ONE SPARK –
COMPOSE YOUR OWN PROJECT MANAGEMENT APP

Sebastian Fröstl, Iris Braun
Technische Universität Dresden
Faculty of Computer Science, Chair for Computer Networks
01062 Dresden, Germany
E-mail: sebastian.froestl@tu-dresden.de

Abstract – In recent years many useful web-based services were introduced, which simplify the work on projects. The idea of One Spark is to include these tools, having no direct project relation, in the scope of a project management software (PMS). One Spark is an easy to use PMS, which allows single users and teams to work efficient on projects. Based on a new plugin concept it allows the user to include these external services and manage them as tools within their own project environment.

I. Introduction

Today many applications for managing projects are existing. Some are desktop versions and others are web-based. The leading thought of developing an own approach was the dissatisfaction, resulting from the usage, of existing PMS solutions. Some points of criticism are the insufficient usability or the distracting amount of functions. This range of functions is on the one hand very large and confusing, on the other hand very limited and not extendable.

In the course of the web 2.0 expansion many providers shipped useful tools, like cloud file storage service Dropbox [1], or SCM platform GitHub [2]. A vast number of these services are accessed via API. This lead to the idea of One Spark, a web-based project platform with focus on an easy to use interface and the possibility to adapt it's functionality by integration of external services. One of the benefits of this approach is a wide range of disposition. The user determines his requirements and is able to adjust his project management application accordingly.

The realization and evaluation of this idea was part of the bachelor's thesis of Sebastian Fröstl [7]. It is based on a market analysis of existing applications as well as a requirements analysis via questionnaires. The proof of concept was realized with a prototype, which was introduced on the exhibition Output DD [3] and is still running online. A subject group was asked to work with the prototype and participate in a survey to evaluate the prototype.

II. Competitors analysis

A group of three PMS was chosen to be examined according to different criteria. Since the scope of a bachelor's thesis is limited, the selected competitors have to be representative and with a similar use case.

- Basecamp [4]: was launched in 2004 by 37Signals and is one of the most popular web-based PMS. It manages over 8 million projects and was completely redesigned in 2012.

- Jira [5]: made by Atlassin is an application for operational project management with focus on software development. About 14.500 customers in 122 countries are using Jira. Especially larger companies apply for this solution.

- Assembla [6]: is a tool for managing commercial or open source software products. It allows to integrate some external tools and is therefore part of the analysis.

The search volume index (SVI) by Google gives a brief insight in the relevance of the selected group. The following figure 1 shows the average divergence of Assembla and Jira based on the search volume index of Basecamp. As a result the factor of Jira was 1.44 above the SVI of Basecamp. To the contrary of Assembla, which factor was only 0.08.

Fig. 1: Search Volume Index of Basecamp, Jira, Assembla

The analysis was done by comparing the three competitors in the following categories [7]:

- Range of functions
- Usability
- Possibility to integrate external services

The range of functions differs strongly. Basecamp has, after it's redesign, a very limited functionality. Jira is very complex and has different internal plugins to extend its functionality. Nevertheless the common feature is the task management, despite their handling is different. Jira and Assembla provide more complex configuration, like a task status, a priority or the estimated amount of work. Basecamp limits itself to a due date and the person responsible. With Jira and Assembla a graphical evaluation of project status is possible.

The consideration of usability could only be done subjective. Basecamp has, regarding its limitation, a very clear and intuitive user interface. There is no confusion and the whole project is displayed on one page. The navigation of Jira is highly nested. The structure is counterintuitive, since the overview is quickly lost and navigation nodes with different children are named similar. The structure of Assembla is more conclusive, however the detail edit views are distracting.

Assembla and Jira offer a concept to integrate extensions. Jira has a central market place, where developers can submit plugins. These plugins work with the internal Jira interface. Assembla provides the user to add tools to his project via the admin area. These extensions are part of Assembla, External developers are able to write a tool and propose it to the Assembla Team. After approval the tool is integrated.

III. Requirements analysis

The requirements for One Spark are phrased as use-cases and are classified according to MoSCoW method [7]. This allows to divide the requirements in „Must",

„Should", „Could", and „Won't have this time maybe later" - categories. The goal is to first accomplish all M and S criteria and then proceed to C and W. All requirements were gathered and categorized like the following example:

M1: The application allows the integration of external services. The prototype connects with a cloud file storage service and a source code management platform.

Since the idea of One Spark is very large application, the strategy to develop a prototype by one person has to be very flexible. Therefor a lean strategy was combined with the agile software development. Every function was implemented with Behavior and Test Driven Development, for every implementation the tests are written at first. Additionally the overall application behavior is tested every time a new feature is added.

IV. Application Concept

One of the key requirements is the extendable architecture of the system. Therefor the basic Model-View-Controller architecture is combined with a service-oriented design. Functionality of One Spark is wrapped in services and accessible via API from the inside or outside. This concept allows different clients to communicate with the central application. The API is realized as a thin RESTful web service, see figure 2.
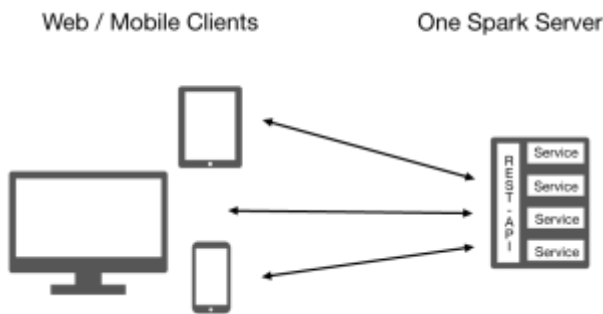

Fig. 2: Service oriented architecture of One Spark

This architecture aims furthermore to bidirectional communication with external services. One Spark is able to integrate external tools via their RESTful API and allows other tools to consume the services of One Spark.
The plugin architecture works as Library Plugin concept, which is a standard design pattern. The new functionality is wrapped in a plugin library and is loaded to the application. The connection to an external service is done via a specific client class within the plugin library. All the detail knowledge is therefor in domain of the plugin, the application calls methods defined in an interface.

V. Prototype

The prototype is based on the framework Ruby on Rails [9] and provides the basic user and project management functionalities. As external services, Dropbox and GitHub are integrated. The tools afford a remote management of these services within your project context.
The biggest challenge for the user interface was to design an extendable layout. This layout should be recognizable across different tools. Therefor the structure is three-stage as seen in figure 3. The first section is the main navigation, where the user selects his tools. The second and third columns structure the tool itself like the common known Folder-Pattern. A main container has child elements.

VI. Evaluation

A short survey was created to evaluate the concept and it's realization. As mentioned before, the prototype was presented on the exhibition Output DD and is accessible via internet [8].
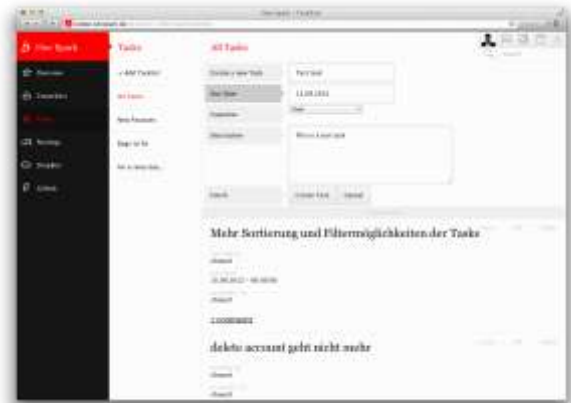

Fig. 3: Prototype of One Spark

The goal of the survey was to gather information about potential users such as:

- The user background concerning experience, usage of other PMS and their satisfaction with these

- Ratings of the concept in general and specific to usability and design

- Ratings to the idea of including external services and it's prototypic realization

- Further suggestions and wishes

The subject contained 34 persons, where 28 are using other PMS. The majority stated their satisfaction with already in use PMS only „average". Overall 76% of subjects rated the concept and the realization of our approach to be „good" or „very good". The point of integrating external services had many positive statements. Suggestions concerned the improvement of the currently basic project and task management.

VII. Conclusion and future steps

The first approach to realize the idea of One Spark led to a working prototype. On the basis of this implementation the evaluation gave a promising prospect to make further development. Despite the fact, that the shipment of One Spark was at a very early state, the user feedback was greatly useful. The briefly cycle of shipment and direct user feedback is a powerful development strategy and will be applied to further iterations.
Further steps will regard the following points:

- Improvement of the project and task management, including a dashboard, a report tool and time tracking.

- Enhancement of the plugin architecture, to assure the integration of more external services

- Specification and implementation of a security concept

References

[1] Dropbox: https://www.dropbox.com
[2] Github: https://github.com
[3] Output DD: http://output-dd.de
[4] Basecamp: http://basecamp.com/
[5] Jira: http://www.atlassian.com/software/jira/overview
[6] Assembla: https://www.assembla.com/home
[7] *Fröstl S.* Entwicklung einer flexiblen Web-Anwendung für die Unterstützung des Projektmanagements. Bachelor's thesis. October 2012.
[8] One Spark: http:onespark.de