ISTANBUL TECHNICAL UNIVERSITY

Q&A Web Page

Project Plan

11/15/2013

VolkanIlbeyli	**040100118**
GökhanÇoban	**040100057**
FarukYazıcı	**040100112**
EmreGökrem	**040100124**
TuğrulYatağan	**040100117**

Istanbul Technical University Computer and Informatics Faculty Software Engineering class Fall 2013 term project / Q&A Web Page

1. Introduction

This project plan includes the development plan of the question and answer (Q&A) site project. The website offers its users a convenient platform for exchanging information by means of questions and answers. A user can sign up and ask a question. Meanwhile the other users may upvote the question to show appreciation of a good question which will make the user who asked the question gain reputation. Based on the reputation points, a user may comment on other questions or answers, upvote a question, downvote a question, propose edit for a question or an answer to keep the maintenance of the website. To encourage its users, the website has an achievement system called badges which are awarded upon completion of certain tasks.

1.1. Scope and Purpose of Document

The purpose of this document is to represent the software application development plan and people who are interested in the development of the project to have information about its development process. The document consists of 6 topics:

1. *Introduction*- gives preliminary information about the project, its objectives, functions, performance issues and constraints
2. *Project Estimates* - gives estimations about the project (such as manpower, cost, and more) to estimate resources needed for the project implementation beforehand
3. *Schedule* - general day to day timetable of the project is described
4. *Project Resources* - all resources needed to carry out the project is debated in this sub-topic
5. *Staff Organization* - individual and general information about the group developing the project
6. *Appendices*

This document is prepared to make management and resource estimation easy. Also project sponsor and development group can benefit from this document according to their requirements.

1.2. Functions and Issues

1.2.1. Major Functions

The Q&A site implemented will provide users a database of questions asked by the users themselves and answered by users themselves as well. The questions will be categorized according to the tags associated with them. The site contains two user groups (admin and public user) and a reputation system. Major functions are as follows:

- Admin will be responsible of tag and badge CRUD operations.
- Users will be able to ask and answer questions, upvote/downvote questions as well as they can comment below the questions and answers. As they gain reputation from the questions, they will be able to suggest edits, flag questions.
- Users will be able to search for other users and questions with corresponding tags.

1.2.2. Performance Issues

There is not a particular performance measure for this Q&A system. However, there is a main issue that affects the performance of the program:

- Size of database: If the size of the database is too large, it will take more time to make a query and produce a result.

2. Project Estimates

2.1. Historical Data Used for Estimates

C# and .NET tools have been used by most of our group members. We have agreed on using .NET and SQL Server together. Some benefits of .NET, such as consistent programming, direct support for security, simplified development for efforts, easy application deployment and maintenance make us choose this platform for this project. Also Entity Framework, which is presented by .NET, is a useful way for accessing database. On the other hand MSSQL Server works with .NET framework properly.

2.2 Estimation Techniques

COCOMO is a model designed by Barry W. Boehm to give an estimate of the number of man month it will take to develop a software product. This "**CO**nstructive**CO**st**MO**del" is based on a study of about 60 projects at TRW, which is a company where many large software projects are being built. The programs examined where ranging in size from 2000 to 100.000 lines of code, and programming languages used range from assembly to PL/I.

COCOMO consists of a hierarchy of 3 increasingly detailed and accurate forms.

Basic COCOMO is a static single-valued model that computes software development effort (and cost) as a function of program size expressed in estimated lines of code.

Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel, and project attributes.

Advanced COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

COCOMO II can be used for:

- Software development approach
- Budget decisions
- Production tradeoffs
- IT capital planning
- Investment options
- Management decisions
- Prioritizing projects
- SPI strategy

2.3. Estimations

FP Value: Since Microsoft .NET platform will be used in development of this project, and C # is selected to be the base language for that environment which has a LOC/FP value of 59.

External Inputs:

- User Login Page
- Search Page
- Question Creation Page
- Answer Input in Question Display Page
- Comments in Question Display Page
- Tag & Badge Creation Page

External Outputs:

- Question Display Page
- Question Search Results Page
- User Profile Page

Estimated line of code is thought as 3500

Work on a Developer per Month
E = a x (KSS)b = 2.41 * 3.5 * 1.05 = 8.85675

Development Span = (Month)
D = c x (E)d = 2.50 * = 8.85675 * 0.38 = 8.4139

Number of Developers
E / D = 8.85675 / 8.4139 = 1,052

The web based application will be developed by five developers; consequently if we calculate Development Span again;
8.4139 * 1,052 = 5 * DS

DS = 1,8 (Months)

Project Type	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Half Dependent	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

3. Risk Management

Project risks shall be defined certainly and correctly. In the project the major risks may be caused by deficiency of planning or lack of intercommunication.

3.1. Project Risks

3.1.1. [R01] Unrealistic software plan or schedule

The development process of the software might be delayed by the unrealistic planning in some parts. Thus an extension of the deadline shall be required. A late submission is not accepted in a strict deadline system. Also latency will result in a loss of the wills of the team members.

3.1.2. [R02] Team communication or meeting problems

Even though the team is connected, problems are still the case. Since the team members are students and some of them are currently working, inevitably some problems while gathering will occur. Also the meeting durations might be insufficient for an efficient work as well. Such problems are likely the most significant ones.

3.1.3. [R03] Lack of Experience on Twitter Bootstrap

Twitter Bootstrap, the CSS framework to be used in the project, is quite functional providing a great user interface. However, it's new to some members. Besides the coding parts of the project, some members shall learn Twitter Bootstrap and practice on it. Hence some extra delay is possible.

3.1.4. [R04] Developing wrong or inadequate functions

Major functions of the software are defined as detailed as possible, however each major function consist lots of function which is not planned yet. In the timeline of planning this kind of function, planning or coding problems will occur.

3.1.5. [R05] Developing wrong or inadequate interfaces

Project may be very well designed with fully functional codes, but the interfaces might not be adequate enough to reflect the functions well. Usability of the software is highly dependent on the interfaces and any faults/lacks will cause inconvenience in the usage of the software.

3.1.6. [R06] Testing shortfall

Project may be very well designed with fully functional codes, but the interfaces might not be adequate enough to reflect the functions well. Usability of the software is highly dependent on the interfaces and any faults/lacks will cause inconvenience in the usage of the software.

3.2. Risk Table

Name	Category	Probability	Impact	Remedy Plan
R01	ES	65%	2	RPoR01
R02	PS	50%	2	RPoR02
R03	PS	90%	2	PRoP03
R04	RQ	40%	3	PRoP04
R05	RQ	30%	4	PRoP05
R06	DB	30%	3	PRoP06

3.3. Risk Mitigation, Monitoring and Management

Probability and the cost of risks may be decreased by means of mitigation, monitoring and management. The main way of management is splitting into two or more groups to work separately. Although this way of working will increase the individual work, it’s still possible to finish the project within the deadline.

3.3.1. [PRoP01] Remedy Plan of [R01] Unrealistic software plan or schedule

3.3.1.1. Mitigation

Because the deadline is tight, the scope of the project must be determined at a sufficient level. It shall not exceed the limits developing capabilities of members or the deadline conditions. Such a risk may lead to great exposure. Hence the project plan and schedule must be monitored. This application will follow the situations of the software. Therefore a possible delay may be detected easily.

3.3.1.2. Monitoring

As mentioned before, monitoring will help in finding the latencies on the project. Latencies that are not subject to the lack of training or technology is probably caused by unrealistic plan.

3.3.1.3. Management

In the case of R1 occurred efficient cost will be so high thus the management plan has lots of strategy.

- The function that cannot be conducted in the time scheduled before may be eliminated from the software scope if and only if it is not a major function.
- If the function is a major function and there is no extra function which can be eliminated from the product, project group working on the function might be split into two groups one of them working on the function the other one following the project schedule. However team members will have much more work to do, the latency of the roadmap can be avoided.

- If the function is a major function, some of the other extra functions may be eliminated from the project or their design and developing time may be tightened.

3.3.2. [PRoP02] Remedy Plan of [R02] Team communication or meeting problems

3.3.2.1. Mitigation

Communication and meeting problems may causes deadly locks on the projects roadmap. Thus the precautions of the risk shall be strict enough. Every meeting must be planned before the project developing timeline. In addition a member of the team might be in charge of intercommunication.

3.3.2.2. Monitoring

Communication or meeting problems can be tracking on the source control mechanism and checkpoints. Branches on the source control mechanism will indicate that team members are developing obliviously of each other.

3.3.2.3. Management

Communication or meeting problems are responsible of communication member and project manager. This kind of problems can be handled by organizing extra meetings and communicating using online tools such as Skype.

3.3.3. [PRoP03] Remedy Plan of [R03] Lack of Experience on Twitter Bootstrap

3.3.3.1. Mitigation

Lack of experience on Twitter Bootstrap Framework has the biggest probability. Therefore detecting this risk shall not release to occur during the developing of the software. Ahead of designing end user interfaces includes Twitter Bootstrap, team members should be trained about Twitter Bootstrap.

3.3.3.2. Monitoring

The trainings on Twitter Bootstrap and interface developing process might be monitored to determine if the risk will occur.

3.3.3.3. Management

The possibility of the risk is so high which means its management conditions must be very effective. In case of the risk one or more member of the team might be responsible of user interfaces and Twitter Bootstrap. Other members will follow the schedule and provide code behind support to these members. So learning or conducting Twitter Bootstrap will cause no hold on the schedule.

3.3.4. [PRoP04] Remedy Plan of [R04] Developing wrong or inadequate functions

3.3.4.1. Mitigation

Since wrong or inadequate functions may be noticed on testing, each module shall have separate testing process. Every check point should designed the way that provides plenty of time in which wrong functions will be fixed or inadequate functions will be developed.

3.3.4.2. Monitoring

Monitoring on checkpoints and function dependencies will provide information about the risk of wrong or inadequate functions.

3.3.4.3. Management

In the case of the risk assigning a team member to fix the wrong function or to develop the missing function might be a way out. Branching on the roadway is also necessary.

3.3.5. [PRoP05] Remedy Plan of [R05] Developing wrong or inadequate interfaces

3.3.5.1. Mitigation

Similar to PRoP04 wrong or inadequate interfaces containTwitter Bootstrap may be noticed on testing. Separate testing processes on each checkpoint will reveal the risk.

3.3.5.2. Monitoring

Monitoring and tracking checkpoints and testing might subscribe enough information to find out the real probability and cost of the risk.

3.3.5.3. Management

Similar to PRoP03, one or more team members will be responsible for Twitter Bootstrap and user interface developing.

3.3.6. [PRoP06] Remedy Plan of [R06] Testing shortfall

3.3.6.1. Mitigation

Any shortfall in testing processes will cause to deliver defected product. As a result of that testing plan and process shall designed and scheduled as comprehensive as possible. Besides delegating one member of team as tester engineer will provide on point controlled test structure and processes. Tracking alpha and beta versions of the product and applying extra test cases on these versions might prevent test shortcuts.

3.3.6.2. Monitoring

As the testing processes are the last step of developing a module or a checkpoint, there is no specific way to define the risk early. Tracking extra tests on the alpha and beta versions might provide information about test fails.

3.3.6.3. Management

Testing shortfalls might be deal with providing more testers over the project.

4. Tasking

4.1. Project Task Set

The Q&A system will provide a platform between users for their questions. Questions are asked and answered by users themselves. Admins are responsible for organizing question tags and badges as well as responding to flagged questions as in deleting and approving edits.

The system will require user sessions, distinguish between regular users and admins, and offer them different interfaces that only allow the actions they would have permission for.

Generic features such as data permanency and intuitive navigation options will be provided by the use of an MSSQL database and Entity Framework. The web page will be coded in C# and ASP.NET.

4.2. Functional Breakdown

A non-exhaustive list of features that the platform would have is given as follows:

- Viewing question listing
- Searching for questions & users
- Commenting on questions and answers
- Preparing the question to be asked
- Defining user accounts
- Defining administrator accounts
- Sessions

4.3. Task Network

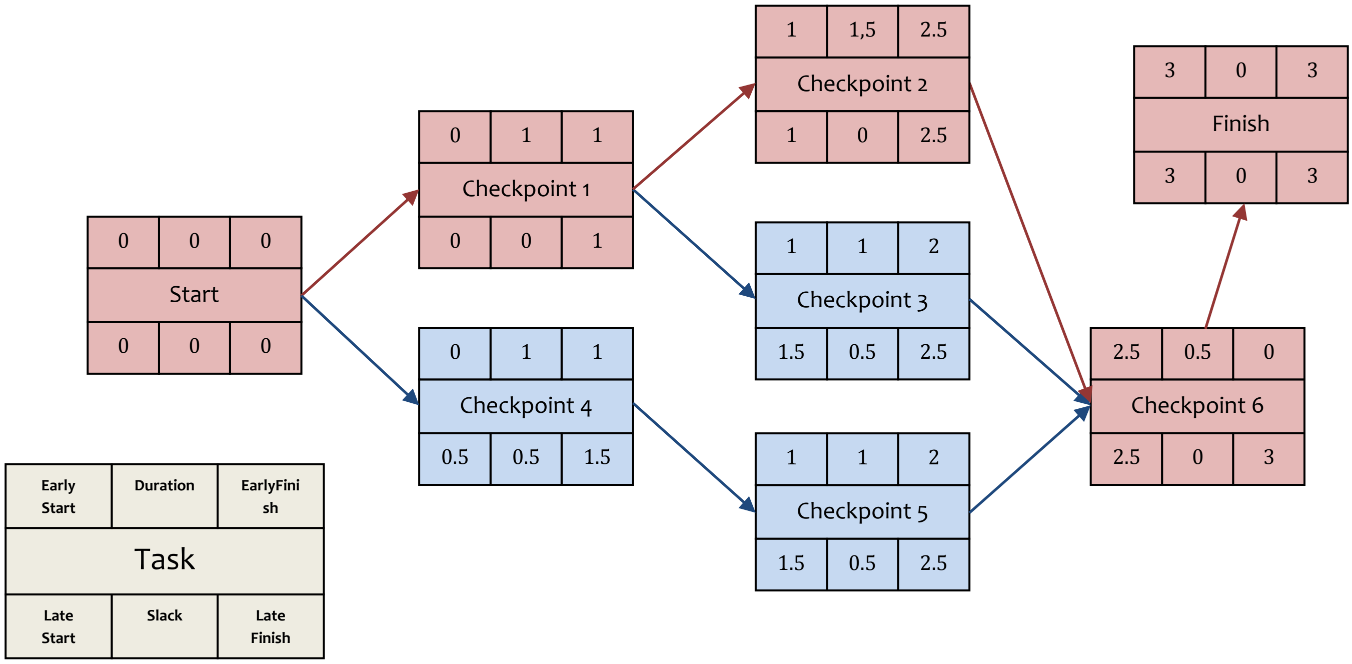
The bulk of the features to be implemented are divided into seven elementary groups, which will be implemented in turn and become checkpoints in the course of development.

There will be no preparation and final testing phases—instead, each checkpoint will be handled modularly, designed, implemented and plugged in to the project, and finally tested. This cycle will be reset at every checkpoint.

The checkpoints are as the following:

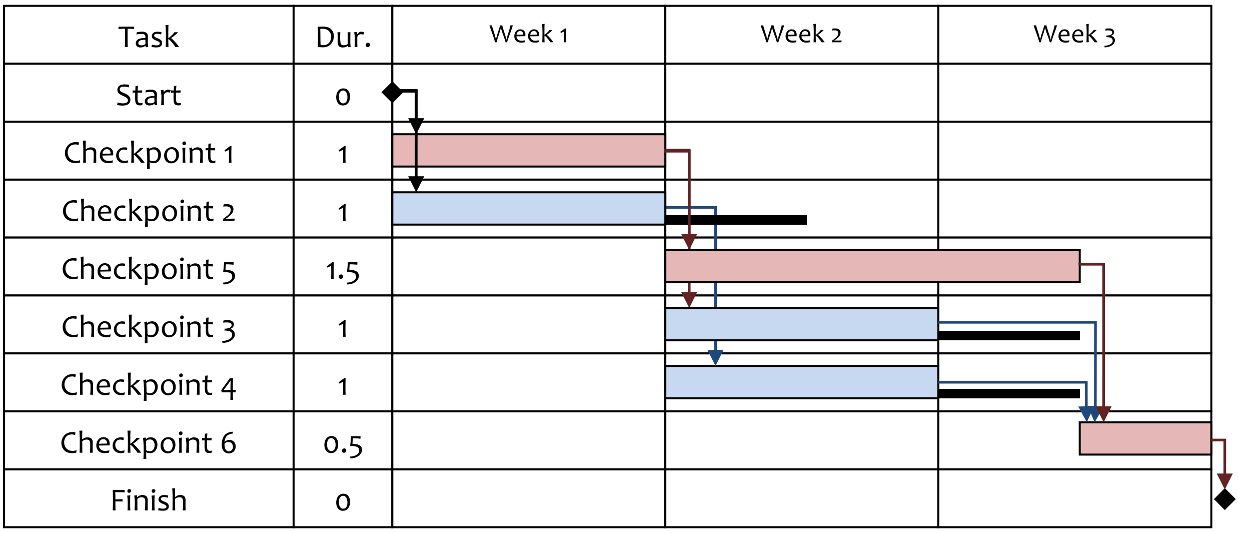
- Checkpoint 1 – Project inception
    - Database modeling
    - Creation of site map
    - Creating data access layer with Entity Framework
    - Mock-up designs of pages
- Checkpoint 2 – Base implementation
    - Creation of master page template
    - Creation of main page
- Checkpoint 3 – User management and login
    - User CRUD and login pages, sessions
    - Management of privileges regarding user types
    - Handling Visibility of pages
- Checkpoint 4 – Admin operations
    - Tag CRUD pages
    - Badge CRUD pages
    - Reputation and badge gain rules
- Checkpoint 5 – Functionalities
    - Question creation page
    - Question display page
    - Comments
    - Answers
- Checkpoint 6 – Advanced functionality
    - Flags of questions and answers
    - Vote-up and vote-down of questions and answers
    - Reputation gain mechanism
    - Badge gain mechanism

The relationships of the feature modules are given in the PERT chart below to illustrate internal dependencies:



4.4. Timeline Chart

The total six weeks available for project design and implementation have been distributed equally among the six sets of tasks, with the exception of that a part of the share of the last module has been assigned to the second module instead. The following is a Gantt chart indicating the timeline for task sets:



Task	Dur.	Week 1	Week 2	Week 3
Start	0			
Checkpoint 1	1			
Checkpoint 2	1			
Checkpoint 5	1.5			
Checkpoint 3	1			
Checkpoint 4	1			
Checkpoint 6	0.5			
Finish	0			

5. Team and Management

5.1. Team Structure

The developer team consists of five programmers. There is no hierarchy among the team members—each member is self-motivated, and assumes designer and programmer roles as necessary. As such, every team member becomes responsible for themself as well as every other team member, and so an agile method becomes even more feasible.

The team members are listed below:

• Volkanİlbeyli	040100118
• GökhanÇoban	040100057
• FarukYazıcı	040100112
• EmreGökrem	040100124
• TuğrulYatağan	040100117

5.2. Management Reporting and Communication

The team holds weekly meetings, where the latest state of the project is discussed and goals are set until the next meeting. Minor progress reports between meetings may be given through informal interactions and in version descriptions pushed to the code repository, TFS.

6. Tracking and Control Mechanisms

In this section, some of the techniques which are going to be used for project tracking and control are specified.

6.1 Quality assurance and control

Software Quality Assurance Plan

6.1.1 Introduction

6.1.1.1 Purpose

This part of the document includes the procedures that the project team members must be aware and also includes the control methods to obtain the best quality software product. Additional information about SQA activities will be added during the project. All managers and developers in the project must read this document.

6.1.1.2 Project Scope

Project shall prepare a web based system with DB components as well to manage a real estate center with multiple agents. The entire system will be developed .NET 4.5 platform.

6.1.2 Management

2.1 Organization

The borders of responsibilities of the project team members are not sharp in this project, never the less every team member has a specific task to do. The developers of the project are responsible for maintaining the process and check whether the project is on time and well. The organization within the project and the responsibilities of each member is determined as below:

Project Manager (PM) **Gökhan Çoban**

PM produces a project management plan, motivate the other members, check budget and control all mechanisms.

Vice Project Manager (vPM) **EmreGökrem**

vPM assists the PM, checks progress of the project and informs PM.

Quality Assurance Manager (QAM) **Volkanİlbeyli**

QAM produces Software Quality Assurance Plan and guarantees that the project will be delivered as desired and good quality.

Configuration Manager(CM) **TuğrulYatağan**

CM manages version and repository, also produces Software Configuration Management Plan.

Team Leader (TL) **FarukYazıcı**

TL plans and coordinates all the team members, make them connect, appoint meetings; maintain project integrity and also informs PM in case of need.

6.1.3. Tasks

The main task of SQA manager is to check if the procedures of the project is followed properly by the team members. Also SQAM must control the consistency between documents.

6.1.4. Project Standards

The standards below are initial consensus on project metrics. Additional standards (if needed) will be added in the document during the project.

For coding:

The methods must be start with capital letter and written in PascalCase.

The loops must have the maximum depth of 3 and the nested-if statements must have the maximum depth of 4.

Comments must be supplied for each class.

For documentation during project:

The project documents use the template provided by the CM and they must be written in English.

SQA manager must check the documents for approval.

6.1.5. Problems and How to Correct

During the project, there may be a lot of issues that have to be solved immediately. For instance, some of the modules may not be completed in given time, the errors in the code, the lack of commentary standards that has been determined before in SQA Plan, incomplete functionality.

- When a team member realizes that there is a problem in any piece of the project, he/she does the necessary tasks and informs it to the PM and all other members.
- TL arranges a meeting with other members and tries to solve problem. Every single member is responsible for any detected problem.
- When the problem is solved by a member the SQA manager would be informed about this.
- If the problem is not solved in a dedicated time and if the problem is discovered after internal approval of the project, than the team members decides if the problem is insignificant or not; if it is important the solution shall be developed.

Changes in requirement

According to user requirement document the changes may done or be discarded with consensus of the team.

6.1.6. Techniques and Methods

SQA manager ensures that the appropriate techniques are used in the project.

It’s supplied that if;

- All the members has the capability to code in .Net platform with C#
- The tools are available for all the members.

6.2 Change management and control

Software Configuration Management Plan

6.2.1. Introduction

6.2.1.1. Purpose

Software Configuration Management Plan helps the developers planning the project and guide them about the conventions.

6.2.1.2. Scope

For all documents of the project, the SCM report provides a standard layout and a structured way to form them.

6.2.2. Management

The Configuration Manager is generally deals with the technical issues, backups etc.

6.2.2.1. Interface Management

The project will use a DB server with SQL Management Studio (MSSQL) and an IIS server. Every team member will connect to these servers and works via the connection.

6.2.3. Configuration Identification

6.2.3.1. Version Tags

The version tags in the documents is denoted as a.b.c

And these are stand for;

a->external(release) version number [when a is incremented both b and c are set to zero]

b->internal version number

c->informal version number[when a new version is created c is incremented]

6.3.3.2. Document Names and ID’s

File names should be short e.g.

(documentName)-(VersionTag).(Extension)

Document ID’s:

All the documents of the project has a document ID which is denoted with “/” following slashes.

E.g.:

ProjectName/documentPathInTheLibrary/versionNumber

6.4. Configuration Control

Any document considered as a single entity (Configuration Item - CI) is stored in two libraries; one of them is development library one of them is archive library.

Development Library

This library contains CI’s that are under development. Every document has its own directory. Configuration Manager is responsible for naming and tagging the documents if any controversy with the standards is seen.

TFS directory tree is more or less like below:

```
/tfs
        /development
        /archive
                /documents
                        /Project Management Plan
                        /Requirements Report
                        /Database ModelRepor
                        /Test Plan Report
                /meetings
                        /weekly-records
                /presentations
```

6.5. Communication

Project members are communicating each other with Facebook. The Team Leader contacts the members in case of a need and arranges demanded meetings for future scheduling. The PM creates issues on TFS and assigns to related members, also checks if the issues are being closed regarding the checkpoints.tfs.visualstudio.com will be used as TFS free repository, and also it supports everything for agile issue tracking and management. Thus; it will fit this project perfectly.