2015

Towards better estimation: Improving and supporting the estimation process in agile team environments

SENG402 – Honours Project

Author: Cade Picard

Student ID: 76874721

Supervisor: Moffat Mathews

UNIVERSITY OF CANTERBURY

TABLE OF CONTENTS

TABLE OF FIGURES

1. ABSTRACT

Estimation occurs when a team of developers approximate how much of each resource (e.g. time, effort) an item (e.g. projects, stories and tasks) will take to complete to a specific standard. The success of a project depends on how accurately each item is estimated. The main deliverable of this project will be a system that supports and guides teams through each phase of the estimation process. To do this, we will research current practices, create distinctly identifiable phases in the estimation process, and design and implement a system that supports these phases. Following that we intended to conduct a small pilot study with teams from SENG302. We will analyse data collected by the process and feedback from the teams to gauge if our process improved the estimation process.

2. INTRODUCTION

2.1 PROBLEM DEFINITION

In Agile software development, the product owner discusses what the customer requires of the software with the customer and the stakeholders to create the product backlog. This contains the features and functionality that the customer would like from the application. In the planning meeting, the development team discusses the stories with the product owner and they try to identify and establish the acceptance criteria for each user story or feature. Once they have this understanding the development team estimate how long they believe it would take their team to implement this feature/functionality and whether it is within their planned iteration workload. Estimation is important to software engineering projects because it is the basis of how much functionality you can deliver to the user in a set period of time. Incorrect estimations lead to changes in expected functionality, reduced quality, or increased resource cost (time, labour, finance) which could lead to project failure.

Estimation of software features and functionality is a complex process. During estimation, a team of developers tries to decipher the customer's wants from a story description to create an implementation plan. During this process, experienced teams reflect on past experiences with similar stories that may help to increase the accuracy of the estimation and to try and identify potential risks or complications in the story. The current practice is to complete this process individually and then draw the team members together to share the estimate they believe represents the story. Due to the diverse knowledge, skills and potentially different programming experience of the development team, these estimations can vary quite considerably.

This leads to further questioning of the product owner and other team members to try and establish why some estimates vary so wildly, which could expose "hidden parts of the iceberg" (complexities or unexpected risks that the product owner or other team members did not realise existed in the story). This part of the process can lead to stories being broken down into smaller stories, which can be achieved in a single iteration of development. Once broken down the new stories will need to be re-estimated.

Team dynamics can also cause problems in estimation, as team members have different personalities. A few developers may have extrovert personalities that enable them to ask the questions. This could provide them with more information or a deeper insight into the task. A few of the developers may have introverted personalities and find it hard to ask questions in large meetings, preferring to have a one on one conversation directly with the product owner. These differing aspects of the development team could lead to incorrect estimations if the introverted developer has the most knowledge about the task.

In Agile software development the resources for an iteration are fixed at the beginning of the sprint. The entire team will be involved in the planning process and the length of the iteration is also fixed before the planning begins. This is because in agile development the functionality to be implemented during an iteration is what is flexible and is negotiated during the planning phase. This means that correct estimation is important to the agile development process. If the estimation of stories is incorrect, this can lead to incorrect size and complexity of tasks. Therefore, leading to tasks taking longer than expected to complete. It can also require more resources than expected taking them away from other tasks, which can lead to the failure of the iteration. This could cost the company if it is a contract obligation or allow competitors get the edge in the market.

Current methods of estimation only help with relative size and through putting a label on a story's size, for example the Fibonacci numbers are often used to describe the sizes of software stories. The process of estimating software stories and tasks needs to be broken down into phases and defined. Then a system will be developed so that there is support for each phase of the estimation process. Improvements to the current systems have largely only tried to change the labels that are being applied to the stories to try and encourage more thought being put into the estimation. No current estimation method gives the development team information to help them improve their estimation process and inform them of any improvements in the team or the individuals.

Estimation is a relatively new practice in software engineering, as it has not had the extensive research that more established agile/software processes have had to support or investigate it. Most of the discussions in which the new ideas for the estimation process occur are in forums or blogs that are run by well-known software engineers and researchers. Because of this we will introduce these sources alongside the peer-reviewed literature in this report.

3. BACKGROUND

3.1 AGILE

Agile software development is based on The Agile Manifesto, this was written in 2001 and has four main points:

1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan

The items on the left and in bold have more value to an agile development team than values on the right. [1]

To follow these four main principles of agile development, there are 12 supporting principles that help to outline the processes that can be classified as agile:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity (the art of maximizing the amount of work not done) is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly. [2]

Agile development has the purpose of providing working defect-free software through a short iteration process. The core difference between agile methodologies and the older style waterfall development models is that the agile methodologies are able to respond to changing user/client requirements quickly. [3]

Agile development has a strong emphasis on teamwork, development teams are designed to be self-organising and cover the program from end to end including tests. Managers of agile development do not micromanage or track the development team extensively, the teams are instead trusted to make the right implementation decisions within the acceptance criteria that is set up during the planning meeting. [4]

Agile software projects studied in 1994 had a 16.2% success rate (finished on time and under budget), a 52.7% partial success rate (finished but was late, over cost or lacking functionality), and 31.1% failure rate (project was abandoned or cancelled). In 2004 these percentages had changed to 29% success rate, 53% partial success rate, and 18% failure rate. [5] This is an improvement, but 71% of the software projects still do not come under the success criteria and correct software estimation is a critical process in delivering software on time and within budget.

3.2 BACKLOGS

3.2.1 PRODUCT BACKLOG

The product backlog is organised by the product owner, who obtains the description of the features that the program needs to have from the clients. They then discuss the feasibility of all of these features with the clients, and order them by importance priority so that the most urgent features addressing the customer's needs will be implemented first.

This list is constantly re-evaluated and reprioritized before each sprint so that the best "return on investment" features can evolve with the product and the team can keep delivering software that satisfies the customer's needs. [6] It is common for the requirements set out by the customer at the start of a project to change as they see the solutions provided by the development team and they get a better understanding of their own requirements.

3.2.2 ITERATION BACKLOG

The iteration backlog is a smaller subset of the product backlog that the product owner and the development team have decided can be achieved in the iteration that they are about to begin. This backlog contains a collection of tasks; as the user stories on the product backlog get estimated and broken down, so that the development team can begin development. [7]

3.3 ESTIMATION

In agile development the fixed aspect of the development is the time for each iteration and the resources available to the development team. The aspect of development that is flexible is the functionality. The functionality is negotiated, modified, and estimated to fit into the time box of the current iteration. Unlike in more traditional approaches to software development, where the functionality was fixed and the time and resources needed to complete the product were negotiated. [3]

Three of the main factors that affect estimation are:

1. The quality of the information on which you are basing the estimates
2. The inherent risk of the functionality you are estimating
3. The availability of representative historical examples that you can draw from [8]

Estimation is full of human error; when everyone has a differing opinion and different skills and talents, how do we know how accurate any one developers estimate is?

3.4 OBJECTIVES

The first objective of the project is to research current estimation practices in agile development. This will allow us to analyse the current process and identify areas that need to be improved. As well as identify practices that are common between many different estimation processes. These common practices will have a higher chance of being included in our improved process.

After researching the current processes, we want to design our own estimation process, using the information gathered in the research phase. Our plan is to create an estimation process that helps improve the estimation accuracy over time. It will give the development team the tools needed to analyse their estimation trends over time. It will also give them a forum in which they can discuss story details with the other team members and the product owner if necessary.

3.5 SCALE AND COMPLEXITY

The scale of this project is immense. In an ideal project with unlimited time to work on the solution, we would like to add an intelligent tutoring system to the solution. This would learn from the estimates of previous stories with similar functionality. This would advise the team of any trends in previous estimations that should influence the current estimation. The tutoring system would also track user interaction with the website and monitor the areas with the most frequent use. This information would be gathered so that the development team can analyse whether they are using the solution to the fullest extent. Due to the size of this endeavour, the intelligent tutoring system would be included in a PhD project timeline. We will be trying to design the current solution with this in mind to make the design and implementation of the tutoring system in the future easier.

The solution would also contain an import function so that the product owners did not have to manually enter the backlog stories one by one. Due to the low priority of this feature and limited time, it may not be implemented in the final solution of this project.

Another ideal feature of the solution would be to walk the developers through the process in a linear workflow, but still allow them to move forward and back through the estimation process. Due to time constraints or possible technology limitations the workflow may consist of an explanation of how the process works and the links to each step in order.

4. PROPOSED SOLUTION

We will implement a software solution that helps teams estimate user stories in an Agile environment during software development. It will act as an interface to any system that a company may use to control their product backlog. They will be able to import their backlog into this system, then their employees will be able to log in to their team's personal backlog and ask questions to the product owner, start team discussions, and estimate the size of stories.

To do this, we will break down the estimation process into separate phases. Once the process is broken down and the estimation process investigated to reveal the key aspects, requirements will be gathered for each phase and software will then be created to meet these requirements.

This software will also keep track of their estimations at a team and individual level. When the story is complete, each employee will be asked to estimate the stories' costs. Now they have the advantage of hindsight, this data will then be used to show trends in estimation accuracy and any improvement over time that the team or individuals experience.

This software will not simply be a program that digitises planning poker. There are a few apps for that already on the market. This program will lead the team through all the phases of the estimation process that we will identify. For example, it will gather estimates from each team member, emphasise the estimation extremes to the team to stimulate discussion, have areas for each story for questions and answers, and then re-iterate through the estimation process until a consensus is reached. It will also collect data for future analyses and learning, particularly during similar future estimates.

The initial solution design for this project has the ability to import the product backlog from the developer's backlog software. As this is not the most important part of the research project it will remain as a simple import text file functionality, in either comma separated format or possibly JSON format. This will import the backlog into the system and allow it to be used by the product owner and the development teams.

The system is being designed to run off a server. It will have a login system, the product owner will act as a system admin and be able to create the product backlog, and also will be able to create and assign users to teams. Each team will have their own backlog which will be a subset of the overall product backlog. This is because the product owner should only have to change one story to have the changes to that story propagate to all team backlogs.

The teams need individual backlogs because each team will have separate priority for each of the stories and they will only want to see the stories that apply to them e.g. UI stories for the UI team. The teams will see the backlog as a list of stories sorted by priority. They will be able to expand the story to read the description and the acceptance criteria. There will be options once a story is selected. These options will include:

- Asking a question of the product owner to elaborate or explain more about the story
- Add comments to the story that can be used to communicate ideas with other team members
- Ask to change the priority of this story
- Estimate the story before the planning meeting for use with retrospectives and learning
- Split the story into smaller stories as it is an epic

5. AGILE DEVELOPMENT PROCESS

Agile development encompasses the entire software development process. Ranging from brainstorming ideas with a customer all the way to version control and maintenance of a finished product. The process is modifiable and extensible, allowing teams to customise the process to suit their organisation and team structure.

The development process has multiple threads that run in parallel. These include; estimation, bug tracking, testing, documentation, customer feedback, sprint and feature planning, resource management, business management, reengineering and refactoring, technical debt management, feature development, and more.

Estimation is but a single thread in this development process. It is a complex process that can be personalised for each agile development team. Estimation can either help or hinder the rest of the agile estimation process through accurate or inaccurate estimations. We wish to improve this process by improving the estimation accuracy of agile teams and enhance the entire agile process through the interconnectivity of the different threads.

5.1 AGILE ESTIMATION PROCESS

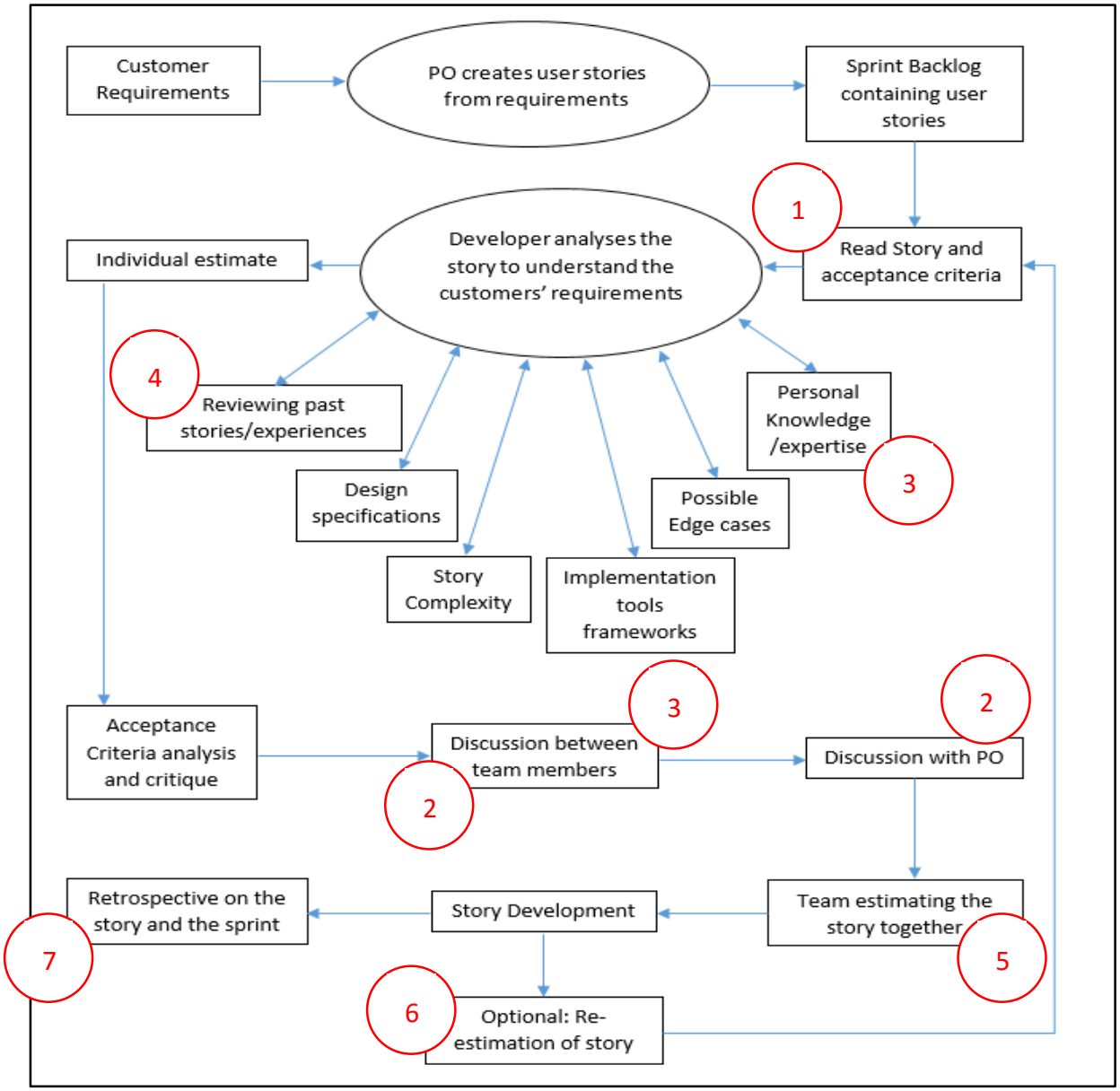


Figure 1: General agile estimation flow chart

Figure 1 above shows the basic agile estimation process. This is the estimation process that we wish to improve upon. This process we have outlined is a general process and not all agile teams will complete every step as shown here. Quite often the estimation process contains only the minimal steps for a coherent workflow. These are the stories on the backlog, team estimation, development, and the retrospective.

We wish to increase the communication between team members before the planning meeting. So that all aspects and edge cases can be explored and analysed before the planning meeting. Meeting times in industry are limited and when planning for a sprint, meetings can easily run overtime if discussions are not discouraged. This increases the chance of the team coming up with an incorrect estimation.

The 1 to 7 numbers in circles on Figure 1 above correspond to the phases of our improved estimation process outlined below. It shows the points where we believe we can improve the current estimation process and the aspects of the current process that are necessary.

5.2 PROPOSED ESTIMATION PHASES

We have broken down the estimation process so that we can gather requirements and begin the planning phase of the project. We had to discuss what we believed the perfect agile estimation process is, and then break down how the process worked into phases as shown in Figure 2 and Figure 3 below. Then we had to design a software solution that would fulfil each identified phase. The system needs to be intuitive and simple so that the process is easy and fast for the development teams. If this system is too complicated or too time consuming then the development teams will not be inclined to use this solution, but the system needs to allow teams the freedom to complete all of the optional phases that we have identified if they are inclined.

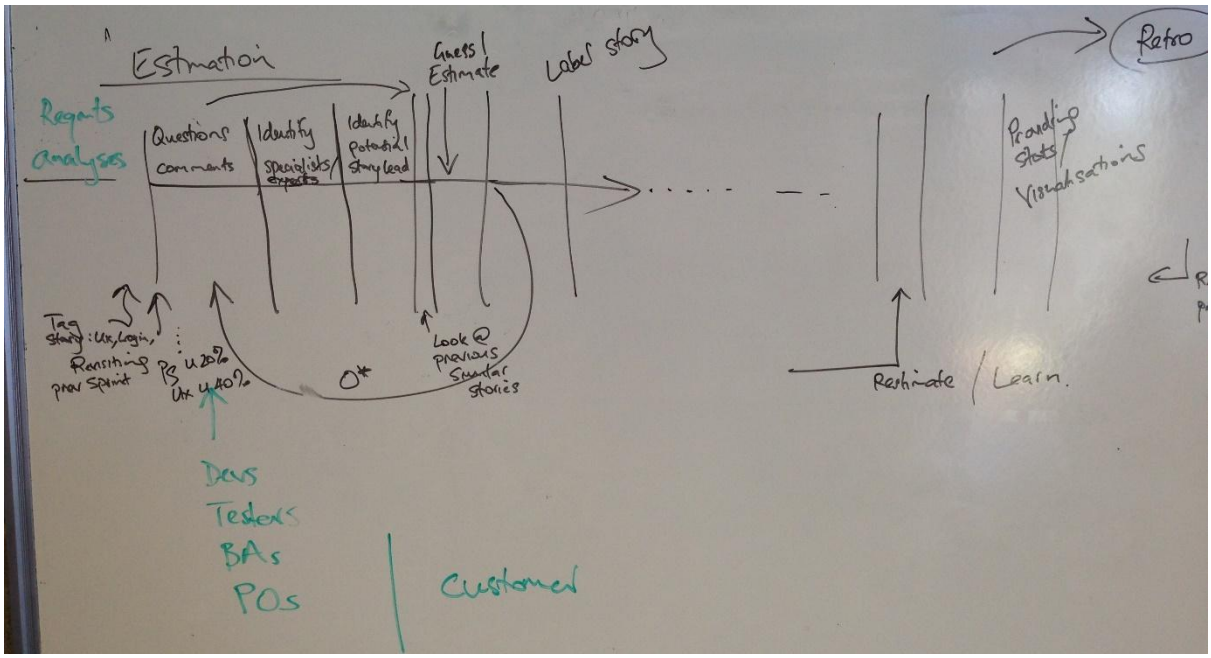


Figure 2: Diagram of the estimation phases produced during brainstorming session

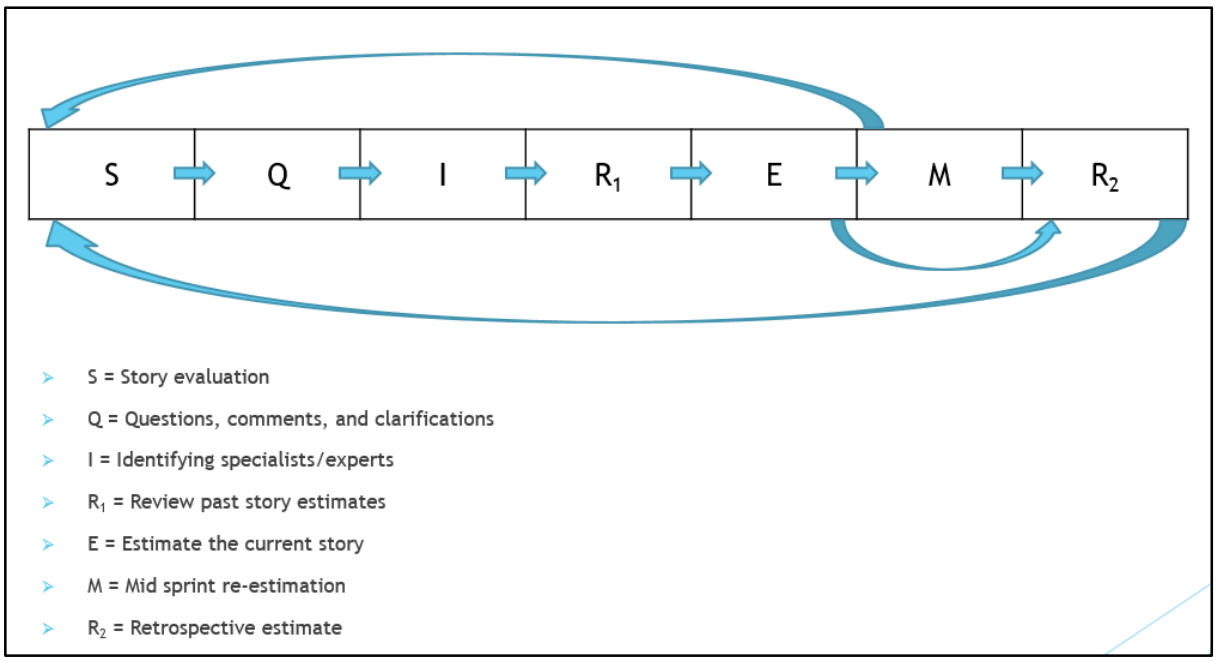


Figure 3: Improved estimation flow chart identifying our seven phases

5.2.1 PHASE 1: STORY EVALUATION

Requirements analysis needs to be completed by the product owner and the stakeholders to complete the product backlog and all of the acceptance criteria. Once this stage is complete then the development team can read through the stories and evaluate the stories. To begin to understand if the story falls into their field of knowledge, then they can either research to find out more information or start brainstorming possible solutions.

This initial reading and brainstorming should be completed by each member of the development team before the start of the planning meeting. This allows the planning meeting to be more productive as the team members come into the meeting with questions and understanding already. This means the meeting time is not wasted explaining the acceptance criteria of the stories. This leaves more time during the planning meeting for the following phases.

5.2.2 PHASE 2: QUESTIONS, COMMENTS, AND CLARIFICATIONS

This phase of the estimation process is to allow the development team to clarify any uncertainties in the story description or the acceptance criteria. The team can ask questions or comment what they think the customer is trying to achieve with the story. Also discuss how the developers believes this will fit into the current iteration of development and the overall product.

These questions may bring to light ideas or information that has not occurred to other team members or the product owner. If the story has complications that the product owner was not aware of then the story may have to be postponed until a future iteration. This is because the product owner will need to re-evaluate the story with the stakeholders with the new information.

This phase makes sure that the development team is all on the same page and that they all fully understand the meaning behind the story.

5.2.3 PHASE 3: IDENTIFYING SPECIALISTS/EXPERTS

In the previous phase the team understands the background and the details of the stories. This allows them to understand which fields of knowledge will be most useful for this story. Team members with the most knowledge or experience can bring other team members up to speed on the complexities of the story. In addition to this the team members may know of other developers in other teams that have knowledge that would be useful to complete this story.

Also the story may require knowledge or input from outside the development company. Depending on the technology or service being developed there may be laws or industry standards that need to be met by the software solution. To get the correct information and how to comply with these laws then experts will need to be consulted. This phase allows the development team the option to specify these experts and how to contact them. As well as the reason for including them, be it knowledge or experience, or that the story is tailored to a specific user that needs to be consulted.

5.2.4 PHASE 4: REVIEW PAST STORY ESTIMATES

As estimation in any field uses educated guesses, the team members will draw on past experiences to help them judge the size of the stories. Using the story's tags to sort similar stories, the team members can access these stories. This will allow them to see if any past stories are similar in content, if they are similar then they can see what they estimated the previous story to be. This will help them make more accurate estimates of the current story.

5.2.5 PHASE 5: ESTIMATE THE CURRENT STORY

The development team should now have a comprehensive idea of what the current story entails. Not all complications or setbacks can be seen during this estimation process, but the team should now be able to put a value against the story.

One of the problems with current estimation methods is that less confident team members let their own estimates be influenced by other team members. To counteract this, the individual team members would make their estimates separately. Then, once each team member has completed their estimate, the list of estimates would be shown to all team members and the average value would also be shown.

Once the estimates are shown to the entire team the process encourages teams to proceed back to phase 2: Questions, Comments, and Clarifications. Looking at the list of estimates there may be some discrepancies. To get a better overall estimate the team members with varying estimates should explain why they gave that estimation. This may bring to light factors that the other team members are not aware of.

5.2.6 PHASE 6: RE-ESTIMATION DURING SPRINT

As an optional phase of the estimation process, development teams may have complications appear during the iteration that require re-prioritisation of the stories. This may require the re-estimation of

the stories currently being worked on in this iteration. This would begin the process again from phase 1.

5.2.7 PHASE 7: RETROSPECTIVE

One other aspect of the estimation process that we would like to improve is the feedback from estimation. If a team constantly underestimates certain stories (e.g. user interface stories are always 20% underestimated), then we would like to bring this to the team's attention. They would be able to see this value during phases 4 and 5 so they can get a more accurate estimate of the current story.

Also during the retrospectives, the development team will want to know if their estimations are improving or not. Being able to see the estimations over time and in a visual way such as in graphs. Allowing the team to easily see if the trend is improving or not, and will help the development teams understand where they need to improve their estimations.

To try and get an estimation value that is as accurate as possible, the development team will be asked to complete an estimation of the stories completed in this iteration after completing the stories. Using hindsight to judge the size of the stories, the team should be able to accurately state the actual estimation value of the story. This can be compared to the original estimation made at the start of the estimation process.

5.3 REQUIREMENTS FOR THE PROPOSED ESTIMATION PROCESS

Table 1: Software requirements identified from our improved estimation process

Ability to tag stories at any time	Stories should be able to be tagged to identify the fields that apply to the details of the story e.g. UI for user interface design or changes.
User Management	There should be 2 levels of users, an Admin user e.g. product owner, and team members. The admin user will be able to create teams and assign users to be team members
Open forum question	Team members will be able to open a story and create a forum under the story with an initial question that the other team members can see and reply to
Create discussion topic	In each forum the team members can create further topics of discussion that the other team members can see and comment on
Close discussion topic	Once the initial issue is explained or resolved then the team can close the discussion topic to show that the issue has been solved
Increase Question Priority	These questions will have a priority system that orders them by importance. The team will be able to increase or decrease the priority of the discussions
Add labels to team members	To help identify team member skills and knowledge the team should be able to apply labels to themselves and each other
Specify other interested parties	Development and knowledge is not restricted to the team members only, the team should be able to

	specify other people outside the team and give a small reason as to why they are being included
Optional story lead	If the teams use story leads in their development processes they will be able to specify a team lead for each story
Resource management	If the story requires special resources then the team can specify what the resources are and when they will be used
Configure estimation labels	To allow the team to custom tailor the estimation values that they wish to use, the labels used in the estimation process will be configurable.
Exclude members from estimation	In case of team members not being present during the estimation process then they can be excluded so that the estimation data is not affected
Show estimates after everyone submitted	To allow all of the team members to express their estimates without influence from other team members the final estimates and the team's estimates should not be shown until all estimates are collected
Estimation page with results and estimation	Once the estimates are collected they should be displayed on a page that show the estimates and the average estimate
Call for re-estimation – list as separate estimation	If complications appear during an iteration the team should be able to call for a re-estimation of the story if needed
Define retrospective estimate milestones	Every story should be re-estimated at the retrospective when the team look back at the sprint using hindsight.
Visualisations	The data should be collected over time and be able to be displayed in data or graphical format.

Table 1 above, outlines the requirements that we generated during our research into the current agile estimation process. We added requirements that we believe would help to improve the current agile estimation process and we want these requirements to be present in the final prototype of this project.

6. ANALYSIS OF THE IMPROVED PROCESS

To help gather information as to the effectiveness of retrospective estimation. We carried out a small investigation into the SENG302 team's estimation practices. We informed them that we would be recording their estimation values that they had agreed upon in the planning meetings. We set out to not interfere in their estimation process as we did not want to influence the team dynamics. After they had finished the sprints and completed the stories, we asked them to individually re-estimate each story that they had completed now that they had the advantage of hindsight. This is the process that we wish to use to aid developers in increasing their estimation accuracy. Getting the developers to retrospectively estimate completed stories.

In Figure 17 in Appendix 3 we can observe that SENG302 team 4 agreed with their planning estimations on certain stories. However, there are other stories where they have changed their estimation through hindsight. For example, the "Basic Basic Search" story was planned as a 20-point story. However, while estimating retrospectively the individuals ranged from 5 to 8 points for the story. The opposite effect occurs in the "Display list order 1" story, where the planning estimate was 2 points and the team's revised estimates range from 3 to 8 points.

In Figure 18 in Appendix 3 we can observe that SENG302 team 5's retrospective estimations match very closely to their planning estimations. In most estimations there are less than half of the team that differ from the planning estimation. This could be due to the fact that their stories are all estimated between 1 and 8 story points. This does not represent a larger, possibly more complex story where incorrect estimations are more likely.

The data for these graphs was gathered over 2 sprints for these teams and is only a small section of their entire projects workload. The teams had been working together for 3-4 sprints at this point of the project and their estimations should be converging, as the teams begin to better understand the team's work ethic and abilities. It helps to show that in certain cases retrospectively estimating a story can lead to very different estimations. This is information that should be used by the development team to learn and improve their estimation accuracy over time.

7. SERVER FRAMEWORK: WAMP

WAMP is being used as the server host for the web tool. This is because we wish for our solution to be web based so that it can be accessed from any computer regardless of operating system or location. In the future the application would be usable/accessible from mobile devices, this limits the application from being a native desktop application. Another feature identified as crucial is the ability to use the application while the development team is not co-located. Some industry businesses have international team and require the development team to have access to the application from anywhere in the world, as well as having access 24/7 from any location.

WAMP is an environment that is easy to install and setup, commonly used to run servers around the world. It also has accompanying Linux and Macintosh versions. Many information technology and software development businesses use a version of WAMP/LAMP to run servers. These are usually used for hosting bug tracking software e.g. Fogbugz/kiln, Jira, or Jenkins. Alternatively, they are commonly used for testing and prototyping. These servers are the environments where our estimation solution would be hosted and used in an industry environment.

The case study that was planned to be undertaken would be run from the university servers. The university servers are Linux machines that run LAMP. The conversion from WAMP to LAMP should be inconsequential, in respect to the development of the project. This will allow the students to have 24-hour access to the service from the university campus and from their homes.

Alternatives to WAMP/LAMP are:

- XAMPP
  - Apache, MySQL, PHP, and Perl
  - Can be used on Windows and Linux
  - Perl will not be used in this project
- AMPPS
  - Apache, MySQL, PHP, Perl, Python, and Softaculous auto installer
  - Works on Windows and Linux
  - Perl and Python won't be used for this project

These alternatives would perform the same role as WAMP/LAMP and may be preferred by other users. The main deciding factor in this decision was the support of LAMP on the university servers that is managed by the system administrators.

8. APPROACH 1: CUSTOM WEB APPLICATION

A web tool was started to be developed in JavaScript and it used the CodeRunner framework with a MySQL database to store the information. This framework and language are chosen because I have some experience with them from a previous web architecture course. Also, the software solution needed to be kept fairly simple to reduce complexity and allow the solution to be developed within the time frame.

Figure 4: Approach 1 custom web application Product Backlog Page of the web application

8.1 MAIN PAGE

The application had a main page with a very simple list to create a product backlog. The product backlog is stored in the MySQL database. A navigation bar is present at the top of the page for page navigation. We had been researching how to get the stories in the backlog to expand when clicked. This is to show the description and the acceptance criteria. We had to research how to get the right hand side of the list area to show a

different view based on which story has been clicked on. This was the basis for the individual story forums and discussions.

Here are our objectives that we wished to complete with this custom web tool:

- On the Main Page the admin user would be able to view and edit the overall backlog, this would then be shown to each team separately. This allows the different teams to filter and order the backlog to suit their team the best. As shown in Figure 5 below.
- Once a team member logs in they would be shown the product backlog and would be able to select the items on the backlog. These will expand to give a more detailed explanation of the story and the acceptance criteria.
- Once a story is selected then the options to create a forum or open a discussion question become available. The team members would be able to give the questions a priority. Also, the option to tag a story will become available, these tags will be shown under the story headings.

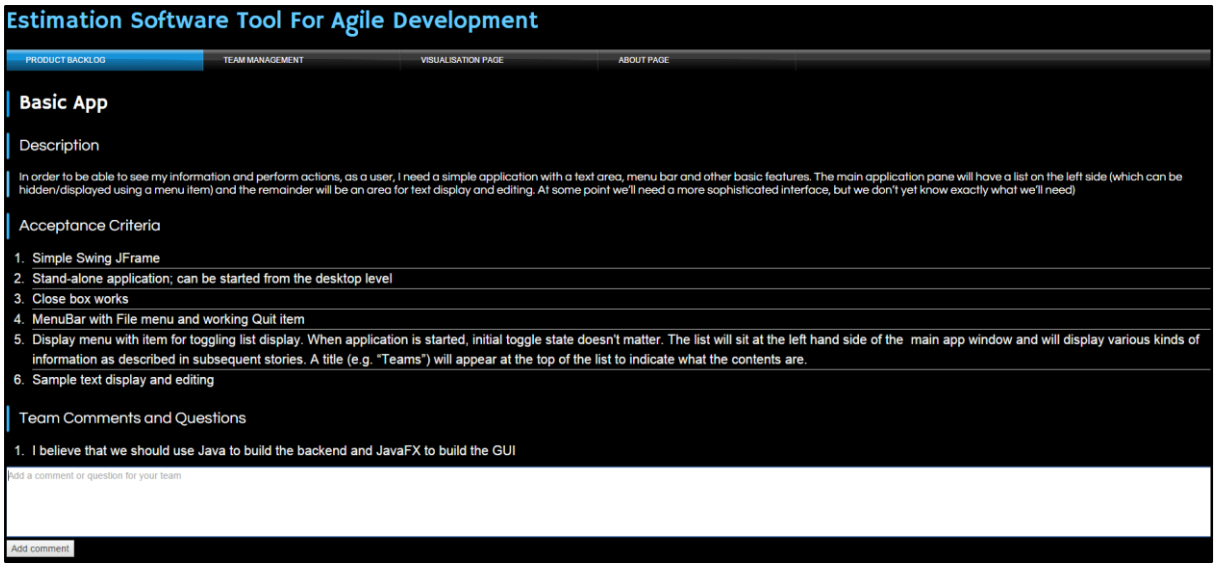


Figure 5: Story page in the original solution development before technical decisions

8.2 LOGIN PAGE

- The application will start with a login page that uses a MySQL database to store the user's information.
- This will allow the users to be shown the correct backlog information, relating to their team or as the product owner.

8.3 TEAM PAGE

- Ability to see who is in the team and to add labels to the team members to show expertise and knowledge.
- Configuration of the estimation labels, let any team member change them and let the team decide how to monitor them.
- Exclusion of team members from the estimation process. Anyone can exclude any team member in case the team member is unable to exclude themselves before their absence.

8.4 ESTIMATION PAGE

- Here is where the estimates for the story would be submitted and shown.
- List of team members and their respective estimations, there will also be an average estimation value.
- Teams will have the ability to do a new estimation at the retrospective.

8.5 VISUALISATION PAGE

- This page will have access to the data from all estimates and will allow the users to see their estimation trends over time through tables and graphs.
- This would be used during retrospectives to help show the team how accurate the current sprint's estimations are and how that compares to their past estimations.

8.6 MVC IN JAVASCRIPT USING CODEIGNITER

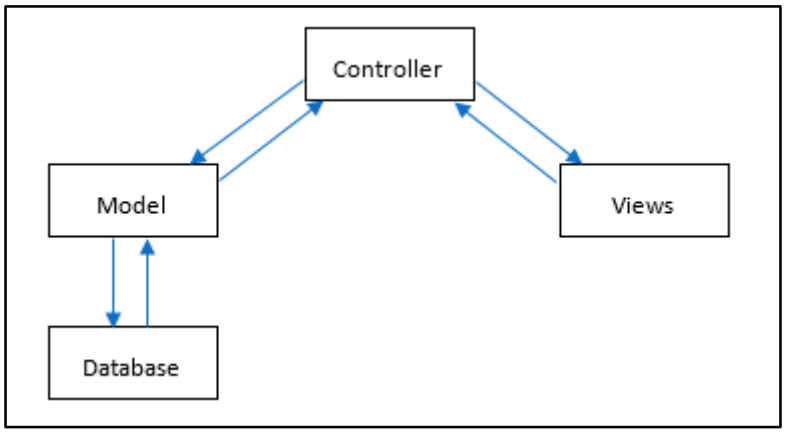


Figure 6: MVC in the CodeIgniter framework

The original solution being explored for this project was using CodeIgniter framework. This helped to segment the JavaScript code into the Model View Controller pattern as seen in Figure 6 above. The view consists of PHP files that utilise PHP, JavaScript and HTML to create the user interface that the user will interact with in the browser. The model contains the PHP file that connects to the database and uses a combination of JavaScript and SQL to extract the information from the database. The controller works as an intermediary between the model and the views. It calls the model functions when needed or prompted by the information coming from the views and can store it in the scope for access by the views. It also controls the responses from the views that help change the page displayed and sends the user's input into the model if necessary.

8.7 TROUBLES WITH POSTING DATA

While setting up a page to show story information in the initial solution design (as seen in Figure 4). This page was written using a combination of HTML, JavaScript and AngularJS. We were trying to add a comments section so that the team members can discuss aspects of the story before the planning meeting. This was a part of the development of our original solution design. Corresponding to phase 2 of our improved estimation process.

We ran across the problem of trying to post data into the database when submitting the comment. This was discovered during the testing of the comment functionality. The data was being posted into the database but because of the post method being used, the spaces were being converted to “%20” and newlines in the text were being ignored. While researching how to properly implement this functionality and fix this dilemma, we looked into forum implementation, comment posting using AngularJS, login systems, and existing software solutions that implement these systems.

During this research the complexity of these systems became more apparent. To develop them ourselves would take a significant amount of time and effort. By the end of this project's life cycle, we would not have completed these features to the point where we could investigate and test the research proposal sufficiently. The focus of this honours project is not to have to implemented all aspects of agile development and web related tools. It is to have the process set up for use and evaluation by developers/experts. With this in mind and the discovery of content management systems such as WordPress, Joomla, and Drupal, we decided to move to the content management system as a base for our web estimation tool.

9. CHOICE OF CONTENT MANAGEMENT SYSTEM

9.1 WORDPRESS

WordPress [9] is a user friendly content management system (CMS). It is used for news and blogging websites, it allows multiple authors and is very easy to customize. It powers sites such as New York Times, CNN, and Forbes. Because it is so popular it has developed a large following community and has had hundreds of thousands of plugins developed for it. [10]

Some disadvantages of WordPress are the security vulnerabilities. WordPress websites are commonly targeted by hackers. Third party plugins are needed to help increase the security of the websites. As our estimation process will have backlog stories that relate to confidential industry intellectual property. We need to make sure that hackers and unauthorised users cannot access sensitive data.

It also has limited content management capabilities; it was originally designed as a blogging platform. Our estimation solution is going to incorporate statistical analysis of user input data. To achieve this in WordPress would be difficult.

9.2 JOOMLA

Joomla [11] is also an open source CMS. It has been downloaded over 30 million times, and it powers sites such as Cloud.com and Linux.com. Joomla is not as user friendly as WordPress, the learning curve is steeper but the UI is polished and flexible. It also has a large development community, not as large as WordPress, but still has a large array of free, open source plugins. Joomla offers a wide variety of extensions e.g. templates, languages, and components. This allows for a more diverse page than WordPress. [12]

Joomla requires more investigation and experience to become proficient. The website lacks a comprehensive access control list (ACL), this is a list of permissions that gives specific users access to the various parts of the website. As our solution is being designed for use in an industry environment, we need to have different levels of permissions for our developers and product owners.

10. APPROACH 2: DRUPAL BASED WEB ESTIMATION TOOL

Drupal [13] is also an open source CMS. It is extremely flexible with a very large array of modules that are open source. The initial installation is very standard and simple, but with the accessibility of the Drupal codebase developers are encouraged to create their own modules. Drupal also comes with strong version control and a built in ACL, these are very important to our solution that will be deployed in an industrial environment. The extremely flexible nature of Drupal can work in the favour of our project. We want to have the basic features of a CMS, a selection of stories that can be read and commented on by the development team. However, we also want to be able to estimate, analyse the estimations, and display them to the users. [12]

Drupal does come with a very steep learning curve and a lack of themes. Themes are not important to this solution at this stage of the development. However, the steep learning curve will be a risk for the project's success.

Using PHP in Drupal modules can be a security risk. If malicious attackers gain access to a node that contains PHP code, if they have edit permissions, then PHP code can access the database and extract sensitive data. To combat this security threat, only authorised accounts such as the product owner, team manager, or developers will have edit permissions for PHP code. Other accounts, such as UI experts, testers, specialists, or anonymous users will have permissions to only view nodes that contain PHP code.

In this project we have elected to create a Drupal hosted web estimation tool and build upon the open source modules that are provided on the Drupal website.

We have researched into a few modules to help with the development of the solution:

- Drupal comes with a search feature already implemented and ready to use, this is very useful for fast navigation to a known aspect of the website.
- Users and the login system are core modules that will be necessary for this project.
- Fivestar:
  - Is a star rating system that can be used to rate stories or comments, shows promise in the field of drawing attention and showing support to important user comments.
- Rate:
  - Module similar to Fivestar that rates stories and comments, it has more configurable options than Fivestar. It also has a wider range of voting widgets that allow for a more informative rating system.
- Advanced Forum:
  - The development team can create a discussion forum on any topic, be it work related or a social forum. It was not ideal to host the backlog as a forum with the stories being forum topics. The layout was too busy and was hard to navigate easily when there was a large amount of backlog stories. Can still contribute to the project as a way of asking questions of the product owner, which are not related to a story.
- Improved Polls:
  - Allows us to add polls to the stories themselves and not have them as a separate entity. This was a promising direction but the module does not allow for multi answer polls and the user interface is confusing and unclear. Due to these reasons we decided to not use this module in our solution.
- Charts:

  - Takes data gathered from PHP functions to create a better user interface to read the data from polls and stories. Graphs can be made for each individual story to show the estimations of the individual team members in the planning, re-planning, and retrospective estimations. Leads to the team being able to see who are the most accurate on particular stories.
  - Created an overall graph that shows the final values for each story and shows them next to each other to show an overall view of all stories. Allows the team to see if their estimations are improving over time.
- References:
  - Small module that enhances the user experience. Allows for nodes such as stories and polls to contain links to other nodes. We have used this module to allow the development team to place links on stories to the estimation polls and vice versa. Saves the user's from having to travel through the estimation lists or the backlog to find a specific poll for voting.
- Views:
  - One of the most used modules in Drupal.
  - Allows for easy development of lists of data types. This is perfect for creating the backlog, as it is a list of nodes of type "Story". The estimation polls can also be grouped into lists through the taxonomy tags.
- Ctools:
  - Background module that enables a large portion of functionality needed by other modules such as views and charts.
- Comment Goodness:
  - Comment extending functionality module. Enables more options to sort and display the comments on nodes such as stories and polls. Allowed us to sort the comments by newest first instead of the default oldest first.
- Core module: Taxonomy terms:
  - Used to add tags to certain nodes in the web tool. Can create multiple vocabularies and terms. For example, story tags (GUI, Model, Database, Networking, Engine), or poll tags (planning, re-planning, retrospective). Used to filter content on views and through PHP for the graphs.
- Core module: Nodes
  - Allows for creation of templates and new node types. Stories can be created and assigned many different fields. Story graphs can be created and assigned a template that contains the PHP code necessary for the graph creation.
- Clean URLs:
  - Clean URLs is a user experience and peace of mind module. It cleans the URL so that it is more human readable and easier to see the relationship between the nodes. It allows for easier creation of links to certain known pages.

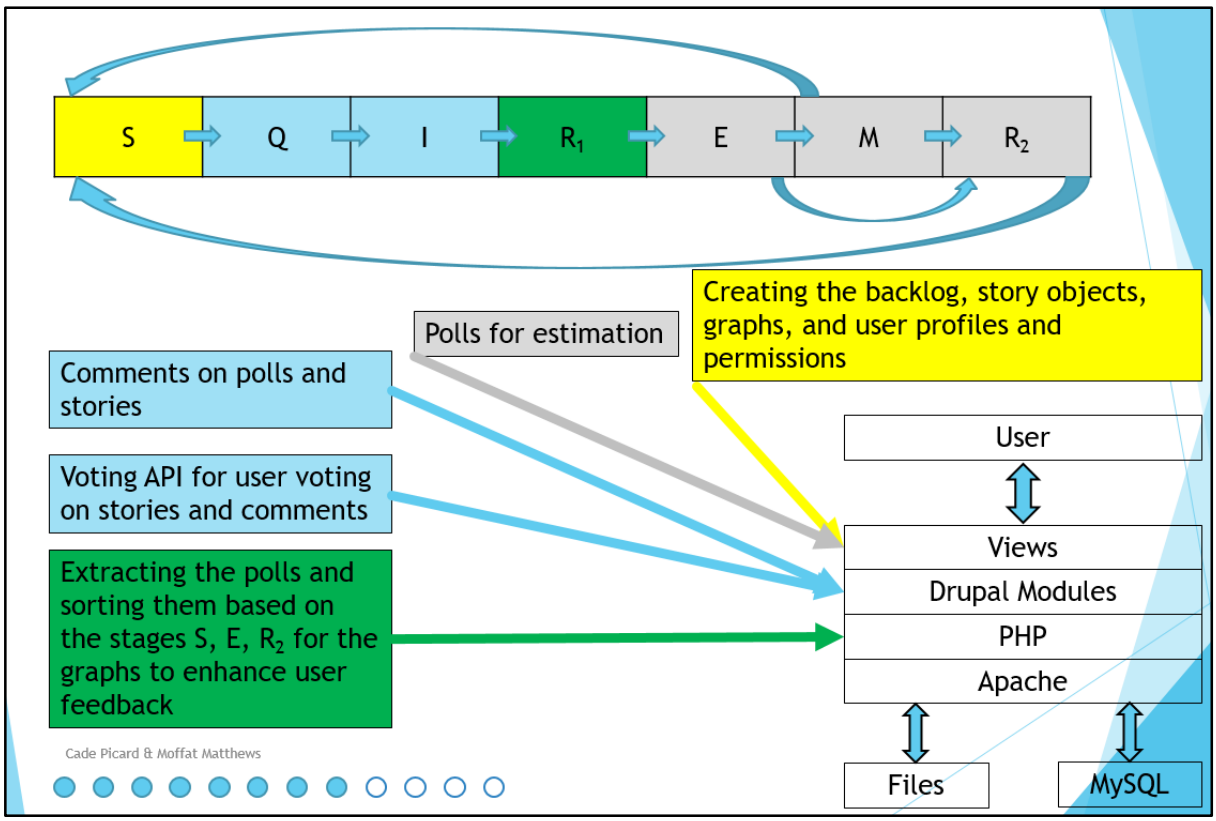


Figure 7: Interaction between the improved estimation process, the Drupal modules, and the Drupal architecture

Figure 7 above outlines the architecture layout of the Drupal based web estimation tool. It relates the different aspects of the Drupal tool to the layers of architecture, as well as to our improved estimation process that is shown in Figure 3 on Page 13.

11. SOFTWARE ENGINEERING PROCESSES

11.1 WEB SECURITY AND MALICIOUS ATTACKERS

When we first investigated Drupal as a possible web based content management system. We started a website on the university servers that hosted a simple forum. We were investigating the forum module to see if it would be a good base to host the product backlog. The website allowed any user, even anonymous users to post content to the forum. Within 36 hours the forum was being spammed by an attacker. This attacker had found that he could post information into the forum anonymously. A script was set up and the forum was getting a new discussion thread started every 20 seconds. Once we saw the traffic flow into our “empty” website we investigated and found out forum to contain thousands of spam topic posts, with titles similar to spam emails. Advertising Viagra pills and get rich quick schemes. This reinforced the idea of web security and access levels to the Drupal web based estimation tool that we ended up implementing.

11.2 TEST PLANS AND CASES

Testing has been focussed around unit testing and integration testing using the manual testing format. While developing our own solution and using MVC in JavaScript, we were testing the functionality that we were adding e.g. the adding of comments, and the links between pages. Using this testing we were able to uncover the problem with the comment posting methods. The data was being posted into the database but was not human readable due to the spaces being replaced with “%20” and newline characters being ignored. No test cases were written for this feature, but we were manually testing the edge cases and usability of the feature during development.

Other testing that has been undertaken is to test the functionality and accessibility of the Drupal website. Due to Drupal having administrator, registered, and anonymous users, there are different actions that each type of user should have access to. The development team would be registered users and be able to create estimation voting polls and add comments to the backlog stories. Other employees would be registered users, created by the administrator as needed. Anonymous users would be anyone outside of this scope, they should not be allowed to edit any of the sensitive information. The product owner would be an administrator user, this would allow them to create and assign user accounts to the development team, create backlog story objects, and change the website aesthetics e.g. page titles or names, themes, or website layout.

During the implementation of the Drupal based solution, the program was in use constantly as manual tests were carried out to ensure the functionality behaved as expected. This helped to uncover missed edge cases such as graph errors occurring due to users not voting on certain estimation polls, or polls missing all together. It also uncovered the lack of support for non-numerical estimation values.

Acceptance testing also occurred as the requirements were tested against the final web based estimation tool. This uncovered missed functionality and exposed requirements that no longer fit the current product design specification.

11.3 DIFFICULTIES WITH TESTING

11.3.1 CHANGING REQUIREMENTS

As the requirements for this project are subject to change at any time, and with the ongoing research into the topic. The likelihood of the requirements changing is high. With this in mind we are aiming for the rapid development process through iterative development. To help get feedback from the industry and experts we are using prototypes as a form of testing. As well as a way to show the interested parties the layout and functionality included.

11.3.2 BUILDING THE SOLUTION USING OPEN SOURCE SOFTWARE

Using the Drupal website to build the project solution, it allows us to build upon open source software developed by other developers. This can be an advantage or a disadvantage depending on the quality of the modules in question. This should reduce the complexity of development of the solution, and increase the functionality of the final project submission.

Another interesting aspect of developing on an open source framework is that if we develop a module to complete a certain task on our website. Then we can publish this module on the Drupal website and contribute to the community.

11.3.3 SELENIUM

Selenium [14] is a possible route to having a robust, browser-based regression automation test suite. This will be explored to see if it is compatible with the WAMP/LAMP server and the Drupal website. This would help to replace the current manual testing that we are currently employing to test functionality.

Selenium Remote Control can be used with any JavaScript enabled browser, and also with any language. It tests the webpage by injecting JavaScript code into the webpage to interact with the components on the page.

12. REQUIREMENTS ELICITATION FROM INDUSTRY

Utilising the input of another university course: SENG404 requirements engineering and software architecture. For the second assignment, we designed a survey to elicit requirements for the estimation process from industry. We will be utilising the survey to help get a more detailed view of estimation from a wider range of software engineers. We wish to draw on the vast experience of agile developers in industry and use their knowledge to improve our estimation process and prototypes.

The University of Canterbury has a professional license with the Qualtrics [15] enterprise survey platform. This allows for unlimited responses and a wider range of survey features and customisation. Qualtrics allows the downloading of response data in multiple formats. For example: comma separated values, statistical package for the social sciences, text file, xml file, HTML file, or as a ZIP export. This along with the report generation supported by Qualtrics allows for a wide range of statistical analysis to be undertaken on the respondent's data.

The survey that I developed for the SENG404 assignment. Has been taken, expanded and revised through the help of Dr. Matthias Galster, Dr. Moffat Mathews and myself working collaboratively. The survey has sections to gather information on a range of topics:

- Personal profile:
  - To understand the agile and programming experience of the respondent.
- Team and organization profile:
  - What methodology of agile in use in the respondent's teams?
  - What domain is the organisation/team in?
- Estimation:
  - What does the agile team estimate in terms of effort e.g. features, stories, or bug fixes?
  - At what planning stage is effort estimated?
  - Which estimation techniques are used, if any?
  - What factors are used when estimating effort e.g. complexity, time, or expertise?
  - Are any tools used to aid the estimation process?
- Estimation accuracy as a team:
  - Are estimations usually over or under estimated?
  - What are some of the causes for these incorrect estimations?
  - Does the team reflect on previous estimations to improve?
- Estimation accuracy as an individual:
  - Are the respondent's personal estimations usually over or under estimated?
- Final open questions:
  - Allows respondents to express thoughts that we have not directly asked.
  - Problems or complaints from their current estimation process.
  - Suggestions for improving the current estimation process.

Using software engineering conferences and mailing lists we will get in contact with a wide array of software engineers, with a myriad of backgrounds and skills. We wish to draw on their experience and knowledge to improve our solution. The results of this survey will be important to this software application but will not directly impact this project due to the time frame. The results of this survey will be published in a separate academic paper after the end of this academic project.

13. DRUPAL BASED WEB ESTIMATION TOOL

13.1 HEADER

The web tool has a header that contains the application title “Estimation Tool”, a search bar for searching for specific content in the web tool, and a set of menu tabs that allow users to navigate through the improved estimation process. As well as the standard web “My Account” and “Logout” buttons.

13.2 HOME

The home page is the default Drupal home page where new content created in the web tool appears. It holds a list that is sorted from newest first and will give a quick link to any new content e.g. new stories created by the product owner, or a new estimation poll for a story for the upcoming sprint.

There is a “Block” in the left hand margin of the page where there are three links. This block is called the “New Content” block. These links appear only on certain pages in the web tool. They are for the creation of new content e.g. a new story, estimation poll, or graph of a stories estimates.

13.3 1-3 BACKLOG

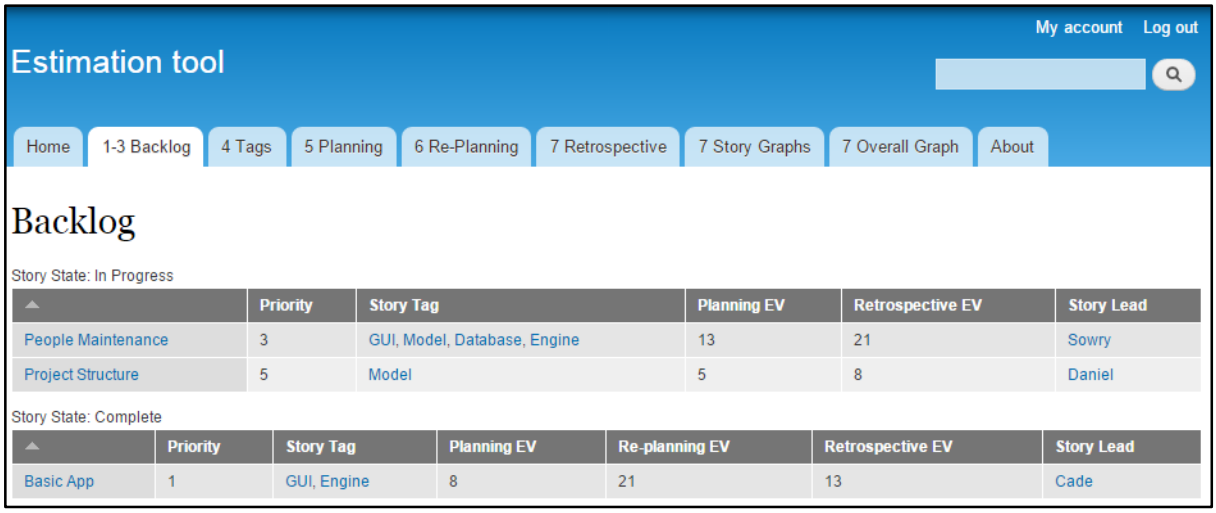


Figure 8: Web based estimation tool backlog page

Figure 8 above show the backlog page of the web tool. It contains a list of all of the stories created by the product owner. The product owner can use SQL to import their backlog from another application such as Agilefant. The main fields for the story type are the title, description, and acceptance criteria. The SENG302 backlog has the story titles and descriptions with the acceptance criteria as part of the description. Because it does not require a story to have acceptance criteria as we have in our tool.

It is created using a view from the view module. The content is filtered down to all story objects, then all of the necessary fields are extracted and displayed in a table. With the table settings allowing for sorting and links to be displayed. The table also allows for columns to be hidden if they are empty, such as the Re-planning EV column in the “In Progress” section of the backlog.

The backlog is where phases 1, 2, and 3 occur in our estimation process. We want developers to have easy access to the stories and frequently used information from the backlog itself. As shown in Figure 8 above, the priority, tags, planning, re-planning, and retrospective estimation values are available as well as the story lead. The lead developer for a story is an optional phase in our estimation process as not all development teams require the aspect. Either due to team size or agile methodology.

The backlog is grouped into two categories, divided by the state of the story. It is either in progress or completed. This allows the development team to see what stories are currently in progress, or upcoming in future sprints. While the completed stories are grouped separately to reduce search time for current stories. Completed stories should be of lower priority than "In Progress" stories, but we still wish for them to be on the backlog and in the system for future reference.

All of the columns in the table are sortable. This allows for the backlog to be ordered by priority, by estimation value, or by story lead. This is for enhanced user experience and customisation.

Once a story is selected the user is taken to the story page, which is shown in Figure 9 and Figure 10 below.

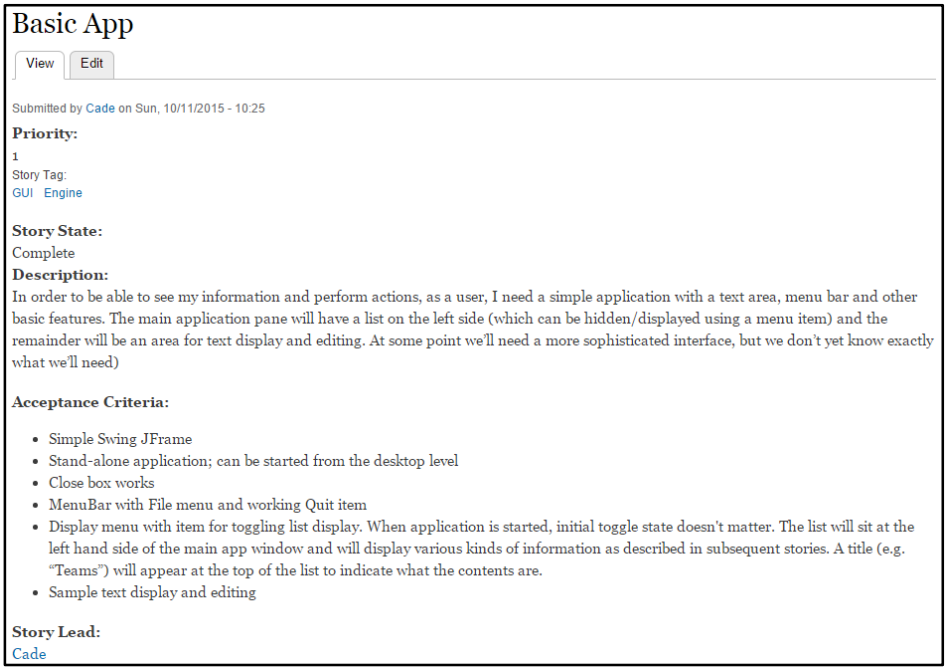


Figure 9: Web based estimation tool story structure part 1

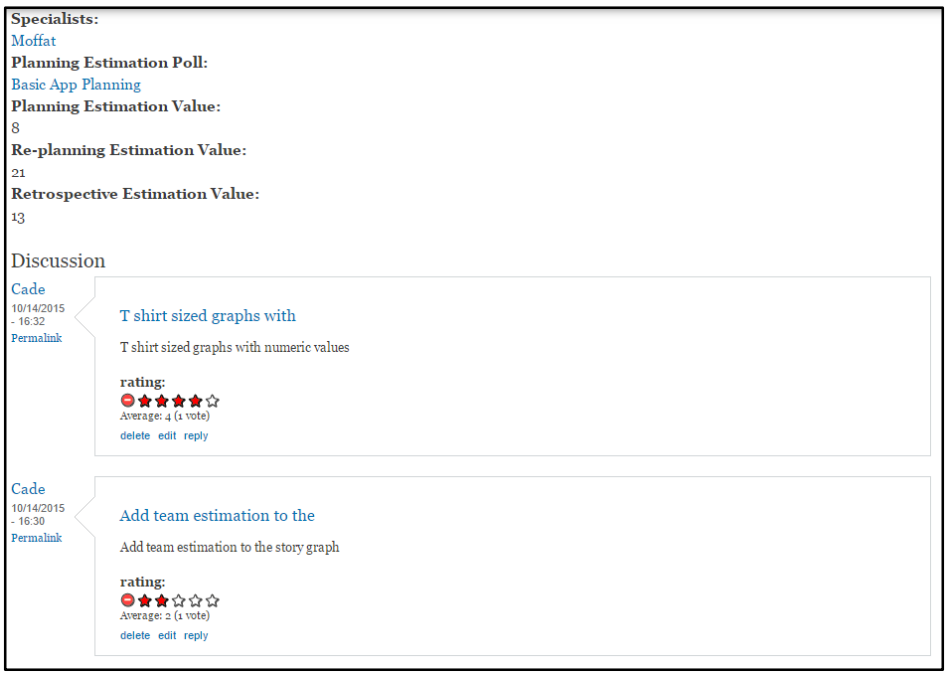


Figure 10: Web based estimation tool story structure part 2

The story page is where the developers get all of the information about the story and the client's requirements. The fields that exist on a story object are:

- A priority ranging from 1-5 with 5 being the highest priority stories.
- A range of story tags that can be assigned to a story and customised by an admin user such as the product owner. These tags are controlled by the taxonomy term module.
- Story state Boolean field to inform the developers of whether the story is complete or not.
- Description of the story that the product owner has written based on the client's requirements.
- Acceptance criteria to outline criteria that should be met by the program once the story is complete.
- An optional story lead, which defines which team member is the point of contact for the story in the team. This is the third phase of our improved estimation process along with the specialists outlined below. This field is created through the references module, which allows for a combo-box of registered users where the team can choose a single member.
- Optional specialists, these can be other team members. This field is also created by the references module allowing for users to have links to their profiles. However other employees or outside specialists can be added, but they will not have a link to their profile as they are not registered users.
- The last 6 fields are the estimation links and values. The links are added by the development team once an estimation poll is created and controlled by the references module. This allows for easier navigation around the web tool. The values are the final estimation values agreed upon by the team. This keeps the necessary information in one place and allows for the graphs to display the overall team estimate after negotiations.
- The final aspect of the story page are the comment fields. This is where the development team undertakes our second phase, they can communicate between themselves, with the product owner, or with specialists. The comments can start new discussions or be replied to. Comments can be rated by the team out of 5 stars, to encourage and draw attention to important discussions and comments.

The comments come built in to Drupal, they have enhanced customisation from the comment goodness module.

13.4 4 TAGS

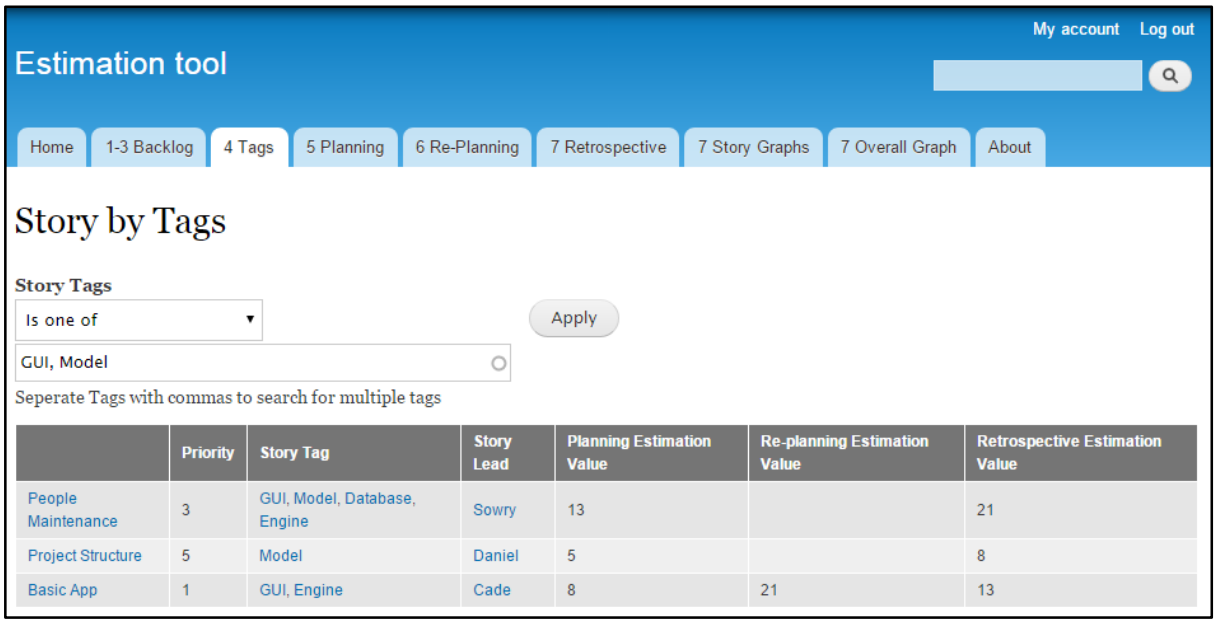


Figure 11: Web based estimation tool tags page

The tags page is where the fourth phase of our estimation process is undertaken. Here the development team members can search through all of the stories on the backlog and sort by story tags as seen above in Figure 11. Using the input field which autocompletes to filter by single tags or multiple tags separated by commas. Once they have completed the search they can analyse past stories to see if any familiar functionality or requirements stand out. The team members can then draw on this knowledge and experience with greater accuracy than solely relying on memory. This will help them to improve the current estimation.

This functionality is only useful if the system has been in use for a period of time as it relies on the information given in previous sprints. It can be extremely helpful if there is an aspect of the development that is being repeated many times over the project's lifetime. For example, in the SENG302 project this year the teams were creating new views for the new features in each sprint, they could use the past estimations as a reference point.

Otherwise this page has similar functionality to the backlog, users can access stories and see important information on the page itself. It is built using the same Drupal modules as the backlog page. The views module allows for the taxonomy term filtering to be exposed to the user and allow for it to function as a filtering search bar. Thus saving the need for a separate view for each possible story tag in the web tool.

13.5 5 PLANNING, 6 RE-PLANNING, AND 7 RETROSPECTIVE ESTIMATION PAGES

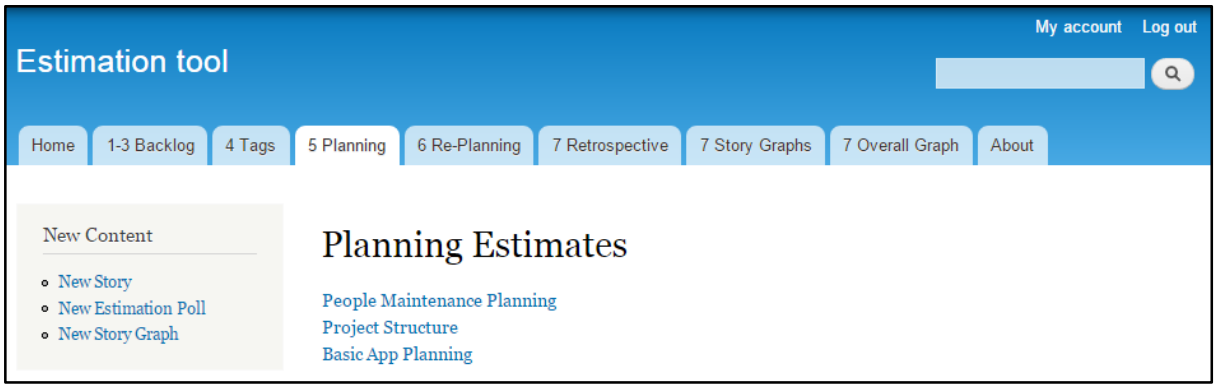
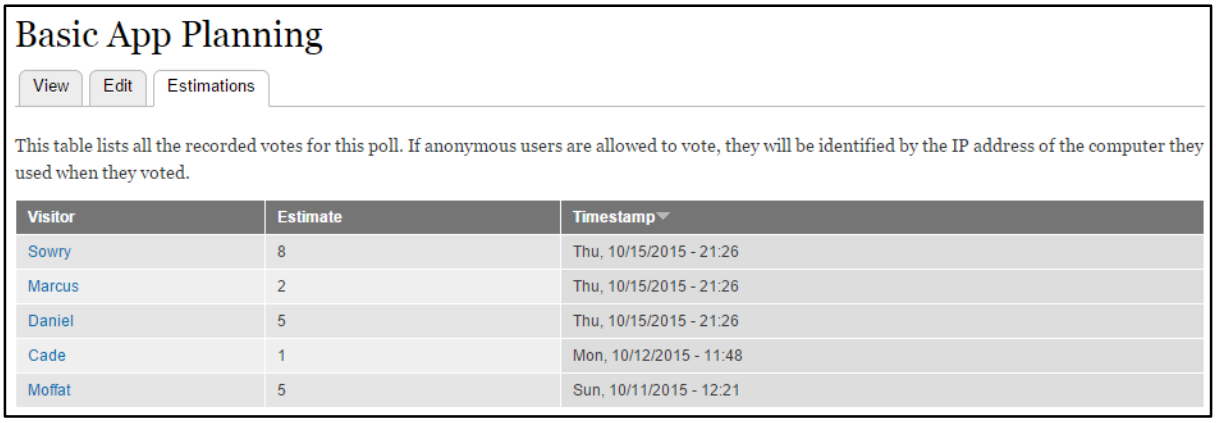


Figure 12: Web based estimation tool planning, re-planning, and retrospective page layouts

Figure 12 above shows the basic layout for the three estimation pages, created using the views module, filtering by poll node types and further filtering by taxonomy terms (planning, re-planning, or retrospective). The "New Content" block is also present on these three pages due to the developers wanting to create new estimation polls during each sprint planning phase. Phases five, six, and seven occur through these tabs. With phase six being optional as it is not always necessary to re-estimate a story mid sprint.

Polls are a standard inclusion in the core of Drupal, there are modules that extend the poll functionality as stated previously. The default poll functionality is enough for our web tool and collecting estimations from the development team. We utilise the module "Comment Goodness" to increase customisation of the polls.



Visitor	Estimate	Timestamp
Sowry	8	Thu, 10/15/2015 - 21:26
Marcus	2	Thu, 10/15/2015 - 21:26
Daniel	5	Thu, 10/15/2015 - 21:26
Cade	1	Mon, 10/12/2015 - 11:48
Moffat	5	Sun, 10/11/2015 - 12:21

Figure 13: Web based estimation tool list of estimations in this poll and the user who estimated each value

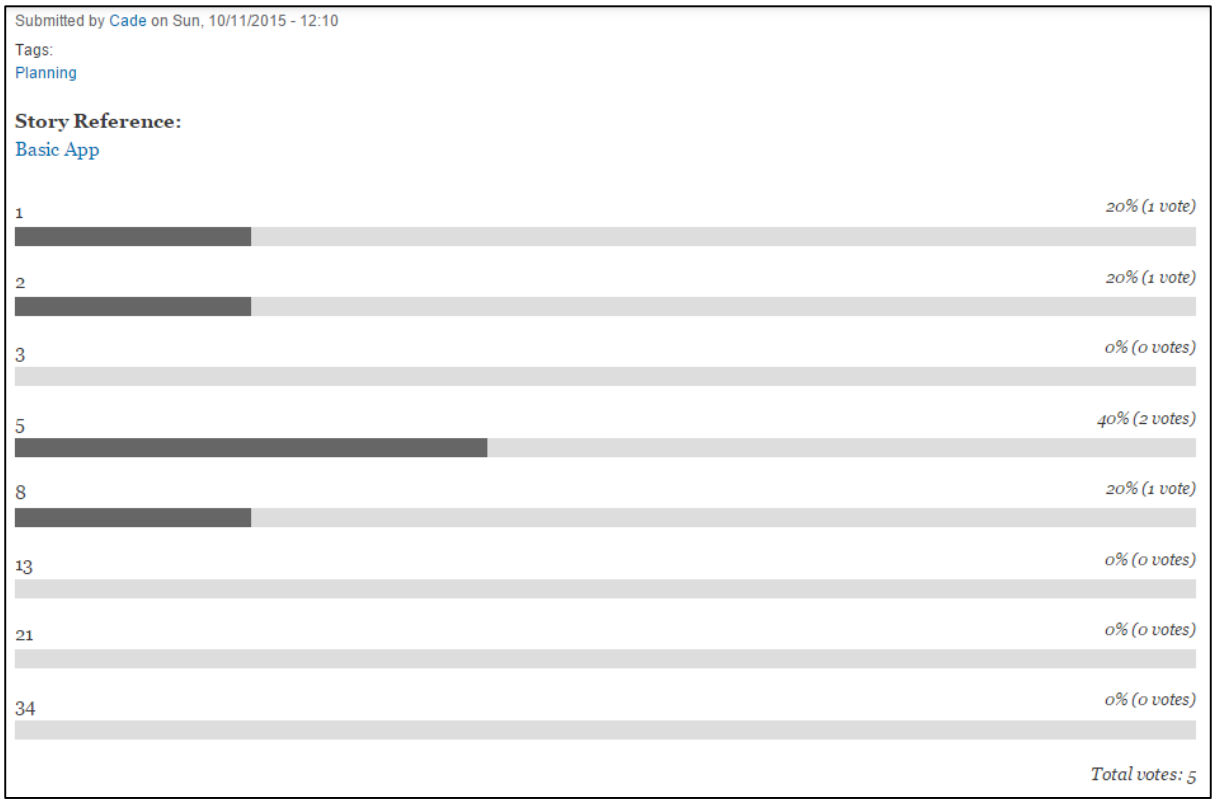


Figure 14: Web based estimation tool estimation poll after estimating

Figure 14 shows the estimation poll interface where the developers get to cast their estimation vote for certain stories. Currently the developers can cancel and change their vote, this is not ideal as we wish to see the changes over time and increasing accuracy. To aid in the discussion of the team everyone can see who voted for which value in the estimation as shown in Figure 13 on the previous page. Allowing for the outlier estimates to be discussed by the team, although this should be less necessary in our improved process as we are encouraging discussion and clarifications in the second phase of the process. Before the team gets to phase five and begins to estimate the story as a team.

13.6 7 STORY GRAPHS

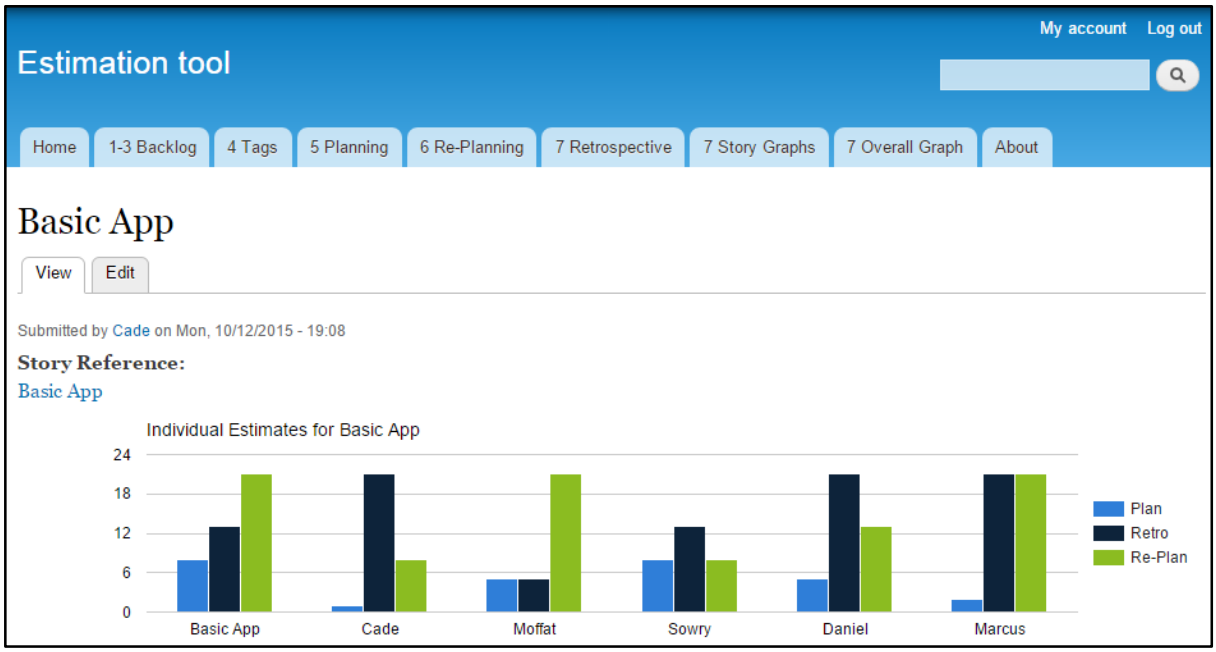


Figure 15: Web based estimation tool story graph page

As part of our retrospective analysis of the estimations, we want the development team to have access to as much useful information as possible. As shown in Figure 15 above, we enable the development team to create graphs for each individual story. This is a view that filters by story graph type and displays the results in a simple list. The “New Content” block is also present to allow developers to create new story graphs.

This plots the planning, re-planning, and retrospective estimations of all the team members against the final team estimation for each phase. Allowing for increased readability as graphs can be much simpler to interpret than tables of data. It also allows the three different estimation phases to be plotted against each other on the same graph. If a team member is away and misses an estimation, their value is set to zero so that the chart library does not produce errors.

These graphs are produced through PHP code and the chart and google chart modules. The data is extracted from the Drupal database through database select calls and then filtered and extracted through the PHP code that comes as part of the story graph template. The development team only need to input the story name into the corresponding field in the story graph node. This is then extracted and used to find the corresponding estimation polls.

13.7 7 OVERALL GRAPH

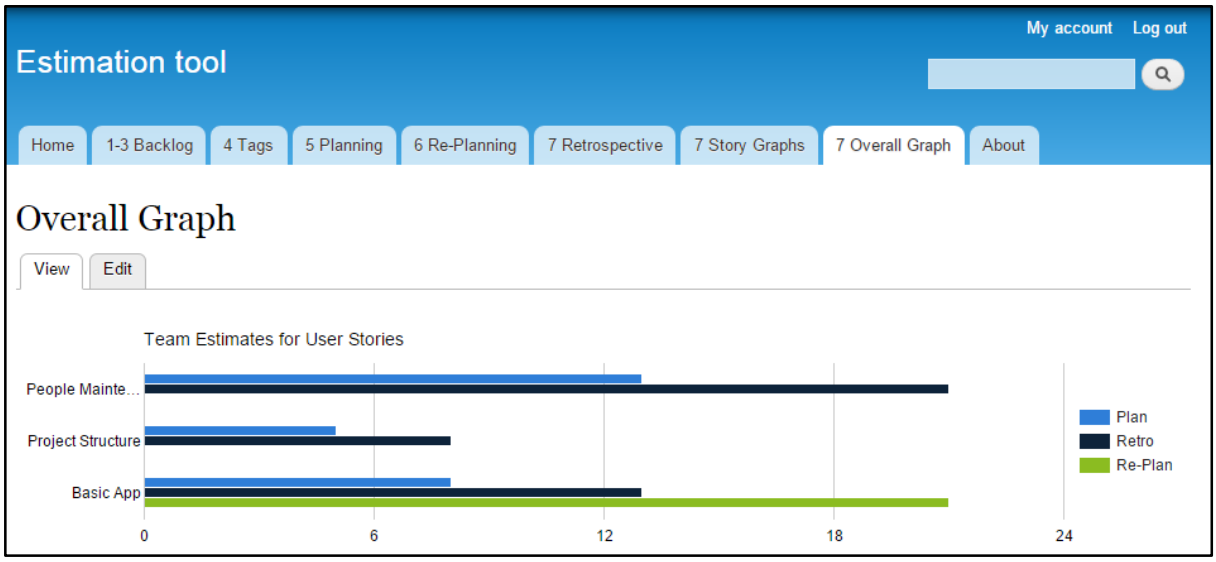


Figure 16: Web based estimation tool overall graph page

The overall graph is a small feature that will become less useful as the number of stories increases. Currently the graph takes all of the stories on the backlog and extracts the planning, re-planning, and retrospective estimations form each story. These are then shown on a bar graph to show the accuracy of the planning estimations compared to the other two estimations. As seen in Figure 16 above.

Once there is a large number of stories created there would be too much information to display effectively on a column graph. A bar graph is selected to display this information due to the fact that the page has virtually infinite vertical scrolling. This allows for new stories to be added to the graph and allow the graph to still show decent information. If the graph was a horizontally expanding graph, then the labels would become incomprehensible.

13.8 ABOUT

The about page explains the estimation process to the developers and outlines how to navigate and use the web based estimation tool. It is a basic tool that uses HTML to format the text and display the information in a coherent structure.

14. LIMITATIONS AND FUTURE WORK

- Estimation values:
  - Currently only numerical estimation values can be graphed. Other estimation value scales such as t-shirt sizes or dog breeds cannot be graphed using the current implementation. This limits the availability of common agile estimation values available to the development teams that use this prototype.
  - To support these values with the current libraries would require setting the non-numerical values to a numerical equivalent and displaying both to the user on the axis.
- User accounts:
  - Currently there are only three levels of accounts. Anonymous, registered, and administrator users. Any account created has at least a registered user level, this lumps together the development team and any specialists that have had accounts created for them. All of these users will appear on the individual story graphs, even though we do not expect specialists outside of the development team to estimate stories as part of the team.
  - To combat this there needs to be more levels of user accounts. Anonymous, specialists, developers, team leader, and product owner. This would allow for better management of the privileges of the different user types as well and increase security.
- Charts and graphs:
  - Currently the individual story graphs have to show the team estimate as though it was another developer. It would be preferable to show the final team estimations as horizontal lines that traverse the column graph and span across the entire development team's estimations. This would increase the readability of the graph and make it more useful to the development team during retrospectives.
  - The overall graph should be able to be filtered through tags to group similar stories together, or at least limit the graph to show the most recent 20 stories for example.
  - As mentioned above the inability to show all of the estimation value scales possible in industry is a limitation of the web estimation tool.
  - Currently the story graphs require the estimation polls to contain the story node title in the name. For example, "Basic App Planning" for the planning estimation poll. The graph checks for any polls containing "Basic App" and extracts any matches. This would cause problems if stories titles contained other story titles. This is an unhandled edge case in this application.
  - Currently the story graph also includes the product owner as a team member who estimates the stories with the rest of the team. This is related to the changing user levels as mentioned above. But in most agile development environments the product owner should not be estimating the stories with the team.
- Automation:
  - Development teams are moving towards automation for almost all aspects of development. Continuous integration, automatic bug tracking through tools such as JIRA/Fogbugz, automated testing and prototyping servers. This prototype requires the team to manually create certain content, such as the estimation polls and story graphs. If once a story was created the corresponding estimation polls and story graph automatically. This would reduce the manual tasks required of the team. It would also reduce the complications of having to create the estimation and story graphs using the same name as the original story, as this would be created automatically.

- Reviewing past stories: Intelligent Tutoring System
  - In the event of infinite time to work on this project, either as a continuing PhD student or academic researcher. We wish to add an intelligent tutoring system to the current web estimation tool. This would automate the reviewing past estimations phase of the process, due to the fact that the tutoring system would suggest similar stories. Through its learning it would advise teams and individuals on the accuracy of their past estimations on similar stories. For example, on average all "GUI" related stories are underestimated by 15%.
  - This tutoring system would be able to build models for the estimation practices of a team as a whole. Through this modelling it would create models for the individual team members and their estimation practices and expertise. As agile development teams can often have members swap around and change, the individual models would be designed to be modular and able to be switched in and out of team models. Aiding in speeding up the time it takes for a newly created development teams to understand the strengths of each member. As the web based estimation tool can suggest experts in the team based on their past experiences and expertise. Almost entirely eliminating the storming phase of team bonding in regards to planning meetings and estimations in the agile environment.
  - Having these models would aid the management of the software business, as it would be easier for the managers to create well balanced teams of developers after looking at the strengths and weaknesses of each developer.
- Mobile usability:
  - We wish for developers to be able to access the web estimation tool from their mobile devices and still have a great user experience. This is due to the fact that most meeting rooms in industry do not come equipped with multiple computers that would allow simultaneous use of our prototype. So developers being able to access the tool from their mobile devices while in the meeting would be a large increase to the usability of the software application.
- Case Study:
  - A case study was not able to be run on the SENG302 students during this project's lifecycle due to time constraints. To truly evaluate the system and get user feedback on the process and user interface design. A case study needs to be undertaken, using a course such as SENG302 is an invaluable resource for this application. It is extremely difficult to find agile development teams in industry that have multiple teams working off a single backlog and working on the same stories. This allows for a much larger range of data that can be collected and analysed.
  - The downside to using SENG302 students as the test subjects, is their lack of industry and programming experience. To increase the effectiveness of our improved process we wish to draw on the experience of the industry developers with years of agile experience and knowledge. To be able to carry out a case study on industry developers would lead to great feedback about the improved estimation process and our prototype.

15. ACCEPTANCE TESTING OF THE REQUIREMENTS

The list of requirements that we created earlier in the project that the final solution meets:

- Ability to tag stories at any time
- User management
  - Requirement is met currently but the requirement needs to change to accommodate more user levels as mentioned previously.
- Open forum question
  - Requirement is met but the terminology needs to be updated to be comments on stories not a forum
- Create discussion topic
- Specify other interested parties
- Optional story lead
- Resource management
  - Can be specified under the specialist's section of the program if necessary but there is no explicit support for this in the current prototype
- Configure estimation labels
- Exclude members from estimation
  - This is not applicable to the current prototype as if a team member is absent then in the visualisations through graphs the value is set to zero.
- Estimation page with results and estimation
- Call for re-estimation – list as separate estimation
- Define retrospective estimate milestones
  - The retrospective estimations are a core part of the improved estimation process and is the phase where the most learning and improving can be accomplished.
- Visualisations
  - There are currently two visualisation sections of the prototype and they have room for improvement. But they are a base where better visualisations can be based off.

Requirements that were not met by the current application:

- Close discussion topic
  - Discussions in the current prototype cannot be closed and remain available to team members to comment and investigate. This requirement is outdated and was related to the forum modules under investigation at the time.
- Increase question priority
  - Questions in the current prototype no longer have priority. But they can be ranked by the other team members to show importance and support. This requirement needs to be updated for the final application.
- Add labels to team members
  - Team members in the current web tool cannot have tags assigned to them to showcase their skills and expertise. This should be implemented when the user levels get re-engineered.
- Show estimates after everyone submitted
  - This requirement was extracted from the fact that developers can become biased after seeing other team members estimate a story at a different value than their current

estimation value. This is not available in the current prototype. Drupal polls displays the votes as they are entered.

- With our improved process we would see any trends in the data over time for individual estimates. While retrospective estimations allow for team members to estimate through hindsight, allowing us to still grasp the expertise of each team member. This is still an important aspect that needs to be considered.

We have achieved most of the requirements that were outlined earlier during the planning of the improved estimation process. One aspect that we missed that is important is the "Show estimates after everyone submitted" requirement. This is a reason that planning poker exists and is in use throughout the agile development world.

16. DISCUSSION

This project has opened our eyes to the world of agile estimation and shown how the simplest of ideas can explode and turn into a mountain of arguing and questions. Starting with the simple how do agile developers estimate? Leads down the slippery slope where we have to ask ourselves what is estimation? How is it implemented in industry? Why is it not researched more? If every team does it differently, how can we possibly improve the process? What could the requirements possibly be?

Some of these questions are hopefully answered through our improved estimation process and the corresponding web based estimation tool prototype. The project had a limited lifespan as it was limited to a one year honours project. But the field of agile estimation needs to be researched more in depth. We believe this project is a good baseline start for any academic or researcher to expand and learn from.

The first approach that was undertaken, opened our eyes to the difficulty of developing a large system completely from scratch. Redefining the wheel as it were, however using existing software such as Drupal with all of its open source modules also comes with limitations. As libraries and pre-made modules can limit the functionality available to the application through the design decisions the module developers chose.

We learnt that redefining a process through academic research can be an exhilarating experience and also terrifying at the same time. We have extremely limited experience and knowledge of agile development and industry development as a whole. And yet here we are trying to take the precious agile methodology and redefine it. The survey that we developed with the help of Dr. Matthias Galster should help to bridge this gap in the knowledge and experience that we have. It is a limitation that the survey could not have been completed before this project began. This would have allowed us to create a more robust and knowledgeable set of requirements to base our solution on. The survey will be an invaluable tool and wealth of data for any further research into the field of agile estimation.

This project expanded my communication skills, developers' knowledge with more experience with web technologies (WAMP, Drupal, PHP, HTML, Javascript, AngularJS), research knowledge in exploring a new field of study for ourselves and software engineering, and time management skills to keep the project on track over the course of the study year.

17. CONCLUSION

We have researched into the current agile estimation practises, created and designed an improved estimation process that consists of seven phases. We analysed the improved process by getting retrospective estimates from SENG302 students, elicited requirements based on this improved process. Designed and implemented a prototype of our custom web tool to try and support this improved estimation process. Researched into web tools and frameworks that could aid in the creation of the web tool, designed and created a Drupal based web estimation support tool. Tested the application through acceptance and manual testing, checking the web based tool against the original requirements and to search for edge cases. Created and refined an industry survey to be sent out to agile developers and collect their responses, to help elicit requirements and gather data on the current agile estimation processes. Lastly we documented and conveyed all of this through academic reports backed by scholarly references.

This project has led to a working prototype of a web based estimation tool, that could be used in a small case study on SENG302 students although there are still improvements that can be made to increase the usability and user experience. Finishing improving the agile estimation process is a much larger task than what could be achieved in this project. However, this research and prototype are a starting point that have been designed to continue on with further research and development.

18. ACKNOWLEDGEMENTS

We would like to thank the Department of Computer Science and Software Engineering at the University of Canterbury for its support throughout this project. The technical team that helped with setting up servers and general computer knowledge. Dr. Neville Churcher for his support with the SENG302 teams and information on the visualisation of data. Dr. Matthias Galster for his support and guidance on the industry survey and requirements elicitation. Dr. Moffat Mathews for his guidance and encouragement throughout this project, without whom the project would not have been completed on schedule. Lastly the SENG302 teams who agreed to participate in the analysis of the improved estimation process and retrospective estimations.

19. BIBLIOGRAPHY

1. Alliance, A., Agile manifesto. Online at http://www.agilemanifesto.org, 2001. 6(6.1).
2. Beck, K., et al., Manifesto for agile software development. 2001.
3. Hunt, J., Agile software construction. 2006.
4. Stober, T. and U. Hansmann, Best Practices for Large Software Development Projects. 2010: Springer.
5. Frese, R. and V. Sauter, Improving your odds for software project success. Engineering Management Review, IEEE. 42(4): p. 125-131.
6. Deemer, P., et al., The scrum primer. Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: http://assets. scrumtraininginstitute. com/downloads/1/scrumprimer121. pdf, 2010. 1285931497: p. 15.
7. Meyer, B., Agile!: The Good, the Hype and the Ugly. 2014: Springer Science & Business Media.
8. Higgins, T. Authoring Requirements in an Agile World. 2009 [cited 2015 11th March]; Available from: http://www.batimes.com/articles/authoring-requirements-in-an-agile-world.html.
9. WordPress. Create your new website for free. 2015 [cited 2015 16th July]; Available from: https://wordpress.com/.
10. Barron, B. WordPress vs. Drupal: Choosing Between Two Platforms. 2015 [cited 2015 16th July]; Available from: https://www.elegantthemes.com/blog/resources/wordpress-vs-drupal.
11. Joomla. The platform millions of website are built on. 2015 [cited 2015 16th July]; Available from: http://www.joomla.org/.
12. Mikoluk, K. Drupal vs Joomla vs WordPress: CMS Showdown 2013 [cited 2015 16th July]; Available from: https://blog.udemy.com/drupal-vs-joomla-vs-wordpress/.
13. Drupal. Get started with Drupal. 2015 [cited 2015 17th July]; Available from: https://www.drupal.org/.
14. SeleniumHQ. What is Selenium? 2015 [cited 2015 17th July]; Available from: http://www.seleniumhq.org/.
15. Qualtrics. The World's Leading Enterprise Survey Platform. 2015 [cited 2015 6th September]; Available from: http://www.qualtrics.com/.
16. Gomaa, H., Software modeling and design: UML, use cases, patterns, and software architectures. 2011: Cambridge University Press.
17. Committee, I.C.S.S.E.S. and I.-S.S. Board. IEEE Recommended Practice for Software Requirements Specifications. 1998. Institute of Electrical and Electronics Engineers.
18. Jalote, P., A concise introduction to software engineering. 2008: Springer Science & Business Media.
19. Voinov, A. and F. Bousquet, Modelling with stakeholders. Environmental Modelling & Software, 2010. 25(11): p. 1268-1281.
20. Schneidewind, N., Software risk analysis. International Journal of Reliability, Quality and Safety Engineering, 2009. 16(02): p. 117-136.

21. Ganssle, J.G., Break points-Creating software prototypes-Prototyping software is an essential step in creating quality embedded systems and a sane work environment. Embedded Systems Design, 2008. 21(8): p. 45.

22. Forward, A., Software Documentation–Building and Maintaining Artefacts of Communication. 2002, University of Ottawa.

23. Trendowicz, A., J. Münch, and R. Jeffery, State of the practice in software effort estimation: a survey and literature review, in Software Engineering Techniques. 2011, Springer. p. 232-245.

20. APPENDIX 1: LEARNING OUTCOMES FOR THIS HONOURS PROJECT

20.1 EXPLORING AND MODELLING COMPLEX SITUATIONS

According to the Object Modelling Group, "modelling is the designing of software applications before coding."[16] We will be exploring current estimation practices that are used in industry and how different teams use these practices to improve their estimation processes and accuracy. We will be modelling estimation practices to see what parts of the current estimation practices are the most used and essential to a good estimation solution. Exploring and modelling will help to increase the accuracy of my proposed solution design and allow my requirements to encompass issues in current estimation solutions.

20.2 REQUIREMENTS ANALYSIS

IEEE standard 830-1998 [17] defines a requirement as "(1) A condition of capability needed by a user to solve a problem or achieve an objective; (2) A condition or a capability that must be met or possessed by a system, to satisfy a contract, standard, specification, or other formally imposed document." We will be analysing current systems and also using requirements engineering practices with my stakeholders to establish requirements for the estimation system that we will be designing and implementing.[18]

20.3 COMMUNICATION WITH STAKEHOLDERS

Stakeholders are individuals or groups of people who either have an interest or will be affected by the software solution. "Stakeholders may represent local, federal, private and public organisations, as well as individual citizens and interest groups, which have very different, oftentimes conflicting interests"[19]

For this area of research, the stakeholders are development teams, testers/quality assurance teams, product owners and their customers, and business analysts. For this project in particular the stakeholders will be the SENG302 students in their respective development teams and the product owners of each team. Also some stakeholders will be developers, product owners and software engineers from the industry that have agreed to cooperate with this research.

We will need to work with the stakeholders of this project so that the requirements are correct and cover all aspects of the estimation problems from as wide a range of views as possible to increase the spread of requirements.

20.4 EXPLORE ALTERNATE DESIGNS

Research will be performed into potential technical and architectural designs to judge their effectiveness as a solution to the estimation problem. Web architectures, mobile technologies and desktop architectures will need to be analysed and judged so that we can provide the best solution to the problem. In completing this research and analysis we will need to produce our favoured design and also some alternate designs for comparison and to show the depth of our research.

20.5 RISK ANALYSIS

Every aspect of software development contains risks, will this feature work as intended? Can our language complete these tasks easily? These are just a couple of high level risks in any project. Every task that will be implemented in a software solution has the chance to deviate from its plan[20] and some of these risks are additive if they are unrelated, while some risks will be multiplicative with other risks. We will have to complete risk analysis to try and reduce the amount of risks associated with my designed solution. This will improve the reliability of the solution and improve the development process if the risks are well defined and documented.

Risks that we have identified already, but need to be investigated further are:

- Time constraints
- Taking on more requirements than feasible for an honours project
- Lack of knowledge of architectures and technology that could simplify the project
- Unforeseen risks e.g. sickness or family matters
- Working with colleagues who also have time constraints e.g. SENG302 students and lecturers, industry contacts

To mitigate these risks, we will be following a timetable as shown in Table 2 in Appendix 2. We will be judging the priority and importance of the requirements with my supervisor. To help get around the time constraints of stakeholders we will be creating a simple questionnaire to help gather information, requirements and different contexts that different company's versions of Agile can create. This will help me to get an initial understanding of their points of view and then provide a basis for my questions; if we decide to have an interview if they can spare the time.

20.6 TECHNICAL CHALLENGES

We will be facing technical challenges when we research and implement the designed solution as we need to research technologies and architectures that we have little to no experience with and construct my solution using these architectures. This research will have to be wide ranging to cover all of the different types of software available that may help to solve this estimation problem in a clean and efficient way.

20.7 PROTOTYPING SOFTWARE SOLUTIONS

In the ever evolving world of software development, user/client requirements can change at any time and quite often the client does not know what they want in a software solution until they see a working product. A solution to this problem, that can help to bridge this gap are software prototypes, which run a minimal subset of the programs final functionality. This code is to be designed to be disposable as prototypes are designed expecting to make mistakes, encounter problems and have differing requirements from the users/clients after they see a prototype. [21]

For this project we will be setting up a virtual machine to run prototypes and the final product when we need to be able to let others access the product or to run some tests. This allows for disposable and easily generated solutions to be prototyped effectively.

20.8 EVALUATION OF DEVELOPED SOLUTIONS

To get an idea of the current solutions that may exist for this specific estimation problem we are researching, we will need to evaluate any current estimation application that may provide a solution and make sure that my research is not leading me towards an already existing solution. This will make sure that my solution is unique and original work.

To evaluate my own solution, we plan to test it on the SENG302 students, this will inform us of its feasibility and usefulness. The software will be designed with the ability to gather logs while it is being used. This will be collected and can be analysed to gain a greater knowledge of the users' experience, complementing their own reviews of the product that can be achieved via questionnaire. We plan to use questionnaires prior to the use of the software as well as after they have used it for a period of time.

20.9 DOCUMENTATION

To make sure that the research project can be explored and investigated by outside sources there needs to be sufficient documentation. "Software documentation is an artefact whose purpose is to communicate information about the software system to which it belongs". [22] For anyone who is interested in this software solution and wishes to learn more about my design and implemented solution, the documentation should provide an informative way to become more knowledgeable about our specific solution.

20.10 CRITICAL APPRAISAL OF SOFTWARE PRACTICES

"Effort estimation is a key factor for software project success, defined as delivering software of agreed quality and functionality within schedule and budget". [3, 23] To correctly identify the flaws and weaknesses in the current estimation practices in agile development, we will need to perform a critical appraisal of the current estimation methods.

20.11 CRITICAL SELF-REFLECTION OF SOFTWARE PRACTICES

To learn as much as possible from this research project, we will be completing a critical reflection of my own skills and practices, in order to improve my abilities as a software engineer. We need to ensure that we are not practicing bad software engineering practices and that any improvements we make to the software engineering practices are reflected in my own behaviour.

21. APPENDIX 2: ORIGINAL TIMELINE

Table 2: Original timeline for the project as outlined in the project proposal

Monday's date	*ISO Week*	*Lecture week*	*Due dates and personal goals*
February 23	9	1	
March 2	10	2	Literature review about estimation processes and techniques used in project management
March 09	11	3	Starting investigation into current software solutions
March 16	12	4	Friday 20th: Project proposal
March 23	13	5	Gathering requirements - creating the backlog
March 30	14	6	
April 6	15	Holidays	Designing the system
April 13	16		
April 20	17		
April 27	18	7	Friday 1st: Progress report 1
May 4	19	8	Main implementation and testing phase until June 7th
May 11	20	9	
May 18	21	10	
May 25	22	11	
June 1	23	12	
June 8	24	Study week	Short expert trial – with an expert from Software Engineering to check that the program is well implemented according to the requirements specifications and behaves correctly.
June 15	25	Exams	
June 22	26		
June 29	27	Holidays	
July 6	28		
July 13	29	13	Friday 17th: Progress report 2
July 20	30	14	Small case study with at least two SENG302 team (remembering initial estimations and getting post estimations for data analysis) lasting until the 7th of

			September
July 27	31	15	
August 3	32	16	
August 10	33	17	
August 17	34	18	Planning for Postgrad conference presentation
August 24	35		Holidays and Project Presentation at the Postgrad conference
August 31	36		
September 7	37	19	Data analysis and seeking improvements from experiment feedback (from teams and from experts)
September 14	38	20	
September 21	39	21	
September 28	40	22	Final report writing and proofreading
October 5	41	23	Planning for practical demonstration
October 12	42	24	Friday 16th: Final Report and Practical demonstration

22. APPENDIX 3: SENG302 TEAM RETROSPECTIVE ESTIMATIONS

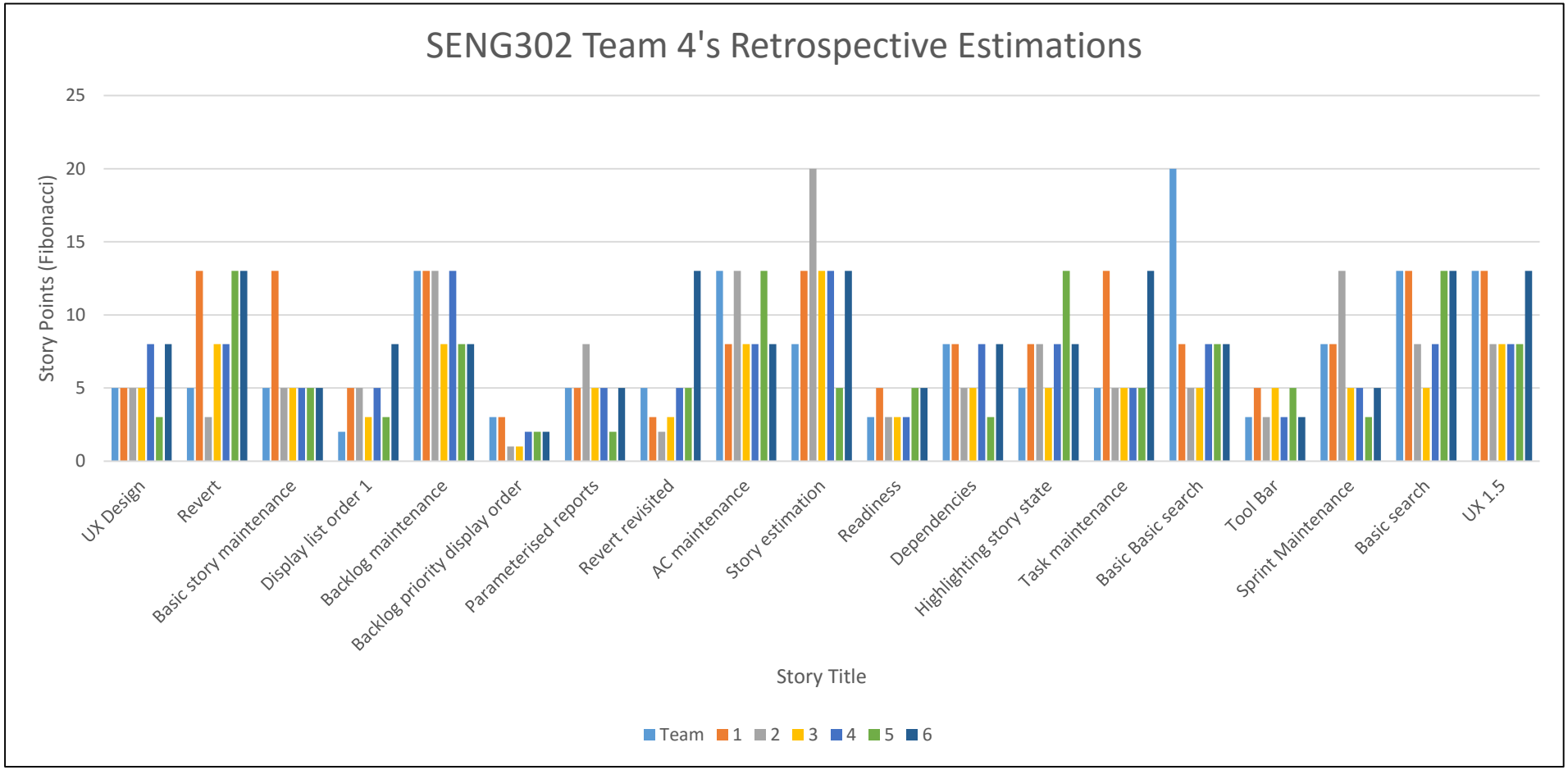


Figure 17: SENG302 team 4's Retrospective Estimations

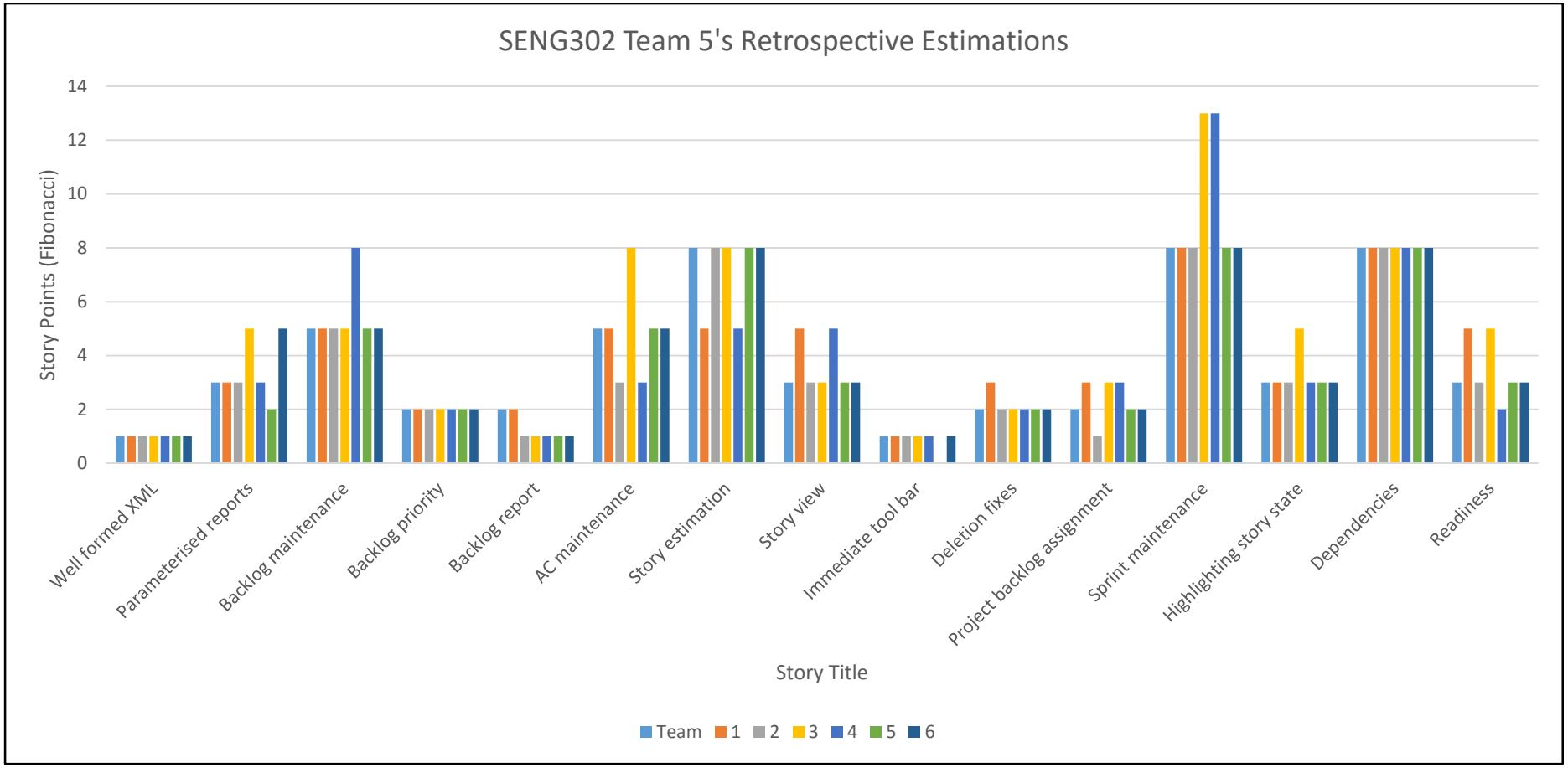


Figure 18: SENG302 Team 5's Retrospective Estimations