.
- [8] Jeffrey Rubin (1994), Handbook of Usability Testing – How to pland, design,and conduct effective tests, John Wiley and sons, Ltd, Chichester, England.
- [9] Soren Lauesen (2002), Software Requirements – Styles and Techniques, Addison-Wesley, Boston
- [10] Mind Tools Ltd, [http://www.mindtools.com/pages/article/newTMC\\_05.htm](http://www.mindtools.com/pages/article/newTMC_05.htm), 2/2 2011
- [11] The Chartered Institute of Personel and Development, <http://www.cipd.co.uk/subjects/corpstrgy/general/swot-analysis.htm>, 2/2 2011
- [12] SWOT analysis, as it is carried out at Flygprestanda AB, 2/2 201
- [13] MIL-HDBK-881A, Departure of Defense Handbook [http://www.acq.osd.mil/pm/currentpolicy/wbs/MIL\\_HDBK881A/MILHDBK881A/WebHelp3/MILHDBK881A.htm](http://www.acq.osd.mil/pm/currentpolicy/wbs/MIL_HDBK881A/MILHDBK881A/WebHelp3/MILHDBK881A.htm) 2/4 2011
- [14] Work Breakdown Structure, [www.workbreakdownstructure.com](http://www.workbreakdownstructure.com), 3/4 2011
- [15] Hipmunk, <http://www.hipmunk.com/>, 4/4 2011
- [16] Monique W.M. Japseprs et al. (2004) The think aloud method: a guide to user interface design. Elsevier Ireland Ltd.
- [17] John P. Chin, Virginia A. Diehl, Kent L. Norman (1998). Development of a Intrument Measuring User Satisfaction of the Human-Computer Interface. Human-Computer Interaction Laboratory, Department of Psychology, University of Maryland, USA.
- [18] Saturnino Luz et al. (2009). Chronos: A Tool For Interactive Scheduling and Visualisation of Task Hierarchies. Department of Computer Science, Trinity College Dublin 2, Ireland.
- [19] Min K. Chung, Donghun Lee, Cheolhyun Jeong (2011). The effects of zoomable user interfaces and user age in searching for a target with a mouse on a two-dimensional information space. Department of Industrial & Management Engineering, Pohag University of Science and Technology (POSTECH), San 31, Hyoja, Pohag, 790-784, South Korea.

[20] Alice M. Mulvehill (1988). A User Interface for a Knowledge-Based Planning and Scheduling System. IEEE Transactions on systems, man, and cybernetics, volume 18.

[21] G. Susanne Bahr, Richard A. Ford (2010). How and why pop-ups don't work: Pop-up prompted eye movements, user affect and decision making. Florida Institute of Technologies, USA.

[22] Daniel Filonik, Dominikus Baur (2009). Measuring Aesthetics for Information Visualisation. Media Informatics Group, University of Munich, Germany.

## 9.2 Inspirational publications

[23] Johan Cahill, Nick McDonald, Captain Gabriel Losa (2010). Intelligent planning and the design of a new risk-based, intelligent flight plan. Springer-Verlag London Limited, England.

[24] Dennis F. X. Mathasiel (1996). Decision support for airline system operations control and irregular operations. Flight Transportation Laboratory, Department of Aeronautics & Astronautics, MIT, Cambridge, MA 02139, USA.

[25] Alice M. Mulvehill (1988). A User Interface for a Knowledge-Based Planning and Scheduling System. IEEE Transactions on systems, man, and cybernetics, volume 18.

[26] Michael McCurdy (2008). Planning Tools for Mars Surface Operations: Human-Computer Interaction Lessons Learned. NASA Ames Research Center

[27] Dandan Huang et al. (2009). Visualization Techniques for Schedule Comparison. Blackwell 25 Publishing

[28] Billur Engin et al. (2008) Maestro: Design Challenges for a Group Calendar. IEEE

# 10 Appendix

## 10.1 Framework used when conducting Observation

Observation - An exploratory observation of Husarrest in their workplace

### Basics

- the Person
- the Place
- the Thing

### Pay attention to:

- SPACE - What is the physical space like and how is it laid out?
- ACTORS - What are the names and relevant details of the people involved?
- ACTIVITIES - What are the actors doing and why?
- OBJECTS - What physical objects are present? eg. furniture
- ACTS - What are specific individual actions?
- EVENTS - Is what you observe part of a special event?
- TIME - What is the sequence of events?
- GOALS - What are the actors trying to accomplish?
- FEELINGS - What is the mood of the group and of individuals?

### Material to be recorded/collected (Ethnography)

- Activity or job description
- Rules and procedures that govern particular activities
- Descriptions of activities observed
- recordings of the talk taking place between parties involved in observed activities
- Informal interviews with participants explaining the detail of observed activities
- diagrams and physical layout, including the position of artefacts
- Descriptions/Photographs/Videos of artefacts used in the course of observed activities
- Workflow diagrams showing the sequential order of tasks involved in observed activities
- Process maps showing connections between activities

### If not covered during the test itself, make certain to address the following issues during debriefing:

- GUI meets their needs?
- Explore specific navigation errors made during test.
- Did they pick up on colour coding?
- Easy to learn (did errors made decrease)?

### Examine similarities between FOCS schedule view and Logic Pro

- Operations
- Graphical layout
- Functions

### How does Logic Pro visualize information?

- How do they differentiate between samples?
- What characteristics of operating Logic Pro change when the information displayed increases?
- What difficulties do they experience?
- How do the users perceive the mapping of adjacent functions and buttons?

**What modifications have been made on the user interface and why?**

What improvements would they like to introduce?

## 10.2 Assessment test

### 10.2.1 Test plan

#### *Purpose*

- Provide insight in efficiency and accuracy of current design implementations as well as their usability
- Provide a standard for future usability testing and evaluation
- Find issues and opportunities regarding interaction

#### *Test environment*

- Workstation in closed of office; computer, two LCD screens, mouse, keyboard and adjustable chair.
- One screen presenting the Schedule View interface, the other test scenarios in a full-screen pdf-doc.
- TMs behind and outside the personal sphere of TS

#### *Roles*

- TS: performs scenarios independently, uses think aloud approach
- TM: passive role during scenarios, only interact if TS abandons task.
- Test observer: Project initiator participates as a silent, passive observer

#### *Test design*

- Pilot test
- Introduction
  - Thank TS for participation – Test anonymity
  - Purpose of the test
  - About the TMs; what we do at Flygprestanda, our roles in the test
  - Purpose of FOCS and Schedule view
  - How the test will be conducted, general logistics
  - Assert the background of TS
- First impression:  
TS present the first impressions TS gets from the Schedule View interface
- Test scenarios
  1. Task description
  2. Perform task
  3. Results

#### 4. Discuss scenario performance

- Last impression: TS present the impressions TS gets from Schedule View after conducting the assessment test.
- Debriefing  
Semi-structured interview; attached
- Test observer comments
- Thank TS for participation, any questions?

#### *Scenario with task specification*

- Attached

#### *Quantitative data to be collected*

- Time to complete task
- Number of complete tasks
- Number of complete tasks with help from TM
- Number of failed tasks with help from TM
- Amount/times when help from TM is required

#### *Qualitative data to be collected*

- Useful quotes
- Learnability
- Appropriateness of prototype implementation

## 10.2.2 Scenario list

### Uppgift 1

1. Skapa ett nytt schema och döp det till "Therese". Schemat skall vara aktivt.
2. Lägg till tre flygplan: ett EMBRAER E-135 och två EMBRAER E-145.
3. Ge flygplanet av typen E-135 en tail.
4. På samma flygplan, skapa en ny flight från ESGG till EGCC som avgår klockan 17:00 idag.

### Uppgift 2

1. Stäng det schemat du skapade
2. Öppna schemat "Evaluation"
3. Byt namn på schemat till "Renamed"

### Uppgift 3

1. Ändra avgångstid på flighten TEST1 till 17:00

2. Ange att flight TEST3 skall ha piloten ”Jimmy Sjöquist”
3. Byt namn på flighten TEST4 till FLIGHT5

#### Uppgift 4

1. Byta tail på flygplanet FP-EAA/E135 till SE-RAA.
2. Ta bort den angivna tailen på flygplanet av typen MD87
3. Ladda in en sparad flight med namn ”TESTFL” till flygplanet FP-EBA/E145
4. Ändra tail på flighten ”TESTFL” som du precis laddade in till SE-RAD/E145
5. Ta sedan bort flygplan SE-RAD/E145.

#### Uppgift 5

1. Ta bort flighten ”TEST5”
2. Gör så att flighten ”TEST2” blir inställd.
3. Ta bort att flight ”TEST2” är inställd (uncancel)
4. Släng schemat ”Renamed”

### 10.2.3 Test monitor guidelines

#### *Introduktion och intervju*

- Tacka för FP's medverkan, det vi har med om försöket i rapport är anonymt
- Fråga om FP's bakgrund
- Berätta vem vi är och vad vi håller på med
- Introducera FOCS och är vilket syfte systemet har.
- Beskriv schema-vyn är och vad det är tänkt den skall lösa
- Beskriv hur testet skall genomföras
  - Schema-vyn vi utvärdera inte hennes genomförande, det går inte att göra fel
  - Genomföra uppgifter/deluppgifter
  - Kortare diskussion efter varje uppgift
  - Avslutande debriefing
- Frågor innan vi börjar?

#### *First impression*

#### *Testgenomförande*

#### *Last impression*

### *Debrief*

- Hur kändes det att navigera omkring i de olika menyerna?
- Var det någon gång oklart att veta vart man skulle vända sig för att lösa en uppgift?
- Gå igenom de problem som uppstod
  - Orsak
  - Hur det påverkade FP
  - Hur man skulle kunna lösa problemen
- Var det några namn på funktioner (labels) som var oklara?
- Vad tyckte FP om playground/work-area, hur påverkade det henne?
- Vilka snabbkommandon, till vilka funktioner, kan du känna du hade behövt?
- Tack för ditt deltagande

### Observatörkommentarer

### Avslutning

- Tacka för deltagandet
- Uppföljning
- Det var allt, frågor? Tack igen, hejdå!

## 10.3 Validation test

### 10.3.1 Scenario list

#### Uppgift 1

1. Skapa ett nytt schema och döp det till "Therese". Schemat skall vara aktivt.
2. Lägg till två flygplan av typen EMBRAER E-145, i schemat du skapade.
3. Lägg till en valfri tail av typen EMBRAER E-135, i schemat du skapade.
4. Skapa en ny flight från ESGG till EGCC som avgår klockan 17:00 idag, på ett valfritt flygplan.

#### Uppgift 2

1. Släng det schemat du skapade
2. Öppna schemat "Evaluation"
3. Byt namn på schemat till "Renamed"

#### Uppgift 3

1. Ändra avgångstid på flighten TEST1 till 17:00
2. Byt namn på flighten TEST4 till FLIGHT5

#### Uppgift 4

1. Ta bort flygplan SE-XX03/E135.

#### Uppgift 5

1. Ta bort flighten "TEST5"
2. Undersök varför "TEST2" orange?
3. Gör så att flighten "TEST2" blir inställd.
4. Ta bort att flight "TEST2" är inställd (uncancel)

#### Uppgift 6 (Extra uppgift)

1. Lägg till en flight utan att använda "+"-knappen i angränsning till flygplan/tail
2. Flytta/byt avgångstid på en flight utan att specificera några klockslag i textfält
3. Flytta valfri flight från ett flygplan/tail till ett annat.

### 10.3.2 Test monitor guidelines

#### *Introduktion och intervju*

- Tacka för FP's medverkan, det vi har med om försöket i rapport är anonymt
- Fråga FP om det är nått angående FOCS hon vill repetera sen sist.
- Beskriv hur testet skall genomföras
  - Schema-vyn vi utvärdera inte hennes genomförande, det går inte att göra fel

- Genomföra uppgifter/deluppgifter
- Kortare diskussion efter varje uppgift
- Avslutande debriefing
- Frågor innan vi börjar?

#### *First impression*

Upplever FP några förändringar sen sist?

#### *Testgenomförande*

- Task+description
- Precondition
- Desired result
- Help FP if needed

#### *Last impression*

Bättre eller sämre än senast?

#### *Debrief*

- Hur kändes det att navigera omkring i de olika menyerna?
- Var det någon gång oklart att veta vart man skulle vända sig för att lösa en uppgift?
- Gå igenom de problem som uppstod
  - Orsak
  - Hur det påverkade FP
  - Hur man skulle kunna lösa problemen
- Var det några namn på funktioner (labels) som var oklara?
- Vad tyckte FP om playground/work-area, hur påverkade det henne?
- Vilka snabbkommandon, till vilka funktioner, kan du känna du hade behövt?
- Tack för ditt deltagande

#### *Avslutning*

- Tacka för deltagandet
- Uppföljning
- Det var allt, frågor? Tack igen, hejdå!

## 10.4 QUIZ Questionnaire with modifications made

Specify your age:

Specify your sex:

Supplemental initial open ended questions:

- What feature in Schedule View has the worst performance? (Please specify the reason)
- What feature in Schedule View has the best performance? (Please specify the reason)
- Do you expect anything of Schedule View that it cannot perform?

**How often work with [the service to be evaluated] schedule view?**

Daily      Weekly      Monthly      Less often

**Overall reactions to the software**

terrible                      wonderful

0 1 2 3 4 5 6 7 8 9

difficult                      easy

0 1 2 3 4 5 6 7 8 9

frustrating                      satisfying

0 1 2 3 4 5 6 7 8 9

inadequate power                      adequate power \*\* (rephrased)

0 1 2 3 4 5 6 7 8 9

dull                      stimulating

0 1 2 3 4 5 6 7 8 9

rigid                      flexible

0 1 2 3 4 5 6 7 8 9

**Screen**

Characters on the computer screen

hard to read                      easy to read

0 1 2 3 4 5 6 7 8 9

Highlighting on screen simplifies task \*\* (rephrased)

not at all                      very much

0 1 2 3 4 5 6 7 8 9

Organization of information on screen

confusing                      very clear

0 1 2 3 4 5 6 7 8 9

Sequence of screens \*\* (non applicable, removed)

confusing                      very clear

0 1 2 3 4 5 6 7 8 9

**Terminology and system information**

Use of terms throughout service

inconsistent                      consistent

0 1 2 3 4 5 6 7 8 9

Labels and terminology is related to the tasks you perform

never                      always

0 1 2 3 4 5 6 7 8 9

Icons on tools describe tool function

not at all                      clearly

0 1 2 3 4 5 6 7 8 9

Position of messages on screen

inconsistent                      consistent

0 1 2 3 4 5 6 7 8 9

Messages on screen which prompt user for input

confusing                      clear  
0 1 2 3 4 5 6 7 8 9

Service keeps you informed about what it is doing

never                              always  
0 1 2 3 4 5 6 7 8 9

Error messages

unhelpful                      helpful  
0 1 2 3 4 5 6 7 8 9

### **Learning**

Learning to operate the service

difficult                              easy  
0 1 2 3 4 5 6 7 8 9

Exploring new features by trial and error

difficult                              easy  
0 1 2 3 4 5 6 7 8 9

Remembering names and use of commands

difficult                              easy  
0 1 2 3 4 5 6 7 8 9

Taks can be performed in a straight-forward manner

never                                      always  
0 1 2 3 4 5 6 7 8 9

Help messages on the screen

unhelpful                              helpful  
0 1 2 3 4 5 6 7 8 9

Supplemental reference materials \*\* (non applicable, removed)

confusing                      clear

0 1 2 3 4 5 6 7 8 9

**System capabilities**

System speed

too slow                      fast enough

0 1 2 3 4 5 6 7 8 9

System reliability

unreliable                      reliable

0 1 2 3 4 5 6 7 8 9

System tends to be\*\* (non applicable, removed)

noisy                      silent

0 1 2 3 4 5 6 7 8 9

Correcting your mistakes

difficult                      easy

0 1 2 3 4 5 6 7 8 9

Experienced and inexperienced users' needs are taken into consideration

never                      always

0 1 2 3 4 5 6 7 8 9

## 10.5 Internal review

<b>CONCEPTS</b>	Level of detail Flight notification	Concept Initial implementation, suboptimal performance. Guidelines developed for future implementation	Additional research needed Additional research needed
	Schedule area warnings	Initial implementation	Additional research needed,
	Short keys Icon usage	Concept Initial implementation, suboptimal performance. Guidelines developed for future implementation	Additional research needed Additional research needed
<b>LOW PRIORITY</b>	Schedule menu	Implemented, released, tested in iteration 2	No action required at this stage
	Search tool	Implemented, released	No action required at this stage
<b>OUTSIDE PROJECT SCOPE</b>	Playground / workarea	Initial implementation, in current version replaced by edit/info toggle-concept	Future implementation. Outside project scope
	Zoom	Initial implementation, suboptimal performance. Guidelines developed for future implementation	New concept to be implemented. Outside project scope.
	Flight turn-around times	Concept, mockups	Future implementation, outside project scope
	Icon design	Mockups, initial implementation	Additional research needed, outside project scope
	Hover and tooltip info	Initial implementation, released	Future implementation, outside project scope
	Aircraft lease management	Concept, mockups	Future implementation, outside project scope
	Aircraft maintenance management	Concept, mockups	Future implementation, outside project scope
	Flight take-off slot times	Concept, mockups	Future implementation, outside project scope
	Schedule area slider window	Concept, low level mockups	Future implementation, outside project scope
	Search options	Concept	Additional research needed, outside project scope
	Role based schedule area	Concept	Additional research needed, outside project scope
	Facilitate crew handling	Framework implemented	Future implementation, outside project scope
	Facilitate trip planning	Concept	Additional research needed, outside project scope
	Hub/airport view	Concept	Outside project scope
	Feature guides / wizards	Concept	Outside project scope
<b>CANCELLED OR REPLACED</b>	Schedule object menu	Implemented, tested in iteration 2	Replace with edit/info toggle-concept
	Flight object menu	Implemented, tested in iteration 2	Replace with edit/info toggle-concept
	Aircraft object menu	Implemented, tested in iteration 2	Replace with edit/info toggle-concept
	Flight options	Implemented, released	Cancelled, replaced by edit/info toggle-concept



## 10.7 Results from interview

### Frågor ur kundens perspektiv

(Kursiva frågor är frågor vars svar kräver mer grundlig verifiering)

- **Hur beskriver kunder sin dagliga verksamhet? (vanliga moment/uppgifter)**  
Operativa uppgifter:
  - Färdplanering
  - Schemaläggning av flyg
  - Pairing
- **Vilka är de viktigaste uppgifterna kunderna måste kunna genomföra?**
  - Ta hand om effekterna av förseningar av flighter
  - Ha översikt över de planerade flighterna
  - Minimera/korta ner procedurer kring skiftövergångar/skiftbyten
- **Hur beskriver kunden problem de upplever i den dagliga verksamheten?**
  - Kunden använder många olika system i sin verksamhet. Operatörerna får då agera gränssnitt mellan dessa olika system då de inte är integrerade. Detta är tidskrävande, det kräver att varje operatör skapar sig egna inlärd rutiner och det kan vara frustrerande.
- **I vilken typ av situation arbetar kunderna under stressade förhållande?** (hur fungerar nuvarande system som *stöd* i dessa situationer)
  - En erfaren operatör får stöd av de nuvarande systemen i sin dagliga rutin
  - I stressade situationer kan systemen bli en belastning
- *Vilka idéer har kunden om framtida system? (eventuella lösningar, funktionalitet, etc.)*
  - Rollbaserad schema-vy: Vyn presenterar olika komponenter beroende på typen av operatör.
  - Systemet borde vara mottagligt för regelverksändringar kring flygrutter och framför allt representera dessa begränsningar med tydlig feedback.
  - Schema-vyn skall kunna presentera information/data från andra domäner (crew, maintenance)
  - Test-schema-vy/"Playground" för att kunna undersöka ny business kring nya rutter, presentera full analys med värdefull data.
  - Presentera/visualisera Slot management
  - Presentera/visualisera flight log/andra doc
- *Har kunderna några synpunkter på vad som är genomförbart och ej?*
  - Manualhantering, revisionshantering sker idag med papper-penna, stoppa in i system
- *Har kunderna uttryckt att det finns konflikter i sin verksamhet som kan påverka vårt projekt?*
  - Goda relationer med konkurrenter på management-nivå
  - Vilka andra system tillåter oss att importera information/data från dem till Focs?
  - Mycket dynamiska arbetsförfarande i operations, lösa rutiner.
  - Stridigheter kring acceptans för Focs mer optimala rutter jämfört mot de traditionella rutter organisationerna arbetat med tidigare.
  - Kostsamma inlärningsprocesser i organisationen

## Frågor ur sales-avdelningens perspekt

- **På vilket sätt marknadsförs FOCS?** (vilka delar av programmet framhävs, vad pushar ni för)
  - Integrerat system med många tjänster samt prestandaberäkningar
  - "påväg mot ett komplett OP"
- **Vilka av FOCS tjänster är det svårt att förmedla nyttan av till kunden?**
  - Driftdownmodulen vilket vi tror beror på okunskap hos kunden och deras gamla rutiner med karta å kompass
- **Vad i marknadsföringen för FOCS visar kunder störst intresse för?**
  - Ruttgenerering som är snabb, mer optimal och har god representation
  - Interaktiv karta i generering
  - Datamerging, automation av datahantering/information/databaser
- **Hur viktig tycker ni att god interaktionsdesign och användarupplevelse är för FOCS?**
  - Bra för att rucka på rutiner
  - Fånga intresse hos kunden
- **Hur ställer ni er till att FOCS ska konkurrera ut liknande produkter på marknaden?**
  - Mycket positivt, optimalt förhållande
- **Finns det något mer ni vill tillägga?**
  - Under utvecklingen av tjänsten skall projektgruppen tänka utanför flygplaneringsbranchens ramar, detta då alla nuvarande system är utvecklade inom den branchen.
  - Handhavandet i schema-vyn skall enligt kunden vara så nära automation som möjligt.

## 10.8 Results from Assessment test

### First impressions

- Lacks colours, FP generally wants more use of colours, example: to let the user know which schedules are "active" (at this stage no flights were displayed) utöka "det ser lite grått ut"
- "just nu ligger vi rätt, va?" FP checks time line/time-scale to confirm
- FP tries to navigate using keyboard left/right arrows

### Tasks

#### 1.1

schedule button → new schedule → names it "therese" → checks active box → save option

No comment, no observation.

Time to complete task: 00:16

#### 1.2

tries drag n drop on aircraft in a different schedule

press dropdown-button → add aircraft option → chooses e-135 → add option

press dropdown-button → add aircraft option → chooses e-145 → add option \*2times

Time to complete task: 00:59, 1 error committed

1.3

took right actions but renew option button confuses FP

right clicks aircraft → assign tail option → chooses tail → load option

Time to complete task: 00:09

1.4

left clicks on aircraft label/right clicks outside label

fills in ADEP in flight no. Box, FP realizes this when ADES box is to be filled

user uses TAB to navigate the create flight dialogue box

when editing ADEP/ADES suggested airports requires double click to be selected, FP single clicks and then ignores the selection drop-down.

right clicks aircraft label → add flight option → fills in data in dialogue box → save option button

Time to complete task: 01:31, 1 errors committed

### **Task 1 – Discussion**

FP is used to be able to drag and drop objects in-between areas. FP is not fond of this behaviour though, since it is error-prone - ”det är så lätt att göra fel”

FP expresses that schedule options are well design - ”mycket tydliga options”

2.1 & 2.2 (FP combines these tasks)

schedules button → unchecks ”therese”-schedule → checks ”evaluation”-schedule

No comment, no obsv.

Time to complete task: 00:31

2.3

schedule drop down-button → edit option → enters new name → save option

renew button confuses FP

Time to complete task: 00:08

### **task 2 – discussion**

no comment, no obsv.

3.1

finds the flight → right clicks → edit → changes the time → save

No comment, no obsv.

Time to complete task: 00:21

3.2

Drags sideways to locate the flight, FP has a little trouble finding the flight because she does not scroll down. After a while she finds the scroll and she finds the flight.

Right clicks flight → assign pilot option → browses to "Jimmy" → save option

Time to complete task: 00:38

3.3

right clicks flight → edit option → changes flight no. Box → save option

No comment, no obsv.

Time to complete task: 00:15

### **task 3 – Discussion**

FP wants some kind of feedback when hovering a flight object information regarding crew i.e which personnel is missing (has not checked in), which personnel is not yet assigned.

FP express that she lacks feedback in general from the flight objects, they are not very informative. Suggest colour mapping of different types of info.

4.1

tries to left click aircraft label,

tries to ctrl + left click aircraft label,

right click aircraft label → unassign option → clicks "+"-button associated with aircraft → browses and selects tail → load option,

FP expresses content after completing task - "tack!"

Time to complete task: 01:23, 2 errors committed

4.2

Right click tail → unassign tail option (learnability from task 4.1)

Time to complete task: 00:11

4.3

FP has a hard time right clicking the label which causes confusion since this approach has been effective in previous tasks. FP tries right clicking other labels of other aircraft with varied result when "missing and hitting" their labels.

Right click aircraft label → load flight option → browses and selects flight → load option

Time to complete task: 00:47

4.4

FP has issues finding the flight, drags the schedule area to locate it.

Right clicks flight → edit → cancel option (after searching for solution to complete task)

Right clicks label of new tail → load option → enters flight no. in search box → cancel option (after search yields no results)

Tries drag and drop flight in-between tails

Right clicks flight → delete flight option → delete option

Right clicks label of new tail → load option → enters flight no. in search box → cancel option (after search yields no results)

Asks test conductor to assist.

When conferring with the test conductor the cause of confusion is in part the fact that after the edit option did not present a solution to the problem the flight drop-down menu as a whole was discarded as a mean to present a solution.

Another factor is that the task description requires the test subject to ”..change tail ..” of a flight and the option for this in the drop-down menu is ”change aircraft”

When the solution to the task, as it is currently implemented, was presented to the test subject she exclaimed: ”Jag var för snabb” and put the blame on her self, which the test conductor was quick to explain that this was not the case.

Time to complete task: - , 5 errors committed, time to surrender task: 02:40

4.5

right clicks aircraft → delete aircraft option → delete option

FP is somewhat confused as to what the orphan status of a flight entails, ponders it for a moment and the clicks the delete button.

Time to complete task: 00:10

#### **task 4 – Discussion**

FP wants to save after actions like delete (of various objects). The test conductor asks FP on input regarding an undo function which is welcomed.

FP wants to modify, move and create aircrafts/flights/schedule without consequence. Expresses wishes similar to the planned implementation of the work area/playground.

FP describes that the reason she wants to use aircraft menus is that she is used to work with crew-systems where most tasks are solved by modifying crew member objects, that are situated in the same general area as aircrafts in FOCS schedule view.

5.1

Locates flight in schedule area → Right clicks flight → delete flight option → delete option

No comment, no obsv.

Time to complete task: 00:06

5.2

Right clicks flight → cancel flight option (system should prompt after cancel flight option)

No comment, no obsv.

Time to complete task: 00:06

5.3

Right clicks flight → uncanceled flight option (system should prompt after uncanceled flight option)

No comment, no obsv.

Time to complete task: 00:03 (learnability from task 5.2)

5.4

Collapses schedule → expands schedule

Clicks schedule drop down-button → delete schedule option → delete option

No comment, no obsv.

Time to complete task: 00:06, 1 error committed

### **task 5 – Discussion**

No comments, no obsv.

### **Last impressions**

“simpelt, nej simpelt ska jag inte säga, inte svårarbetat program”

“jag är inte rädd, [jag] klickar gärna”

The renew option/button in some dialog windows/boxes caused confusion throughout the test.

### **Debrief**

FP wants to be able to search/find specific flights.

Label design was good “unassign är ord som förekommer i den här världen”

FP would like the top bar of the active schedule to be highlighted (i.e blue).

FP mentions that it might be nice to have some sort of status bar where information regarding current flight (hovered flight) could be displayed.

FP wants to be able to navigate using keyboard arrows.

FP also mentions short-cuts for jumping to the next or the previous flight for a specific aircraft/tail.

FP thinks the schedules button has a functional and natural design.

FP didn't experience any particular moments of irritation

Playground: “dra in verkligheten [i playground] .. trolld för att få ihop den här skiten och sen ladda in lösningen i ett aktivt schema”

FP describes her wish of colour mapping the different types of information in the schedule area.

All available information regarding flights must be visualized in association with the flights

FP expresses an idea regarding flight info being shown in a separate window when hovering the mouse over a flight.

### **Observers thoughts**

More of the following:

- Short keys to frequently used options
- single click actions
- re-design the renew function/option/button

## **10.9 Results from Validation test**

### **First impressions**

TS has the impression that the GUI is more grey than when conducting the previous tests.

### **Tasks**

#### **Scenario 1**

##### 1.1

schedule button → new schedule → names schedule “Therese” → checks active box → save button

TS is somewhat confused since the schedule is not displayed when created. She checks the box in the schedule drop down menu and resolves this issue on her own. This is done after task, by definition, is completed.

The notion of an “active/non-active” schedule is confusing to TS

Time to complete task is prolonged due to typing errors committed, these are not recognized as test errors committed.

Time to complete task: 00:37

##### 1.2

clicks “+”-button of an aircraft in a different schedule → cancel button

clicks edit-mode button → clicks drop down-menu button → adds three aircrafts → clicks activate

expands schedule → clicks edit-mode button → clicks delete aircraft button → clicks activate

When TS added aircrafts the schedule was collapsed and the changes made was not visible. When TS expands the schedule she finds that she has added three aircrafts and corrects this error.

Time to complete task includes TS rectifying the error committed.

Time to complete task: 1:50, 2 errors committed

##### 1.3

collapses schedule → expands schedule

clicks edit-mode button → clicks drop down menu button → adds the tail → clicks activate

No comment, no observation.

Time to complete task: 00:22, 1 error committed

1.4

clicks “+”-button for specified aircraft → types ADES-code in flight no-field →types ADEP-code in ADEP-field → specifies STD → clicks cancel button

clicks “+”-button for specified aircraft → types ADES-code in flight no-field →types ADEP-code in ADEP-field→ specifies STD → checks different options → clicks cancel

unable to proceed → TL provides guidance → TS specifies data in correct fields → clicks save-button

TS aborts task after 03:05, she can not add the flight since no ADEP has been specified. After TL informs TS that ADEP is specified in the wrong field TS completes the task.

Time to complete task: 03:20, 2 errors committed

### **Discussion – Scenario 1**

TS has a hard time finding the dropdown menu for the schedule after entering editmode, she exclaims: “Dääääääääär va den” when it is found.

When entering edit-mode the schedule will not automatically expand, this should be implemented.

The “+”-button should only be visible in edit-mode.

### **Scenario 2**

2.1

clicks edit-mode menu → clicks cancel button

clicks drop down menu → clicks delete schedule → clicks prompt popup-window delete button

No comment, no obsv.

Time to complete: 00:26, 1 error committed

2.2

clicks schedule menu button → checks specified schedule

No comment, no obsv.

Time to complete: 00:04

2.3

clicks edit-mode button → clicks dropdown menu button → clicks edit schedule option → enters new name in name-field → clicks save-button

Time to complete affected by type-o, not recognized as error.

Time to complete: 00:15

### **Discussion – Scenario 2**

No comment, no obsv.

### **Scenario 3**

3.1 + 3.2

right clicks flight

clicks “+”-button for aircraft → clicks cancel button

clicks edit-mode button → clicks flight “TEST1” → enters new STD in STD-field → clicks save-button → clicks edit-button → clicks discard button

clicks edit-mode button → clicks flight “TEST1” → enters new STD in STD-field → clicks save-button

clicks flight “TEST4” → enters new flight no in flight-no-field → clicks save-button → clicks activate button

TS is uncertain if she has edited the flight, added a new flight or deleted the flight.

TS is confused after clicking save-button in edit popup-window if this constitutes as saving the changes to the schedule, which it does not.

Time to complete 03:52, 4 errors committed

### **Discussion – Scenario 3**

TS does not like the label “Activate”, suggests “save” or something similar.

The concept of the ghost flight object from the state the flight was in before a change is not clear at first. This is compounded by the fact that TS chooses to merge tasks 1 and 2.

### **Scenario 4**

4.1

clicks aircraft

clicks edit-mode button → clicks delete aircraft button → toggles delete aircraft button → clicks activates button

TS toggles the delete aircraft button because she feels that this will confirm the delete action.

Time to complete: 00:48, 1 error committed

### **Discussion – Scenario 4**

No comment, no obsv.

### **Scenario 5**

5.1

right clicks the specified flight object

clicks the “+”-button for the aircraft that has the specified flight

clicks edit-mode button for the schedule with specified flight → clicks specified flight object → clicks delete button → clicks activate button

Time to complete: 00:14, 2 errors committed

5.2

clicks the specified flight → remarks on that the “Flight plan”-box is coloured orange

TS abandons the task after 00:25, and is unable to complete it.

TL shows TS that she has the option to click the “Flight Plan”-label and box, which expands the popup-window and here additional information is displayed.

Time to complete task: -

5.3

clicks specified flight → clicks “cancel flight”-button → confirms action when prompted → clicks edit-mode button → clicks activate button

There is a known bug in the system that does not update the graphics of the flight object when the flight is cancelled, there is therefore no visual feedback from the action.

Time to complete:00:12, 1 error committed

5.4

clicks specified flight → clicks “uncancel flight”-button → confirms action when prompted → clicks edit-mode button → clicks activate button

Time to complete:00:04, 1 error committed

### **Discussion – Scenario 5**

5.3 & 5.4 Due to the graphical bug not providing feedback until the view is altered or some other action is taken, we feel that the extra action of entering edit-mode and pressing the activate-button is done by TS. This is the behaviour that she has learned, during the previous task, that changes made requires activation to take effect. TS does not distinguish between being in edit-mode and view-mode. If the task in scenario 5 would have been sequenced before the tasks regarding edit-mode, or if the interface provided direct feedback the TS might not have made the errors in these tasks.

### **Scenario 6 (extra scenario)**

6.1

clicks edit-mode button → right clicks schedule area → clicks aircraft/tail object → clicks arbitrary flight object

clicks edit-mode button (exiting edit-mode) → clicks arbitrary flight object → clicks edit-mode button

clicks schedule area → fills in required text fields → clicks the add-button → clicks activate-button

Time to complete: 01:38, 4 errors committed

6.2

clicks edit-mode button → drags and drops flight → clicks activate-button

No comment, no obsv.

Time to complete: 00:02

6.3

clicks flight object → clicks “edit flight”-button → clicks edit-mode button (exiting edit-mode)

clicks edit-mode button → drag and drops flight to another aircraft/tail → clicks activate-button

No comment, no obsv.

Time to complete: 00:26, 1 error committed

## Discussion – Scenario 6

6.1 & 6.2: These features have a new design unaccustomed to the TS. She has, through this and the previous evaluation test, grown accustomed to the dialogue window approach where changes require her to fill out text fields. The more hands on approach was therefore new and required trial and error to figure out, but was mastered more or less at once after having been figured out.

6.3: The hands on approach was at this time known to the TS, but the notion of edit-mode versus view mode was not. As soon as TS knew that moving flight objects is done while in edit-mode she mastered the approach at once.

### Debrief

TS felt that the use of colour had improved since the last test. The feedback received from the colour coding greatly increased the visualization of the information vital for the user.

On the point of menu design TS is accustomed to the approach of right clicking objects to display the options menu. TS feels that our design of menus in the form of panels is not lacking, the use of either approach is controlled by habit.

We ask TS which approach of adding flights she preferred, the “+”-button adjacent to the aircraft/tail or clicking the schedule area. TS preferred the “+”-button since this displays the old “add flight” popup-window. TS is of the opinion that the “add flight”-panel, that is displayed when clicking the schedule area, has an incomplete feel and is not as “real” as the old popup window. It lacks authenticity and when using it TS feels that she is not actually making changes to the schedule.

Quote: “den rutan är grå, det känns inte som om den är verklig, inte som att det gäller”

We ask TS for a comparison between the design of info-panel and edit-panel. TS states that if edit-panel had a design similar to info-panel, TS would be more inclined to use it. TS reasons that there is a closer parallel between the old popupwindow, with its design, containing a header, labels and icons, and info-panel with its flight component, appropriate mapping and dividers. Furthermore TS likes that the labels of the buttons in info-panel clearly explain their action.

TS remarked that the service needs more unambiguous, consistent labels on buttons. An example TS gives is the confusion created with “cancel”- and “cancel flight”-button, as well as the “Activate”-button suggesting this be renamed to “Save changes” or simply “Save”.

We discuss the concept of edit-mode instead of the planing area concept with TS. The notion of the schedule having two modes, information- and edit-mode, was not apparent to TS during the test but after a brief explanation she grasps the concept. TS is used with working with tasks in a planing area and then exporting the resulting schedule into an active real-time view, but sees the benefits with different types of users being able to use the same service with the edit/info-mode.

Lastly TS adds a suggestion to the flight drag-and-drop feature. She wants a realtime update on the STD- and STA times in association with the upper left- and right corners of the manipulated flight object. TS points out that this will improve the feedback of the service and with this help the user accurately move a flight to another point in time.