

# Student Information System

*Team 05*

David Jew: david.jew@csu.fullerton.edu

Catherine Do: catherd@csu.fullerton.edu

Holly Do: hdo88@csu.fullerton.edu

Noemi Quezada: noemi01@csu.fullerton.edu

CPSC 546 - Modern Software Management

Professor Bin Cong

Submission Date: 05/15/2016

# Table of Contents

## [Table of Contents](#)

### [Process Definition](#)

#### [Overview](#)

#### [PreGame: Planning Phase](#)

##### [Purpose](#)

##### [Activities](#)

##### [Work Products](#)

#### [PreGame: Staging Phase](#)

##### [Purpose](#)

##### [Activities](#)

##### [Work Products](#)

#### [Development Phase](#)

##### [Purpose](#)

##### [Sprints](#)

##### [Activities](#)

##### [Work Products](#)

#### [Release Phase](#)

##### [Purpose](#)

##### [Activities](#)

##### [Work Products](#)

### [Pre-Game: Planning Phase](#)

#### [Vision](#)

##### [Description of system to be built](#)

##### [Project Goals](#)

##### [Business/Organization Goals](#)

##### [Budget](#)

##### [Major Features](#)

##### [System Context Diagram](#)

### [Product Backlog](#)

### [Pre-Game: Staging phase](#)

#### [Product Backlog](#)

#### [Release Plan](#)

### [Burndown chart - Release](#)

### [Done Criteria - Top 10 User Stories](#)

### [Sprint 1](#)

#### [Backlog Refinement](#)

#### [Sprint Planning](#)

#### [Top 5 Risks](#)

#### [Sprint Backlog](#)

#### [Sprint Burndown chart](#)

[Refined Product Backlog](#)  
[Refined Release Plan](#)  
[Refined Release Burndown chart](#)  
[Retrospective Meeting](#)  
[Sprint 2](#)  
[Backlog Refinement Meeting](#)  
[Sprint Planning Meeting](#)  
[Top 5 Risks](#)  
[Sprint Backlog](#)  
[Sprint Burndown chart](#)  
[Refined Product Backlog](#)  
[Refined Release Plan](#)  
[Refined Release Burndown chart](#)  
[Retrospective Meeting](#)  
[Sprint 3](#)  
[Backlog Refinement](#)  
[Sprint Planning Meeting](#)  
[Top 5 Risks](#)  
[Sprint Backlog](#)  
[Sprint Burndown Chart](#)  
[Refined Product Backlog](#)  
[Refined Release Plan](#)  
[Refined Release Burndown Chart](#)  
[Retrospective Meeting](#)  
[Team Practices vs CMMI V.1.3's PP and PMC specific practices](#)

## Process Definition

### Overview

Our software development process incorporates Scrum and XP practices to develop high-value solutions to complex problems through time-boxed deliverables, with a continuous emphasis in collaboration, simplicity, feedback and customer satisfaction.

At the core of our process, we use Scrum practices such as Sprint Planning, Sprint Review, and Daily Scrum meetings to manage the schedule and budget of the project, while incorporating XP practices such as pair programming, acceptance tests, and CRC cards for the design and engineering of the product. To avoid technical debt by going too fast and creating messy codes, we utilized technical practices such as test driven development, continuous integration, pair programming, collective ownership, and refactoring. These practices serves as a way for us to document and measure the codes so

that we can prevent technical debt from occurring. This produce quality codes which leads to a successful Scrum team.

Our process is composed of four phases, Pre Game - Planning Phase, PreGame - Staging Phase, Development Phase which consists of  $N$  number of time-boxed iterations called Sprints based on the complexity of the product and a Release Phase. In the following sections we discuss some of the Activities and Work Products that cover these phases.

## **PreGame: Planning Phase**

### **Purpose**

To understand client and business objectives, establish the product vision and understand the budget of the project.

### **Activities**

During the PreGame Planning phase, we:

- Identify the vision of the product
- Understand the budget of the project
- Draft a rough overall scope of the product
- Draft the initial product backlog and estimate items using tools such as the Planning Poker activity.

### **Work Products**

During the PreGame Planning phase, the work products that will be created are:

- **Scope and Vision document** - where we identify the application we are building, goals of the project, and the organization or business. We also identify major system features, and stakeholders.
- **Context Diagram** - Visually illustrate the application to be built in a black box approach by identifying external entities that interface with the system through data, and control flows, which help identify the scope boundary of the system and major system features.
- **Product Backlog** - Initial list of requirements with initial estimates.

## **PreGame: Staging Phase**

### **Purpose**

To create a plan prior to entering the first sprint. Identify any additional requirements, estimate and prioritize the requirements and update the Product Backlog and Release Plan.

### **Activities**

- Identify more requirements and user stories.

- Using the Planning Poker activity, reestimate additional requirements.
- Prioritize Requirements
- Update Product Backlog
- Update the Release Plan
- Conduct an exploratory design or prototypes as needed.

### Work Products

- **Product Backlog** - A prioritized list of all work to be done for the project derived from the gathered requirements.
- **Release Plan** - After the Product Backlog is estimated and prioritized, we estimate the velocity of the scrum team and create plan that estimates project completion date based on user stories that have been defined.
- **Prototypes**.

## Development Phase

### Purpose

To create timeboxed incremental releases of the product. The cycle repeats until no more items are left in the product backlog, the budget is depleted or a deadline approaches.

### Sprints

Our sprints are a timeboxed iteration of two weeks that consist of Backlog Refinement, Sprint Planning, development work, Sprint Review and Sprint Retrospective meetings culminating to a small release of the product.

### Activities

- **Backlog Refinement Meeting** - Held a few days before the Sprint Planning Meeting so that the Product Owner has time to revise priorities before commitments are made. The main purpose is to clarify and decompose higher priority Product Backlog Items so that Product Backlog Items are Independent, Negotiable, Valuable, Estimable, Small and Testable.
- **Sprint Planning Meeting** - A Sprint Planning meeting should be conducted with the entire team to plan the work to be performed and identify a sprint goal, which is the objective set that can be met at the end of the sprint.
- **Daily Scrum Meetings** - A daily scrum meeting will take place for the team to review what was done yesterday, what is planned for today, and if there are any impediments.
- **Development** - During development we go through Requirement Analysis, Design, Code, Integration, Testing, until Deployment.

- XP practices such as Pair Programming, Continuous code integration, collective ownership for coding, CRC cards for design and Acceptance and Regression testing for Testing are utilized during development.
- **Sprint Review Meeting-** A Sprint Review Meeting stakeholders who are interested in the product are welcome to join. The agenda consists of a product demonstration of a potentially shippable product increment to the Product Owner. Product Owner analysis of the product backlog items committed to the Sprint, the Sprint goals and the Product Backlog is updated accordingly and a Sprint Velocity is counted. Finally stakeholders can give feedback and this may result in new Product Backlog items that are prioritized based on the Product Owners vision.
- **Sprint Retrospective Meeting-** Conducted by the Scrum Master after a just-concluded sprint and it helps to discuss what can be changed that might make the next sprint more productive.

### Work Products

- **Small Testable Potentially shippable product increment** - A subset of the entire system to be built that has been tested and that covers the product backlog items defined in the Sprint.
- **Revised Product Backlog** - A revised product backlog produced after the Sprint Review Meeting.

## Release Phase

### Purpose

Composed of two processes that help deliver the Accepted Deliverables to the customer and identifying, documenting and internalizing the lessons learned during the project.

### Activities

- **Ship Deliverables** - Deliverables that meet the acceptance criteria and definition of done are delivered to the stakeholders.
- **Retrospect Project** - Organizational stakeholders and the Scrum team get together to retrospect the project and identify and document the lessons learned.

### Work Products

- **Accepted Deliverables**
- **Agreed Actionable Improvements document** - Lessons learned can lead to an Agreed Actionable Improvements document that can help be implemented in future projects. (link: <http://www.scrumstudy.com/blog/exploring-the-release-phase-of-a-scrum-project/>)

# Pre-Game: Planning Phase

## Vision

### Description of system to be built

For Cal State University, Fullerton students, professors and administrative staff who have a need to manage various aspect of the education system the Student Information System is an online informational system that maintains the schedule of course offerings, register students in classes, show class rolls, and record grades. Unlike other Student Information Systems that only focus on creating a robust functional system our system focuses on the usability and accessibility of the system to students, professors and administrative staff through various tactics such as notification that alert students when courses and grades are available through email and text message notifications to aid in the accessibility of information and an improved user experience.

### Project Goals

The objective is to build a web-based student information system that includes course schedule maintenance, student course registration, grade reporting, graduation checks, and student and professor management. The system needs to be operational by May 15, 2016 and needs to cost less than \$100,000.

### Business/Organization Goals

The goals of California State University, Fullerton are:

- Create an easy to use SIS that increases productivity and decreases workload and the need for manually created documentation in hard copy or digital form
- Ensure that the system is accessible to staff and students alike on all mobile or web platforms
- Automate the course and student management process
- Reduce costs and streamline system processes to save time and money

### Budget

California State University, Fullerton has dedicated \$100,000 of the annual school budget to the development of the Student Information system.

### Major Features

FE-1: Create, view, modify and remove courses.

FE-2: Create, view, modify students and professors.

FE-3: Register and drop courses.

FE-4: Update student grade levels.

FE-5: View current student schedule and course history with grades earned.

FE-6: Drop student from a class.

FE-7: Enter grades.

FE-8: View class row call and student course history.

FE-9: Assign passwords to users of the system.

FE-10: Remove student and professor authorizations from the system.

FE-11: Create, view, modify course schedules for a semester.

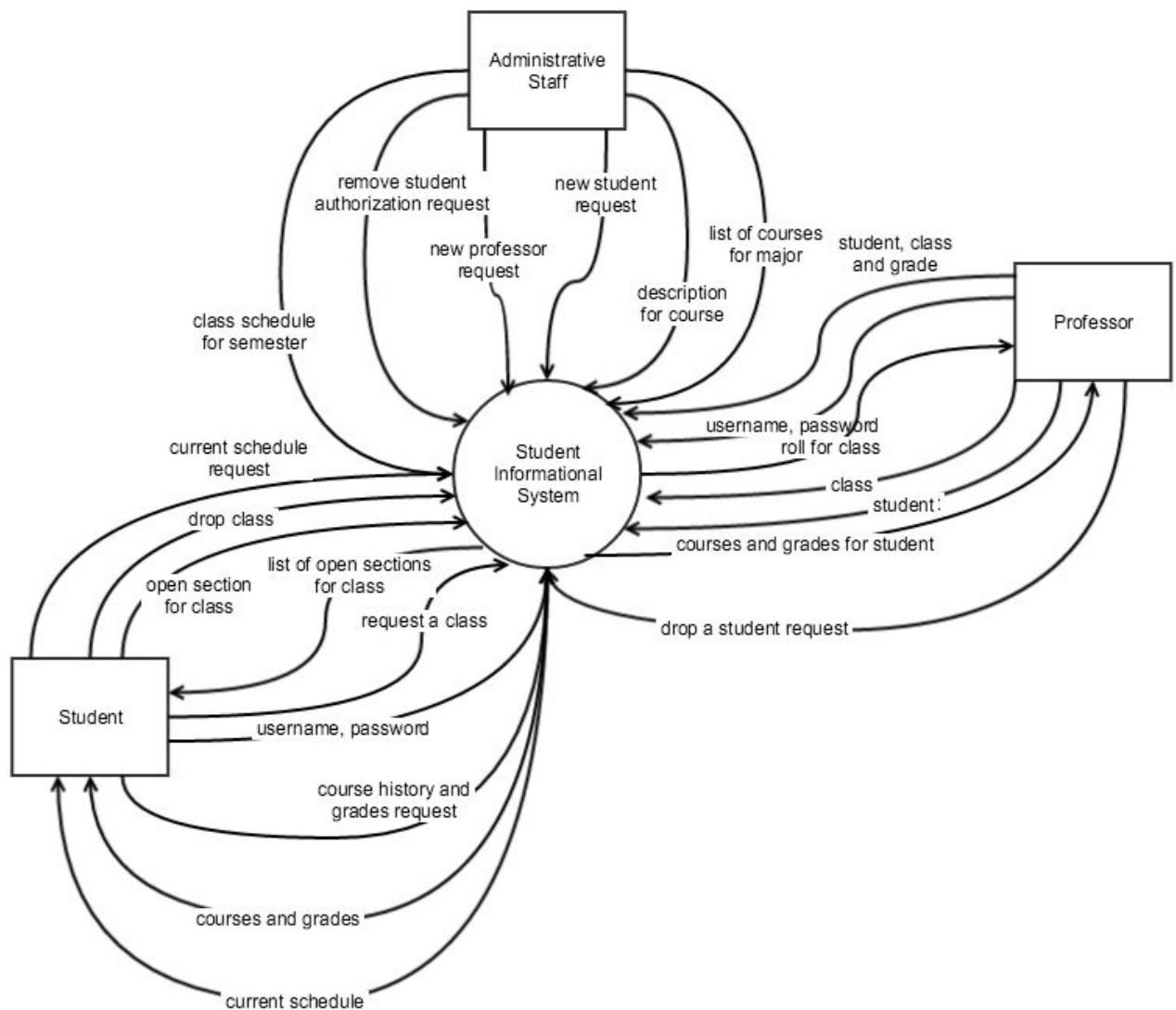
FE-12: Add, view, courses to a major.

FE-13: Perform graduation checks.

FE-14: Produce student transcripts.

FE-15: Notify students when grades have been posted and registration date opens.

### System Context Diagram





## Product Backlog

This is the initial product with initial estimates based on the features and requirements given by the product owner. As of now they are not organized by priority, this will be done later.

Story points are a relative measurement of the complexity of a task. It is necessary to utilize story points because people are better at relative measurements than precise measurements. Thus story points are assigned to tasks based upon the relative difficulty to each other. For example, we first assigned the task “Search for Classes” 20 story points. Then when we looked at the “Drop Courses” task, we estimated its story points based upon the perceived difficulty it is to implement in comparison with the “Search for Class” task.

Name	Description	Story Points
Search for Classes	As a student, I can search for classes so that I can see what classes are open.	20
Drop Courses	As a student, I can drop a course so that I can unenroll from a class.	8
View Schedules	As a student, I can view my current schedule and my previous course history so that I can see my the courses I have this semester and my grades from previous semesters.	20
View Student Schedules	As a professor, I can view the schedule of one of the students enrolled in my class so that I can see what classes they are enrolled this semester.	8
View Roll Call	As a professor, I can view the roll call of one my classes so that I can see what students are in class and what students are not.	5
Drop Students	As a professor, I can drop students from my class so that they are no longer enrolled in my class.	8
Enter grades	As a professor, I can enter grades so that I can record which students have passed and which students have failed my class.	8
Add Students	As an administrative staff, I can add new students to the system so that they can register for classes.	8

Add Professors	As an administrative staff, i can add new professors to the system so that they can teach classes.	8
Remove Authorizations	As an administrative staff, I can remove student and professor authorizations so that they can no longer access the system.	5
Add Class Schedules	As an administrative staff, I can add class schedules for a semester so that students can register for classes.	13
Add Descriptions for New Classes.	As an administrative staff, I can add descriptions for new classes so that students can view what the classes are about.	5
Add Class List to Majors.	As an administrative staff, I can add a list of classes for a major so that a class can be related to a major.	8
Request Student Transcripts and Student Histories.	As an administrative staff, I can request student transcripts and history of courses taken with grades so that I can inspect the academics of a student.	20
Student Notifications	As a student, I can get a notification when my grades have been posted and when my registration has opened up.	13

## Pre-Game: Staging phase

Prior to entering the first sprint, we analyzed the initial product backlog that has been created in the Pre-Game: Planning Phase and we added any missing requirements and estimated the stories and prioritized them based on the product owners vision.

### Product Backlog

Top priority product backlog items are on top.

The Product Owner expressed that his vision for this system is to be more friendly accessible to students. Therefore, most of his top priority items are aimed towards students.

Name	Description	Story Points
Search for Classes	As a student, I can search classes so that I can see what classes are open.	20

Register for Classes	As a student, I can register for classes so that I can be enrolled as a student this semester.	20
Drop Courses	As a student, I can drop a course so that I can unenroll from a class.	8
View Current Schedule	As a student, I can view my current schedule so that I can see what classes I am enrolled this semester.	5
Enter Grades	As a professor, I can enter grades so that I can record which students have passed and which students have failed my class.	8
View Course History	As a student, I can view my course history so that I can see what classes I was enrolled in previous semesters and the grades I received.	13
Drop Students	As a professor, I can drop students from my class so that they are no longer enrolled in my class.	5
Add Class Schedules	As an administrative staff, I can add class schedules for a semester so that students can register for classes.	13
Add Students	As an administrative staff, I can add new students to the system so that they can register for classes.	8
Add Descriptions for New Classes	As an administrative staff, I can add descriptions for new classes so that students can view what the classes are about.	5
Add Prerequisites for New Classes	As an administrative staff, I can add prerequisites to classes so that students know what classes need to have completed before registering for a class.	5
Student Notifications	As a student, I can get a notification when my grades have been posted and when my registration has opened up.	13
Request Student Transcripts	As an administrative staff, I can request student transcripts and history of courses taken with grades so that I can inspect the academics of a student.	20
Add Professors	As an administrative staff, i can add new professors to the system so that they can teach classes.	8
View Roll Call	As a professor, I can view the roll call of one my classes so that I can see what students are in class and what students are not.	5

Add Class Lists to Majors	As an administrative staff, I can add a list of classes for a major so that a class can be related to a major.	8
Remove Authorizations	As an administrative staff, I can remove student and professor authorizations so that they can no longer access the system.	5
Assign Passwords	As an administrative staff, I can assign passwords to students and professors so that they can login into the system.	5
View Student Schedules	As a professor, I can view the schedule of one of the students enrolled in my class so that I can see what classes they are enrolled this semester.	8

### Release Plan

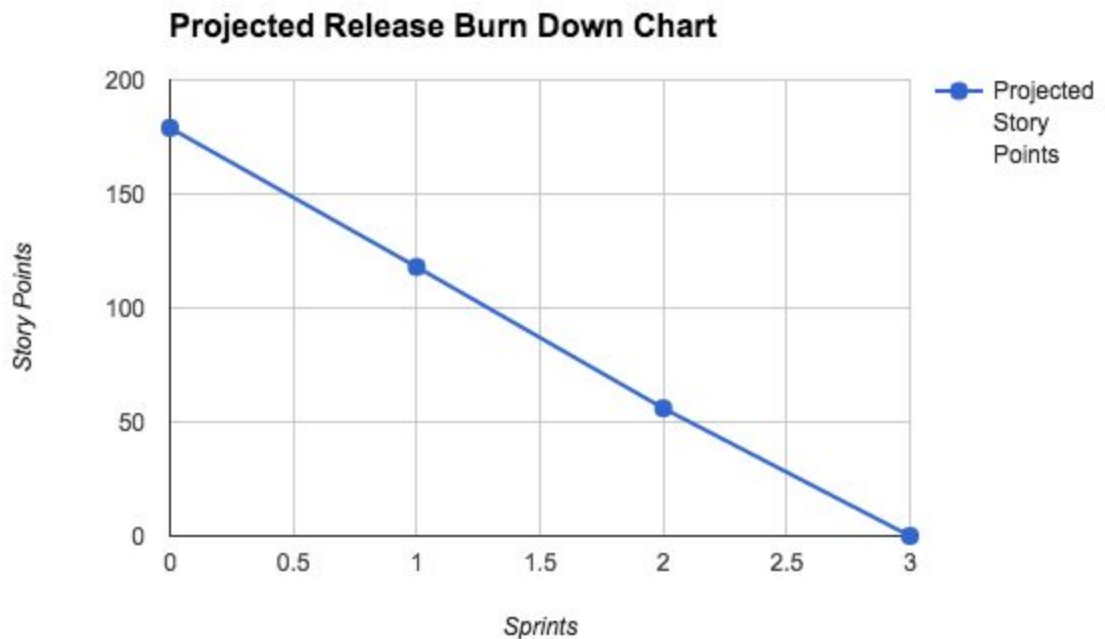
After providing rough estimates to the product backlog items and it has been prioritized by the Product Owner we estimated the velocity of the scrum team. Since our project is planned with 3 sprints timeboxed at two weeks iterations. To finish within 3 sprints, we are planning to average 58 story points per Sprint, so we set a capacity of 65 story points per sprint to be able to reach this goal.

	<b>Iteration1</b>	<b>Iteration 2</b>	<b>Iteration 3</b>
	Search for Classes (20)	View Course History (13)	Request Student Transcripts (20)
	Register for Classes (20)	Drop Students (5)	Add Professors (8)
	Drop Courses (8)	Add Class Schedules (13)	View Roll Call (5)
	View Current Schedule (5)	Add Students (8)	Add Class Lists to Majors (5)
	Enter Grades (8)	Add Descriptions for New Classes (5)	Assign Passwords (5)
		Add Prerequisites for New Classes (5)	View Student Schedules (8)
		Student Notifications (13)	Remove Authorizations(5)
<b>Plan</b>	61	62	56

<b>Capacity</b>	65	65	65
-----------------	----	----	----

## Burndown chart - Release

The burndown chart below shows the ideal line in blue of how many story points should be completed per sprint to complete the project on time.



## Done Criteria - Top 10 User Stories

### Definition of Done

1. Search for Classes:
  - a. Search code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Search code is fully integrated with Student Information Systems
  - d. Search function passes functionality tests on all platforms
  - e. Search UI and results pages pass UX testing
  - f. Full search documentation is added to the main document
2. Register for Classes:
  - a. Register for classes code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Register for classes code is fully integrated with Student Information Systems
  - d. Register for classes function passes functionality tests on all platforms
  - e. Register for classes page passes UX testing
  - f. Full register for classes documentation is added to the main document
3. Drop Courses:

- a. Drop courses code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Drop courses code is fully integrated with Student Information Systems
  - d. Drop courses function passes functionality tests on all platforms
  - e. Drop courses page passes UX testing
  - f. Full drop courses documentation is added to the main document
4. View Current Schedule:
- a. View current schedule code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. View current schedule code is fully integrated with Student Information Systems
  - d. View current schedule function passes functionality tests on all platforms
  - e. View current schedule page passes UX testing
  - f. Full view current schedule documentation is added to the main document
5. Enter Grades:
- a. Enter grades code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Enter grades code is fully integrated with Student Information Systems
  - d. Enter grades function passes functionality tests on all platforms
  - e. Enter grades page passes UX testing
  - f. Full enter grades documentation is added to the main document
6. View Course History:
- a. View course history code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. View course history code is fully integrated with Student Information Systems
  - d. View course history function passes functionality tests on all platforms
  - e. View course history page passes UX testing
  - f. Full view course history documentation is added to the main document
7. Drop Students:
- a. Drop students code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Drop students code is fully integrated with Student Information Systems
  - d. Drop students function passes functionality tests on all platforms
  - e. Drop students page passes UX testing
  - f. Full drop students documentation is added to the main document
8. Add Class Schedules:
- a. Add class schedules code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Add class schedules code is fully integrated with Student Information Systems
  - d. Add class schedules function passes functionality tests on all platforms
  - e. Add class schedules page passes UX testing
  - f. Full add class schedules documentation is added to the main document

9. Add Students:
  - a. Add students code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Add students code is fully integrated with Student Information Systems
  - d. Add students function passes functionality tests on all platforms
  - e. Add students page passes UX testing
  - f. Full add students documentation is added to the main document
10. Add Descriptions for New Classes:
  - a. Add descriptions for new classes code functionality is 100% complete
  - b. Code passes internal QA tests with 0 bugs
  - c. Add descriptions for new classes code is fully integrated with Student Information Systems
  - d. Add descriptions for new classes function passes functionality tests on all platforms
  - e. Add descriptions for new classes page passes UX testing
  - f. Full add descriptions for new classes documentation is added to the main document

## Sprint 1

Timebox: 3/21/2016 - 4/4/2016

### Backlog Refinement

According to the release plan for Sprint 1, our focus is on Search for Classes, Register for Classes, Drop Courses, View Current Schedule and Enter Grades user stories. We held a Backlog Refinement meeting to further discuss and refine these users stories prior to starting the sprint and came up with a refined product backlog as below, which includes some acceptance criteria.

Name	Description	Story Points
Search for Classes by Class Number	As a student, I can search classes so that I can see what classes are open. <b>Acceptance Criteria / DONE:</b> Search by class number	20
<del>Register for Classes</del>	<del>As a student, I can register for classes so that I can be enrolled as a student this semester.</del> <i>Has been broken down to: Add Classes to Cart, Checkout Cart</i>	<del>20</del>

Add Classes to Cart	<p>As a student, I can add a class to my cart so that I can keep track of which classes I want to enroll in.</p> <p><b>Acceptance Criteria / Done:</b></p>	8
Checkout Cart	<p>As a student, I can checkout my cart so that I can enroll in the classes in my cart this semester.</p> <p><b>Acceptance Criteria / DONE:</b></p> <p>Only allow to checkout cart if they have a valid registration date.</p>	8
Drop Courses	<p>As a student, I can drop a course so that I can unenroll from a class.</p> <p><b>Acceptance Criteria / DONE:</b></p> <p>Provide a warning before allowing a student to drop a course.</p>	8
View Current Schedule	<p>As a student, I can view my current schedule so that I can see what classes I am enrolled this semester.</p> <p><b>Acceptance Criteria/ DONE:</b></p> <p>Be displayed in a list format.</p>	5
Enter Grades	<p>As a professor, I can enter grades so that I can record which students have passed and which students have failed my class.</p> <p><b>Acceptance Criteria / DONE:</b></p> <p>Professors should be able to input grades for multiple students at the same time.</p>	8
View Course History	<p>As a student, I can view my course history so that I can see what classes I was enrolled in previous semesters and the grades I received.</p>	13
Drop Students	<p>As a professor, I can drop students from my class so that they are no longer enrolled in my class.</p>	5
Add Class Schedules	<p>As an administrative staff, I can add class schedules for a semester so that students can register for classes.</p>	13
Add Students	<p>As an administrative staff, I can add new students to the system so that they can register for classes.</p>	8
Add Descriptions for New Classes	<p>As an administrative staff, I can add descriptions for new classes so that students can view what the classes are about.</p>	5



Add Prerequisites for New Classes	As an administrative staff, I can add prerequisites to classes so that students know what classes need to have completed before registering for a class.	5
Student Notifications	As a student, I can get a notification when my grades have been posted and when my registration has opened up.	13
Request Student Transcripts	As an administrative staff, I can request student transcripts and history of courses taken with grades so that I can inspect the academics of a student.	20
Add Professors	As an administrative staff, i can add new professors to the system so that they can teach classes.	8
View Roll Call	As a professor, I can view the roll call of one my classes so that I can see what students are in class and what students are not.	5
Add Class Lists to Majors	As an administrative staff, I can add a list of classes for a major so that a class can be related to a major.	8
Remove Authorizations	As an administrative staff, I can remove student and professor authorizations so that they can no longer access the system.	5
Assign Passwords	As an administrative staff, I can assign passwords to students and professors so that they can login into the system.	5
View Student Schedules	As a professor, I can view the schedule of one of the students enrolled in my class so that I can see what classes they are enrolled this semester.	8

## Sprint Planning

For Sprint 1 our main goal was to focus on the most important Student features; to be able to enroll in classes, drop classes and view their schedule and incorporate basic functionality for Professors to enter grades.

## Top 5 Risks

The top 5 Risks we identified for this Sprint are:

1. Fully test the Checkout cart process, which enrolls a student in a course since not all courses have been populated in the system.
2. Properly implementing a multi-input of grades for Professors without allowing the session to time out.

3. Product Owner will be attending a Conference on March 22, 2016 and will not be available to be reached that day.
4. Applying a test-driven development approach and pair programming practice when it is not something the team is use to doing during development.
5. Getting prompt feedback from the Product Owner on Design pieces prior to implementing them.

### Sprint Backlog

Below is a snapshot of the Sprint Backlog for Sprint 1 before our Sprint Review Meeting.

Committed Backlog Items	Tasks	To Do	In Progress	Done
Search for Classes by Class Number				
	Create Tests			X
	Interface to Access Classes from the Database based on Class Number			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Integrate search feature with user authentication			X
	Implement Page Design			X
	Update the staging site with the latest version			X
Add Classes to Cart				
	Create Tests			X
	Model Cart system to add classes			X
	Access Classes to a users cart from the Database			X
	Interface with the database to Add a new Class to users cart			X

	Get restrictions from Product Owner of cases where a student can and can't add classes to a cart.			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Integrate add to cart feature with user authentication			X
	Implement the Page Design			X
	Update the staging site with the latest version			X
Checkout Cart				
	Create Tests			X
	Interface with the database to allow a student to checkout their cart to register for courses			X
	Gather restrictions from Product Owner of cases where a student can't checkout their cart.			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement Checkout cart feature with user authentication			X
	Implement the Page Design			X
	Update the staging site with the latest version			X
Drop Courses				
	Create Tests			X

	Interface with the database to allow a student to drop a course		X	
	Gather restrictions from Product Owner of cases where a student can't drop a course.			X
	Expand the Page Design			X
	Get Feedback on Page Design			X
	Implement the Page Design		X	
	Using Test-driven development write the code to interface with the database and the template		X	
	Implement User Authentication for the drop course feature		X	
	Update the staging site with the latest version	X		
View Current Schedule				
	Create Tests			X
	Interface with the database to retrieve a student's current registered courses			X
	Expand the Page Design			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement User Authentication so that a student can view only his current schedule			X
	Update the staging site with the latest version			X
Enter Grades				
	Create Tests			X

	Interface with the database to input grades for a class and particular student			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Update the staging site with the latest version			X

### **Sprint Burndown chart**

The Sprint burndown chart is used to track the progress of the product development by showing the amount of work remaining per day. The Scrum team can use the sprint burndown chart to estimate if their sprint goal will be completed in time. In Sprint 1, we used the first day to talk about the product backlog and understand user requirements. Therefore, no progress was made on the first day of the sprint. From day 2-9 we made great progress since our actual amount of work remaining is below that of what was estimated. However, we started to slow down and on the 10th day we could not finish our sprint goal in time because the product owner was not available to provide some crucial information.



### Refined Product Backlog

During our Sprint Review meeting we presented to the Product Owner what was completed during Sprint 1 and based on the DONE criteria specified the Product Owner had deemed the following stories to be DONE:

Stories	Story Points
Search for Classes by Class Number	20
Add Classes to Cart	8
Checkout Cart	8
View Current Schedule	5
Enter Grades	8
<b>Story Points Completed</b>	<b>49</b>

This shows that our sprint velocity for Sprint 1 was 49 story points. The absence of the Product Owner during the first Sprint caused a setback on the development and its one of the reasons we were not able to fully implement Drop a Course story due to delayed information from the Product Owner and unforeseen issues implementing few edge cases identified by the Product Owner. It was moved back to the Product Backlog and some stories were rearranged based on priority. As of now this is the state of the Product Backlog.

Name	Description	Story Points
Drop Courses	<p>As a student, I can drop a course so that I can unenroll from a class.</p> <p><b>Acceptance Criteria / DONE:</b></p> <p>Provide a warning before allowing a student to drop a course.</p>	8
View Course History	As a student, I can view my course history so that I can see what classes I was enrolled in previous semesters and the grades I received.	13
Drop Students	As a professor, I can drop students from my class so that they are no longer enrolled in my class.	5
Add Class Schedules	As an administrative staff, I can add class schedules for a semester so that students can register for classes.	13
Add Students	As an administrative staff, I can add new students to the system so that they can register for classes.	8
Add Descriptions for New Classes	As an administrative staff, I can add descriptions for new classes so that students can view what the classes are about.	5
Add Prerequisites for New Classes	As an administrative staff, I can add prerequisites to classes so that students know what classes need to have completed before registering for a class.	5
View Roll Call	As a professor, I can view the roll call of one my classes so that I can see what students are in class and what students are not.	5
Student Notifications	As a student, I can get a notification when my grades have been posted and when my registration has opened up.	13
Request Student Transcripts	As an administrative staff, I can request student transcripts and history of courses taken with grades so that I can inspect the academics of a student.	20
Add Professors	As an administrative staff, i can add new professors to the system so that they can teach classes.	8
Add Class Lists to Majors	As an administrative staff, I can add a list of classes for a major so that a class can be related to a major.	5

Remove Authorizations	As an administrative staff, I can remove student and professor authorizations so that they can no longer access the system.	5
Assign Passwords	As an administrative staff, I can assign passwords to students and professors so that they can login into the system.	5
View Student Schedules	As a professor, I can view the schedule of one of the students enrolled in my class so that I can see what classes they are enrolled this semester.	8

### Refined Release Plan

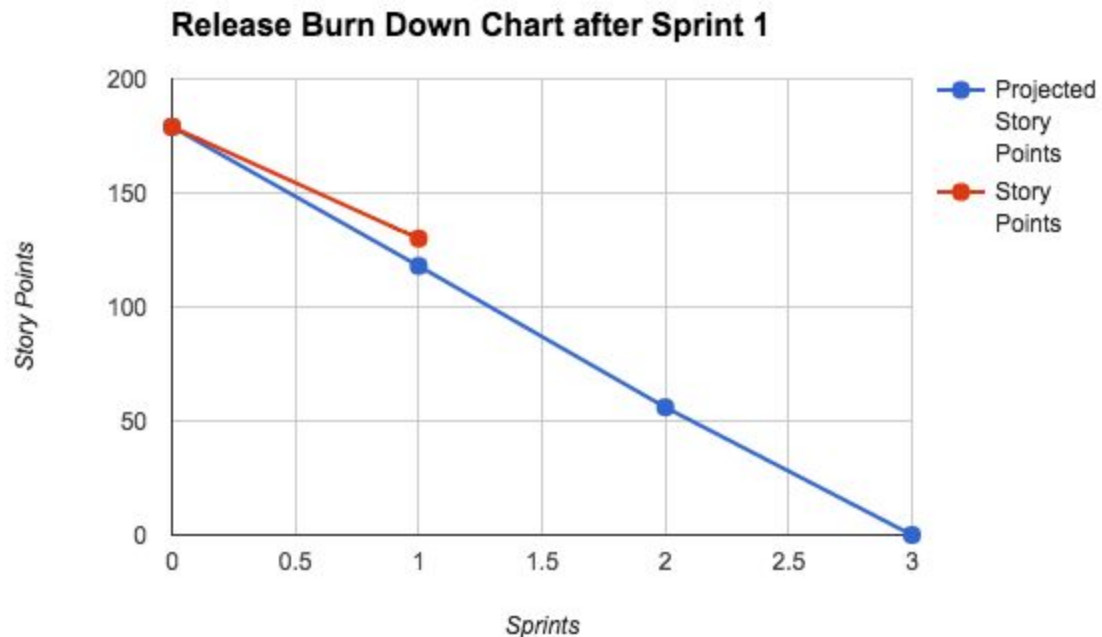
Even though in Sprint 1 we were not able to deliver all stories identified in the Release Plan and had to put back Drop a Course story back to the product backlog we decided to still continue with the 65 Story Point Capacity per sprint, and even though Drop a Course got moved back in terms of development, it did not affect the 3 Sprint deadline.

	Iteration1	Iteration 2	Iteration 3
	Search for Classes (20)	Drop Courses (8)	Student Notifications (13)
	Add Classes to Cart (8)	View Course History (13)	Request Student Transcripts (20)
	Checkout Cart (8)	Drop Students (5)	Add Professors (8)
	View Current Schedule (5)	Add Class Schedules (13)	Add Class Lists to Majors (5)
	Enter Grades (8)	Add Students (8)	Assign Passwords (5)
		Add Descriptions for New Classes (5)	View Student Schedules (8)
		Add Prerequisites for New Classes (5)	Remove Authorizations(5)
		View Roll Call (5)	
<b>Plan</b>	49	62	64
<b>Capacity</b>	65	65	65



## Refined Release Burndown chart

Our Project Velocity for Sprint 1, based on the stories DONE, is 49 story points. The Release Burndown chart below shows how we under delivered based on the ideal projection of story points.



## Retrospective Meeting

During our Sprint Retrospective Meeting we had identified that one of the reasons we did not completely fulfilled our committed product backlog items was because we were not able to reach the Product Owner in a timely manner during development to gather the important edge cases for Dropping a course. Due to this setback, we have implemented the proper improvements to our process, which are outlined below:

- Perform a small discovery phase before diving into full implementation of a story early during the Sprint, so that any proper knowledge and information is gathered before implementation.
- All team members, including the Scrum Master and Product Owner should attend their emails two times a day at minimum.

## Sprint 2

Timebox: 4/4/2016 - 4/18/2016

## Backlog Refinement Meeting

According to the release plan for Sprint 2, our focus is on Drop Courses, View Course History, Drop Students, Add Class Schedules, Add Students, Add Descriptions for New Classes, Add Prerequisites for New Classes, View Roll Call user stories. We held a Backlog Refinement meeting to further discuss and refine these users stories prior to starting the sprint and came up with a refined product backlog as below, which includes appropriate acceptance criteria for the user stories in question.

Name	Description	Story Points
Drop Courses	As a student, I can drop a course so that I can unenroll from a class. <b>Acceptance Criteria / DONE:</b> Provide a warning before allowing a student to drop a course.	8
View Course History	As a student, I can view my course history so that I can see what classes I was enrolled in previous semesters and the grades I received. <b>Acceptance Criteria / DONE:</b> Allow to see courses up to 4 years back. Search by semester.	13
Drop Students	As a professor, I can drop students from my class so that they are no longer enrolled in my class. <b>Acceptance Criteria / DONE:</b> Provide a warning before allowing a professor to drop a student.	5
Add Class Schedules	As an administrative staff, I can add class schedules for a semester so that students can register for classes.	13
Add Students	As an administrative staff, I can add new students to the system so that they can register for classes.	8
Add Descriptions for New Classes	As an administrative staff, I can add descriptions for new classes so that students can view what the classes are about.	5
Add Prerequisites for New Classes	As an administrative staff, I can add prerequisites to classes so that students know what classes need to have completed before registering for a class.	5
View Roll Call	As a professor, I can view the roll call of one of my classes so that I can see what students are in class and what students are not.	8

Student Notifications	As a student, I can get a notification when my grades have been posted and when my registration has opened up.	13
Request Student Transcripts	As an administrative staff, I can request student transcripts and history of courses taken with grades so that I can inspect the academics of a student.	20
Add Professors	As an administrative staff, i can add new professors to the system so that they can teach classes.	8
Add Class Lists to Majors	As an administrative staff, I can add a list of classes for a major so that a class can be related to a major.	5
Remove Authorizations	As an administrative staff, I can remove student and professor authorizations so that they can no longer access the system.	5
Assign Passwords	As an administrative staff, I can assign passwords to students and professors so that they can login into the system.	5
View Student Schedules	As a professor, I can view the schedule of one of the students enrolled in my class so that I can see what classes they are enrolled this semester.	8

### **Sprint Planning Meeting**

For Sprint 2, our main goal was to focus on the most important Student features; to be able to enroll in classes, drop classes and view their schedule and incorporate basic functionality for Professors to enter grades.

### **Top 5 Risks**

1. Coordination with Product Owner to get all appropriate information before beginning to implement a story since the Product Owner will be absent for two days due to doctors appointments in the first week of Sprint 2.
2. Understanding the story scope for Adding Class Schedules, since this seems to be a multi-part story.
3. Writing tests, before beginning to implement.
4. Loading all courses into the system before implementing Add Prerequisites for New Courses story.
5. System must be able to handle numerous enrollment request during the start of each semester.

### **Sprint Backlog**

Below is a snapshot of the Sprint Backlog for Sprint 2 before our Sprint Review Meeting.

Committed Backlog Items	Tasks	To Do	In Progress	Done
View Course History				
	Create Test			X
	Integrate User Authentication to view Course History			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Integrate search feature with user authentication			X
	Implement Page Design			X
	Update the staging site with the latest version			X
Drop Students				
	Create Tests			X
	Set Professor Permissions			X
	Incorporate User Authentication			X
	Interface with the database to Drop student from class			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement the Page Design			X
	Update the staging site with the latest version			X
Add Students				
	Create Tests			X
	Set administrator permissions			X
	Interface with the database to Add a student to the system.			X
	Design Page Layout			X

	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Incorporate user authentication			X
	Implement the Page Design			X
	Update the staging site with the latest version			X
Drop Courses				
	Create Tests			X
	Interface with the database to allow a student to drop a course			X
	Gather restrictions from Product Owner of cases where a student can't drop a course.			X
	Expand the Page Design			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement User Authentication for the drop course feature			X
	Update the staging site with the latest version			X
View Course History				
	Create Tests			X
	Interface with the database to retrieve a student's course history			X
	Expand the Page Design			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement User Authentication so that a student can view only his course history			X

	Update the staging site with the latest version			X
Add Descriptions for New Classes				
	Create Tests			X
	Implement User Authentication			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Update the staging site with the latest version			X
Add Prerequisites for New Classes				
	Create Tests			X
	Implement User Authentication			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Update the staging site with the latest version			X
View Roll Call				
	Create Tests			X
	Interface with the database to retrieve roll call for a class for a professor			X
	Gather restrictions from Product Owner of cases where a professor cannot access roll call for a class that is not his own			X
	Design Page design and interactivity			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X

	Implement User Authentication for viewing roll call			X
	Update the staging site with the latest version			X

### Sprint Burndown chart

The sprint 2 burndown chart showed our progress in a day to day basis. We started off strong the first day with some work completed because we knew that our team need to catch up from the last sprint. However, we could not catch up to where we wanted to be until day 6 because the product owner was away for 2 days early on in this sprint. Although it was hard to work without the Product Owner's input, our team managed to implement the user stories from the product backlog in time to realize our sprint's goal by day 10.



### Refined Product Backlog

During our Sprint Review meeting we presented to the Product Owner what was completed during Sprint 2, in this Sprint Review we also had a professor and administrator attend the meeting and based on the DONE criteria specified the Product Owner had deemed the following stories to be DONE:

Stories	Story Points
Drop Courses	8
View Course History	13

Drop Students	5
Add Class Schedules	13
Add Students	8
Add Descriptions for New Classes	5
Add Prerequisites for New Classes	5
View Roll Call	5
<b>Story Points Completed</b>	<b>62</b>

This shows that our sprint velocity for Sprint 2 was 62 story points. The team has reached their Sprint 2 goal of completing all committed product backlog items even though the Product Owner was absent two days early in the sprint. However, the new improvements to our process helped us reach our goal. However, the administrator after seeing the Add Class Schedules functionality, requested for an improved user experience by allowing an administrator to create multiple course sections on one screen. The Product Owner added this request as a story to the product backlog for consideration. As of now, this is the state of the Product Backlog.

<b>Name</b>	<b>Description</b>	<b>Story Points</b>
Student Notifications	As a student, I can get a notification when my grades have been posted and when my registration has opened up.	13
Request Student Transcripts	As an administrative staff, I can request student transcripts and history of courses taken with grades so that I can inspect the academics of a student.	20
Add Professors	As an administrative staff, i can add new professors to the system so that they can teach classes.	8
Add Class Lists to Majors	As an administrative staff, I can add a list of classes for a major so that a class can be related to a major.	5
Remove Authorizations	As an administrative staff, I can remove student and professor authorizations so that they can no longer access the system.	5
Assign Passwords	As an administrative staff, I can assign passwords to students and professors so that they can login into the system.	5



View Student Schedules	As a professor, I can view the schedule of one of the students enrolled in my class so that I can see what classes they are enrolled this semester.	8
Add Multiple Course Sections at a Time	As an administrator, I can add multiple sections of a course at a time so that I can increase my efficiency in creating course schedules.	N/A

### Refined Release Plan

Since we were able to complete all estimated committed product backlog items identified in the Release Plan, no changes were made to the release plan at the moment. Because the new story Add Multiple Course Sections at a Time requested by an administrator seems to be an improvement to what was implemented, it is not considered in the refined release plan since it is out of scope based on the objectives and vision of the system.

	Iteration1	Iteration 2	Iteration 3
	Search for Classes (20)	Drop Courses (8)	Student Notifications (13)
	Add Classes to Cart (8)	View Course History (13)	Request Student Transcripts (20)
	Checkout Cart (8)	Drop Students (5)	Add Professors (8)
	View Current Schedule (5)	Add Class Schedules (13)	Add Class Lists to Majors (5)
	Enter Grades (8)	Add Students (8)	Assign Passwords (5)
		Add Descriptions for New Classes (5)	View Student Schedules (8)
		Add Prerequisites for New Classes (5)	Remove Authorizations(5)
		View Roll Call (5)	
<b>Plan</b>	49	62	64
<b>Capacity</b>	65	65	65

### Refined Release Burndown chart

Our Sprint Velocity for Sprint 2, based on the stories DONE, is 62 story points. Our Project Velocity is 55.5 story points per sprint. The Release Burndown chart shows that in

comparison to Sprint 1, we had achieved our estimated story points completed per sprint. Even though we have been planning for max 65 story point sprints, our realistic outlook is finishing 55.5 story points per iteration, which puts us in a 4 Iteration project.



### Retrospective Meeting

During our Sprint Retrospective Meeting we had identified that even though we did meet our Sprint 2 goals that we could have had more solid testing since it was one of the biggest time consumers in Sprint 2. Due to this finding, we have implemented the proper improvements to our process, which are outlined below:

- Build incrementally and continuously integrate.
- During each integration, it is required to run all tests, if any fail, address the issue as soon as possible and integrate again.

### Sprint 3

Timebox: 4/18/2016 - 5/2/2016

#### Backlog Refinement

According to the release plan for Sprint 3, our focus is on Student Notifications, Request Student Transcripts, Add Professors, Add Class Lists to Majors, Assign Passwords, View Student Schedules, and Remove Authorizations user stories. We held a Backlog Refinement meeting to further discuss and refine these users stories prior to starting the sprint and came

up with a refined product backlog as below, which includes appropriate acceptance criteria for the user stories in question.

Name	Description	Story Points
Student Notifications	As a student, I can get a notification when my grades have been posted and when my registration has opened up. <b>Acceptance Criteria / DONE:</b> Email/Text Message notifications	13
Request Student Transcripts	As an administrative staff, I can request student transcripts and history of courses taken with grades so that I can inspect the academics of a student.	20
Add Professors	As an administrative staff, i can add new professors to the system so that they can teach classes.	8
Add Class Lists to Majors	As an administrative staff, I can add a list of classes for a major so that a class can be related to a major. <b>Acceptance Criteria / DONE:</b> Multiple Select classes to be associated with a major	5
Remove Authorizations	As an administrative staff, I can remove student and professor authorizations so that they can no longer access the system. <b>Acceptance Criteria / DONE:</b> Send email to user after authorization has been disabled.	5
Assign Passwords	As an administrative staff, I can assign passwords to students and professors so that they can login into the system. <b>Acceptance Criteria / DONE:</b> Send email to user after password has been assigned.	5
View Student Schedules	As a professor, I can view the schedule of one of the students enrolled in my class so that I can see what classes they are enrolled this semester. <b>Acceptance Criteria / DONE:</b> View Current Student Schedules for enrolled semester.	8
Add Multiple Course Sections add a Time	As an administrator, I can add multiple sections of a course at a time so that I can increase my efficiency in creating course schedules.	N/A

## Sprint Planning Meeting

For Sprint 3, our main goal was to focus on stories with deal with management of the system, some low priority functionality related to Professors and extra feature that enhances the user experience of students, notifications.

### Top 5 Risks

1. There's an upcoming holiday that will reduce the amount of work time available.
2. While there are some extra feature that will enhance the system. We must be careful not to gold-plate the system by adding unnecessary features.
3. One of the developers is leaving the company for another job, which would reduce the amount of developers working on the project.
4. System must be able to handle multiple requests for students grades near the end of term for graduation checks.
5. Most University uses a third party to request official transcript. It is necessary to clarify with the administration staff if the user story Request Transcript is for official or unofficial transcript.

## Sprint Backlog

Below is a snapshot of the Sprint Backlog for Sprint 3 before our Sprint Review Meeting.

Committed Backlog Items	Tasks	To Do	In Progress	Done
Student Notifications				
	Create Tests for the various Notifications			X
	Integrate User Authentication to view Course History			X
	Design Email / Text content for notifications			X
	Design Notification interaction			X
	Using Test-driven development write the code that triggers the notification.			X
	Using Test-driven development write the code that is triggered after a notification has been received.			X
	Implement Email / Text message design			X
	Update the staging site with the latest version			X
Request Student Transcripts				
	Create Tests			X
	Design Transcript page template			X
	Incorporate User Authentication			X

	Interface with the database to retrieve transcript information			X
	Design Page Layout and Interaction			X
	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement the Page Design			X
	Update the staging site with the latest version			X
Add Professors				
	Create Tests			X
	Set administrator permissions			X
	Interface with the database to Add a professor to the system.			X
	Design Page Layout			X
	Get Feedback on Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Incorporate user authentication			X
	Implement the Page Design			X
	Update the staging site with the latest version			X
Add Class Lists to Majors				
	Create Tests			X
	Interface with the database to associate courses to a major			X
	Design Page and interactivity			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement a many to many relationship between courses and a major			X
	Update the staging site with the latest version			X
Remove Authorizations				
	Create Tests			X
	Implement User Authentication			X
	Design interactivity			X

	Using Test-driven development write the code to interface with the database and the template			X
	Update the staging site with the latest version			X
View Student Schedules				
	Create Tests			X
	Interface with the database to retrieve a student's current registered courses			X
	Expand the Page Design			X
	Get Feedback on Page Design			X
	Implement the Page Design			X
	Using Test-driven development write the code to interface with the database and the template			X
	Implement User Authentication so that a student can view only his current schedule			X
	Update the staging site with the latest version			X

### Sprint Burndown Chart

The sprint backlog showed that although we started out slow, our team was able to finish up with the product development on time by day 10. From day 0-4 we did not have a lot of progress due to an issue with the implementation of student notifications. One of the developers had trouble with the notifications sending too many notices to the users and having it marked as spam. After conducting pair programming, our team was able to overcome this issue and our progress improved.



### Refined Product Backlog

During our Sprint Review meeting we presented to the Product Owner what was completed during Sprint 3, in this Sprint Review we also had an administrator attend the meeting and based on the DONE criteria specified the Product Owner had deemed the following stories to be DONE:

Stories	Story Points
Student Notifications	13
Request Student Transcripts	20
Add Professors	8
Add Class Lists to Majors	5
Remove Authorizations	5
Assign Passwords	5
View Student Schedules	8
<b>Story Points Completed</b>	<b>64</b>

This shows that our sprint velocity for Sprint 3 was 64 story points. The team has reached their Sprint 3 goal of completing all committed product backlog items. The continual integration and testing during integration helped reduce the number of bugs introduced to the system. At the Review Meeting the administrator after seeing the Request Student Transcripts, requested for a batch functionality when requesting student transcripts. The Product Owner added this request as a story to the product backlog for consideration. As of now, this is the state of the Product Backlog.

Name	Description	Story Points
Add Multiple Course Sections add a Time	As an administrator, I can add multiple sections of a course at a time so that I can increase my efficiency in creating course schedules.	N/A
Request a batch of student transcripts	As an administrator, I can request the transcripts of multiple students at once so that I don't have to request them individually.	N/A

### Refined Release Plan

Even though there are still two new items that have been added to the Product Backlog we have reached Sprint 3, and we have delivered all functionality identified in the Release Plan and we have reached out 3 iteration deadline. Due to this the release plan will not be modified at the moment. Both additional stories left in the product backlog can be discussed among the product owner, stakeholders and team to either continue development or deliver the product.

	Iteration1	Iteration 2	Iteration 3
	Search for Classes (20)	Drop Courses (8)	Student Notifications (13)
	Add Classes to Cart (8)	View Course History (13)	Request Student Transcripts (20)
	Checkout Cart (8)	Drop Students (5)	Add Professors (8)
	View Current Schedule (5)	Add Class Schedules (13)	Add Class Lists to Majors (5)
	Enter Grades (8)	Add Students (8)	Assign Passwords (5)
		Add Descriptions for New Classes (5)	View Student Schedules (8)
		Add Prerequisites for New Classes (5)	Remove Authorizations(5)
		View Roll Call (5)	
<b>Plan</b>	49	62	64



Capacity	65	65	65
----------	----	----	----

### Refined Release Burndown Chart

Our Sprint Velocity for Sprint 3, based on the stories DONE, is 64 story points. Our Project Velocity is 58.33 story points per sprint. The Release Burndown chart shows that our delivery based on our release plan improved in comparison to Sprint 1. Even though we have been planning for max 65 story point sprints, our realistic outlook is finishing 58.33 story points per iteration as the project continues.



### Retrospective Meeting

We were able to accomplished the most story points in sprint 3 as the team learned to collaborate with each other better. In addition, pair programming promote collective ownership of the code. As a result, the lost of one developer was not detrimental to our progress. It was also helpful that the developer who was leaving, stayed long enough to provide training for his replacement.

The difficulty that one developer had in implementing the student notification story caused a delay which fortunately was solved after pair programming was utilized. To prevent such issues from impeding the team we've decided to improve upon the following:

- Use Test Driven Development to create a test case first before coding
- Encourage team members to ask for help sooner

## Team Practices vs CMMI V.1.3's PP and PMC specific practices

### Team Practices (Scrum & XP):

In this project, we utilized both Scrum and XP for this project as both had useful and unique qualities that contributed to a better end-product for the project. We used Scrum to find and plan for the stakeholder's needs, to prioritize requirements, and to adapt to problem areas as they came up through iterations. Scrum was especially useful for its Sprint Planning, Sprint Review, and Daily Scrum meetings. Also, we found that XP was useful for developing high-value solutions to complex problems with time-boxed deliverables while keeping a focus on collaboration, simplicity, feedback and customer satisfaction. XP also added an emphasis on pair programming, acceptance testing and CRC cards to develop and design solutions to the SIS.

To best illustrate this, we will outline the steps in our team practice below.

- Pregame: planning phase
  - Identify the vision of the product
  - Understand the budget of the project
  - Draft a rough overall scope of the product
  - Draft the initial product backlog and estimate items using tools such as the Planning Poker activity
- Pregame : staging phase
  - Identify more requirements and user stories
  - Using the planning poker activity, reestimate additional requirements
  - Prioritize requirements
  - Update product backlog
  - Update the release plan
  - Conduct exploratory design or prototypes as needed
- Development phase
  - Backlog refinement meeting
  - Sprint planning meeting
  - Daily Scrum meeting
  - Development
  - Sprint review meeting
  - Sprint retrospective meeting
- Release phase
  - Ship deliverables
  - Retrospective project

Pregame: planning stage is similar to CMMI in regards to their purpose of planning out a project (in the form of a scope & vision document for our practice), and it's also similar in regards to the stakeholder's needs and budget to a degree. Things differ when it comes to a product backlog (as opposed to WBS) and usage of Planning Poker, etc.

Pregame: staging phase is the first attempt at prioritizing all of the requirements & user stories as the backlog is refined. Further refinements take place during the later development stages as well, though. It is also unique for the “exploratory” nature with designs and prototypes, which contrasts with CMMI since it’s not as reliant on figuring everything out at the onset.

The development phase is the most unique step out of the four, and it involves backlog refinement, sprint planning (which CMMI does not have), daily Scrum meetings, reviews and retrospectives. In this step, we iterated, acknowledged issues, dealt with issues in subsequent sprints and later reviewed our progress.

The release plan is also unique for our practice since CMMI is more focused on getting/keeping a plan, keeping the project on track and improving the process while ours is focused on shipping a mature product that adds value to the customer.

The biggest notable difference with our practice was a heavy emphasis on finding out the requirements in the early stages of the project, planning out the requirements, prioritizing those requirements, iterating, and finally refining/reviewing before releasing a finished product. This follows the overall goal of figuring out the stakeholder’s needs as early as possible so that we could focus our resources on the most important requirements so that we could release a mature product that adds value to the customer.

### **CMMI v1.3 Project Planning (PP):**

Capability maturity model integration (or CMMI for short) takes a different approach. The main goal of CMMI is to continuously improve upon a well established plan while maturing the processes in a uniform and controlled way. Both project planning (PP) and project monitoring and control (PMC) follows this philosophy in their respective implementations.

Project planning (PP) in particular is notable for its three goals. To establish estimates for the project, to develop a project plan, and to obtain commitment to the implementation of plan. Below is a listing of the steps and substeps:

- Establish estimates
  - Estimate scope of project
  - Establish estimates of work product/task attributes
  - Define project lifecycle phases
  - Estimate effort/cost
- Develop a project plan
  - Establish budget/schedule
  - Identify project risks
  - Plan data management
  - Plan project resources
  - Plan needed knowledge/skills
  - Plan stakeholder involvement

- Establish project plan
- Obtain commitment to the plan
  - Review plans that affect project
  - Reconcile work/resource levels
  - Obtain plan commitment

Establishing estimates is fairly similar to our practices with Scrum in regards to estimating the scope of the project, estimating the work needed for work products or attributes, defining the lifecycle (or steps to release) phases, and estimating effort. CMMI puts more emphasis on cost and budgeting, though.

Additionally, developing a plan is similar to our practice in regards to schedule, ID of project risks, plan out project resources, involving stakeholders, and establishing a project plan. CMMI is unique in regards to having a focus on budget, planning data management, focusing on knowledge and skills and waiting to involve stakeholders.

Finally, obtaining commitment to the plan is similar to our practice by reconciling work/resource levels (in the form of a check of story point status after each sprint), but it is unique in regards to reviewing the plans and obtaining commitment to the plan.

### **CMMI v1.3 Project Monitoring and Control (PMC):**

Project monitoring and control (PMC) has some similarities with the status checks that are involved in our practice, but it is a longer iteration cycle and it involves much of the same subsections as the previous (project planning) example. PMC has two main goals: monitor the project against plan and manage corrective action to closure. Below is a list of the steps and substeps:

- Monitor the project against plan
  - Monitor project planning parameters
  - Monitor commitments
  - Monitor project risks
  - Monitor data management
  - Monitor stakeholder involvement
  - Conduct progress reviews
  - Conduct milestone reviews
- Manage corrective action to closure
  - Analyze issues
  - Take corrective action
  - Manage corrective actions

Monitoring the project against the plan also has similarities with our practice in a lot of ways, but there are some differences as well. Monitor project planning parameters is similar to tracking progress with the story points and burn-down charts, monitor project risks is similar to top 5 risks,

conduct progress/milestone reviews is also similar to refinement process, CMMI differs in regards to monitoring commitments, data management and stakeholder involvement.

Finally, manage corrective action to closure is somewhat similar to retrospective meetings, but through a longer iteration process. Both CMMI and our practice deals with analyzing issues, taking corrective action and managing corrective action, but our practice does so after each sprint. CMMI implements changes that can be applied on a wide scale once it reaches a similar point, and it does so until the problem is solved. Those lessons can be added to internal documentation to improve the process.

### **Compare & Contrast**

Our Scrum/XP practice and CMMI have similar goals in regards to discovering the needs of the project, estimating the work that might go into certain tasks, planning out those tasks, and tracking the progress while trying to achieve set milestones. The method to reach that point differs in the details, though. For example, our practice involves involving the stakeholder in a more collaborative relationship as we iterate. We collectively figure out the needs, the priorities, the problem areas and the plan. We track progress through burndown charts and story point totals, and we review to see what we must do better for the next sprint. CMMI has an emphasis on following the plan to minimize risk. CMMI also breaks down the project into a work breakdown structure (WBS) that is then broken down further into logical units. This is somewhat similar to story points, but with the addition of external/internal work units as well. CMMI also has some other differences such as the emphasis on budgets, skills, data management and stakeholder involvement.

Generally speaking, our practice emphasized collaboration, finding/ranking/planning the priorities, iterating, reviewing, and finally releasing a finished product. We would plan for issues that we could see, and we would deal with the unforeseeable issues as they came up. CMMI is similarly focused on planning, tracking and improving, but with more of an emphasis on minimizing risk through the creation and adherence to a plan throughout. Also, improvements are implemented in the next sprint in our practice while it is integrated into a more heavily documented practice with CMMI.