1 Project Requirement

Table of Contents

1.1 ProjectOverview

Background

Software Projects Subject (COMP90082) is the subject for students to perform necessary Software Engineering practices to successfully build a high-quality software product for real-world clients. To monitor, evaluate, and provide feedback to students, a supervisor will observe the Software Projects activities of a student team through the software development tools used and through student-supervisor meetings. The involved software development tools include a Version Control System (Git, GitHub/BitBucket/GitLab), a Task Management Tool (JIRA/Trello) and a Documentation tool (Confluence). Some processes would be considered to assess students include Software Testing, Code Review, Continuous Integration, and Deployment.

Problem

The tools and processes that students use for these projects (as stated in Background) are diverse and are all accessed on different websites/platforms. Currently, supervisors have to manually open each tool to analyse the complete tasks on each of them. It is an inefficient and tedious process, especially if a supervisor monitors more than ten teams. Although some development tools (e.g., JIRA) provide visual summaries of a team's work, the visualisation only shows data from a single activity. Thus, it is difficult and time-consuming for supervisors to offer detailed and insightful feedback to students, especially if a supervisor has more than one team to supervise.

Project

This project aims to have a software system that automatically gathers and summarise all aspects of the software engineering activities of CIS (Computing and Information Systems) Software Projects (COMP90082) developed. This will allow course coordinators to monitor and promptly provide meaningful feedback to teams, improve the quality of processes used in development and software systems developed, thus ensuring software deployment at the end of every project.

Project Scope

Expected Functionalities

1. It is a web application.
2. The app can automatically collect data from the software development tools used by students that cover various aspects of Software Engineering. The expected tools are:
    a. Version Control System: Git + online repository of the student teams' choice (GitHub, GitLab, Bitbucket)
    b. Task Management System: JIRA
    c. Documentation: Confluence
    d. Code Review: Pull Requests in GitHub/GitLab/Bitbucket
    e. Continuous Integration and Testing Activities: GitHub/GitLab/Bitbucket
    f. Communication: Comments from JIRA and meeting minutes on Confluence
3. The app can analyse and summarise the gathered data within four key aspects:
    a. Product Quality (e.g., measuring code quality, test code quality, collecting released versions)
    b. Process Quality (e.g., sprint velocity and burn-down chart, code review frequency and coverage, code commits, test results from Continuous Integration)
    c. Communication (How frequently the team has a meeting)
    d. Individual Contribution (summarise the activities of an individual, e.g., tasks completion, communication, code reviews, documentation)
4. The app can visualise the statistical analysis report in a friendly visualised manner according to the four key aspects stated above.
5. The app provides extensible for future development.

Expected Pages

Page	Sub-page	Description
Home		Shows the list of team members and their corresponding projects
Project /Team	Overview	• Summarises the activities of a project and displays the student information. • Changes the link of Jira, Git and Confluence manually.

	Product Quality	Includes statistical and/or graphical summaries by code analysed tool (TBC) on: - Code quality - Test code quality Software Engineering tools include: - Git (Team's choice of GitHub, GitLab or Bitbucket)
	Process Quality	Includes statistical and/or graphical summaries on: - Sprint velocity and burndown - Documentation on Confluence - Code review frequency and coverage - Code commits Software Engineering tools include: - GitHub - JIRA - Confluence
	Communication	Includes statistical and/or graphical summaries on comments from engineering tools including: - GitHub - Confluence
	Individual Contribution	Includes statistical and/or graphical summaries of every team member's activities: - Task completion on JIRA - Code reviews on Git - Documentation contribution on Confluence Software Engineering tools include: - Git (Team's choice of GitHub, GitLab or BitBucket) - JIRA - Confluence

Out of Scope

- Login authentication
- Adding students to the system
- Creating projects and assigning students to projects
- Collecting information from Slack
- Providing feedback to teams on the application

1.2 Quality Requirements

Note: table refers to collection, analysis and summary of data

Requirement	Relevant Features	Category
Accurate	Collection	Functional Suitability
Up to date	Collection, Analysis	
Visually Stimulating	Summary	
Instructive	Summary	
Efficient	Analysis, Project Management	Performance
Intuitive	Analysis	Usability
High Usability	Analysis, Project Management	
Available	Collection, Summary	Reliability
Authenticated	Collection	Security
Modifiable	Project Management	Maintainability
Extensible	Collection, Analysis, Summary	
Replaceable	Summary	Portability

1.3.0 Persona Draft

Coordinator

(This page is the same as the Persona Draft from the previous team except that supervisor Persona is deleted because we decide that we do not need supervisor role in our project.)

- Has strong knowledge about IT and the student's ongoing projects.
- Main goal is to use the system to organise and analyse different student's data across various sources.
- Frustrations:
    - Have to browse different sites/sources to collect data.
    - Cannot view students' work explicitly.

- Motivations:
    - Wants to minimise time with browsing different sites to collect data and focus more on marking.
    - Wants to be able to easily assess quantitative parts so that they can focus on qualitative side (e.g., code/algorithm quality)
- Tasks:
    - Collect all activities from various tools.
    - Measure/assess efforts of different groups.
    - Visualise data to simplify marking process.

1.3.1 Coordinator Persona

Profile
Name: **Kristen Baker** Age: **35** Occupation: **Subject Coordinator at The University of Melbourne**

Motivation
Kristen is the subject coordinator of the Software Engineering Project Subjects at the University of Melbourne. She is constantly trying to find new ways to improve the subject by helping the students learn more. She is also very conscious about the hard work being done by the subject supervisors in managing these various group projects. One area she hopes to improve is the complicated marking process which involves the analysing not only the final product but also the processes the students use to build it.

Goals
• Wants to minimise amount of time spent on administrative tasks and focus more on marking and providing useful feedback to her students. • Wants to have easy access to the quantitative data she needs to mark her students so that she can spend more time analysing the qualitative aspects (e.g., code) • Wants to have a single source summarising a group's work so she can quickly see what the group has been working on recently.

Frustrations
• Spends too much time looking through different software development tools to see if the group has been using the correct processes in their project. • Unable to closely analyse a group's work as she is spending too much time taking note of administrative things.

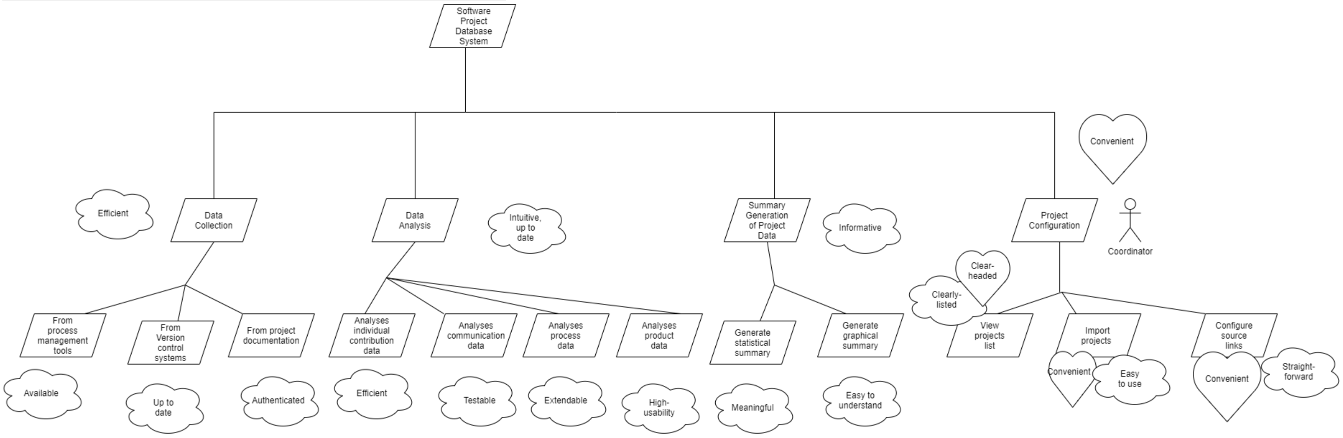
1.4 Goal Models

Goal model from previous team.pdf

Changes from the goal models from the previous team:

- Delete the supervisors role
- Delete "assign supervisors" functionality
- Delete "Collects data from communication tools" functionality
- Delete "View projects visual summary' functionality
- Replace "Add configuration files to project" with "Import projects" functionality
- Add "Configure source links" functionality

goal model.drawio



1.5 User stories

ID

Each user story is identified with a unique ID. The format is set as US_{ID}.

As a, I want, So that

Indicates the user role, goal, and benefit, respectively.

Priority

The priority field follows the MoSCoW scale/method. Under this approach, user stories are prioritized as either 'Must have', 'Should have', 'Could have' or 'Won't have (this time)' according to their importance.

Epic	**ID**	**As a**	**I want**	**So that**	**Priority**	**Story Points**
Home Page	US_1	Coordinator	To import student projects from confluence	I can assess students' work	Must have	2
	US_2	Coordinator	To see those imported projects	I can view each project	Must have	2
	US_3	Coordinator	to delete a project that does not belong to my subject.	I can only keep the projects from my subject(s).	Could have	1
Product Quality	US_4	Coordinator	to see some metrics from a third-party code analysis tool	I can assess the product quality of each team based on the summary	Must have	3
	US_5	Coordinator	to view up to date statistical data	I can view the progress they made each week	Must have	4
Process Quality	US_6	Coordinator	to see the process quality summary of my project teams	I can assess the process quality of each team based on the summary	Must have	12 (break down)
	US_7	Coordinator	to view graphical data on a project team's Confluence activities	I can measure and assess a project team's process quality in terms of task management	Must have	4
	US_8	Coordinator	to view graphical visualisations on a project team's JIRA task activities (i.e., total vs. completed vs. remaining tasks)	I can measure and assess a project team's process quality in terms of task management	Must have	4
	US_9	Coordinator	to view graphical visualizations on the number of code commits with proper code reviews and Continuous Integrations tests in GitHub/Bitbucket/GitLab	I can measure and assess a project team's process quality in terms of codebase management	Must have	4
Communication Quality	US_10	Coordinator	to see the communication summary of my project teams	I can assess the communication of each team based on the summary	Must have	10 (break down)
	US_11	Coordinator	the system to filter out irrelevant messages (emoji only messages) when gathering assessable communication data for each project team	I can measure and assess a project team's communication quality in Slack accurately by excluding irrelevant messages.	Should have	2
	US_12	Coordinator	to view statistical data on the total number of messages sent over assessable Slack channels for each project team	I can measure and assess a project team's communication quality in terms of chat frequency in Slack	Should have	2

	US_13	Coordinator	to view each meeting information including meeting minutes for each project team	I can measure and assess a project team's meeting frequency and meeting quality	Must have	3
	US_14	Coordinator	to view statistical data on comments on GitHub	I can measure and assess a project team's communication quality in GitHub	Must have	3
Individual Contribution	US_15	Coordinator	to see the individual contribution summary of each team member in my project teams	I can assess the individual contribution of each team member based on the summary	Must have	13 (break down)
	US_16	Coordinator	to view statistical data on the number of pull requests involving peer reviews made per team member in GitHub/Bitbucket/GitLab	I can assess the individual contribution of each team member based on their pull requests	Must have	3
	US_17	Coordinator	to view statistical data on the number of lines of executable code made per team member	I can assess the individual contribution of each team member based on the number of lines of code they have written.	Could have	5
	US_18	Coordinator	to view statistical data on the number of completed tasks/tickets on JIRA by each team member	I can assess the individual contribution of each team member based on their task management /completion	Must have	3
	US_19	Coordinator	to view statistical data on the number of complete tasks from JIRA by each team member	I can assess the individual contribution of each team member based on their individual contribution	Must have	3
	US_20	Coordinator	to view statistical data on publishing and editing activity made by each team member on Confluence	I can assess the individual contribution of each team member based on their documentation work	Must have	8
	US_21	Coordinator	to view statistical data on meeting attendance of each team member	I can assess the individual contribution of each team member based on their attendance in meetings	Can have	5
Configuration	US_22	Coordinator	to change the URLs of JIRA, Confluence and Git.	I can make any changes to the links of JIRA, Confluence and Git in case some project teams might want to change their JIRA /Confluence/Git destination.	Must have	3

1.6 Project Analysis Metrics

This document is a general outline about the metrics we will be using to represent the progress teams/individuals have made on their project.

JIRA

- Team
    - Number of Sprints
    - Sprint start/end date
    - Tasks created/completed in each sprint.
- Individual
    - Number of Tasks assigned
    - Number of Tasks completed

Confluence

- Team
    - Project Overview
    - Number of documentations
- Individual
    - Number of comments
    - Number of activities

Git

- Team
    - Number of all lines
    - Number of blank lines
    - Number of classes
    - Number of declarable statements
    - Number of executable statements
    - Number of files
    - Number of functions
    - Number of inactive lines
    - Number of lines containing comments
    - Number of lines of source code
    - Number of pre-processor lines
    - Ratio of comments lines to code lines
- Individual
    - Number of commits
    - Number of comments