Customer Driven Project

Rock Concert Audience as a Screen

Project Report

Netlight AS

Agnethe Soraa, Tomas Dohnalek, Jan Bednarik, Milos Jovac
Project adviser: Anh Nguyen Duc

September 14, 2013

Abstract

This report will give the reader an insight into the details of the design, development and implementation of the task given in the course TDT4290 - Customer Driven Project, taught at NTNU - the Norwegian.

University of Science and Technology. The customer is Netlight and they have presented the group with the task of breathing new life into the console.

Web-applications these days are leaning against a mouse-controlled, web-fronted design. This has taken away much of the efficiency of power users, who have traditionally used terminal applications on a daily basis, and had the system in their fingers.

A hybrid web-fronted/console design would be a possible solution to this problem: The power user can make use of their full potential through a console whilst the objects are presented in the web-interface.

This is a proof-of-concept task, and all research done will be documented and used to argue for and against the solutions used and not used. Everything from the planning of the project startup and preliminary-study to the complete conclusion is described in this report.

The approach to investigate and solve this problem starts with a thorough study of relevant technologies, and how this can be made possible. The conclusion of this study allows us to create a system which showcases the real potential of our soution. Through this whole process we have a close work-relationship with our customer to ensure his desires and expectations for the project are met, and that our conclusions and findings boosts future research in this field.

Preface

Contents

1 Introduction

1.1 General information

1.2 Terminology

1.3 Structure of report

1.4 Project and project name

1.5 Project purpose and concept

1.6 Project goal

1.7 Stakeholders

1.7.1 Customer

1.7.2 Customer contact

1.7.3 Development team

1.7.4 Advisor

1.8 Project background

2 Planning

2.1 Project plan

2.2 Methodology choice - Scrum

2.3 Organization

2.4 Risk Management

2.5 Quality Assurance

2.6 Measurement of project effects

2.7 Duration and workload

2.8 Gantt diagram

2.8.1 Description

2.8.2 Result schedule

2.8.3 Roles

2.8.4 Version Control

2.8.5 Textual documentation

3 | Preliminary studies

3.1 Similar projects

3.2 Market investigation

3.3 Existing technologies and frameworks

3.4 Evaluation of alternative solutions

3.5 Outcome of research - Our decision

3.6 Constraints

3.7 Chosen development technologies??

3.8 Evaluation criteria

4 | Requirements

4.1 Description/scope

4.2 Definitions/general terms

4.3 Business Requirements

4.3.1 Functional

4.3.2 Non-functional

4.4 Use cases?

4.5 Product backlog

4.6 Summary

5 | Testplan

5.1 Approach

5.2 Templates

5.3 Responsibilities

5.4 Test criteria

6 | Software Architecture

6.1 Introduction

6.2 Selection of architectural viewpoints

6.3 Views

6.4 Tactics

6.5 Patterns

6.6 Data Storage

7 | Tools and strategy

8 | Sprint 0

8.1 Sprint planning

We have embraced Sprint 0 as a preliminary sprint, when we can set up all necessary collaboration tools, equipment, prepare templates for meetings and mainly to acquaint ourselves with Scrum methodology. The original plan was to finish sprint 0 on 8th of September, but we have decided to terminate it prematurely due to finishing sprint goals in shorter time than we had expected. Other reason for terminating the sprint was desire to start actually working on the product itself.

The actual user stories are listed in table 8.1. Since we started to use the software collaboration tool only during the sprint we did not manage to estimate the time needed to complete each story beforehand and thus the column **Est.** is left empty.

8.1.1 Sprint 0 User-stories

ID	Description	Est.	Sp.
259	**I as a developer need to prepare LaTeXtemplate for minutes, project plan, weekly status report.**		**5**
	Meeting minutes		2
	Project report		2
245	**We as a team need to give a project and team name.**		**2**
	Team name		1
	Product name		1
248	**I as a developer need to agree on customer, advisor and internal meetings.**		**2**
247	**I as a developer need to agree on daily working hours.**		**1**

243	**I as a developer need to set up the video conferencing.**	**2**
249	**I as a developer need to add goals for Sprint 0.**	**4**
250	**I as a developer need to decide which collaboration technologies to use.**	**20**
258	**We as a team need to assign roles to team members.**	**1**
258	**I as a developer need to write a project plan.**	**90**
258	**I as a developer need to research the older reports.**	**30**
258	**I as a developer need to summarise the requirements.**	**4**
	SUM:	**161**

Table 8.1: User stories selected for Sprint 0.

8.2 System Burndown

Since we managed to establish the proper collaboration tool Target Process 3 only during the sprint the software was not able to generate relevant burndown chart. We at least tried to estimate how much time we spent working on each of the user stories listed in table

8.3 Architecture

8.4 Implementation

8.5 Testing

8.6 Occurring risks

8.7 Retrospective

8.7.1 Pros

8.7.2 Cons

8.8 Evaluation

9 | Sprint 1

9.1 Sprint planning

After assembling all the tools in Sprint0, we decided to start with the implementation of core modules. As our understanding of task improved, we were able to come up with user stories from the perspective of user, customer, developer and student. All user-stories were given to the customer so they can be prioritized. All but user-stories concerning our student obligations, like writing project plan, minutes, meetings with supervisor and attending lectures. Those were mandatory and already added as user-stories of sprint1. On Monday 02.09.2013. we had the meeting with a customer where we estimated time we need for every user story. The result of that meeting was the list of the rest of the user-stories for sprin1. All user stories for finishing our first prototype were on the sprint1 list so we also agreed date for presentation and showing the running demo - Thursday 12.09.2013. After that ,at a group meeting, we decoupled user-stories into tasks and we were ready to start with the imlementation of client-server core module.

9.1.1 Sprint1 User-stories

9.2 System Burndown

9.3 Architecture

Choosing client-server arhitecture was very intuitive to do. Our project has user application that depends on commands for what to play, on one side, and application that is responsable of detecting and sending commands to that users on the other. Every aplication(user) have to be either one or another.

Write about Android NSD, create class diagram,

9.4 Implementation

9.5 Testing

9.6 Occurring risks

9.7 Retrospective

9.7.1 Pros

9.7.2 Cons

9.8 Evaluation

Figure 9.1: Sprint1 Burn Down Chart

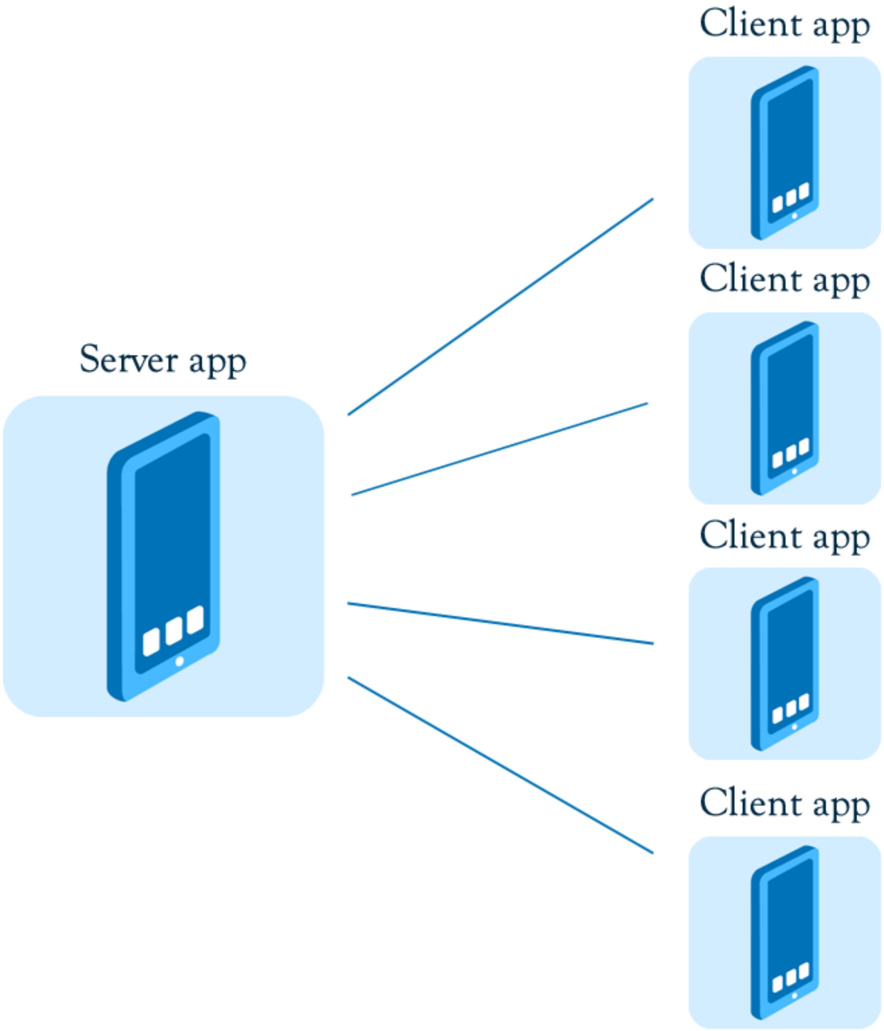

Figure 9.2: Sprint1 Arhitecture

10 | Sprint 2

10.1 Sprint planning

10.1.1 User-stories

10.2 System Burndown

10.3 Architecture

10.4 Implementation

10.5 Testing

10.6 Occurring risks

10.7 Retrospective

10.7.1 Pros

10.7.2 Cons

10.8 Evaluation

11 | Sprint 3

11.1 Sprint planning

11.1.1 User-stories

11.2 System Burndown

11.3 Architecture

11.4 Implementation

11.5 Testing

11.6 Occurring risks

11.7 Retrospective

11.7.1 Pros

11.7.2 Cons

11.8 Evaluation

12 | Sprint 4

12.1 Sprint planning

12.1.1 User-stories

12.2 System Burndown

12.3 Architecture

12.4 Implementation

12.5 Testing

12.6 Occurring risks

12.7 Retrospective

12.7.1 Pros

12.7.2 Cons

12.8 Evaluation

13 | Sprint 5

13.1 Sprint planning

13.1.1 User-stories

13.2 System Burndown

13.3 Architecture

13.4 Implementation

13.5 Testing

13.6 Occurring risks

13.7 Retrospective

13.7.1 Pros

13.7.2 Cons

13.8 Evaluation

14 | Sprint 6

14.1 Sprint planning

14.1.1 User-stories

14.2 System Burndown

14.3 Architecture

14.4 Implementation

14.5 Testing

14.6 Occurring risks

14.7 Retrospective

14.7.1 Pros

14.7.2 Cons

14.8 Evaluation

15 | Testing

15.1 Types

15.2 Unit testing

15.3 Integration

15.4 System testing

15.5 Usability

15.6 Acceptance

16 | Evaluation

16.1 Group evaluation

16.1.1 Group dynamics

16.1.2 Role assignment

16.1.3 Risk evaluation

16.1.4 Customer and project task

16.1.5 Advisor

16.2 Project Evaluation

16.2.1 Planning

16.2.2 Preliminary Studies

16.2.3 Scrum

16.2.4 Meetings-Summary

16.2.5 Course feedback

16.2.6 Testing

16.2.7 Time usage

16.3 Technology evaluation

16.3.1 Skype

16.3.2 Github

16.3.3 Facebook

16.3.4 Testflight

17 | Conclusion

17.1 Introduction/Final product/description

17.2 Results

17.2.1 Functionalities

17.3 Evaluation criteria

17.4 Evaluation Results

17.5 Conclusion

17.6 Discussion

17.7 Further work

17.8 Reflection

17.9 Summary

18 | References

19 | Attachments

A | User Manual

B | Installation Guide

C | Glossary

D | XML Scheme?

E | Customer meetings

F | Group meetings

G | Supervisor meetings

H | Evaluation Questioner