Capability Maturity Model Integration Project

Team:
Jason Ebueng
K. Ngo
K. Nguyen
T. Yau

1. Develop a Product Backlog for the system using User Story format, and prioritize them according to feature values.

Product Backlog

1. As a system administrator, I would like to remove authorizations from student and professor accounts.
2. As a system administrator, I would like to enter new students and professors into the system.
3. As a system administrator, I would like to enter new courses for the semester and set descriptions for them.
4. As a student, I would like to be able to register open classes when my enrollment window is open.
5. As a student, I would like to be able to drop classes before the second week of classes.
6. As a system administrator, I would like the option to set prerequisites for courses on the course enrollment system so that only students with prerequisites can register for a course.
7. As a student, I would like to be able to view my current schedule.
8. As a student, I would like to be able to view my course history and grades earned.
9. As a student, I would like to get notification when my registration window is opened.
10. As a professor, I would like to be able to enter students' grades into the system.
11. As a student, I would like to be able request my transcript.
12. As a student, I would like to be able to view the number of open seats in a course.
13. As a professor, I would like to be able to drop students from my class.
14. As a professor, I would like to view students schedules.
15. As a professor, I would like to be able to view students' course history.

2. Estimate the features using story points for the entire product backlog.

Using Fibonacci numbers:

Story ID	Description	Story Points
S1	As a student, I would like to be able to register open classes when my enrollment window is open.	21
S2	As a student, I would like to be able to drop classes before the second week of classes.	8
S3	As a student, I would like to be able to view the number of open seats in a course.	5
S4	As a student, I would like to be able to view my course history and grades earned.	13
S5	As a student, I would like to be able to view my current schedule.	5
S6	As a student, I would like to be able request my transcript.	8
S7	As s student, I would like to get notification when my registration window is opened.	8
S8	As a system administrator, I would like the option to set prerequisites for courses on the course enrollment system so that only students with prerequisites can register for a course.	8
S9	As a system administrator, I would like to enter new students and professors into the system.	13
S10	As a system administrator, I would like to remove authorizations from student and professor accounts.	21
S11	As a system administrator, I would like to enter new courses for the semester and set descriptions for them.	13
S12	As a professor, I would like to view students schedules.	8
S13	As a professor, I would like to be able to view students' course history.	13
S14	As a professor, I would like to be able to drop students from my class.	8

S15	As a professor, I would like to be able to enter students' grades into the system.	5
	Total	157

*3. Develop a release plan based on your team's velocity. Refine the plan after each simulated sprint. Please conduct three simulated sprints. **

*** = Step 9, identify top risks, is also covered within this section*

Total Story Points : 157

Release Plan
We estimated the team's velocity and developed the the three sprints based on priorities of the items and the story points.

Each iteration should have about 52 points each

	Iteration 1		**Iteration 2**		**Iteration 3**	
	Story ID	**Points**	**Story ID**	**Points**	**Story ID**	**Points**
	S9	13	S1	21	S7	8
	S10	21	S2	8	S6	8
	S11	13	S5	5	S3	5
	S8	8	S4	13	S14	8
			S15	5	S12	8
					S13	13
Total	**55**		**52**		**50**	

4. Develop an issue tracking excel table and use it to track issues identified each sprint.

Prefix Meaning
“ISS” : Issue
“CH” : Change request

Issue ID	Iteration Identified	Severity	Issue description
ISS1	1	Medium	Story cannot be tested completely without student “add class” feature.
ISS2	1	High	Users are unable to access system when they are not connected to the school network on campus.
ISS3	2	Minor	Register button is too high on page.
ISS4	2	Medium	Course history does not refresh after dropping classes.
ISS5	3	High	Students are not notified when they are dropped from the class by their professor.
CH1	1	Minor	Customer requested for professors to be able to register themselves with

			approval by administrator.
CH2	3	Minor	Customer requested that students get additional notifications prior.

5. Develop a process Scrum like process that your team will use to conduct the project.

My team will be using a scrum-like process that involves 3 iterations consisting of 2 week sprints. The self-organizing team will consist of the engineers as well as the scrum master. The scrum master will facilitate 15 minute scrum meetings every morning in which each member of the team will briefly talk about what he/she did yesterday and what he/she will do today. The project manager and product owner may attend the meeting as they please. The project manager will assist the team in removing blockers. There is a sprint review at the end of the sprint where finished stories are demonstrate to all relevant stakeholders. Furthermore, a sprint retrospective will be held in order to assess what went well and what did not. Team progress is displayed on a giant screen visible to the entire team.

Sprint 1

3. Refined Release Plan

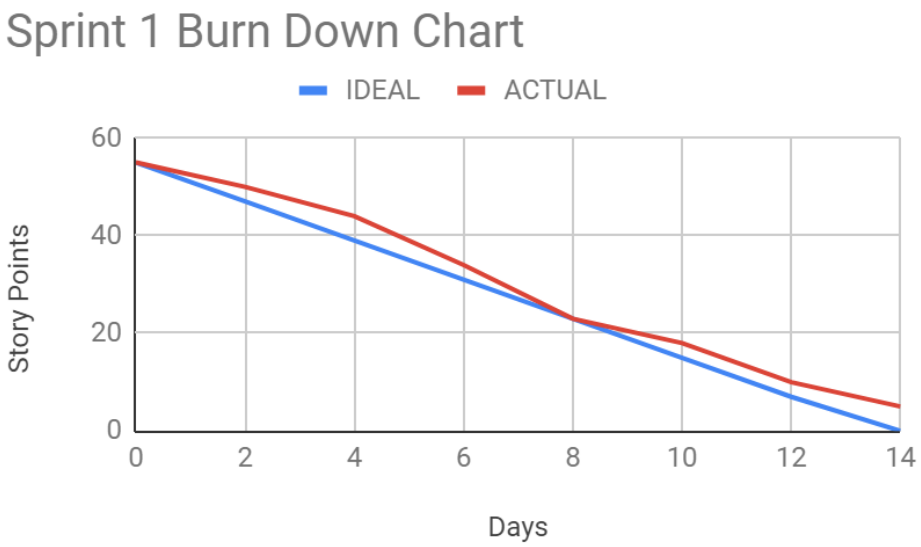
	Iteration 1		Iteration 2		Iteration 3	
	Story ID	**Points**	**Story ID**	**Points**	**Story ID**	**Points**
	S10	21	S9	13	S7	8
	S1	21	S11	13	S6	8
	S2	8	S8	8	S3	5
			S5	5	S14	8
			S4	13	S12	8
			S15	5	S13	13
Total	**50**		**57**		**50**	

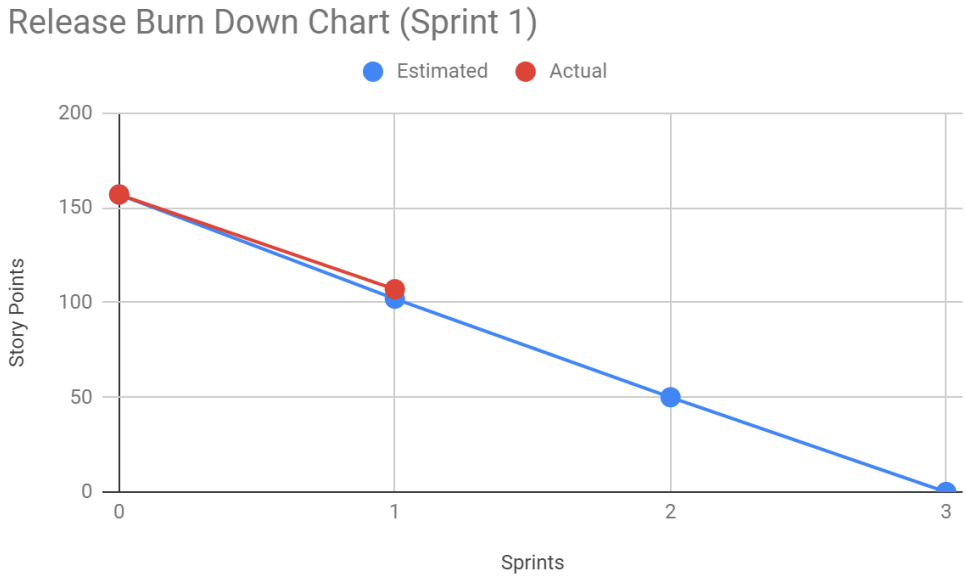
During the beginning of Sprint 1, our team came across the following issue:
Story cannot be tested completely without student "add class" feature.
Therefore, during the current sprint's iteration planning meeting we have decided to swap S8 and S11, the testing of which is dependent on students being able to add/drop courses, with S1 and S2 (originally planned for Sprint 2). Moving S1 to Iteration 1 also allows us to test another issue we've discovered:
Users are unable to access system when they are not connected to the school network on campus.
We also moved S9 to Iteration 2 to balance out the workload across sprints, based on story points.

6. Improvements Record at retrospective meetings

What went well	What can be improved
Story estimation accuracy was well done for this sprint.	Stories can be grouped and prioritized better so that they can be completely tested.
	Testing suites can be implemented to cover more stories

7. Burndown Charts for Sprint and Release





Sprint 1 Risks

1. System Administration stories are too long for two-week sprint
2. Student and User stories and requirements are dependent on System Administration
3. Student Enrollment system may require further decomposition into smaller stories
4. Student Enrollment system are too long for two-week sprint
5. Teacher, Student, and Administrator user database architecture is too large of a story

Sprint 2

3. Refined Release Plan

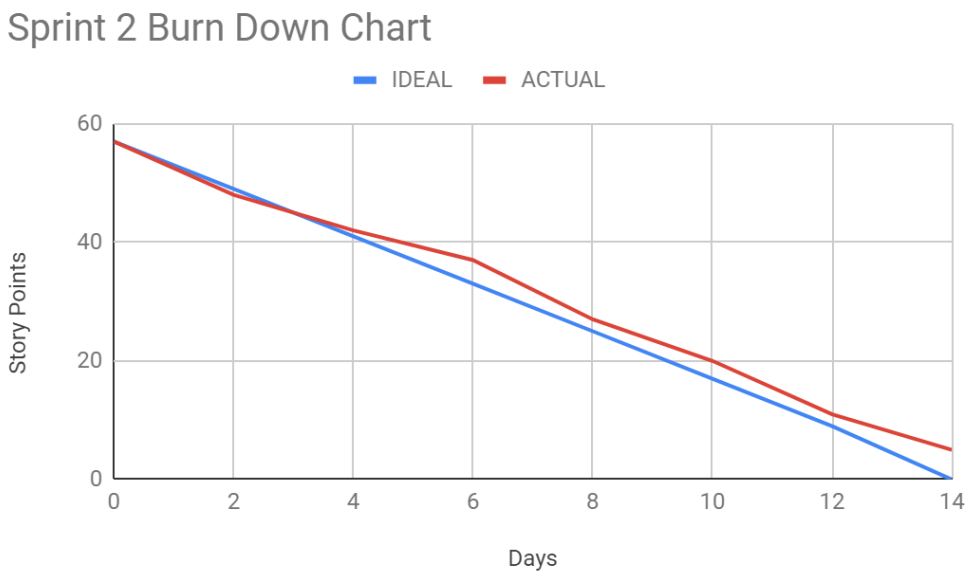
	Iteration 1		Iteration 2		Iteration 3	
	Story ID	**Points**	**Story ID**	**Points**	**Story ID**	**Points**
	S10	21	S9	13	S7	8
	S1	21	S11	13	S6	8
	S2	8	S8	8	S3	5
			S5	5	S14	8
			S4	13	S12	8
					S13	13
					S15	5
Total	**50**		**52**		**55**	

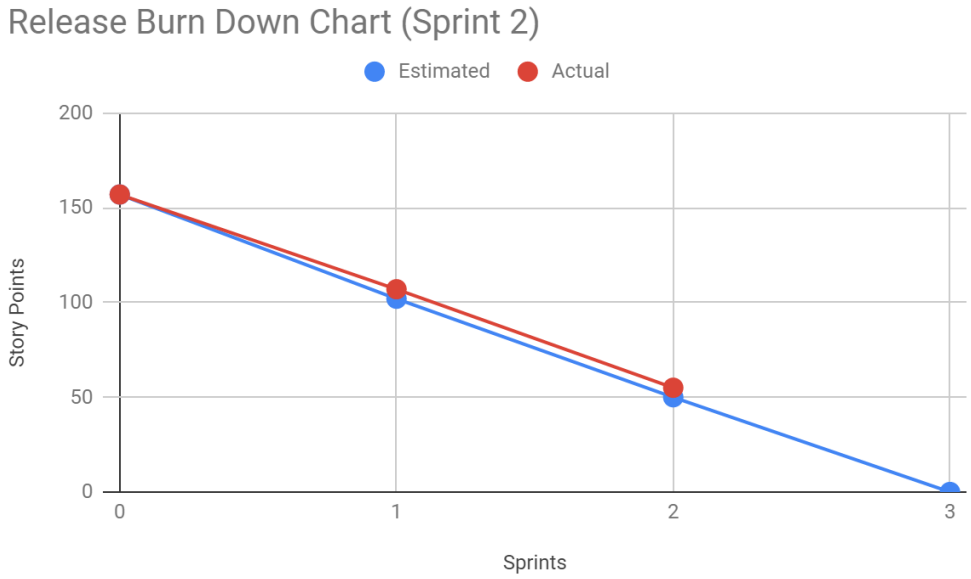
The first issue we've discovered:
Register button is too high on page.
The placement of the Register Button from S1 turned out to be too high, so minor adjustments need to be made to the code for that story.
Another issue we've discovered during the development of S4 in the current iteration:
Course history does not refresh after dropping classes.
To allow our developers enough time to create the fixes for these issues in addition to completing most of the stories already assigned to this sprint, we've decided to push S15 to Iteration 3.

6. Improvements Record at retrospective meetings

What went well	What can be improved
Stories were more thoroughly tested due to better grouping and prioritization	Need to follow test-driven development more strictly to minimize the number of bugs
Test automation initiative went well, now have tests to run against stories which gives us more confidence in our product	Need to delegate tasks better so that each developer has enough time to complete their assigned stories

7. Burndown Charts for Sprint and Release





Sprint 2 Risks

1. Teacher, Student, and Administrator user database architecture is too large of a story
2. Course, Class, Student, and Teacher data objects rely on incomplate University data
3. Implementing S9 will take more time over S11
4. S10 is tied too heavily to S9, so S9 must also have reliable S10 implementation
5. S8 and S5 should be quick wins, but not prioritized

Sprint 3

3. Refined Release Plan

	Iteration 1		Iteration 2		Iteration 3	
	Story ID	**Points**	**Story ID**	**Points**	**Story ID**	**Points**
	S10	21	S9	13	S7	8
	S1	21	S11	13	S6	8
	S2	8	S8	8	S3	5
			S5	5	S14	8
			S4	13	S12	8
					S13	13
					S15	5
Total	**50**		**52**		**55**	

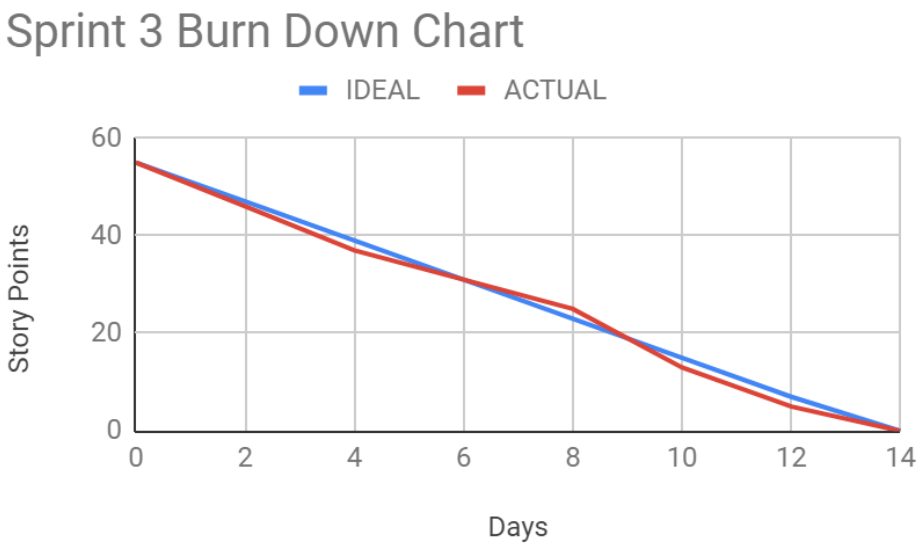
S14 (issue found) to be continued in Iteration 4.

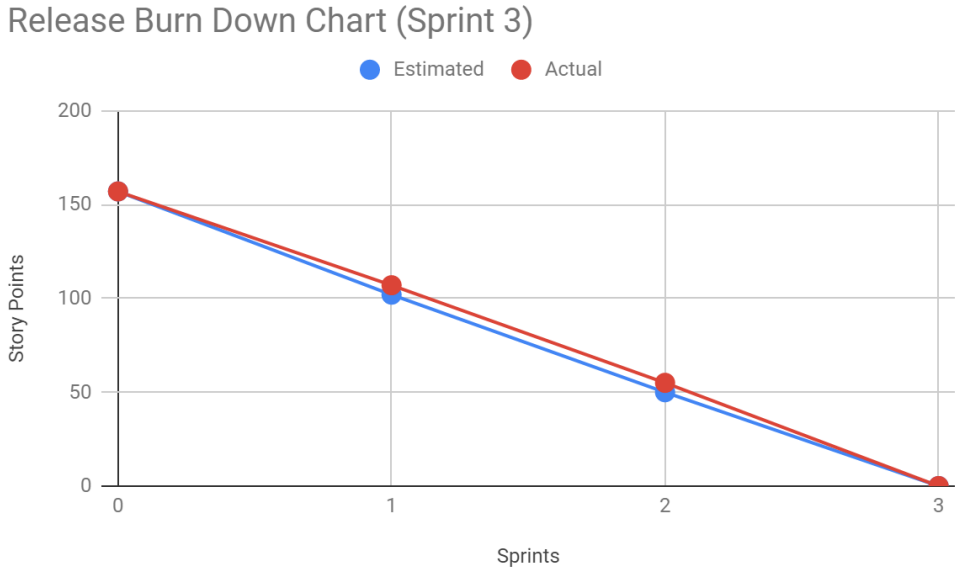
The following issue was discovered after development of S14:
Students are not notified when they are dropped from the class by their professor.
Since developers are already preoccupied with other stories in the current iteration, S14 is now planned to be closed out (upon fix) in Iteration 4.

6. Improvements Record at retrospective meetings

What went well	What can be improved
Test-driven development was followed more strictly, decreasing the number of bugs that occurred this iteration	Need better documentation on features and technologies used for non-technical audience to understand
Developers were more satisfied with the way tasks were delegated this iteration	Need to timebox standups better, discuss less of the details and more of what got done

7. Burndown Charts for Sprint and Release





Sprint 3 Risks

1. S13 story is actually worth more than 13 points
2. Current Sprint stories rely heavily on previous story implementations and testing
3. S13 course history data objects relies on incomplete University data
4. Storing thousands of course documents relies on more testing infrastructure
5. Testing thousands of course data relies on more testing infrastructure

8. Define the DONE criteria for top 10 user stories defined in your product backlog.

1. As a system administrator, I would like to remove authorizations from student and professor accounts.
    a. 100% completion of code functionality for removing authorizations.
    b. Quality assurance tests for codes detect 0 bugs
    c. Removing authorizations function is able to perform on all platforms.
2. As a system administrator, I would like to enter new students and professors into the system
    a. 100% completion of code functionality for entering new accounts.
    b. Quality assurance tests for codes detects 0 bugs
    c. Entering new accounts function is able to perform on all platforms.
3. As a system administrator, I would like to enter new courses for the semester and set descriptions for them.
    a. 100% completion of code functionality for entering new courses with descriptions.
    b. Quality assurance tests for code detect 0 bugs
    c. Entering new courses with descriptions function is able to perform on all platforms.
4. As a student, I would like to be able to register open classes when my enrollment window is open.
    a. 100% completion of code functionality for registering classes.
    b. Quality assurance tests for codes detects 0 bugs
    c. Registering classes function is able to perform on all platforms.
5. As a student, I would like to be able to drop classes before the second week of classes.
    a. 100% completion of code functionality for dropping classes.
    b. Quality assurance tests for code detects 0 bugs
    c. Dropping classes function is able to perform on all platforms.
6. As a system administrator, I would like the option to set prerequisites for courses on the course enrollment system so that only students with prerequisites can register for a course.
    a. 100% completion of code functionality for setting prerequisites
    b. Quality assurance tests for codes detect 0 bugs
    c. Setting prerequisites function is able to perform on all platforms.
7. As a student, I would like to be able to view my current schedule.
    a. 100% completion of code functionality for viewing current schedule.
    b. Quality assurance tests for codes detect 0 bugs
    c. Viewing current schedule function is able to perform on all platforms.
8. As a student, I would like to be able to view my course history and grades earned.
    a. 100% completion of code functionality for viewing course history and grades.
    b. Quality assurance tests for codes detect 0 bugs

    c. Viewing course history and grades function is able to perform on all platforms.
9. As a student, I would like to get notification when my registration window is opened.
    a. 100% completion of code functionality for receiving notification for registration.
    b. Quality assurance tests for codes detect 0 bugs
    c. Receiving notification for registering function is able to perform on all platforms.
10. As a professor, I would like to be able to enter students' grades into the system.
    a. 100% completion of code functionality for entering grades
    b. Quality assurance tests for codes detect 0 bugs
    c. Entering grades function is able to perform on all platforms.

10. Check and see if your team's practices meet the intents/requirements defined in CMMI V1.3's PP and PMC specific practices.

Project Planning

Practice	Artifacts	Gaps
SP 1.1 Estimate the Scope of the Project	Refined Release Plans	Task Descriptions, Work Packages, WBS
SP 1.2 Establish Estimates of Work Product and Task Attributes	Product Backlog, User Stories	Technical Logistic Reports and Estimations
SP 1.3 Define Project Lifecycle Phases	Scrum Process Description	
SP 1.4 Estimate Effort and Cost	Product Backlog, User Stories	Estimation Rational, Project Logistics Estimates
SP 2.1 Establish the Budget and Schedule	Product Backlog	Schedule Dependencies, Project Budget
SP 2.2 Identify Project Risks	Sprint Risks	Project Risks
SP 2.3 Plan Data Management		Data management plan, Business Requirements, Data content and format description
SP 2.4 Plan the Project's Resources		Work Packages, Task Dictionary, Staffing Requirements, Critical Resources
SP 2.5 Plan Needed Knowledge and Skills		Team Charter, Inventory of Skills, Databases
SP 2.6 Plan Stakeholder Involvement		Team Charter
SP 2.7 Establish the Project Plan	Project Plan	
SP 3.1 Review Plans That Affect the Project	Refined Release Planning	
SP 3.2 Reconcile Work and Resource Levels	Sprint Retrospectives, Refined Release Planning	
SP 3.3 Obtain Plan Commitment		Commitment Charter

Using the Findings figure table above, Project Planning specific practices are measured by any associated Artifacts. Any missing work-products are suggested under the Gaps column.

With this analysis, we can see there are several practices lacking work-products from the current assignment. Achieving Project Planning goals will require more production toward the later goals, where more analysis and work-products can satisfy the missing gaps.

Project Monitoring and Control

Practice	Artifacts	Gaps
SP 1.1 Monitor Project Planning Parameters	Project Plan, Sprints, Burndown Charts	
SP 1.2 Monitor Commitments		Commitment Reviews, Interviews
SP 1.3 Monitor Project Risks		Project Risk Monitoring. Risk Mitigation Plan. Interviews.
SP 1.4 Monitor Data Management		Data Management, Tools and Operation, Interviews
SP 1.5 Monitor Stakeholder Involvement		Stakeholder Involvement, Interviews
SP 1.6 Conduct Progress Reviews	Improvements Record, Sprint Retrospectives, Burndown Charts	
SP 1.7 Conduct Milestone Reviews	Improvements Record	
SP 2.1 Analyze Issues	Issue Table	
SP 2.2 Take Corrective Action	Refined Release Plans	
SP 2.3 Manage Corrective Actions	Refined Release Plans	

Using the Findings figure table above, Project Monitoring Control specific practices are measured by any associated Artifacts. Any missing work-products are suggested under the Gaps column.

With this analysis, we can see there are several practices lacking work-products from the current assignment. Achieving Project Monitoring Control goals will require more production toward the later goals, where more analysis and work-products can satisfy the missing gaps.

11. Design and develop a project summary report that covers topics discussed here and anything else your team think is interesting.

Project Summary Report

This assignment has allowed us to manage a simulated project in which we applied common Agile and Scrum practices to deliver a Student Information System for Cal State Fullerton. The Student Information System is an online system built to help facilitate some of the more common administrative tasks that professors and administrative staff typically perform, like adding/dropping courses and establishing course prerequisites, as well as providing students a convenient portal to design course schedules and access all their academic information.

We started off by developing a Product Backlog that stores all of the User Stories necessary to establish a minimum viable product for the Student Information System. The User Stories were written in the standard User Story format and convey the types of functions that a particular member of the intended audience (e.g. student, professor, administrative staff) would be able to exercise once the User Story has been delivered. The Product Backlog was also prioritized so that the most essential features, as well as User Stories that other stories depend on, are to be delivered first.

The next step was to use Story Pointing to estimate the expected workload for each of the User Stories in the Product Backlog. Story pointing adds a numeric dimension to the User Stories, allowing the team to better allocate the tasks amongst the team members as well estimate the whole team's velocity per sprint. We decided to use the Fibonacci sequence as possible values to assign to the user stories, and deliberated over Story Point values for each story until we reached a consensus value for the story.

After all the User Stories were estimated, we developed a Release Plan in which tasks were roughly evenly distributed across the three planned iterations. Each planned iteration was assigned about 52 story points worth of user stories. Note that this allows for flexibility in the sprint's plan due to possibility for bugs and requirements changes to occur which may delay or necessitate the reallocation of stories.

We also developed an issue tracking table used to maintain a history of issues that occurred during each sprint. Each issue is traceable to its corresponding user story, allowing developers to easily figure out in which feature the bug exists. Each issue is also evaluated for its severity so that developers can prioritize which bugs to patch first. The issue tracking table allows project managers to allocate work to other sprints in case developers are too occupied with developing their WIP features or fixing bugs.

As a preliminary to executing our Scrum process, we created a summary of the proposed Scrum process that lays out all the methodologies we would be using to conduct our project. We defined our iterations as 2 week sprints and described the different Scrum meetings we would conduct, like the daily morning stand-ups to discuss the tasks for the day, and the Sprint Retrospective to allow team members to review the strengths of the Scrum Process in the current iteration as well as evaluate points of improvement for the following sprint. The top risks were also identified before each sprint, in order to gauge how the upcoming Sprint was going to be like. Although we identified huge stories with points, explicitly identifying concerns helps us keep in mind potential issues and timesinks. This turned out to be a good exercise even within a simulated project.

A sprint retrospective is conducted at the end of each sprint. During these meetings, the team reflects on the last sprint. Each member in the team shares and reflects upon what went well or could be improved. These reflections are recorded in a sheet and is used to make improvements to the process. Things that went well will be kept in the process while adjustments are made to improve on the things that did not.

In this process, burn down charts were created to track the degree of plan deviations. The burn down charts include an ideal story point burn down and the actual story point burn down. These deviations help the team determine whether adjustments are needed to the process. This may include adjustments in plan, estimation, or resources. With these in mind, release plans can be refined and adjusted to suit the project's needs. Afterwards, the DONE criteria was explored for the top ten user stories from our evaluations. Simulated criteria was mainly surmised from previous experience, but we all had different backgrounds in tech ie one of us was more well versed in quality assurance while others were more knowledgeable about web development.

For the final step, the project's practices were compared against CMMI V1.3, specifically the practices of Project Planning (PP) and Project Monitoring Control (PMC). This was a welcome exercise after performing it for a previous project. A similar Findings table was used to compare artifacts and gaps according to example work-products from CMMI, which helped visualize what other processes we can do if we want to further develop the online Student Information System for Cal State Fullerton.