Designing an interface for creating report templates in a test management tool using competitive analysis and prototype testing

Malin Persson
malpe323

Linköpings universitet
Linköping
2013-05-19

ISRN: LIU-IDA/KOGVET-G--13/033--SE
Abstract
Test management tools are used to structure and manage test processes in software and hardware development. The purpose of this project was to design a test report template editor for a test management tool with a focus on usability.

Requirements analysis was made using domain analysis and stakeholder interviews. In the design phase, usage scenarios were developed as well as two paper prototypes. A final paper prototype was created based on feedback from the stakeholders. TDR reports is a WYSIWYG editor that provides a set of features that allow users to choose which information to include in their test reports. A basic usability evaluation was performed at the end of the study that indicated that the initial requirements were fulfilled.
Preface
I would like to express my great appreciation to the employees at Syntronic for spending time and resources for finding this project for me and for being helpful during my work with it. I especially want to thank my supervisor at Syntronic, Malin Seeger, for teaching me most of what I know about TDR, for always being willing to help when I had questions and for giving feedback on every detail in my prototypes. Jonas Karlsson at Syntronic has also provided great help with practical and technical questions.

Advice and feedback given by my supervisor Magnus Bång at Linköping University have been very appreciated during the whole project.

I would also like to thank Torsten Stolpmann at Verit informationssysteme and Adam Sandman at Inflectra for being helpful, polite and providing me with information about their products via e-mail.

Finally, I would like to thank Sebastian Jansson for proofreading this thesis; giving feedback and helping me improve the language.

Malin Persson
malpe323@student.liu.se
List of contents

List of figures ...................................................................................................................... 6
1 Introduction ..................................................................................................................... 1
  1.1 Purpose ...................................................................................................................... 1
  1.2 Limitations ............................................................................................................... 1
2 Background .................................................................................................................... 2
  2.1 Different approaches to design problems ................................................................. 2
  2.2 Competing test management tools ............................................................................ 2
    2.2.1 Microsoft SQL Server Report Builder (used together with Team Foundation
            Server) .................................................................................................................... 3
    2.2.2 HP LoadRunner .................................................................................................. 3
    2.2.3 SilkCentral Test Manager ................................................................................... 4
    2.2.4 Klaros-testmanagement ..................................................................................... 4
    2.2.5 SpiraTest ............................................................................................................ 5
    2.2.6 Summary of the products .................................................................................... 5
3 Method ............................................................................................................................ 6
  3.1 Requirements elicitation ........................................................................................... 7
  3.2 Domain analysis ......................................................................................................... 7
  3.3 Design process .......................................................................................................... 8
    3.3.1 Creating and testing the two early prototypes ...................................................... 9
    3.3.2 Feedback on the two early prototypes ................................................................ 12
    3.3.3 Creating the final prototype ............................................................................... 13
    3.3.4 Usability test ..................................................................................................... 13
4 Results ............................................................................................................................ 15
  4.1 Requirements and features ....................................................................................... 15
  4.2 TDR reports .............................................................................................................. 16
    4.2.1 Preview of a report template .............................................................................. 16
    4.2.2 Create a new report template ............................................................................ 19
    4.2.3 Make changes in an existing report template .................................................... 26
    4.2.4 Create a new report template based on an old one .......................................... 27
    4.2.5 Save a draft ...................................................................................................... 28
  4.3 Results from the usability test ................................................................................... 29
5 Discussion ....................................................................................................................... 31
  5.1 Result ......................................................................................................................... 31
  5.2 Method ....................................................................................................................... 31
  5.3 Ideas for the future .................................................................................................... 32
6 Conclusion ...................................................................................................................... 35
References ......................................................................................................................... 36
Appendix 1: Scenarios ....................................................................................................... 39
Appendix 2: Photos of early prototypes ............................................................................ 44
List of figures

Illustration 1: An overview of the process. ................................................................. 6
Illustration 2: Overview of templates in Prototype 1. For each template you can see its name,  the date when it was created, description, status and type (Activity or SIT). The buttons below are used for editing existing reports templates and creating a new one. .................. 9
Illustration 3: Creating a new report template in Prototype 1. Counters are selected using checkboxes. Beside the names of the counters there are fields and anus for creating formulas, changing the desirable values of the counters and choosing whether the value is a count or an average. ........................................................................................................ 10
Illustration 4: Overview of templates in Prototype 2. The left column contains a list of all the existing report templates and whether they are customized for activity reports or SIT reports. The right column shows a preview of template G13A, i.e. what it will look like when the template is used for a report. ................................................................................................. 11
Illustration 5: Creating a new report template in Prototype 2. A report template is created in a separate window, where you can name the prototype and select counters to include by starting to write their names. .......................................................................................... 12
Illustration 6: Overview of report templates in TDR Reports. (The template overview, in the middle, is new and a result of this project. The top and left menus are used to navigate the application and was already part of TDR before the start of this project.) ............ 17
Illustration 7: A preview of a report template with example data ...................................... 18
Illustration 8: New template button .............................................................................. 19
Illustration 9: Creating a new template in TDR. In this illustration, the main header “QSI”, the level 2 heading “Circuit switched” and the level 3 header “QSI CS” are maximized .... 20
Illustration 10: This is what it looks like when you search for a counter a get a suggestion. In the right column you can see three counters that you have already added. You can click on the trashcan to remove them from the template .......................................................... 21
Illustration 11: Editing desirable values, formulas and charts. ........................................ 22
Illustration 12: Enter a desirable value of a counter. You can enter an interval (upper and lower limit) or only one value. If you want only one value, check the box "No upper limit". The unit can be set to count or percent. ................................................................. 23
Illustration 13: When you start writing the name of a counter in a formula, you get suggestions. ......................................................................................................................... 24
Illustration 14: When you have written your formula it looks like this. Now you only have to save ......................................................................................................................... 25
Illustration 15: Create a chart. You can click the small left and right arrows to change the order of the bars. To remove a counter from the chart (but not from the template), click the trashcan ................................................................. 26
Illustration 16: Close-up of the overview of templates shown in Illustration 6. Click the pencil icon to edit a template, the paper icon to make a copy of a template, or the trashcan to delete a template ................................................................................................. 27
Illustration 17: The user can also edit a report template from the preview mode, by clicking the Edit button on top of the preview ................................................................. 27
Illustration 18: When you duplicate e.g. a template called G14A, the copy will be called G14A(1). The name can be changed afterwards. ......................................................... 28
Illustration 19: When you create a new report template, you can make it based on an existing template. The new template will then become an identical copy of the existing one, and you can change the things you want to change ........................................... 28
Illustration 20: Drafts cannot yet be used as templates. In the overview, they are written in italic in order to be distinguished from the finished report templates ................................. 29
## 1 Introduction

Software and hardware testing are investigations conducted to provide information about the quality of a product or service. The definition of software and hardware testing is broad, but in short it could be described as validating that a product works as expected and meets the requirements of the user. There are many different testing techniques depending on what you want to test; it could be simply checking that the right thing happens when you click a button, it could be measuring how a system manages many users at the same time and it could be testing the computer performance. Another example of a testing technique is executing the program with the intent of detecting software bugs. Some companies make test reports to present the outcome of their test cases.

Test management tools are used to structure automated tests and manual test processes for software and hardware testing. A test management tool can be used as a single application for managing cases, environments, automates tests, defects and project tasks. They also allow quick access to data analysis and facilitate collaboration and communication with other project teams. Many test management tools also create reports of the test cases. When you create reports, you may want to make different versions of the reports for different employees, so that everyone can see only the information that is relevant to them. For this, you can use report templates, i.e. templates for different ways of showing data from a test case.

Test Data Repository (TDR) is a test management tool developed by the company Syntronic, a consulting company that provides services within electronics, electro-mechanics, technical and administrative software. Syntronic provide both internal and external consulting. TDR is an internal consulting project, i.e., it is developed at Syntronic’s office and not at the customer’s. TDR is developed and optimized for one customer, a company in the telecom industry that collects test data from their base stations.

### 1.1 Purpose

The purpose of this thesis is to design the user interface for the functionality of the test management tool TDR where a user can create and edit report templates, and to analyze the design process. The goal of the design process is to create an interface with high usability.

### 1.2 Limitations

TDR is a large and complex test management tool with many functions, one of which is to create report templates. There are two kinds of reports in TDR: Activity reports, which are reports from a single test case, and SIT reports, which merges several activity reports to show the test results over time. In the prototype designed in this project, focus is on creation of activity report templates.
2 Background

Syntronic have developed a browser based test management tool called Test Data Repository (TDR) for a customer in the telecom industry. The customer has recently started using it, but some features they want are still missing. One of them is a usable interface for creating custom report templates. One reason why you might want to create different report templates is that different people need different information, and you may want everyone only to have the necessary information. Today the customer creates report templates by editing XML files and there is only one person doing this. The customer has expressed a desire for a usable interface that allows them to create report templates directly in the web application. This could make the task easier and possible to be performed by more employees.

In order to make a test, the customer names the test case. They choose parameters; what they want to test and how long to wait between the tests. Then they press a button, and the data from the test goes automatically into the TDR database. There are two kinds of reports: activity reports, which are reports from a single test, and SIT reports, that is a merge of several activity reports that shows the test results over time. An activity report is generated when you click an activity to view it. A SIT report is created manually from several activity reports.

2.1 Different approaches to design problems

TDR is not the only product of its kind; there are many different test management tools on the market. Brown (2007, cited in Arvola, Lundberg and Holmlid, 2010) describes a method called competitive analysis, which is used by designers to learn from existing examples and precedent designs. The idea of competitive analysis is to line up different competitors side-by-side to find similarities and differences between them. This reveals some of the expectations from customers who are used to other products, and can also work as an inspiration for the designer as you can compare different existing solutions of the same problem. Some competitive analyses are broad and give a feel for the product landscape; some are narrow and focus on solutions of specific design problems. A common way of presenting a competitive analysis is to use a table with two dimensions, with competitors at the top row and features along the left column.

Another common design method is making interviews with potential users of the product, finding out what their goal would be with the product and what features they need. Data from the interviews can be used to create one or more personas, which are prototypes of users, based on the people you have interviewed. You make different personas for different target groups, and the personas’ problems and goals should match the problems and goals that the target group they represent typically have. Starting from the personas, the designer can write some different scenarios and finally make design solutions that make these scenarios possible (Goodwin, 2009).

2.2 Competing test management tools

In this thesis, competitive analysis is used to see different companies’ solutions of allowing the user to create report templates in a test management tool. The intent is to summarize what features the competing products usually has and does not have, and to get inspiration from different existing designs. It is rather narrow, as it focuses only of the functionality of creating report templates and includes a selection of specific features.
The following chapters describe five different test management tools and their features for creating report templates. The tools included are SQL Server Report Builder from Microsoft (used together with Team Foundation Server), HP LoadRunner, SilkCentral Test Manager from Micro Focus International, Klaros-testmanagement from Verit informationssysteme and SpiraTest from Inflectra.

2.2.1 Microsoft SQL Server Report Builder (used together with Team Foundation Server)

Team Foundation Server (TFS) is a test management tool developed by Microsoft. It is intended for collaborative software development projects and provides source control, data collection, reporting and project tracking.

When you start a new project in TFS, you can use different standard reports and add your own custom reports. (Microsoft, 2013b.) To create your own report templates you need another Microsoft program called SQL Server Report Builder (SSRB) (Microsoft, 2013c). The user interface of Report Builder is similar to the well-known Microsoft Office environment (Microsoft, 2013a).

When designing a report in Report Builder, you choose where to get data from, which data to get and how to show the data in the report. You can create tables, matrices and charts to show the data visually.

SSRB provides a feature called auto-generated clickthrough reports; a kind of temporary reports that show user-related data and allow the user to explore data interactively. For clickthrough reports, there are two default templates, one for single instance data (e.g. about a specific customer) and one for multi-instance data (e.g. show all orders for a specific product). These templates cannot be modified, but you can supplement or replace clickthrough reports with predefined reports that has the style and layout you want. (Microsoft, 2013d.)

You can easily create and change formulas in your report. For example, you can get the sum of the values in some different fields (Microsoft, 2013e), (Microsoft, 2013f). The interface for creating and changing formulas in SSRB looks like in Illustration 2.

You can create charts and format them in many different ways, for example you can change the chart type, the colors, the data points, the axis labels and legends; e.g. change the order and text of legends (Microsoft, 2013g; Microsoft, 2013h; Microsoft, 2013i).

2.2.2 HP LoadRunner

HP LoadRunner is, according to their marketing department, the industry standard for application performance testing (Hewlett-Packard, 2013). It is used for examining system behavior and performance. With HP LoadRunner you can simulate thousands of users to test how the system manages many users at the same time. You can collect information from key infrastructure components like web servers or databases.

The report system of HP LoadRunner provides fully customizable report templates where you can modify the content and format of the report (Hewlett-Packard, 2010).
You can add, modify, import, export, or duplicate a report template. The templates are saved as XML files (Hewlett-Packard, 2012).

There is an overview of all report templates, and you can also create your own, delete, export, import and duplicate a report template. When creating a report template in HP LoadRunner you fill out the same form as when creating a report, but you leave out the information about the author. You can choose from a list, with help from checkboxes, which counters to include in the report. You can customize the colors, fonts and other layout properties as you like.

2.2.3 SilkCentral Test Manager

SilkCentral Test manager is a test management solution developed by Micro Focus International. Its purpose is to promote effective collaboration between teams, processes and tools (Micro Focus, 2012).

To create a new report template in SilkCentral Test Manager, you download an existing report in Excel format, modify it in Excel, and finally upload the modified file to SilkCentral Test Manager again.

SilkCentral Test Manager allows you to choose which parameters to include with help from SQL (Micro Focus n.d.). Using Excel you can also create charts, change the name and order of counters in a chart and also create and change formulas for different parameters. It is possible to share report templates with your colleagues.

2.2.4 Klaros-testmanagement

Klaros-testmanagement is an Ajax-based web application developed by the German company Verit informationssyteme. The key areas of Klaros-testmanagement are gathering, managing and evaluating all relevant data of a test process (Verit informationssyteme, n.d.). On their company web site there is an online demo easily available for anyone who wants to try the system (Verit informationssyteme, 2009-2013).

A report template in Klaros-testmanagement consists of three different parts: A script (in Groovy or Java) which gathers data from a database, an XML based rendering template describing the document structure and layout which is used to render values from your value context, and, finally, a set of optional parameters entered by the user (Stolpmann, 2013).

Thanks to the online demo that Verit informationssyteme provide, you can easily examine the features of the reporting part. The start page of the report template part consists of an intuitive overview of all report templates. Next to every listed template there are three small icons for changing, deleting or exporting the template as a PDF. You can also create new report templates. There is a text field for adding parameters to a template, but most other changes requires knowledge of Groovy/Java and XML. It is possible to make a new report template based on an old one by duplicating the content of an old template (Verit informationssyteme, 2009-2013).
2.2.5 SpiraTest

According to the web site of Inflectra Corporation, SpiraTest “provides a complete Quality Assurance solution that manages requirements, tests, bugs and issues in one environment” (Inflectra, 2006-2013a).

The report templates in SpiraTest are XML based and can consist of multiple sections. The WYSIWYG editor allows you to add custom headers and footers to the entire report and to each section. WYSIWYG stands for What You See Is What You Get, and in this case it means that the editor looks like the report will look.

Using a templating language called XSLT, you can define the layout and style of the report (Inflectra, 2006-2013b). With XSLT you can also add logic to the reports, for example you can choose which counters to include in the report or to make simple mathematical formulas. Editing, deleting and creating new templates is possible, and so is creating new templates based on old ones. You can also make graphs. SpiraTest does not include embedded charts in the report, but you can create charts on a separate dashboard, which is also customizable (Sandman, 2013).

2.2.6 Summary of the products

All of the above products supports choosing which counters you want to show in a report. All of them can create charts, although some of them require external tools like Excel for that. All of the products also allow you to edit the colors and layout of the report. All systems also support sharing templates with your colleagues, with the possible exception of SQL Server Report Builder where there wasn’t any clear information on this on their web page.

Most of the systems require knowledge of some language like SQL, XSLT or XML. Here HP LoadRunner has an advantage over the other products. All functions for making report templates in HP LoadRunner, found via web searching, can be performed through a simple graphical user interface.
3 Method

The project started with a domain analysis, i.e. finding information about how TDR works right now and how competing products work. The features of different test management tools were compared and summarized in a table (shown in the appendix). The next step of the project was to decide how important each of these features is for TDR, prioritize them and, based on this evaluation, draw two different early prototypes. After getting feedback on the two early prototypes, the next task was to draw a final prototype and evaluate it through a usability test. Illustration 1 shows a brief overview of the work process.

![Illustration 1: An overview of the process.](image)

Saffer (2007) describes four approaches to interaction design, of which two are called User-centered design and Activity-centered design. User-centered design starts from the idea that the user knows best and you should study the user to make the best design. When designing, the goal of the user is in focus. Activity-centered design focuses on an activity rather than a user. The goal of the design process is that the user should be able to perform certain activities with the product, but you do not focus as much on what goals the user can approach performing these activities. Saffer (2007) gives an example of activity-centered design: When using a rake, the gardener’s goal might be to get a clean garden, but the rake is simply designed for one activity: collecting leaves. When designing from an activity-centered perspective, you can take advantage of affordances, i.e. visual hints of what something is used for (Goodwin, 2009). For example, a protruding handle looks like it should be pulled and a three-dimensional look of an element of pixels indicates that it is a clickable button. When you know what functions your product should have, affordances can facilitate the user’s understanding of how to use it.

This project can be said to use a merge of user-centered design and activity-centered design. The usability test in the end of the process is typical for user-centered design, but a great part of the rest of the design process starts from a collection of activities that the user should be able to perform. Many of these activities are requests that Syntonic previously have received from the customer, though, so you could say the method has a user-centered perspective too.
3.1 Requirements elicitation

The first step of the design project, to get a good understanding of what kind of product TDR is and whom it is for, was to make stakeholder interviews with four different employees at Syntronic who have worked with developing TDR. The approach and questions of the stakeholder interviews followed the example of the book *Designing for the Digital Age* (Goodwin, 2009), which means that we interviewed the stakeholders one-by-one and in person. The benefit of interviewing each stakeholder separately is that different people often have different perspectives and opinions about the product or service to be designed. As a designer you want to hear it from everyone’s point of view. The questions asked to the stakeholders were partly about the stakeholders themselves, what they had worked with before and what their role were in this project, and partly about TDR, what kind of product it is and for whom. There were also questions about the interviewees’ expectations and concerns of TDR and what the product will mean to Syntronic as a company. Finally there were some technical questions since most of the stakeholders involved were developers.

All stakeholder interviews took place at Syntronic’s office in Linköping. The interviews were recorded and later listened to while thorough notes were taken. According to Goodwin (2009) it is preferable to be two people interviewing, one leading the discussion while the other one is taking notes. But since only one person made this project, the interviews were recorded instead. This made it possible to take even more thorough notes, as you can listen to an interview several times.

Reading Syntronic’s documentation and demos about TDR and navigating around in the application to examine its existing functionality was an additional important part of the information gathering process. Examining the database tables of TDR and the design of existing reports gave a good grasp of what is possible to do with TDR given the technical structure of the system. Malin Seeger at Syntronic, who have had the most contact with the customer, also provided a lot of helpful information about the customer’s needs, in addition to her stakeholder interview.

3.2 Domain analysis

When having a good grasp of what TDR is and whom it is for, it was time to make a competitive analysis of different existing test management tools. Chapter 2.2 contains a brief description of five different test management tools. These five tools were analyzed and compared to each other to see which features they have and to get inspired from their design. All the above-mentioned test management tools support the function of creating custom report templates, but most test management tools do not. Therefore, finding products where you can make custom report templates required quite a lot of searching. Table 1 summarizes some of the features of all five tools. It gives an overview of all the tools and what features they have and do not have. The choice of features to compare was based partly on the requests of TDR from the customer, and partly on features that most of the competing products happened to have.

Most of the information about the products was found by searching the web and by e-mailing the companies who have developed them. The Wikipedia page about test management tools includes a long list that was useful in the process. The main method for finding report template tools was to search for each item in the Wikipedia list, and then reading the results briefly to see which of them seemed to support the creation of report templates. E-mails were sent to the companies asking about their products’ features. The companies who answered the
e-mails were Verit informationssysteme who developed Klaros-testmanagement, and Inflectra who developed SpiraTest. The information about the rest of the tools is based on what could be found on the web.

<table>
<thead>
<tr>
<th>Change existing report templates</th>
<th>SQL Server Report Builder +TFS</th>
<th>HP Loadrunner</th>
<th>SilkCentral Test Manager +Excel</th>
<th>Klaros-Test Management *</th>
<th>SpiraTest *</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO, at least not the clickthrough reports. Others unclear.</td>
<td>YES</td>
<td>?</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

| Delete report templates | ? | YES | ? | YES | YES |

| Create new report template based on an old one | ? | YES | SORT OF, you can create a new template based on an old report. | SORT OF, you can duplicate the contents of old templates | YES |

| Choose which parameters to include in report | YES, with SQL | YES, in a form with checkboxes | PROBABLY, with SQL | YES (and create new ones) | YES, with XSLT language |

| Create charts | YES | YES | YES, in Excel | YES, with XML | YES, but not embedded. There is a separate chart dashboard. |

| Add and delete counters from a chart | ? | ? | YES, in Excel | YES, with XML | YES, you can change which fields to include in the graph. |

| Change name and order of the counters in a chart | YES | ? | YES, in Excel | YES, with XML | UNCLEAR about order. You can change names on custom fields but not standard fields. |

| Create and change formulas | YES | ? | YES, in Excel | YES, with Groovy or Java | YES, with XSLT language |

| Change "desirable values" | ? | ? | ? | YES, by adding a user defined parameter in the code. | YES |

| See an overview of all report templates | ? | YES | Only when you create a report and choose template. | YES | Only as XML files in a folder. |

| Choose colors and layout | YES, you can choose between some different layout templates. | YES, you can edit however you like. | YES, a little. | YES, with XML | YES |

| Share templates with colleagues | ? | YES | YES | YES | YES |

Table 1: A comparison of five existing test management tools. The asterisk (*) means that the information about this product was provided to me by e-mail replies from the company. The information about the other products is based on what could be found on the web.
3.3 Design process

Based on what capabilities were supported by the TDR database and Syntronic’s knowledge about the needs of the customer, some different user scenarios were written. The scenarios were used as inspiration when designing, and later as tasks in the usability testing. The scenarios were written in Swedish, which felt natural, as the people who took part in the usability test are Swedish. In the appendix you can see the scenarios both in Swedish and translated to English.

Based on the scenarios, two early prototypes were drawn by hand. Two Syntronic employees gave feedback on both of them. Afterwards, a final design solution was drawn on the computer, and a usability test was performed on three different employees at Syntronic.

3.3.1 Creating and testing the two early prototypes

In this chapter I will describe the two early prototypes and show a few pictures of them. More pictures can be found in the Appendix.

To create report templates in the prototypes that were produced, you have to log in to the already existing settings part of TDR. In the leftmost menu of the settings part there are (since before) buttons for navigating between all the different settings functionalities. Here another button was added, saying “Report templates”, which will link to this new report templates functionality. When clicking the “Report templates” button, the first you will see is an overview of all existing report templates. This applies for both prototypes, but the overviews look different in the different prototypes. The overview of templates in prototype 1 is inspired by the overview of templates in Klaros-testmanagement, which lists all the templates in a kind of table with names and information about them (Illustration 2). This is supposed to provide a quick overview of what report templates you have and some information about them.

Illustration 2: Overview of templates in Prototype 1. For each template you can see its name, the date when it was created, description, status and type (Activity or SIT). The buttons below are used for editing existing reports templates and creating a new one.
The creating and editing of templates in prototype 1 is supposed to be WYSIWYG (i.e. it should look similar to the design of a complete report). The reason why we chose to use a WYSIWYG design is that the user is supposed to know intuitively where to look for what if the interface looks similar to the familiar design of a report. E.g., when you edit the desirable value of a counter, you do it in the corresponding place as where the desirable value is displayed in a report. When you choose which counters to include in the template, you can see the headers, subheadings and counters as they look in a complete report. The headers and subheadings can be minimized and maximized in this view, to not take too much place. The counters, placed under the subheadings, have checkboxes beside them, so you can check the counters you want to include. This is inspired by HP LoadRunner where you also choose counters using checkboxes. It seemed simple and intuitive to use, so we wanted to try this and see what Syntronic thought of this idea for TDR. Illustration 3 shows what it looks like when you create a new report template in Prototype 1.

Illustration 3: Creating a new report template in Prototype 1. Counters are selected using checkboxes. Beside the names of the counters there are fields and anus for creating formulas, changing the desirable values of the counters and choosing whether the value is a count or an average.

The overview of templates in Prototype 2 is inspired by the interface of HP Loadrunner. In the left column there is a list of all existing report templates, displayed with only their names and types (Activity or SIT report). When you click one of them, a preview of that template is shown in the right column (Illustration 4). Before clicking the name of a report template, you can see no information about it apart from name and type, unlike Prototype 1 where the
overview looks like a table and contains more information. We wanted to try both of these concepts, as both give a nice overview of what templates you have. The Prototype 1 overview contains more information, while that of Prototype 2 is more sober and simple, so we thought it would be interesting to see which of them is preferable for TDR.

In prototype 2, new templates are created in a separate window (Illustration 5), unlike Prototype 1 where they are created in the same window as the rest of the application. This difference, too, was an experiment, we wanted Syntronic to decide what they thought would be preferred. When you choose which counters to include in Prototype 2, you do not choose them via checkboxes, instead you start writing the name of a counter in a text field, get suggestions and choose from the suggestions what you want to include. This interface could be useful for a user who knows the name of a counter (or at least what letters it starts with) and want to find it quickly.

Illustration 4: Overview of templates in Prototype 2. The left column contains a list of all the existing report templates and whether they are customized for activity reports or SIT reports. The right column shows a preview of template G13A, i.e. what it will look like when the template is used for a report.
Illustration 5: Creating a new report template in Prototype 2. A report template is created in a separate window, where you can name the prototype and select counters to include by starting to write their names.

3.3.2 Feedback on the two early prototypes

This section explains the most important parts of the feedback from the two Syntronic employees who evaluated the two early prototypes. Everything in this chapter is either their opinion or facts they know from developing TDR and from their contact with the customer.

One problem that applied for both of the two early prototypes was the way of selecting which counters to include. In prototype 1 you could see a list of all counters, listed under their headers, and select counters by checking the checkboxes beside them. According to the Syntronic employees this was not a very good solution, because the list of counters can be very long, sometimes including hundreds of counters. In prototype 2, the counters were not listed under any headers; there was only text field and a field where suggestions came up. You chose counters by starting to write the name of a counter in the text field, get suggestions in the suggestion field and then choose one of the suggested counters. This was a better solution, but also caused problems, as the names of the counters are not necessarily unique. Some counters have the same names but are found under different headers.

When creating charts in both of the early prototypes, you could drag and drop to change the order of the bars. According to the Syntronic employees, this will be difficult to implement, so they recommended making a different solution.
When you create or edit a report template, the interface looks the same regardless of whether you are creating a new template or editing an existing one. There should be some kind of indicator that shows whether you are creating or editing a template.

### 3.3.3 Creating the final prototype

Based on the two early prototypes and the feedback from Syntronic, a final paper prototype was made. This prototype was created on the computer by copying from screenshots of TDR and drawing in Paint. The prototype was printed, put together and usability tested on three Syntronic employees. In chapter 4.2, the final prototype and its functionalities will be described.

### 3.3.4 Usability test

Three Syntronic employees took part in the usability test, which were performed on the basis of the example described in Goodwin (2009), i.e. the participants were to work through a series of realistic tasks. A task should be based on user goals, be related to issues of importance for product success, have an appropriate scope and a limited and predictable set of possible solutions (Snyder, 2003; cited in Goodwin, 2009). Prior to a usability test, Goodwin (2009) gives the advice to determine whether you are most interested in ease of learning or efficiency of use over time. For example, it is interesting to test whether a passenger can use an airport kiosk at first try, but it is irrelevant to test whether an air traffic controller can use his or her complex system without training.

For this usability test, the focus was on testing whether a person who is familiar to TDR could immediately understand how to create report templates using the prototype. That is, a person who has used TDR before, knows what a report looks like and what counters they are interested of, should not have to learn how to create report templates. They should be able to guess it from the interface, since it does not have to be a very complex task.

One of the tests took place in a group room at Linköping University, and the other two tests were performed in a conference room at Syntronic’s office in Linköping. Both rooms were calm and quiet, light, and with a big table to put the prototypes on. The participants’ task was to use the final paper prototype to do the tasks described in the scenarios mentioned in chapter 0. The sound of the conversation was recorded during the tests, which facilitated the work for the designer. Instead of having to take notes during the test, focus could be kept on managing the prototype and listening to the users. Besides performing the tasks, the users were also allowed to give spontaneous feedback about anything in the prototype.

The usability was measured through a method called binary success, i.e. for each task the user tries to perform, you observe whether he or she succeeds to complete the task or not. Tullis and Albert (2008) describe four different types of “task failures”. These are:

1. Giving up – the participant indicates they would not continue if they were doing this on their own.
2. Moderator “calls it” – the study moderator stops the task because it is clear that the user is not making any progress.
3. Too long – sometimes a task is considered successful only if they can be accomplished within a given time period.
4. Wrong – the user thinks they completed the task successfully, but they did not.
In this usability test, you could say that the most common type of failure was “moderator calls it”. The ambition of this usability test was to improve everything in the interface that was not obvious and easy to understand at first try. A task was therefore considered a failure when it became clear that the user could not intuitively know how to solve the task. Indications of task failure could be:

1. The user tried to solve the task in a way that would not work.
2. The user tried to solve the task in an unnecessarily complex way.
3. The user commented the button or link that was supposed to be used, but did not use it because they thought it was the wrong one.

The usability tests were summed up listening to the recordings while notes were taken. The notes described everything that had come up during the usability test that could be improved; i.e. tasks that had been difficult to perform for the test users and spontaneous feedback given during the tests. The parts of the design that were considered good as they are and not in need of any change were not mentioned in these notes. The reason for only mentioning the things to be changed is that this should give a good overview of what needs to be done in the future. When these notes are handed to Syntronic for future work, they can focus on fixing these things and keep the rest as it is; they will not have to search for the things to change among a list of things that are already good.
4 Results
This chapter presents the features of the system as well as the final prototype named TDR reports. Moreover, the results from the usability testing of TDR reports are presented.

4.1 Requirements and features
The result from the requirement elicitation phase was a set of features needed in the tool. Table 2 shows a selection of features and an estimation of how important each of them is for TDR. The selection of features is the same that were used to compare the different competing test management tools. As you can see in Table 2, each feature has been assigned a value from 1 to 10, telling how important the feature is. These values are based on Syntronic’s knowledge about the customer’s needs, gained from previous contact with them.

The final prototype includes every feature in the table except changing names of counters in a chart and choosing the colors and layout of a report, which were, according to Syntronic, not very important for the customer. The possibility of sharing templates with colleagues does not require a function in TDR, as everyone has access to every template per default. Editing the colors and layout is a function that is available in all of the products that were compared, but despite that, this feature will not be included in the TDR prototype. The reason for not providing color and layout functionality is that the present graphic design of TDR, including the reports, is already very uniform. The colors, layout, and fonts of the reports melt in so well with the style of the rest of TDR, so changing the design of some particular reports would probably only make it look non-uniform and strange.

In the final prototype, all the included functions are available through a simple graphical user interface, i.e. you do not have to use SQL, XML or any other language.
Table 2: Importance of different functions for TDR.

4.2 TDR reports

The final prototype looks like a merge of the two early prototypes. The overview is that of Prototype 2, the one that is inspired by the overview in Klaros-testmanagement. It also has the WYSIWYG view with the extendable menu from prototype 1. The function for finding the counters you want to include in the template is the same as in prototype 2; the user starts writing the name of a counter and gets suggestions. These choices were made based on what the Syntronic employees preferred during the feedback meetings.

To get to the report template page in TDR, you go to “Settings” in the top menu and log in with a username and a password. Then click the button “Report templates” second lowest in the left menu. You will see an overview of all existing report templates, if there are any, and a button for making new report templates (Illustration 6). As you can see, this is the same idea as I Prototype 1.

4.2.1 Preview of a report template

By clicking on the name of a report template (e.g. G12A in Illustration 6), you can see a preview of it. The preview can be shown with or without example data. By showing the example data, you can check that all your formulas are calculated correctly, and also get a
better grasp of what an actual report will look like using your template. Illustration 7 shows a preview of a report template with example data.

Illustration 6: Overview of report templates in TDR Reports. (The template overview, in the middle, is new and a result of this project. The top and left menus are used to navigate the application and was already part of TDR before the start of this project.)
Illustration 7: A preview of a report template with example data.
4.2.2 Create a new report template

To create a new report template, click the “New template” button below the list of templates (Illustration 8).

![New template button](image)

Illustration 8: New template button.

You will see an extendable menu of all headers and subheadings. To add a counter to the report template, maximize the header under which you want the counter to be (Illustration 9). There you will see two fields: a search field where you can search for the name of a counter and a bigger field with a long list of counters. When you start writing the name of a counter in the search field, you get suggestions from the list in the bigger field (Illustration 10). You can double-click a counter to add it, or select it and click the “Add” button. As you might have noticed, this design is a merge of Prototype 1, where you have an extendable menu and choose counter via checkboxes, and Prototype 2, where there is only see a text field for searching the name of a counter, without any heading or subheadings. This is a solution to the problem that counter names are not always unique. When you search for them under a specific heading, only the counter that is under that heading can be found. Another counter with the same name must be under another heading. This design also solves the problem with checkboxes, that the list can be too long. You do not have to read a long list of counters in this interface; instead you start writing the name of the counter you want.
Illustration 9: Creating a new template in TDR. In this illustration, the main header “QSI”, the level 2 heading “Circuit switched” and the level 3 header “QSI CS” are maximized.
Illustration 10: This is what it looks like when you search for a counter and get a suggestion. In the right column, you can see three counters that you have already added. You can click on the trashcan to remove them from the template.

When you are done adding all the counters you want to include in the report template, you click the “Next” button. In this next step, you can add charts, formulas, and enter desirable values of the counters (Illustration 11).
Illustration 11: Editing desirable values, formulas and charts.

Some counters have desirable values, i.e., the value of a certain counter must be within a certain interval to pass a test case. When you create or edit a report template you can change desirable values of counters. To do this, you click the pencil icon under “Desirable value” beside the counter whose values you wish to change. In the window that appears (Illustration 12) you can enter the lowest and highest acceptable value for the counter. If you only want to have one desirable value and not an interval, you can check the box “No upper limit” and enter only the lower limit.
Illustration 12: Enter a desirable value of a counter. You can enter an interval (upper and lower limit) or only one value. If you want only one value, check the box "No upper limit". The unit can be set to count or percent.

Some counters have formulas, that is, their value is calculated from one or more other counters. To enter or edit the formula of a counter, click the pencil icon beside the text field “Formula”. A window like that in Illustration 13 will appear. You enter a formula by writing it in the formula field. When you start writing the name of a counter you want to include in the formula, you get suggestions from a list, just as when you choose a counter to add to the template. Operators or signs in formulas can be written or chosen from the field to the right. Numbers in the formulas are simply written down. Illustration 14 shows what it looks like when you are done entering a formula. This way of writing formulas are meant to be intuitive for the user. You do not have to remember the whole names of the counters to include them in the formula, you only have to know part of them and recognize them when you get the suggestions. You also never have to choose a counter form a long list. Operators and counters can be written via the keyboard directly in the formula field. The intent is to make it as similar as possible to simply writing the formula in a text editor.
Illustration 13: When you start writing the name of a counter in a formula, you get suggestions.
Illustration 14: When you have written your formula it looks like this. Now you only have to save.

For each counter, you can choose whether to show the count or the average. Count is only used for a few counters, and shows how many events of a special kind that have occurred. The average, used for most counters, is used if you have made several measures of the same thing during a test case and want to show the average of the results. The rightmost dropdown menu beside each counter allows you to easily choose whether to show the count or the average of that counter (Illustration 11).

To create a chart of the counters under a certain header, just click the icon “Create chart” at that header. A window like the one in Illustration 15 will appear. You can click the small left and right arrows to change the order of the bars. To remove a counter from the chart (but not from the report template) you can click the trashcan. This design is a little bit less intuitive than the “drag and drop” function in the early prototypes, but this will be easier to implement and is still intuitive. You can guess what the arrows and trashcans do, and there are not really any other functions. When you have created a chart, the “Create chart” button (Illustration 11) will turn yellow and the text will change to “Edit chart”, so the user can easily see where there are charts and not.
Illustration 15: Create a chart. You can click the small left and right arrows to change the order of the bars. To remove a counter from the chart (but not from the template), click the trashcan.

### 4.2.3 Make changes in an existing report template

To make changes in an already existing report template, you can either click the pencil icon beside the template in the overview (Illustration 16), or watch a preview of the template and click the “Edit” button from there (Illustration 17). You will come to the same interface as when you create a new report template, but the information you have entered earlier will be prefilled. You just have to change the information you want to change.
<table>
<thead>
<tr>
<th></th>
<th>In use</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>use</td>
<td>2</td>
<td>Activity</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>SIT</td>
</tr>
<tr>
<td>0</td>
<td>Activity</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SIT</td>
<td></td>
</tr>
</tbody>
</table>

*Illustration 16: Close-up of the overview of templates shown in Illustration 6. Click the pencil icon to edit a template, the paper icon to make a copy of a template, or the trashcan to delete a template.*

*Illustration 17: The user can also edit a report template from the preview mode, by clicking the Edit button on top of the preview.*

### 4.2.4 Create a new report template based on an old one

Sometimes you want to create a new template that is very similar to an already existing one. It might be annoying to have to start over and make a new report template from scratch when this new report template is supposed to be almost identical to one you have already created. Fortunately, you can make a copy of an old report template and then just make a few changes to it. There are two ways of doing this. One way is to use the “Duplicate” icon in the overview (the one that looks like two pieces of paper in Illustration 16). This will create a new template that is identical to the template whose duplicate icon you clicked. The new template immediately appears in the overview of templates, as shown in Illustration 18. You can then rename the new template and make changes in it as you like.
Illustration 18: When you duplicate e.g. a template called G14A, the copy will be called G14A(1). The name can be changed afterwards.

The other way to create a new report template based on an old one is to use the regular “New template” button. From there, you can use the dropdown menu “Based on template” to choose the template you want to start from (Illustration 19). The headers and counters from the chosen template will automatically be added to your new template. You can then make changes as you like.

Illustration 19: When you create a new report template, you can make it based on an existing template. The new template will then become an identical copy of the existing one, and you can change the things you want to change.

4.2.5 Save a draft

If you want to save a template before you are done creating it, but do not want anyone to use it yet, you can save a draft of the template. That means that you save the template as it is, and could open it again to continue creating and editing it, but nobody can use it for a report. When you have saved a template as a draft, the overview will look like in Illustration 20.
Illustration 20: Drafts cannot yet be used as templates. In the overview, they are written in italic in order to be distinguished from the finished report templates.

4.3 Results from the usability test

During the usability test, it was found out that most of the tasks could be accomplished easily, but there are some things in the design that should be changed. This chapter will contain a description of the most important design problems and proposed solutions to them.

Table 3 shows the result of the usability test in terms of binary success. The tasks from the scenarios are listed at the leftmost column and the participants are listed along the top row.

<table>
<thead>
<tr>
<th>Name</th>
<th>Create date</th>
<th>Description</th>
<th>Status</th>
<th>In use</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>G12A</td>
<td>2012-08-01</td>
<td>First draft</td>
<td>Do not use</td>
<td>2</td>
<td>Activity</td>
</tr>
<tr>
<td>G12B</td>
<td>2012-08-02</td>
<td>Changed counter</td>
<td>Active</td>
<td>14</td>
<td>SIT</td>
</tr>
<tr>
<td>G12A</td>
<td>2013-01-23</td>
<td>New counter</td>
<td>Active</td>
<td>0</td>
<td>Activity</td>
</tr>
<tr>
<td>G14A (Draft)</td>
<td>Today</td>
<td>First version</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3: Result from the usability test, measured with binary success. 1 means that the participant succeeded to accomplish the task, 0 means that they did not.

As you can see in Table 3, all participants completed most of the tasks successfully. There are some tasks, though, that seemed a little bit difficult to perform. We are now going to describe some of them more closely.

<table>
<thead>
<tr>
<th>Task</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview a report</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Name the report</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Choose the report type ”Activity report”</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Set description</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Choose headings, subheadings and counters</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Choose ”count” or ”average”</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Save draft</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Go back to main menu while creating template</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Create chart</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Switch places of bars in a chart</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Set desirable value of a counter</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Set formula</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Save template</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Edit formula</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Change desirable value</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Undo last action</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Finding a function in step 2 when you are in step 1 or vice versa</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Create a new template based on an old one</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
When creating or editing a report template there is an “Undo” button for undoing your last action. Some of the test users did not dare to click this, as they thought it would undo everything they had done since last time they saved. This could be made more obvious, e.g. by renaming the button to “Undo last action”.

In the first step of creating report templates you select the counters you want to include. When you are done selecting counters, you should click the “Next” button to come to the page where you can create charts and change the formulas or desirable values of counters. It was not obvious to everyone what would happen when you click the “Next” button. One way to make it more obvious this could be to name these two steps something more describing, e.g. “Select counters to include” and “Set properties of counters”.

The user must be able to add new headers and subheadings. The customers already have a lot of headers and subheadings in their system that they use for the reports, but they may want to add more of them or change the structure of the headers. One could solve this by simply adding links to the interface where you start creating a report template (Illustration 15). You could add links saying “New main header”, “New level 2 header” or “New level 3 header”, placed where the next header would be if there were one. For example, in the present prototype there are two main headers, Events and QSI. Below them you could add a link saying “New main header”, and if you click there you can write the name of your new main header. When you expand the header “QSI” you see three level 2 headers: “Traffic”, “Circuit switched” and “Measurements”. Below them there could be a link saying “New level 2 header”, which you can click to add a header in the same way as the main headers. The same thing could apply for the level 3 headers. You should also be able to remove headers and subheadings from a template.

A larger text field for writing the description could be desirable, in case the user wants to write longer descriptions.

When creating or editing a template, you can click the “Cancel” button to go back to the overview. This was not obvious to the test users; they had difficulties finding how to do this. Renaming the Cancel button to “Back to main menu” or “Back to overview” could probably solve this.
5 Discussion
This chapter will include a discussion of the method used in this project, some thoughts about the result, and some ideas for the future of TDR.

5.1 Result
The result of this project is a paper prototype that, according to the usability test, is mostly usable and intuitive, but on some aspects there is room for improvement. Most of the tasks in the scenarios turned out to be easy to perform, the user could easily understand how to do them, and so they were considered success. Some tasks, though, described in chapter 4.3, were not entirely obvious how to do, but we have discussed solutions to those problems that are also mentioned in chapter 4.3. Since the purpose was to make a usable design for creating report templates in TDR, we therefore think that the project have achieved its purpose well, although not perfectly. On the other hand, you seldom manage to make a perfect design; there is always some room for improvement.

One aspect of the design, the functionality of creating and editing formulas, turned out to be good from the user perspective but can be difficult to implement. The people who took part in the usability test really liked this solution as users and solved the task easily, but one of them pointed out that this might be a complicated solution for the programmer. In the prototype, the system takes input from the keyboard and recognizes whether the user writes a string (and whether it is the name of a counter), a number or an operator sign, and that is not a very trivial thing for a programmer to do. Therefore, it might be better to have an alternative design of that function that would be easier to implement. This idea was discussed with Malin at Syntronic, and there are some early ideas of an alternative solution for formulas. Making the formula functionality design easier to implement will probably make it less optimized for the user, though.

As there was no possibility to test the prototype on the real future user, the reliability of the usability tests can be questioned. Anyway, they gave some good hints about the usability of the prototype.

5.2 Method
The original idea was to make interviews with the customer and to observe them working with TDR, but unfortunately the customer had no time for meetings during the project. To know the users’ needs and goals, the best available option was to start from what Syntronic knew, from earlier contact with the customer, about their work process and requirements.

The lack of contact with the customer was a main reason for the choice of using competitive analysis, as this is something you can do, and get valuable information from, without meeting any users. Competitive analysis still makes it possible to get some good hints of what the user will expect from the design, since the customer is likely to have used or looked up similar products before ordering yours. Maybe you can even guess what users generally need by looking at the design of other products, as the competitors might have made user researches for their designs. And if they have not, you can at least compare different design solutions and get inspired from the ones that you think seem most intuitive. Of course you would not know what is most intuitive to your user, but at least you can see some examples of solutions and make a guess.
A disadvantage of competitive analysis in comparison to user interviews is, that if you design for a very specific target group, as in this case, you can of course not find out your customer’s specific needs. Therefore, I think a combination of competitive analysis and user interviews would have been the best solution for this project. Or maybe only user interviews would have been enough. If you design for a very broad target group, on the other side (which is not the case in this project), it is often better not to make very much user research, because if you do there is a risk of making the design too specific (Saffer, 2007). You may optimize the design for the target groups you have interviewed, without realizing that you might make it more difficult for other people to use it. In such a situation, the best option might be only to make a competitor analysis and to focus on what kind of tasks should be possible to perform with your product. This would be similar to the activity-centered design approach described in chapter 3 (Saffer, 2007) and would result in a very general design. Not an optimal design for anyone, but hopefully good enough for everyone.

Throughout the design process, paper prototypes were used, as it would save time in comparison to making hi-fi prototypes from HTML and JavaScript. This was a good thought, but as the design turned out to contain some rather complex functions, e.g. extendable menus, the prototypes became very ambitious and required creative solutions. Maybe the paper prototypes did not save any time after all, as it takes more time to make changes in such ambitious paper prototypes than to change some rows of code.

Only three people took part in the usability test, and these people were Syntronic employees developing the product, not potential users. The results from the usability test may have been more reliable if it were possible to test it on the customer, as they probably know their needs and problems best. Syntronic employees were considered to be the second best choice, though, as they have had contact with the customer, know some of their needs and also have knowledge of the product. It would have been better if it were possible to include more people in the usability test, as that would give us a grasp of how significant the problems were. If five out of ten people cannot accomplish a task, few people would doubt that it is a problem (Goodwin, 2009). But if one out of three people cannot accomplish a task you cannot be equally sure it is a problem – maybe it is just that person having a bad day. In our usability test, anyway, we chose to consider it a problem if one person failed to solve a task; to be sure we were not ignoring something important.

5.3 Ideas for the future

The next step of the process is to hand everything over to the person who will implement the design, who will also be a student writing his thesis. It will be necessary to communicate to this person both how the prototype works, and what changes he will need to make to it.

During the project, many ideas came up, but within the limits of the course, it was not possible to include every idea in the prototype. There are also some things that are likely to be needed later but were not possible to include right now. Here are some of the things that are not included in the prototype, but that could be done in the future.

- As mentioned in the Limitations chapter (chapter 1.2), the prototype only includes an interface for creating activity report templates and not SIT report templates. This should not be a problem, though, as the design of the SIT reports are very similar to that of the activity reports. Probably you can use exactly the same design for creating SIT reports, maybe excluding a few features.
Discussions have come up about what to do if the user is not sure about the name of a counter, but still wants to include it in a report template. Right now, this will be a problem because you have to search for a counter’s name to find it. As you get suggestions when you start writing the name of a counter, you only have to know part of the name, but what to do if you do not even know part of the name? A proposed solution for this problem is to include a function where you can add a description of each counter to the database. Then you could make it possible for the user to search for keywords from the description of the counter instead of searching for the name. This will be helpful if the user knows what a counter is and what it does, but not its name.

A future suggestion that could improve the usability of TDR even more is evaluating it through Nielsen’s Heuristics for User interface Design (Nielsen, 2005). Heuristic evaluation is a quick and cheap way of evaluating a user interface design and finding its usability problems. The method can be used as part of an iterative design process and involves having a small set of people examine the interface and judge its usability based on some usability principles – the “heuristics” (Nielsen Norman Group, 1998-2013). These heuristics are meant to work as rules of thumb for good usability. The ten most general heuristics are:

1. **Visibility of system status**
   The system should give feedback showing the user what is going on.

2. **Match between system and the real world**
   Use words and concepts familiar to the user rather than system-oriented terms.
   Information should be presented in a natural and logical order.

3. **User control and freedom**
   Support undo and redo functions if the user make mistakes.

4. **Consistency and standards**
   Use the same terms for the same things. Follow platform conventions.

5. **Error prevention**
   Prevent mistakes, e.g. by presenting the user with a confirmation option before committing an action.

6. **Recognition rather than recall**
   Instructions should be visible or easily accessible all the time, so the user does not have to remember information from one part of the dialogue to another.

7. **Flexibility and efficiency of use**
   Use accelerators, such as shortcuts and command keys, to speed up the process for the expert user without making it difficult for the novice user.

8. **Aesthetic and minimalist design**
Eliminate information that is irrelevant, otherwise it will compete with the relevant information and diminish its relative visibility.

9. **Help users recognize, diagnose, and recover from errors**
   Error messages should be expressed in an understandable language and constructively suggest a solution.

10. **Help and documentation**
    It is best to have a system that can be used without documentation, but it may be necessary to have it anyway. Make all such information easy to find, make it focus on the user’s task and list concrete steps to be carried out.
(Nielsen, 2005)

Nielsen’s Heuristics is a good alternative if you need a cheap and quick method of evaluating usability and do not have time or resources to prepare a proper usability test and perform it on people from the right target group. It does not measure the usability as good as e.g. a usability test according to Goodwin’s (2009) example, but it is still better than nothing and it takes little time and effort. It might be a good idea to do a heuristic evaluation on TDR, both on the prototype created in this project and on the entire system. This might improve the usability of the system even more, and it can be done even if Syntronic still cannot have contact with the customer.
6 Conclusion

The purpose of this project was to design a usable interface for creation of custom report templates in a test management tool. Requirements for the design were collected via domain analysis and stakeholder interviews. Two early prototypes were created based on user scenarios. These were discussed with field experts who have participated in the development of the test management tool in question. The feedback from the experts was incorporated in a final prototype based on the earlier prototypes. TDR reports is a WYSIWYG editor that provides a set of features that allow users to choose which information to include in their test reports. A basic usability evaluation was performed at the end of the study that indicated that the initial requirements were fulfilled. Future work for the design of TDR could include validating the system through Nielsen’s heuristics.
References


Appendix 1: Scenarios

Scenarion till TDR-prototyperna

1. Förhandsgranska en rapportmall

Titta på en förhandsgranskning av rapportmallen som heter G12A.

2. Skapa en ny rapportmall

Nu ska du skapa en ny rapportmall. Den ska se ut som på bilden på nästa sida. Dina deluppgifter är följande:

1. Döp rapporten till "G14A".
2. Välj att rapportmallen ska vara en Aktivitetsrapport.
3. I beskrivningen ska det stå "First version".
5. På räknaren CS_TCH står det att antalet (count) ska visas. Det är fel. Ändra det så att medelvärdet (average) visas i stället.
10. Ange en formel för räknaren CS_TCH. Formeln ska vara KPI_DURATION * TCH_TRAFFIC.

3. Ändra i befintlig rapportmall

Du kommer på att du vill ändra några saker i rapportmallen du just skapade.

1. Ändra formeln på parametern CS_TCH så att den i stället blir KPI_DURATION * TCH_TRAFFIC.
2. Ändra "Desirable value" för QSI CS. Här vill du inte sätta ett högsta och lägsta värde, utan bara ett värde. Sätt värdet till 95,00. Sätt enheten till procent.
3. Du ängrade dig! Du kom just på att du inte alls ville ändra "Desirable value" för QSI CS. Vilken tur att det finns en ångra-funktion. Använd den för att ångra din senaste handling.
4. Nu när du gjort dessa ändringar, ändra beskrivningen så att det i stället står "Changed formula".
5. Spara dina ändringar.
4. Skapa en ny rapportmall baserat på en gammal

Nu vill du skapa en **ny** rapportmall som heter G14B och är baserad på G14A. Den ska se ut exakt som G14A, **förutom**:

1. Räknaren `QSI_CS` ska inte vara med.
2. Beskrivningen ska vara "First version" i stället för "Changed formula".

Hitta ett sätt att göra detta **utan** att behöva skapa en helt ny rapportmall från grunden.
## Activity Report

### Report Information
- **Activity:** TDR-Test Activity 99
- **Time:** 12-05-25 15:00 -- 12-05-25 21:00
- **Report created:** 2013-03-06 09:00
- **Status:**

### Events

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Desirable Value</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP_EVENT</td>
<td>0-4</td>
<td>1</td>
</tr>
<tr>
<td>RP_EVENT</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### QoS

#### Traffic

<table>
<thead>
<tr>
<th>Traffic Volume</th>
<th>Desirable Value</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS_TCH</td>
<td>-</td>
<td>1464.00</td>
</tr>
<tr>
<td>CS_SDCCH</td>
<td>-</td>
<td>288.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Duration</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

#### Circuit Switched

<table>
<thead>
<tr>
<th>QoS CS</th>
<th>Desirable Value</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>&gt; 99.00 %</td>
<td>97.41 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Traffic</th>
<th>Desirable Value</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCH TRAFFIC</td>
<td>-</td>
<td>244.00</td>
</tr>
<tr>
<td>SDCCH TRAFFIC</td>
<td>-</td>
<td>46.00</td>
</tr>
</tbody>
</table>

### Measurements

#### CP Restarts (seconds)

![Graph showing CP restarters](image.com)

- **Average:**
  - CP Restart SMALL: 13500.00
  - CP Restart LARGE: 13500.00
Scenarios for the TDR prototypes

1. Preview a report template

Watch a preview of the report template named G12A.

2. Create a new report template

Create a new report template. It should look like the picture on the next side. Your subtasks are following:

1. Name the report template "G14A".
2. Select the report to be an Activity report.
3. The description should say "First version".
4. Choose which headers, subheaders and counters to include. Go to the next step.
5. At the counter CS_TCH it says that the count should be shown. That is wrong. Make it show the average instead.
6. As you don't want to lose what you have created so far, you want to save a draft of your report template. Save a draft.
7. Go back to the main menu (where you started) to see what the overview looks like when you have saved you draft. Then return to finish your report template.
8. Create a chart where there should be a chart. In the chart you should switch places of the counters to make CP Restart LARGE be shown to the left in the chart.
9. Enter a “Desirable value” for CP_EVENT. Set the lowest value to 0 and the highest value to 4. Set the unit to "count".
10. Enter a formula for the counter CS_TCH. The formula should be KPI_DURATION * SDCCH_TRAFFIC.
11. Save the report template “for real” (i.e., not as a draft) and preview it. Preview it both with and without example data. When previewing with example data, make sure that the formula makes sense.

3. Edit an existing report template

You realize that you want to make some changes in the report template you just created.

1. Change the formula for the counter CS_TCH to be KPI_DURATION * TCH_TRAFFIC.
2. Change the “Desirable value” for QSI CS. You don't want to set a lower and a higher limit, but only one value. Set the value to 95,00. Set the unit to percent.
3. You changed your mind! You realized that you did not want to change the desirable value for QSI CS. What a luck there is an undo function. Use it to undo your last action.
4. Now that you have made these changes, change the description of this report template to read “Changed formula”.
5. Save your changes.
4. Create a new report template based on an old one

Now you want to create a new report template called G14B, that is based on G14A. It should look exactly like G14A, except for:

1. The counter QSL_CS should not be included.
2. The description should be "First version" instead of "Changed formula".

Find a way to do this without having to create a whole new report template from scratch.
Appendix 2: Photos of early prototypes

**Photos of early prototype 1**

![Figure 1: Overview of report templates in Prototype 1.](image1)

**Figure 1:** Overview of report templates in Prototype 1.

![Figure 2: Watching a preview of a report template.](image2)

**Figure 2:** Watching a preview of a report template.
Figure 3: Creating a new report template.

Figure 4: Creating a new report template.
Figure 5: Creating a new report template.

Figure 6: Adding a formula to a counter.
Figure 7: Entering the desirable value of a counter.
Photos of early prototype 2

Figure 8: Overview and preview of templates.

Figure 9: Creating a new report template, step 1. Choosing counters.
Figure 10: Creating a new report template, step 2. Editing the properties of counters.

Figure 11: Creating a chart.
Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Malin Persson.