Evaluating a Practical Decision Support System from a Usability Perspective

Steve Dillan, Anna Prenzel and Georg Ringwelski

Hochschule Zittau/Görlitz, Brückenstr.1 , 02826 Görlitz, Germany
sistdill@stud.hs-zigr.de, sianpren@stud.hs-zigr.de, gringwelski@hs-zigr.de

Abstract: If optimization software creates better solutions than any other available product it doesn't imply that users are willing to use this product. The user has to be part of the software development process to guarantee the acceptance of the final product. This paper describes our approach to achieve this critical success factor of an industrially used decision support system.

1 Introduction

Background. In this work-in-progress paper we describe the requirements- and usability engineering processes of a software product for companies operating in large areas such as water, power or telecommunication suppliers. The functionality of the product includes scheduling vehicles while minimizing driving costs subject to:

- given resources and tasks with time windows: the usual VRP constraints [CDD+01]
- a predictive consideration of repetitive tasks
- a partial order of tasks: e.g. $task\ A$ has to be finished before $task\ F$ starts but after $task\ N$ is completed
- lunch breaks: within a certain time window there has to be a lunch break of certain duration. A vehicle must neither perform any task nor travel during its lunch break.
- further customer specific issues which are too specific to be implemented in the product and therefore need user interaction: e.g. adjustment of the original schedule due to a vehicle breakdown, traffic jam or unexpected new tasks

Problem. Even if all requirements a customer specified for a software product are satisfied it is not clear, whether it will provide the expected added value. In our context this means, that even if we are able to provide optimal solutions to practical VRP, we don't know whether users will be able and willing to use the tool to create them. In particular we can't predict the benefit of various features the product provides. For a successful product it is thus not sufficient to satisfy requirements such as "I can optimize the schedule by pressing a button" which are often asked for by certain stakeholders. This is just a prerequisite

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

for a critical success factor (CSF) namely the acceptance by the user and her intention to improve her own working results by the system. In our current research we search for ways to achieve this CSF in the domain of optimization and decision support systems.

Our idea. User participation is a key aspect of usability engineering [Dah06]. Therefore customers and their employees respectively are interviewed as two separate groups of stakeholders. Based on these interviews use cases are created. To evaluate the quality of the resulting software product the users are considered again. We implement this on the basis of usability tests in which not only the usability, but also the outcome of the user-interaction is quantified. Since customers cannot be expected to provide us access to their employees on a large scale, we have to run the tests with non-expert users we recruit ourselves. These users will perform a predefined set of tasks with our system. To evaluate the benefit of the functionality we provide we offer features to groups of testers and hide them from other groups. Available features for the interactive creation of schedules are:

- automatic optimization
- display the scope of action for a selected task — based on the domain of it's start time
- display travelling times dynamically while the user moves a task over the chart
- pin a task (to fix a task to a certain time and vehicle)
- group tasks (to perform actions such as optimization, remove from plan to a set of tasks)
- display costs dynamically. We consider two metrics: accumulated driving time and standard deviation of workload among different vehicles
- undo/redo

Using these features our test persons are asked to create valid and high quality schedules in a two dimensional chart.

Contribution. In this paper we reflect on a usability- and requirements engineering process and its artifacts for the creation of a practical decision support system. We describe some of the requirements specified by real customers and describe the setup of evaluating the effect of the above mentioned features.

2 Requirements

In a standard requirements engineering process [Sch02] it is important to identify the various sources that have knowledge of the problem or are able to provide requirements on the software. These sources may be statutes, regulations or standards as well as the currently used system or competing products. But furthermore every involved person (stakeholder) is a potential source of requirements.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

There can be a large number of stakeholders and it may be impossible to take each into account individually. Therefore representatives of various interest groups are chosen for interviews. But it is important to consider every interest group since these groups may have conflicting expectations as well as exclusive knowledge of certain aspects. In a VRP system such representatives could be:

- a manager that wants to save money due to efficient tours
- a maintenance engineer that wants sufficient lunch breaks and a manageable workload
- a team leader who wants adjustable schedules

People usually prefer real-life examples to abstract descriptions [Som07]. And often this is how stakeholders articulate their requirements. Based on the gathered information use cases can be created. Use cases itemize the functionality of the software product. They describe what will be implemented.
Both usability and functionality are different kinds of requirements. The stakeholders usually neither don't know nor are interested in any classification of requirements. For that reason there may not be a distinction between functionality and usability in the use cases. Furthermore stakeholders usually don't really know what exactly they want. Therefore requirements have to be validated to ensure they define a system the stakeholders want. One way to do that is prototyping (others could be requirements reviews or test-case generation) where users interact with an executable model of the system.

2.1 Use cases in case study

In our project we interviewed several managers and team leaders to gather information on possible ways to use the system in order to enhance the quality of their daily work. This information was captured in use cases which were afterwards verified by the stakeholders. In the following we describe the scenario and stakeholders of some of these use cases.

First scenario. There is an unexpected new task which has to be added to the current schedule. Usually, the user wants the addition of this task to affect the existing schedule as little as possible. Furthermore, he doesn't want the system to make a decision. All he seeks is decision support. That means the user is looking for suggestions and he wants to know his scope of action and the possible consequences of a decision.

Stakeholders are a) the user who wants to keep control and have full freedom of his decision b) the customer who wants to reduce driving while being able to work on all such unexpected tasks

Second scenario. There is a certain number of resources (e.g. maintenance engineers) and tasks that have to be performed within a certain time window. The resources may vary in the qualification required for the several tasks. The resulting schedule should end as soon as possible. But lunch breaks and possible overnight stays have to be considered as well.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Stakeholders are a) the user who wants all tasks finished as soon as possible b) the customer who wants sufficient breaks and considered overnight stays or additional journeys if a tasks takes more than one day to be performed.

Third scenario. Since the duration of a task is basically a speculative value, it is possible that a scheduled task cannot be done because another necessary task hasn't been finished yet. In that case the user wants to alter the schedule if possible. The team that cannot start its task on time should perform a different task in the meantime. Therefore the system should suggest an altered schedule.

Stakeholders are a) the company that wants to increase its productivity b) the customer who wants his work to be done soon by reducing avoidable waiting periods.

Fourth scenario. Usually there are several tasks to be done at various locations. Therefore the order of execution will determine the covered distance. This distance should be as short as possible to save time and money.

Stakeholders are a) the company that wants to increase its productivity b) the user who wants to keep control and have full freedom of his decision by being able break off the computation or alter the schedule manually c) the customer who wants to reduce avoidable travels.

Fifth scenario. Sometimes it is of interest to know when all tasks will be done after a schedule was created. The amount of resources, their qualification as well as the time windows of the several tasks should be considered in this schedule. The overall duration should be as short as possible.

Stakeholder is the user who wants to know when all tasks are processed.

3 Implementing decision support system requirements

In this section we describe general guidelines [TALS11] we considered to implement the general features described in the introduction. In Figure 1 you can see a screenshot of the tool we use for evaluation purposes.

Humans control the process. Our system doesn't replace the decision maker. We let the user keep control and offer decision support. Therefore ***current costs*** of a schedule (displayed as Gantt chart) are always computed and displayed. Furthermore ***travelling times*** are shown dynamically at the mouse curser when the user moves a task over the schedule. By highlighting the background of the Gantt chart as shown in Figure 2 we also display the ***scope of action*** for the user if the holds a task with the mouse.

Effectiveness and efficiency. Since users create schedules manually as before the resulting schedules may be capable of improvement. We implemented a button for ***automatic optimization***. It optimizes the current schedule and moves all tasks from a panel of unplanned tasks to the Gantt charts and thus assigning all tasks to resources and a starting time.
For each scheduled task the user can turn non-deterministic decisions made by himself or

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

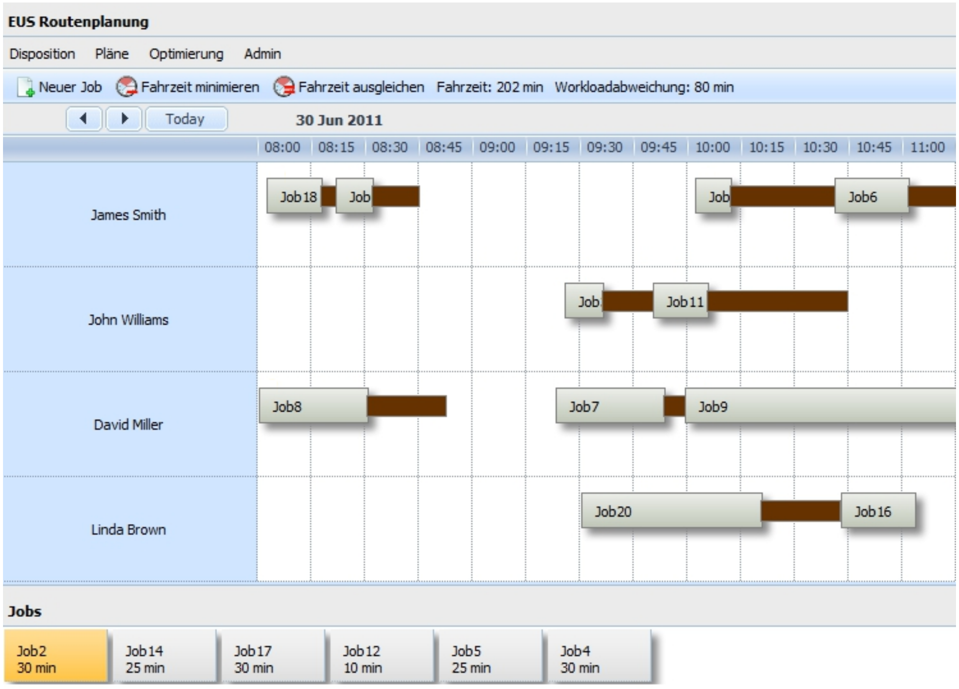

Figure 1: The tool the participants will use — containing the current schedule and a tray with yet unassigned tasks.

the automatic optimizer into constraints by ***pinning*** tasks to positions in the chart (also shown in Figure 2). This way we balance human control and effectiveness.

Semistructured or unstructured problems. In semistructured or unstructured situations decision support should be a combination of human judgement and computerized information. The ***current costs*** and the ***travelling times*** are such kind of computerized information.
The user can also draw rectangles over sets of tasks with the mouse to ***group*** them, by right-clicking he can then perform tasks to each element of the group. It is possible to perform ***optimization*** only on this group of tasks instead of all tasks. It doesn't matter whether this group consists of assigned tasks or of yet unplanned task from the tray.

Interactive, ease of use. Letting the user keep control may not be enough to ensure usability of a decision support system. We tried to make use of graphical possibilities to increase user friendliness and intuitivity. When we're highlighting the background of the Gantt chart to show the ***scope of action*** the user can see: a) red: where constraints prohibit its placement b) yellow: where (non-deterministic) decisions of the placement of other tasks prohibit its placement and c) green: where he can be almost certain that no conflicts will result of placing the task there.
Furthermore ***undo and redo*** are implemented in the usual way.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Figure 2: Scope of action for a new task in an existing schedule with some pinned tasks

Adaptable and flexible. Each of our mentioned features can be enabled and disabled via checkboxes in the tools options window. The tool is designed to satisfy the objective of the user by supporting the completely manual creation of schedules. This is what users have and use today. In our system we add new functionality without replacing what is already in use. Thus the system can be used operationally in the way these users are used to and will thus yield acceptance. We expect that just providing additional features will not lead to a negative attitude of the users to the new system. However, the additional features are provided and users may learn how to use them gradually. On this basis we let the users decide on the way they use their tool.

4 Features evaluation

To evaluate the value of our system we set up usability tests to get quantitative information on the "performance in practice", which depends on different factors:

- the cleverness of the user in decision making and her ability to input these decisions in the system
- the amount of usage of system features for automatic reasoning
- performance of these features, which is the usual quality metric of optimization systems

The participants will be students such that we can create statistically significant data for different peer groups. Unfortunately it is not possible to get real users of the system for these tests due to organizational reasons. All participants will get an introduction to the software including our implementations of the features mentioned in Section 1 of this paper. Afterwards several peer groups will be formed. All participants will have to deal with the same tasks but each group will have a different set of available features. This way the effect of the various features can be determined. In the manner of a regular usability tests each participant will be recorded on video while using the system. Furthermore the participants will make notes of their discoveries or results. The software will save the solution after each test. Based on this information the tests will be analyzed and evaluated to determine:

- the time the user needed to solve a task
- the quality of the achieved solution
- the use of the available features

On this basis we can measure the effect that certain features have on the objectives of the customer. We described these objectives in the paragraphs on the stakeholders in Section 2. As we described at the end of Section 3, we expect that the requirements of the user are satisfied a priori.

4.1 Evaluation in case study

First scenario. The participants will have to solve the following task:
There are three teams and a schedule has been created for each of them. Therefore all teams are assigned to tasks which they're working on right now. At 10:15 am you receive a fault message: The hydrant in the Robert-Bosch-Straße 1 in 02828 Görlitz is out of order. Within 30 minutes one of the three teams must be on the spot and start to repair the hydrant. Determine which changes in the schedule are necessary and note down the resulting activities (e.g. calling employees).
Initial situation:

- three teams are already assigned to various tasks; each team has its list of tasks to be done (the participants will have tables where this tasks are listed)
- the task "Hydrant Bosch Straße 1" hasn't been created yet
- the teams are qualified to take on the task

Second scenario. The participants will have to solve the following task:
Assign all tasks of a given pool of tasks to two teams. The overall duration should be as short as possible. The distances between the tasks don't matter at all.
Consider that there has to be a 30 minutes lunch break for each employee between 11:00

erschienen im Tagungsband der INFORMATIK 2011 weitere Artikel online:
Lecture Notes in Informatics, Band P192 http://informatik2011.de/519.html
ISBN 978-3-88579-286-4

am and 1:30 pm.
Initial situation:

- there is a certain start- and destination point (the participants will have a table that among others contains this location's address and duration)
- team 1 and 2 are selected but there are no tasks assigned yet
- there is a pool of unassigned tasks (as mentioned earlier the participants will have a table containing several locations and durations)

Third scenario. The participants will have to solve the following task:
Team 1 is working on its schedule. The task at 10:00 am required civil engineering to be done first. The team calls to inform you that there will be a delay of 90 minutes. Alter the task and adjust the schedule to fit the new conditions. The task can also be done by a different team.
Initial situation:

- three teams are already assigned to various tasks; each team has its list of tasks to be done (the participants will have tables where this tasks are listed)
- there are unassigned tasks (the participants will have a table containing several locations and durations)
- all teams are qualified to take on the tasks

Fourth scenario. The participants will have to solve the following task:
Assign all tasks of a given pool of tasks to two teams. The cover distances as well as the idle times should be as short as possible. The overall duration or a balanced assignment of tasks don't matter at all.
Consider that there has to be a 30 minutes lunch break for each employee between 11:00 am and 1:30 pm. Start- and destination point is the Furtstraße 3 in Görlitz.
Initial situation:

- there is a certain start- and destination point (the participants will have a table that among others contains this location's address and duration)
- team 1 and 2 are selected but there are no tasks assigned yet
- there is a pool of unassigned tasks (as mentioned earlier the participants will have a table containing several locations and durations)

Fifth scenario. The participants will have to solve the following task:
There are team 1 and team 2 with qualification A and team 3 and team 4 with qualification B. Assign the teams their qualifications. Now assign all tasks of a given pool of tasks to these teams. The overall duration should be as short as possible. The distances between the tasks don't matter at all.
Consider that there has to be a 30 minutes lunch break for each employee between 11:00

erschienen im Tagungsband der INFORMATIK 2011 weitere Artikel online:
Lecture Notes in Informatics, Band P192 http://informatik2011.de/519.html
ISBN 978-3-88579-286-4

am and 1:30 pm. Start- and destination point is the Furtstraße 3 in Görlitz.
Initial situation:

- teams 1 to 4 are shown
- neither of the teams have a qualification assigned
- there is a pool of unassigned tasks (the participants will have a table containing several locations and durations)

5 Conclusion & future work

After these tests we will analyse and evaluate the collected data to draw appropriate conclusions. We expect that there will be a benefit to the users based on the provided features. The available features should increase the usability as well as the quality of the achieved solution. Therefore in comparison with other participants a participant that uses a particular feature should either be able create a better solution or a similar solution in less time and with less effort respectively. If the tests indicate such benefits the mentioned features are likely to be used in the final product.
Of cause it is also entirely possible that features may not have the benefits we strove for. In this case we should take the following questions into consideration:

- Will there be a benefit if the user is an employee instead of a non-expert user?
- Are there any possible adjustments on the feature concerned?
- Which (new) feature could provide the desired benefit?

The obvious thing to do then would be to perform new test — e.g. with employees instead of students, with adjusted features or with new features.
If we recognize a general quality of a developed feature it is conceivable that this feature will be used in similar projects to ensure their usability.

References

[CDD+01] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis. VRP with Time Windows. In *The vehicle routing problem*, pages 157–193. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[Dah06] Markus. Dahm. *Grundlagen der Mensch-Computer Interaktion*. Pearson Studium, München, 2006.

[Sch02] Bruno Schienmann. *Kontinuierliches Anforderungsmanagement: Prozesse, Techniken, Werkzeuge*. Pearson Education, 2002.

erschienen im Tagungsband der INFORMATIK 2011 weitere Artikel online:
Lecture Notes in Informatics, Band P192 http://informatik2011.de/519.html
ISBN 978-3-88579-286-4

[Som07] Ian Sommerville. *Software Engineering*. Addison-Wesley, Harlow, England, 8 edition, 2007.

[TALS11] Efraim Turban, Jay E Aronson, Ting-Peng Liang, and Ramesh Sharda. *Decision Support and Business Intelligence Systems (9th Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2011.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4