**CS673S16 Software Engineering**

**Team 3 - ProPal**

**Project Proposal and Planning**



|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Xiang Chen | Configuration Leader/ Environment Integration Leader | *Xiang Chen* | 09/14/2017 |
| Yansen Liu | Design Leader | *Yansen Liu* | 14/09/2017 |
| Yuhao Wu | Implement Leader | *Yuhao Wu* | 09/04/2017 |
| Weicheng Yu | Backup Team Leader/ Security Leader | *Weicheng Yu* | 09/14/2017 |
| Chen Shou | Requirement Leader | *Chen Shou* | 14/09/2017 |
| Dawei Li | Team Leader | *Dawei Li* | 14/09/2017 |
| Lu Min | QA Leader | *Lu Min* | 14/09/2017 |
|  |  |  |  |
|  |  |  |  |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Overview](#_87t9hln2vjz0)

[Related Work](#_mps353x5ezyl)

[Detailed Description](#_fg3z0hpd4q9v)

[Management Plan](#_ds8oyr75pnh1)

[Process Model](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.27177f40uci)

[Risk Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.a4oqwntk3mw)

[Monitoring and Controlling Mechanism](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.ywdoc2clc9yt)

[Schedule and deadline](#_tadq5mb0pici)

[Quality Assurance Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.72e1f4uawy2r)

[Metrics](#_b2haznn3yyz2)

[Standard](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.vc72k6dweldv)

[Inspection/Review Process](#_f1c69ifi68h7)

[Testing](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.r5d5mhtlf0kq)

[Defect Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.54a4wuncjg1c)

[Process improvement process](#_jhct37ebxxpn)

[Configuration Management Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.hw41vg4ykxen)

[Configuration items and tools](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.bwlb4d4vdox2)

[code commit guidelines](#_yyauft6zr9hw)

[References](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.8mva2050iy7t)

[Glossary](#_ty3i2nqffhtc)

# Overview (By Weicheng)

The motivation of this project came from the shortage of a simple and easy to use management tool when developing a large piece of software. The main purpose of Propal is providing a user friendly website application that members of a software development group can use to check current prior objective, track working progress, as well as log in work hours. The potential users of the software can be students who just start to learn software engineering, software developers who intend to implement a small scale software as a group, and also managers who want a robust platform to track a software project. In general, just like our project name, Propal, our software is a good pal for software development project.

# Related Work (By Xiang)

(describe any similar software systems, and the difference from them)

* 1. [***Aha!***](http://www.aha.io/?ref=quora.com)

Aha! allows you to manage a sprint like Pivotal. But it is designed for product teams and based around the idea that you need to link your product strategy to your roadmaps to your development work. Otherwise, you can quickly build yourself to nowhere

* 1. [**Fusioo**](https://www.fusioo.com/?utm_campaign=tb&utm_content=89278&utm_medium=a&utm_source=quora)

Fusioo is a customizable CRM and project management tool that allows teams to work the way they want to. Unlike other project management tools, with Fusioo you can set up your custom workspace to track and manage the information you need.

# Proposed High level Requirements(By Yansen)

Pivot tracker link: <https://www.pivotaltracker.com/n/projects/2108674>

# **Functional Requirements**

* + 1. Essential Features

1. All users shall be able to create accounts
2. All users shall be presented with the site’s Terms and Conditions and Privacy Policy
3. Registered users shall be able to have a drop-down menu to choose a project.
4. Product owner shall be able to have a horizontal navigation bar to manage Stories, Members, Burndown Chart and Settings.
5. Registered users shall be able to have a notice window before they leave the webpage (close the window or click backward).
6. Registered users shall be able to create new projects
7. Product owner shall be able to create more than one requirements in each projects
8. Product owner shall be able to set the role for invited register users.
9. Registered users shall be able to login
10. Registered users shall be able to reset their password
11. Product owner shall be able to invite other registered users to their projects
12. Product owner shall be able to edit their projects(name, requirements, etc)
13. Product owner shall be able to delete projects
14. Product owner shall be able to create requirements
15. Product owner shall be able to edit requirements
16. Product owner shall be able to delete requirements
17. Product owner shall be able to assign requirements(one or more) to other developers(one or more)
18. Registered users shall be able to access their accounts
19. Developer shall be able to view requirements
20. Developer shall be able to finish assigned requirements
21. Scrum master shall be able to view requirements
    * 1. Desirable Features
22. Product owner shall be able to view finished requirements
23. Product owner shall be able to restart finished requirements
24. Product owner shall be able to view deleted requirements
25. Product owner shall be able to restore deleted requirements
26. Product owner shall be able to set due date for requirements
27. Registered users shall be able to view finished projects
28. Registered users shall be able to view deleted projects
29. Registered users shall be able to restore deleted projects
    * 1. Optional Features
30. Registered users shall be able to view invited people’s projects
31. Registered users shall be able to set and change security questions
32. Registered users shall be able to use security questions to reset password
33. Registered users shall be able to create a biography for their profile
    1. **Nonfunctional Requirements**
34. The language used shall be English.
35. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
36. Privacy of users shall be protected and a Privacy Policy shall be effectively communicated to the users. The Privacy Policy shall disclose what user data is collected and how it used.
37. The site shall have a response time of less than 3 seconds.
38. Date shall be stored in the MySQL database on team server.
39. Application shall be served from the team’s account
40. Application shall be hosted and deployed on Digital Ocean Host Services.
41. Application shall be developed using local LAMP environment: MAMP
42. Application shall be developed using Intellij as IDE and Fillezilla as FTP.
43. Site security: basic best practices shall be applied. In addition, the site shall salt and hash user passwords and prevent basic SQL injection.
44. The site shall accommodate up to 100 concurrent users accessing the application at any time.
45. The site shall be governed by a set of Term and Conditions, which shall include, but not be limited to (1) a statement limiting ProPal liability; (2) a notice of copyright; (3) a Privacy Policy.
46. The website shall prominently display the following text on all pages “BU/CS673/Software Engineering Project, Fall 2017/Team 3 Brotherhood. For Demonstration Only”
    1. **Implemented Features (to be completed at the end of project)**

# Management Plan (Dawei)

# (For more detail, please refer to SPMP document for encounter example)

## Process Model

## Roles

* + - 1. Product Owner (To be discussed and assigned) (Scrum.org)
         * Clearly describe product backlog (user story)
         * Assign priority to each backlog
         * Plan Sprint backlogs
         * Ensure every developer understand the meaning of his/her backlog
      2. Developers (All Team Members)(Scrum.org)
         * Self organizing (Most backlogs are based on the other)
         * No leader for developers team
         * No sub-teams
         * Personal skill is team skill
      3. Scrum Master (Dawei Li)(Scrum.org)
         * Watch you guys coding and do nothing HAHAHAHA
         * Serve to the whole team

Lead by example, if i noticed i do something wrong, I confess immediately

Read the room (know the thought of everybody).

Zero tolerance if somebody hindered scrum management

* + - * + Serve to the Product Owner

Help owner manage backlog efficiently

Help developers understand backlogs

Help owner arrange priority and order backlog sequence

Arrange scrum events

* + - * + Serve to developers

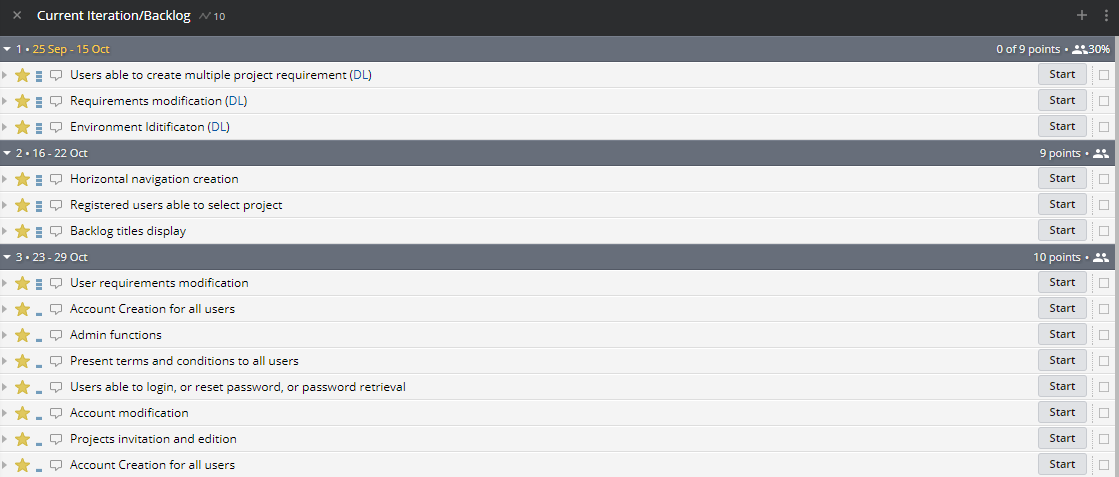
Teach developers how to self-organization

Arrange scrum events

* + 1. [Basic framework of Scrum](https://s3.amazonaws.com/scrumorg-website-prod/drupal/2016-06/ScrumFramework_17x11.pdf) (Click to see)
    2. Tool: pivotaltracker.com:<https://www.pivotaltracker.com/n/projects/2108674>
    3. Specific To Do List for each sprint see the figure below

## Objectives and Priorities

1. We decide



## Risk Management (need update constantly)

RM notes

Note 1: Xiang Chen will provide a basic tutorials to members who are not familiar with PHP

Note 2: Xiang Chen will design the interface between PHP and other language

Note 3: We will discuss and simplify our product to meet the deadline of commitment

Note 4: Yansen Liu will lead the UI design group to improve UI design until satisfied

## Monitoring and Controlling Mechanism

* + 1. Burndown Chart
    2. Daily Sprint
    3. Weekly report meeting
    4. Github
    5. Pivotal Tracker
    6. Immediate communicating(Wechat, Slack)

## Schedule and deadlines (need update constantly)

Deadlines:  
 Requirements 10.1

Model design 10.4

UI design 10.18

Database Design 10.15

Iteration 1 10.18

First demo

Database creation

Features:

Create new story

Finish story

Delete story

Add description

Add points

Iteration 2 11.08

Features:

Sprint(add, rename, delete)

description (show, hide, quick edit)

Story(clear, restore)

tag (add, delete, filter)

Sort(Hand, Point, Date created, Date modified)

Show all stories

Advanced add new story

Search story

Iteration 3 12.13

Login

Logout

Sign up

Access control

# Quality Assurance Plan( Chen, Lu)

# (For more detail, please refer to SQAP document for encounter example)

## Metrics

* + 1. Definition (e.g. define what metrics will be used, , how to keep track of metrics, and how to analyze the metrics for process improvement. Two types of metrics should be included: product metrics and process metrics. Particularly include product complexity (LOC, # of files, # of classes, # of methods etc.) cost (in terms of man hours), defect and defect fix rate etc.

Definition: From Software testing perspective, metrics can be defined as standards of measurement. Metrics will be used: bugs per line of code, code coverage, comment density, number of classes and interfaces, number of lines of code, number of lines of customer requirements, program execution time, program load time, program size. To track metrics mentioned above, quality activities are reviewed with senior management on a period basis, also, they will be reviewed with project leader both periodically and as needed. Process analysis will cover model analysis and analysis of existing process qualitatively, mainly focus on quality assurance metrics, techniques of process improvement includes: publish process models and process standards (determinant on metrics), questionnaires and interviews, ethnographic analysis involving assimilating process knowledge by observation.

* + 1. Two types of Metrics To Be Included:
       1. *Product metrics*: Schedule Variance, Effort Variance, Size Variance, product productivity, productivity for test case preparation, productivity for test case execution, productivity of defect fixation, productivity of defect detection.
       2. *Process metrics*: Number of methods and classes used, Effort measured in person months/person's hour, Errors/KLOC, Defects/KLOC, Cost/LOC, Documentation Pages/KLOC, size of product produced, Lines Of Code(LOC), 1000 Lines of code (KLOC).
    2. Results (to be completed at the end of each iteration)

At the end of each iteration quality assurance results need to be finished to track on both progress and product-wise productivity. Every programme need to hand in their own process metrics based iteration report, and quality assurance leader need to analyze entire product based on product metrics. These reported will be reviewed by senior management leaders and project leaders, so the entire management group can predict and improve qualities of product/process.

## Standard (e.g. documentation standard, coding standard etc. )

## Documentation standards:

The IEEE software engineering standards will be observed in designing the software requirements specification (SRS) and the software quality.

* + 1. Coding Standards:

The source codes of entire development team will follow the guidelines in Java/JavaScripts/PHP/SQL coding standards.

## Inspection/Review Process

## (e.g. describe what are subject to review, when to conduct review, who do the reviews and how ?)

Each week the project team need to have a meeting to report problems(any problems encountered throughout the duration of the project, software developer can discuss and report problems with major leaders). Subject to be viewed on weekly meeting: check through product/process metrics report, discuss about current issues encountered and possible solutions, if any two members have conflicting opinions might have further discussion or discuss with professor. Also need to focus on current productivity and progress of entire project, and discuss about any adjustments if needed. Formal technical inspections on the product architecture design will be conducted by other group leaders and provide a formal report. Each leader will review the produced documentation and make comments and suggestions during each iteration. Each milestone must be approved by each leader to proceed to next milestone. Each milestone will be indicated by the presentation of each iteration. If problems unable to be solved by group leaders, will have a discussion with main professor.

## Testing

## (e.g. who, when and what type of testing to be performed? How to keep track of testing results?)

A separate document about testing result should be linked here. ([link here for testing results](https://docs.google.com/document/d/1WIsB_gzWCiuO3Np2FZKiA1FLnAww0IR0eO6VMNCSluA/edit))

Who : team members.

When: after we finished the every iteration of the project, or we need to upgrade our software, or when we need to add features to the software.

What type of testing:

1. *Unit testing* - the testing of an individual unit or group of related units.
2. *Integration testing* - check if a group of components are combined to produce expected output.
3. *Functional testing* - the testing to ensure that the specified functionality required in the system requirements works.
4. *System testing* - check if our web app works in different environment.
5. *Stress testing* - the testing to evaluate how system behaves under unfavorable conditions. Or when there are many users managing their project at the same time, our app would not breakdown.
6. *Performance testing* - the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements.
7. *Usability testing* - the testing to check if web app is user-friendly, if users are easy to understand and handle the functions of our app.

We would perform testing mainly in testing environment, mainly by tool MAMP. After we have done this part of testing, we would test in product environment to make sure it works. We would keep updating testing logs during this procedure.

## Defect Management

(e.g. describe the criteria of defect, also in terms of severity, extend, priority, etc. The tool used to management defect, actions or personnel for defect management)

* + 1. Criteria of defect (mainly two parts):
       1. *Defect Severity* is a classification of software defect (bug) to indicate the degree of negative impact on the quality of software
          - **Severity 1 - Critical (S1)**: A defect that completely hampers or blocks testing of the product/ feature is a critical defect.
          - **Severity 2 - Major (S2)**: A major defect occurs when the functionality is functioning grossly away from the expectations or not doing what it should be doing.
          - **Severity 3 - Moderate (S3)**: A moderate defect occurs when the product or application doesn’t meet certain criteria or still exhibits some unnatural behavior, however, the functionality as a whole is not impacted.
          - **Severity 4 - Low (S4)**: A minor bug occurs when there is almost no impact to the functionality but it is still a valid defect that should be corrected.
       2. *Defect priority* determines how quickly the defect turnaround time must be.
          - **Priority 1 – Critical (P1)**: This has to be fixed immediately within 24 hours. This generally occurs in cases when an entire functionality is blocked and no testing can proceed as a result of this.
          - **Priority 2 – High (P2)**: Normally when a feature is not usable as it’s supposed to be, due to a program defect, or that a new code has to be written or sometimes even because some environmental problem has to be handled through the code, a defect may qualify for a priority 2.
          - **Priority 3 – Medium (P3)**: A defect with this priority must be in contention to be fixed as it could also deal with functionality issues which are not as per expectation.
          - **Priority 4 – Low (P4)**: A defect with low priority indicates that there is definitely an issue, but it doesn’t have to be fixed to match the “exit” criteria. Sometimes defects with priority low are also opened to suggest some enhancements in the existing design or a request to implement a small feature to enhance user experience.
    2. Tools:
       1. For defect management, it is essential to have:
          - Reporting facility – complete with fields that will let you provide information about the bug, environment, module, severity, screenshots etc.
          - Assigning – What good is a bug when all you can do is find it and keep it to yourself.
          - Progressing through the life cycle stages – Workflow.
          - History/work log/comments
          - Reports – graphs or charts
          - Storage and retrieval – Every entity in a testing process need to be uniquely identifiable, the same rule applies to bugs too. So, a bug tracking tool must provide a way to have an ID, which can be used to store, retrieve (search) and organize bug information.
       2. And there are many tools online, we may use the following tools, and update as needed.
          - *Plutora Test* - It uses a single instance for all projects consolidating testing design, planning, manual and automated execution, defect tracking and progress reporting and improves efficiency every step of the way.
          - *Backlog* - It’s easy for anyone to report bugs with a full history of issue updates, comments, and status changes. Reported issues are easy to find with search and filters.

# Configuration Management Plan ( By Xiang)

(For more detail, please refer to SCMP document for encounter example)

## Configuration items and tools

* + 1. Development:

Version Control Tool: Git and Github

Github : <https://github.com/buse2017fall/SoftwareEngineering>

IDE : Intellij and Eclipse

Environment : Windows, Linux and Mac OS

* + 1. Deployment:

Host: Digital Ocean

Domain: Free Domain

FTP: Fillezilla

* + 1. Architect:

Web: LAMP ( Linux + Apache + MySQL + PHP )

Test on Local : MAMP

## Change management and branch management

* + 1. Only configuration leader have the access to main account (Github)
    2. Developers use their own github account to fork the project repository
    3. Developers can create master and child branch on their own github and local machine
    4. Developers can push to their own github account freely
    5. If developers want to push to the main account, he or she has to pull new request on github, then configuration leader will check the code and decide if merge request or not
    6. Since we use git and github, it is easy to rollback to specific version

Branch:

* Master branch - final and deliverable branch
* Feature branch - used to test new feature
* More branches may be needed based on the future requirements

## Code commit guidelines

**Tutorial:**

**Each time before you change your local code , please following those steps**:

1. Git fetch upstream
2. Git merge upstream/master
3. Git add <file>
4. Git commit -m “message”
5. Git push
6. Go to your own Github to create new pull request

**What if there is conflicts ? How to solve it** ?

After fetch and merge, if there is conflicts , please follow those steps

1. open <file\_where\_conflicts\_occur>
2. Delete >>>>>> , ======, <<<<<<<
3. Save and quit
4. Git add <file>
5. Git commit -m “message”
6. Git push
7. Go to your own Github to create new pull request

# References

“15 Best Bug Tracking Software (Reviews of Top 2017 Defect Tools).” Software Testing Help 15 Best Bug Tracking Software Reviews of Top 2017 Defect Tools Comments, SOFTWARE TESTING HELP, Web. 1 Jan. 2016.

Nelson, Megan. "An Intro to Git and GitHub for Beginners",Web. Accessed 18 Sep 2017.

Nadig, Sneha. “How to Set Defect Severity and Priority: Defect Triage Process.” Software Testing Help How to Set Defect Severity and Priority Defect Triage Process Comments, SOFTWARE TESTING HELP, 1 Feb. 2015.

“What Is a Scrum Master?” Scrum.org, Scrum.org, 1 Jan. 2017, Web.Accessed 19 Sep 2017.

“What Is a Scrum Development Team?” Scrum.org, Scrum.org, 1 Jan. 2017, Web.Accessed 19 Sep 2017.

“What Is a Product Owner?” Scrum.org, Scrum.org, 1 Jan. 2017, Web.Accessed 19 Sep 2017.

Zafar, Rehman. “What Is Software Testing? What Are the Different Types of Testing?” What Is Software Testing? , Code Project, Web. 20 Mar. 2012.

# Glossary