Distributed Polling System
Project Plan Document

Version 1.0

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this document is to present the description of the project which is a decision making system for predefined group of members. This project is a part of the distributed software development course organized in Malardalen University, Vasteras. This document will server as an input to score supervisor and evaluators. This document also discusses the detailed plan of the project such as the project schedule, roles and responsibilities, different constraints and plan for activities and so on.

1.2 Intended Audience

- Supervisors (SCORE and Steering Group), who need to know how the project will be delivered and when to expect deliverable.
- Customers, who need to know the basic Project plans.
- Team members, who need to know the deadlines and project deliverable.
- Future users and developers of DPS application.

1.3 Scope

The scope of the document is to demonstrate the overall project perspective and the entire project schedule. This includes time schedule for different activities, defining the roles and responsibilities, assumptions and constraints. The description of the requirements is outside the scope of this document.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Project Manager	A person in charge of organizing the team and communicating with the customers/steering group

1.4.2 Acronyms and abbreviations

Acronym or Abbreviation	Definitions
SCM	Software configuration management
SVN	Subversion
ASAP	As soon as possible

2. Organization

2.1 Project management

Project Manager: Jenis kavadiya
Responsibilities:
1. Communication with SCORE customer and course supervisors
2. Ensuring project is on-track as per plan.
3. Resource management and work distribution
4. Acting as a Team member (due to small team size, a dedicated project manager cannot be assigned)

2.2 Project group

Name	Initials	Responsibility (roles)
Jenis Kavadiya	JK	Project Manager, Scrum Master
Avijit Dutta	AD	Architect, Developer
Aparna Vijaya	AV	Database Designer, Documentation Coordinator

Farahnaz Yekeh	FY	Test coordinator ,Developer
Rishabh Gupta	RG	SVN Coordinator, Developer
Sajjad Ali Khan	SK	Developer, Designer

These are the main responsibilities of the team members. But finally we work as a team and every member is responsible for every phase of the project.

2.3 Steering group

Rikard Land, Aneta Vulgarakis

2.4 Customer

Miguel Felder, prof.dr.sc. Ivica Crnković

2.5 Others

Xiaoping Jia

3. Project Overview

3.1 Executive Summary

Distributed Polling System is a distributed software system that facilitates important business decisions to be taken without having any conference call or adopting any other media of conversation, where decision makers are scattered across the world. DPS allows decision makers to compose a business issue, to intimate the members present anywhere in the world and to collect voting responses, through SMS and Email. In addition, the DPS architecture supports further medium of intimation like voice mail or video stream.

In project start-up, we faced few ambiguities in our understanding of the requirements and we had to start successive phases like design, implementation and so on at early stage in parallel to requirements engineering due to time limitation. However we started with the portion of the requirements that we had pretty much clarity and started designing the architecture along with a little bit of implementation. We used waterfall model for project planning purposes and followed Scrum approach because of the small team size, more reliance on informal communication and less requirements clarity in inception phase.

From SCORE perspective DPS focuses more on architecture, design, algorithms and interfaces, which are the most critical parts for evaluation. Considering the same we have revealed a robust architecture design that facilitates scalable, technology independent, loosely coupled and component based system.

To fulfill, we designed DPS as a software system that consists of software applications. SMS Gateway, web-DPS application and Email server are the software applications in distributed polling system aka DPS. All the software applications in DPS are inter-connected through JBOSS middleware which introduces high level of loose coupling. In the system middleware is in the hub and each application is situated on the perimeter of a bicycle-wheel linked through spoke called 'adapters' in software terminology.

Time limitation, complexity of the system and adoption of such architecture introduces a number of independent components whose integration is a challenging task

DPS is packaged software that can be installed and further configuration will ensure communication among its software applications.

A handful of technological challenges have been faced during the implementation phase. Major challenges include understanding telecom network, Integration of Software applications using middleware - the core of our system, software interaction with mobile handset and Unavailability of university Email server.

Resolution includes hard work of team members to grasp telecom network functionalities and to

expertise in JBOSS middleware product suite, implementation of java service to interact with the mobile handset through COM port and successful installation, configuration of Microsoft WINMAIL server version4.6 respectively.

However, our confidence has been boost-up after successful integration of this huge number of components that has produced system's expected functionalities through proper testing using well formulated test design techniques as per our architecture design.

Our project team is versatile in nature from various perspectives like cultural differences, work experience, preferences of time in project work and so on. Team consists of six members from India, Iran and Pakistan perusing Masters Degree from Sweden and Croatia. We have achieved our success by investing twenty hours per person per week for almost ten weeks, due to having few other courses in parallel in the same curriculum.

3.2 Project Scope and Main Challenges

DPS is part of distributed software development course which runs for a duration of ten weeks as a part of Masters in software engineering program jointly between Mälardalen University, Sweden and University of Zagreb, Croatia. The aim of this course is to enable students gaining experience in distributed development environment. The course consists of students from both Sweden and Croatia in order to achieve distributed environment. Participation in SCORE and its result is one of the major evaluation criteria of this course. Both universities have international students which makes project team representatives from various countries. This provides ideal environment for a distributed project in terms of geography and different cultures.

The major challenges we have identified in the whole process were:
- Team members are from different countries having different work cultures which may lead to misunderstandings during the development process.
- Communication among team members is another main challenge because of distributed environment and also many team members does not have English as a mother tongue which can cause problems during intercommunication among team members.
- The contact with customer (SCORE supervisor in this case) is limited and through emails only. This is a major challenge for requirement engineering phase. The clarification of requirements is time consuming as mode of communication is only email.
- The actual time we have for completing the whole project is ten weeks. This leads careful selection of development process because time constraint is one of the major factors while deciding upon development process in any project.
- Another important challenge is selection of technology as different team members are skilled in different technologies and are good in different phases of development cycle. It is really important to create and assign roles based on skills we already posses in order to produce most productive and efficient outcomes.

3.3 Development Process

We have used agile methodology and in particular Scrum for development of our project.

Scrum requires frequent and sometimes face to face interaction with customer which was not possible for us. Thus we assigned the role of Customer manager (who will act as customer for our project) to one of our members. One person is selected as the Scrum Master. We reserved first four weeks for Planning, Analysis and Design. During these four weeks we came up with an initial version of task/story list, core architecture and partial design of the system. Initial task list consisted of up to 54 different tasks that had grown to 132, when we finished project implementation. Each task was assigned a priority (on scale of 1 to 5 from urgent to optional respectively), task description, remark and an example scenario.

The project was developed in multiple sprints; each lasted 4 to 5 days of around 20 hours each. Before starting each sprint we had a Sprint meeting and before starting each days work we had a

Scrum/standup meeting (max of 10-15 min).

Sprint meetings: In this, our Customer manager selected those stories which had maximum value to the project, which were more risky and some were those on which many other tasks were dependent. Our first sprint consisted up of tasks email and SMS communication.

Standup meetings: In the standup meeting, the Scrum master selects the tasks from the sprint, which are to be finished on the current day. Members or group are assigned tasks from the task list and their names are written along with the tasks on the whiteboard. For some critical tasks we even assigned 2 members (pair programming for example we did it in task: result calculation for polls). The members then start implementing them. On the next day standup meetings all members stand together for short time and each one speak the following three things:

What tasks the member has achieved after the last meeting?
What are the problems they are facing?
What tasks the team members are planned to do now?

Once the standup meeting is over, the scrum master updates the excel sheet of story list.
Then the scrum master discusses the problem faced by individuals. Our experience is that during the first sprints we hesitate to communicate with each other. But in later sprints team members start proactively helping each other. The Infrastructure setup is like a virtual class room between both universities, and each student can see and hear other student from the other country. Video conferencing using Microsoft Net-meeting is established. Students are provided with number of time slots where they can discuss and communicate with other students located 1500 KM apart from other student of the same project team.

3.4 Management Plan

The DPS process involves different activities. These activities need to be distributed and managed among team members. As we are a distributed team, communication among different team members becomes one of the major areas of focus from project management perspective. Various methods are used for proper communication among team members. Weekly formal meetings using video conferencing was organized to discuss weekly status report of tasks assigned to individuals. Google groups, Skype chat are other tools we used for communicating quick issues.

We have various scrum meetings, start-up meetings during implementation phase. For document and code sharing, we used a configuration management tool SVN. One of the team members is assigned as SVN coordinator whose responsibility includes managing directory structure of the SVN repository, taking regular backups to counter server crashes. The access to the repository is strictly restricted and team has specific access rules defined for them. All these rules and access information is maintained in a SVN policy document. We use Google groups for creating topics regarding ongoing activities and for discussing and providing feedback about other's work in the team. In addition to all these, we have different meetings on requirement basis such as knowledge sharing sessions.

Some formal meetings are documented by writing Minutes of Meetings (MOM). Each MOM has clear description of the meeting and actions assigned on individuals before the next meeting. We also have weekly meetings with DSD course supervisor which includes review of tasks done in previous week. We also had two project status presentations as part of the DSD course. This presentation includes discussion about all challenges we faced, activities we have completed, and planning for the coming weeks. Using all these methods, we kept track of all activities and ensured that we are working on the correct track, kept every team member on same level and pace. We have created different roles for different activities. Every team member has ownership of task corresponding to the role he has assigned to. For example one of the team members was assigned as document manager and all project related documents are his responsibility and he is the one who keep track of status of the documentation tasks assigned to other team members. This way we have created sense of ownership among team members which really helped us a lot during the whole development process.

We used a structured approach for communication among the team members. All members were able

---

to contact project manager directly. And only project manager is responsible for communicating with DSD and SCORE supervisor. Any issue and doubt of the team members has to go through project manager.

4. Deliverables

Following will be the deliverables that will be produced as a part of entire software application:

- Project Plan and Description document
- Requirement Specification document
- Architecture and Design documents
    - o High Level Design document
    - o Low Level Design document
        - ▪ Algorithm
        - ▪ DPS Error handling document
        - ▪ Message Routing and Transformation document
- Technical document for Mobile Communication
- Implementation
    - o Software Code and Scripts
    - o External Software's used by DPS Application
- Verification and Validation
    - o Test Plan Document
    - o Test Case Document
- Installation Manual
- User Manual

5. Project Risks

Risk	Impact	Risk Mitigation and prevention
Time shortage	High	Use of Agile (scrum) process, Proper planning.
Need of SMTP email server IP, port and log-in credential details	High	Install and configure trial version of Microsoft WINMAIL server v4.6
Software interaction with mobile handset	High	Java program (service/API) has been coded that interacts with the mobile handset through COM7 port
System integration using middleware	High	Hard work and efforts of team members to expertise in JBOSS middleware product suite
Competence in technologies	Medium	Knowledge sharing sessions and Training. Pairing less competent members with more experienced one.
Limited contact with the customer	Medium	A team member played the role of customer manager (who act as customer for our project)
Miscommunication	Low	All main communications and decisions will be written down in MOM's and circulated through and Emails and Google group.
Server crash (database/SVN)	Low	Generation and updating the database scripts periodically. Regular backups.

6. Communication

6.1 Language

All the project communication will be in English. When corresponding locally, members are free to use local language, but all the conclusions should be made public in English. Also, all the documents and presentations will be in English only.

6.2 Asynchronous communication

For asynchronous communication we will use e-mail. Maximum response time for e-mail is 24 hours,

but the problems should be addressed as soon as possible.

6.3 Synchronous communication

As deadlines are strict, it will be important to have a real-time communication to solve current problems and issues ASAP. We will have text chat meetings on Skype and we will have weekly meeting usually on Mondays or when needed. After the meetings, the Minutes of Meeting (Summary) document will be available on FERWeb project page.

6.4 FER Web project page

DPS has its own web page at: http://www.fer.hr/rasip/dsd/projects/distributed_decision, where members, steering group and customers can upload/download important documents concerning the project and publish news.

7. Configuration management

We will use SVN as Software Configuration Management (SCM) tool for configuration management. The Subversion repository provides the DPS (Distributed Polling System) team with a very powerful means to share files and record their changes over time.

However, the ability to publish documents also puts a lot of responsibility on the shoulders of those who commit changes to the repository. Committing something to SVN has serious consequences. All other developers will get your changes once they are in SVN.

7.1 Rules for using SVN repository

There are several rules that ensure that the Subversion repository is used a way that we respect privacy and gain the maximum benefit.[1]

- Please do not upload documents that are not under your custodian. These files tend to get outdated, which contradicts the notion of version control. Borrowing files from other sources also bares the risk of storing them multiple times at different places. This might cause conflicts.
  Moreover, please also do not upload files that can be generated automatically from other files. Only real "source" files should be in the repository. For instance, there is no benefit to upload PDF files next to their corresponding Word documents. In the same way, software files like *.jar, *.o, *.class, and *.exe should not be part of the repository (in particular, as they contribute significantly to the overall size of the repository)
- Please do not add change logs within files. Furthermore, do not activate change tracking when using MS Word or PowerPoint. Subversion provides it own powerful mechanisms to keep track of changes.
- Also, please avoid version numbers in file names. They become inconsistent anyway by the next commit.
- When committing files, please provide a log message. Log messages should be understandable to someone who sees only the log. They should not depend on information outside the context of the commit. Try to put the log messages only to those files which are really affected by the change described in the log message.
- Subversion allows you to commit more than one file at a time. Therefore, please commit all related changes in multiple files, even if they span over multiple directories at the same time in the same commit. This way, you ensure that the repository always stays in a consistent state.
- When committing files, always take update from repository before committing because SVN allows concurrent modification of files. So, always update your local copy before committing your changes otherwise it will overwrite the changes done by another team member. And in case of any conflict, both members should sit together and perform manual merging of their respective changes.

---

[1] https://trac.fkie.fgan.de/MTRS/wiki/SVNPolicy last accessed on 08-11-2008

- No member should change the directory structure of the repository. Team members are allowed to add and modify files but they are not allowed to modify directory structure. If any change is required in directory structure, please contact the SVN coordinator.

7.2 Access Information

URL of the repository: svn://lapis.rasip.fer.hr/svn/DSD/Distributed_decision.
The user name and password for accessing the repository will be same as the users FER web account.

8. Project plan

8.1 Activity plan and Timelines

Activity	S1/ w45	S2/ w46	S3/ w47	S4/ w48	S5/ w49	S6/ w50	S7/ w51	S8/ w52	S9/ w53	S10/ w54
Project preparations and planning	Major	Minor	Optional	Optional	Minor	Optional				
Requirements analysis and definition		Major	Minor	Minor	Major	Minor	Optional			
Architecture and design		Minor	Major	Major	Minor	Minor		Optional	Optional	
Environment Setup				Major	Major	Minor	Minor	Optional	Optional	
Communication layer implementation				Minor	Major	Major	Major	Optional	Optional	
Business layer implementation				Minor	Minor	Major	Major	Major	Major	Optional
GUI and database layer implementation					Major	Minor	Minor	Major	Minor	Optional
Unit testing and Integration testing					Major	Major	Major	Major	Major	Optional
Functional testing					Minor	Minor	Minor	Major	Major	Optional
System and Alpha testing					Minor	Minor	Minor	Minor	Major	Optional
Project Report									Minor	Major

S1: Sprint 1, W45: Week 45th

■ (Red) Major activity: Dedicated team members assigned for the task and will be focus for the sprint
■ (Orange) Minor activity: This can be done along with major activity and not the main focus of the sprint
■ (Yellow) Optional activity: They are not predictable at time of planning but might happen during the sprint.

9. Outcomes Challenges and Lesson Learned

9.1 Outcomes

The outcome of the entire process is a working functional prototype with most of the desired functionality and artifacts including project plan, architecture, design and test cases document along with number of byproducts and end products such as user and installation manual and scripts.
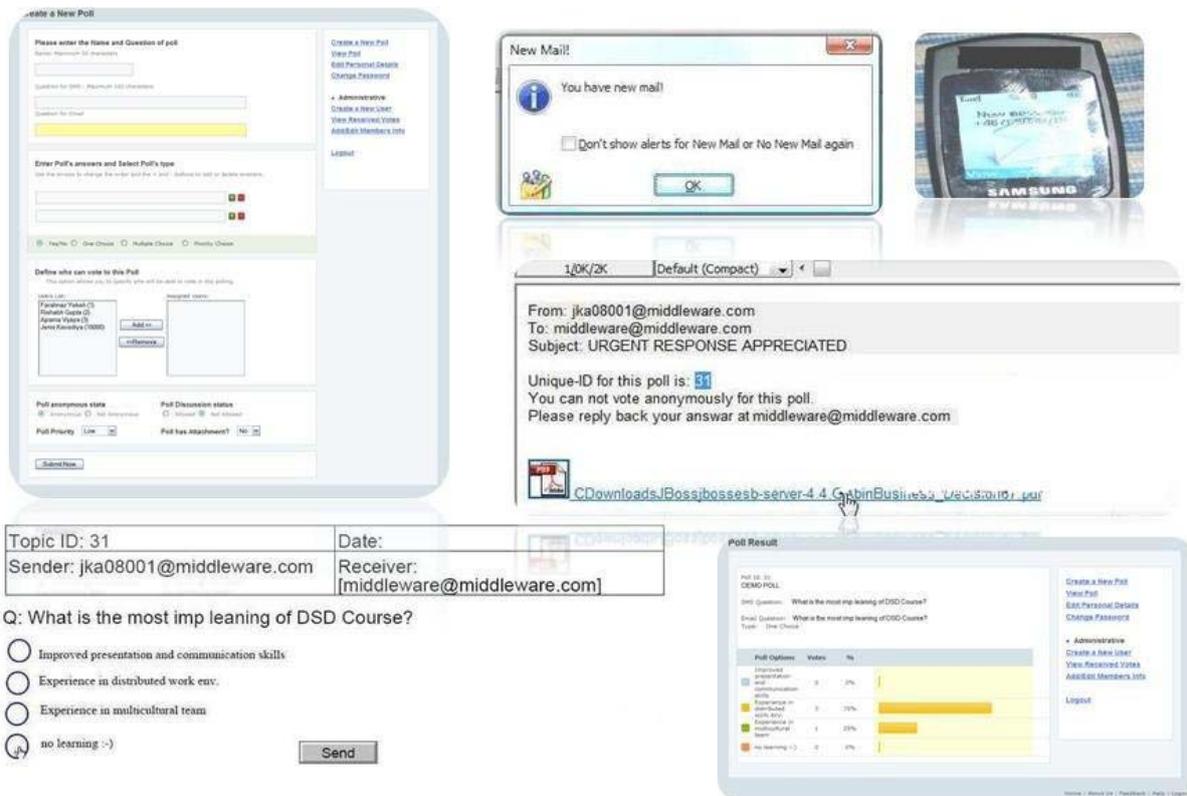
Figure 1: Snap-shot of some DPS functionalities
 (Create poll, receive mail alert, receive SMS, received mail, attached PDF info and view poll result)

9.2 Challenges

Main challenge we faced was during requirement engineering. The only way of communication between team and customers is thorough email which led to some misunderstood requirements. The initial requirement definition was created using functional requirement document provided by SCORE. We had misunderstanding in one of the requirement of having anonymous user in the system; this was sorted out late in the development cycle which cost us significant amount of time because the solution of this problem included changing the design (Database and GUI changes).

We found some problems within the team during early stage of the project because of different communication and cultural difference barriers. Due to time constraints we had some team meeting on weekend which created problem for some team members because they were not comfortable working on weekends and late night because of their work culture. But with mutual cooperation, understanding, respect and helping each other, we successfully overcome all problems. Inefficient competency and configuration problem for different software like email server, email clients, application server, web server and integration of these software with rest of the core application were a major issues in the project. By self learning, knowledge sharing and organizing sessions on various technology related issues we led our development process in a smooth way.

Finally, we are able to finish our job in a smooth and systematic manner. The team is able to implement all major functionalities of SCORE and able to develop a fine working prototype. The project is under DSD curriculum and team of different diverse culture and nationality members are working in distributed fashion to face overcome different real life challenges and work in distribute manner towards one goal. This project can be further enhanced by adding some new pretty good features if there is some more time for the implementation. However, with the current features and functionalities it is ready for the end users to use it in the real environment. Although we could not invest much more time due to having parallel courses in the same curriculum, we would be grateful to work further on this software to make it commercially successful.

9.3 Lessons Learned

Our project "Distributed Polling System" is a huge learning experience for us which has taught us to invest more time in proper architecture designing which in-turn reduces heavily the implementation time for any system.

In addition to that, the architecture has guided us to segregate the whole system into granular components that has enriched maximum reusability and utmost team activity. Even the architecture has helped us to invest less time for component integration which has become evident while we started getting expected system behavior after our first time component assembling.

We made a mistake in prioritizing among the requirements without involving the customer, enough. This is our one of the most important learning from the project. Earlier we hey have reduced some functionality required, like the anonymous vote, with no agreement of the stakeholder. But now we have implemented those functionalities, after receiving the requirements from the reviewers and are a part of DPS application.

During the project we have a good chance to learn some new technologies. Meanwhile, we also faced some problems during the system integration due to strict deadlines. Moreover, there was less time for testing of project. By all team working together, we were able to implement most of the system functionalities in a very limited development time.

In spite of having six team members from different cultural background and practices, by investing twenty hours per person per week for almost ten weeks, we successfully designed, developed, integrated and tested the functionalities.

It's our honor and privilege to be part of such kind of real time project implementation and we believe we could do further enhancement to this software in different aspects to make it a commercial software.