Your Project

A Group Project For

SE391 / SE491

Software Engineering Studio

400 Points

Overview

As a group, you will work on a project of your own devising.

You will come up with a concept and try to attract developers to your project. You will prepare a project plan with the intent of having an initial roll-out of your product by the end of SE491. You will evolve your product to incrementally add value and to address operational concerns such as monitoring, notification, disaster recovery, etc.

The Tools

I am standardizing on some tools for this project, but will try to leave many decisions up to you. These are the tools that you must use:

- **Trello**. Trello will be used for all product management work including the product backlog and the sprint backlogs. It can also be used to organize your retrospectives and general TODOs. It's simple and free for our purposes.

- **Git and GitHub**. You will use GitHub for all of your code storage. Most IDEs allow you to integrate with Git, but you need to understand the commands. For this class all you really need to know are: basic file commands (adding, moving, removing), committing, branching, and merging. Git supports complex variations on these basic commands, but you shouldn't need them.

  In addition, GitHub provides many additions to Git, such as pull requests, markdown files, and GitHub Actions, which you'll use for your continuous integration pipeline.

  If you need a primer on Git or GitHub, you might want to start here: https://github.blog/developer-skills/programming-languages-and-frameworks/what-is-git-our-beginners-guide-to-version-control/

Here are the things that I am not standardizing. These must be determined by each group. However while I am not standardizing on particular tools or frameworks in this category, each group must choose at least:

- **Programming language**. You all need to do some hands-on development and I expect each person in the team to contribute. This language must support unit testing and automated builds.

- **Unit testing**. Any modern programming language will have some kind of unit testing and mock object framework available.

- **Automated build**. As with unit testing, any modern programming language will have some kind of build automation framework available.

- **IDE**. Make sure that you can use whatever IDE you choose.

- **Cloud provider**. This is up to you. Just make sure you keep it free.

Scrum

You will use Scrum to manage your project and its progress. You will use four (4) 2-week sprints for your project. These will be described in more detail below. At the end of each sprint, each group will present a sprint recap in-class to cover their progress and lessons learned.

Constraints

As with all projects, this one has a set of constraints that you need to follow:

1. Each student should plan to spend anywhere from 8-12 hours per week outside of class to work on the project. You will work with your team to make your commitment for each sprint.
2. You must use only Java technologies for the implementation. This means the Java JDK as well as any additional libraries that you think might be useful.
3. You may only use freely available tools and frameworks.
4. You may be given other technologies to use during your project. No substitutions are allowed. These additions will be free.

Activities and Ceremonies

You will use the basic activities and ceremonies of Scrum. You will need to adapt them to your particular group and the schedule of its members.

Product Backlog Management

You will need to keep your product backlog groomed throughout the term. That means that you should always have a prioritized, estimated supply of stories. This will help sprint planning go much faster.

The team will estimate in story points using the modified fibonacci sequence: 1, 2, 3, 5, 8, 13, 20, 40, 80, 100. Do not estimate in hours, story counts, or anything other than points.

The product backlog will be managed within Trello. How the boards are managed is up to each group.

Sprint and Release Planning

At the start of each sprint your team will carefully choose the stories that will add the greatest value for your minimum viable product (MVP). Plan on spending 1-2 hours per sprint for this activity.

Release planning is an ongoing process where you manage the expectations and contents of the first release that is scheduled for the end of the term. Release planning will begin as a guesstimate, but, using the techniques we'll discuss in class, the group will refine their release plan to be based on the average number of points delivered in each sprint.

Daily Scrum

This is the most challenging ceremony. Normally full-time teams meet daily to discuss their progress, but that may not be feasible for your team. You will need to arrange matters with your team members so that you can make solid progress on the project. This might be a combination of on-line or in-person meetings supplemented by useful technologies such as email, IM, and other collaborative work products.

Review

At the end of each sprint, you will conduct a project review for the class that showcases what you've accomplished during the sprint. You would ordinarily plan on 30-60 minutes for this ceremony for a 2-week sprint, but since we're on a condensed schedule, plan for about 15-20 minutes instead. This review will include a demonstration of your potentially shippable product increment, which means that you must be able to demonstrate your code outside of your text editor, just like an actual user would use it.

Retrospective

At the end of each sprint you will, as a team, conduct a sprint retrospective. You should plan on 30-60 minutes for this if you intend to take it seriously, which you should.

During each in-class sprint review you will present the results of your discussion using the “Stop, Start, Continue” framework: Stop – things you're doing that you shouldn't; Start – things you're not doing that you should; and Continue – things that are working well that you want to retain.

Schedule

This project will run all term. You will deliver your project in phases using the Scrum and eXtreme Programming frameworks. The timeline and associated list of activities and deliverables for each sprint is provided on the pages below.

Week	Date	Sprint	Milestones
01	2025-01-08	—--	❖ Discuss the course, its logistics, and the project
02	2025-01-15	—--	❖ Complete and present project pitches to the class ❖ Instructor to form groups
03	2025-01-22	1	❖ Start sprint 1 ❖ Start release planning
04	2025-01-29		❖ Complete release plans ❖ Prepare sprint 1 presentation.
05	2025-02-05	2	❖ Sprint 1 presentation ❖ Start and work on sprint 2
06	2025-02-12		❖ Sprint 2 work ❖ Prepare sprint 2 presentation ❖ Prepare for sprint 3
07	2025-02-19	3	❖ Sprint 2 presentation ❖ Start and work on sprint 3
08	2025-02-26		❖ Sprint 2 work ❖ Prepare sprint 3 presentation ❖ Prepare for sprint 4
09	2025-03-05	4	❖ Sprint 3 presentation ❖ Start and work on sprint 4
10	2025-03-12		❖ Sprint 4 work ❖ Prepare sprint 4 presentation
11	2025-03-19	—--	❖ Sprint 3 presentation

Sprint 1 (2025-01-22 – 2025-02-05): Bootstrapping

Goal: Planning and Stories

Release Planning: During this sprint, you have a lot of up-front work to do to get ready for the remainder of the project. You'll plan the remaining sprints around your product backlog.

1. Prepare your product backlog.
2. Prepare a release plan for the term that describes which stories you think you'll complete in each of the next three (3) sprints. Your goal is to have a usable, if incomplete, product by the end of the term, whereas the goal of each sprint is to have demonstrable progress toward that product.
3. Prepare your architecture including your GitHub repo and all major tools and frameworks your group intends to use.

Review: Your initial product backlog in Trello with initial epics and stories. The next sprint's stories should be estimated and have detailed acceptance criteria.

Retrospective: A retrospective where you discuss how you can improve as a team. You will use the "Stop, Start, Continue" format described in the section, Retrospective above.

Evaluation: This sprint will be evaluated on:

- **[20 pts.] Backlog Management.** Your product backlog in Trello featuring epics and stories. You have defined priorities of these stories in a way that leads you to an MVP at the end of the term and potentially shippable product increments at the end of each sprint.
- **[20 pts.] Release Plan.** Your initial release plan, showing the product backlog items that you expect to complete in each remaining sprint as well as the items you do not expect to complete.
- **[20 pts.] Technology.** Describe your technology stack, and show that you setup the GitHub repository including a markdown file called "the team" that provides a 2-3 sentence background of each group member. Each group member should commit their own background to this file.
- **[20 pts.] Review.** An in-class presentation of your project plan and the first sprint's stories, your technology stack and architecture, your product backlog Trello board, your Github repo along with the markdown file and its commit history, and your retrospective.
- **[20 pts.] Retrospective.** The documentation of your team's lessons learned for the sprint and how you intend to apply those lessons to later sprints. You should use the "Stop, Start, Continue" format.

Submit: A presentation that captures the major items described by the evaluation. Be sure to include links to external systems like Trello and GitHub. You only need to submit a single presentation for the team.

Sprint 2 (2025-02-05 – 2025-02-19)

Goal: Team’s Choice

Release Planning: Work on the stories identified for sprint 2, and maintain your product backlog to reflect new stories, removed stories, and changed stories.

1. Prepare an updated release plan for the term that describes which stories you think you'll complete in each of the next two (2) sprints. Your goal is to have a usable, if incomplete, product by the end of the term, whereas the goal of each sprint is to have demonstrable progress toward that product.

Review: Demonstrate your progress to date. Briefly go through the stories that you were able to complete and those you were not. Discuss your updated product backlog in Trello with epics and stories. The third sprint’s stories should be estimated and have detailed acceptance criteria.

Retrospective: A retrospective where you discuss how you can improve as a team. You will use the “Stop, Start, Continue” format described in the section, Retrospective above.

Evaluation: This sprint will be evaluated on:

- **[20 pts.] Backlog Management.** Your updated product backlog in Trello featuring epics and stories. You have defined priorities of these stories in a way that leads you to an MVP at the end of the term and potentially shippable product increments at the end of each sprint.
- **[20 pts.] Release Plan.** Your updated release plan, showing the product backlog items that you expect to complete in each remaining sprint as well as the items you do not expect to complete. Provide an updated release burndown chart.
- **[20 pts.] Technology.** Demonstrate that you have unit tests for your classes and code as well as an automated build script.
- **[20 pts.] Review.** An in-class presentation of your updated release plan and the second sprint’s stories, your technology stack and architecture, your updated product backlog Trello board emphasizing changes since Sprint 1, your GitHub repo with committed code, and your retrospective.
- **[20 pts.] Retrospective.** The documentation of your team's lessons learned for the sprint and how you intend to apply those lessons to later sprints. You should use the “Stop, Start, Continue” format.

Submit: A presentation that captures the major items described by the evaluation. Be sure to include links to external systems like Trello and Github. You only need to submit a single presentation for the team.

Sprint 3 (2025-02-19 – 2025-03-05)

Goal: Team’s Choice

Release Planning: Work on the stories identified for sprint 3, and maintain your product backlog to reflect new stories, removed stories, and changed stories.

1. Prepare an updated release plan for the term that describes which stories you think you'll complete in the next sprint. Your goal is to have a usable, if incomplete, product by the end of the term, whereas the goal of each sprint is to have demonstrable progress toward that product.

Review: Demonstrate your progress to date. Briefly go through the stories that you were able to complete and those you were not. Discuss your updated product backlog in Trello with epics and stories. The fourth sprint’s stories should be estimated and have detailed acceptance criteria.

Retrospective: A retrospective where you discuss how you can improve as a team. You will use the “Stop, Start, Continue” format described in the section, Retrospective above.

Evaluation: This sprint will be evaluated on:

- **[20 pts.] Backlog Management.** Your updated product backlog in Trello featuring epics and stories. You have defined priorities of these stories in a way that leads you to an MVP at the end of the term and potentially shippable product increments at the end of each sprint.

- **[20 pts.] Release Plan.** Your updated release plan, showing the product backlog items that you expect to complete in each remaining sprint as well as the items you do not expect to complete. Provide an updated release burndown chart.

- **[20 pts.] Technology.** Demonstrate that you have continuous integration defined using Github actions. Show a successful and failed build.

- **[20 pts.] Review.** An in-class presentation of your updated release plan and the third sprint’s stories, your technology stack and architecture, your updated product backlog Trello board emphasizing changes since Sprint 2, your GitHub repo with committed code, and your retrospective.

- **[20 pts.] Retrospective.** The documentation of your team's lessons learned for the sprint and how you intend to apply those lessons to later sprints. You should use the “Stop, Start, Continue” format.

Submit: A presentation that captures the major items described by the evaluation. Be sure to include links to external systems like Trello and Github. You only need to submit a single presentation for the team.

Sprint 4 (2025-03-05 – 2025-03-19)

Goal: Wrap-Up

Release Planning: Work on the stories identified for sprint 3, and maintain your product backlog to reflect new stories, removed stories, and changed stories. Prepare your finished release for demonstration to the class.

1. Prepare an updated release plan outlining how many additional sprints, if any, you think you'd need to complete the game and which stories you would place into each one.

Review: Demonstrate your progress to date. Briefly go through the stories that you were able to complete and those you were not. Discuss your updated product backlog in Trello with epics and stories.

Retrospective: A retrospective where you discuss how you can improve as a team. You will use the "Stop, Start, Continue" format described in the section, Retrospective above.

Evaluation: This sprint will be evaluated on:

- **[20 pts.] Backlog Management.** Your updated product backlog in Trello featuring epics and stories. You have prioritized these stories in a way that leads you to an MVP at the end of the term and potentially shippable product increments at the end of each sprint.
- **[20 pts.] Release Plan.** Your updated release plan with additional sprints and the stories that you would add to each one. Provide an updated release burndown chart. Use the techniques studied in class to estimate the remaining effort.
- **[20 pts.] Technology.** Demonstrate that your build fails based on insufficient code coverage. Show a successful and failed build.
- **[20 pts.] Review.** An in-class presentation of your updated release plan and the fourth sprint's stories, your updated product backlog emphasizing changes since Sprint 3, and your retrospective.
- **[20 pts.] Retrospective.** The documentation of your team's lessons learned for the term in a "Do, Don't Do" format. You do not need a traditional sprint review.

Submit: A presentation that captures the major items described by the evaluation. Be sure to include links to external systems like Trello and GitHub. You only need to submit a single presentation for the team.

Peer Evaluation

At the end of the term, you will be asked to submit a peer evaluation of each of your teammates. This evaluation is intended to reflect your peers' understanding and ability to apply the agile practices, technologies, and techniques that make up this project.

The peer review determines how many of the project contribution points you will receive for your work during the term. Not everyone contributes equally to a project and thus not everyone should earn the same rewards for its successful completion. Your total peer evaluation score will be based on the median (middle value) of the peer evaluations provided by your team, including your own.

If you do not submit a peer evaluation form, then you will receive a “no show'” (0%) for your self-evaluation, while each of your team members will receive “excellent'” (100%) for your evaluations of them. Late evaluations will not be accepted.

Please fill out the form on the following page for yourself and each of your team members and submit them as PDFs to D2L for the “Peer Evaluation” assignment once it becomes available. Guidance on the evaluation criteria and scale is provided on the next page to help you in your assessment. You may provide additional comments below the table if you wish, but only your scores in the final table will be used in the score.

Group Member Name:

DEPENDABILITY	Always	Usually	Sometimes	Rarely	Never
Comes to meetings prepared	5	4	3	2	1
Completes their tasks well	5	4	3	2	1
Completes their tasks on time	5	4	3	2	1
Makes timely responses to project communications	5	4	3	2	1

TEAM PLAYER	Always	Usually	Sometimes	Rarely	Never
Invested in the group doing well	5	4	3	2	1
Exhibits friendly professionalism	5	4	3	2	1
Is respectful of others	5	4	3	2	1
Did a fair share of the work	5	4	3	2	1

PROJECT PARTICIPANT	Always	Usually	Sometimes	Rarely	Never
Proposes, but doesn’t force, new ideas, suggestions, and courses of action	5	4	3	2	1
Builds on or extends others’ proposals	5	4	3	2	1
Expressed support for other group members’ opinions or ideas	5	4	3	2	1
Disagreement with group members’ opinions or ideas is respectful	5	4	3	2	1
Accepts or respectfully negotiates when other group members disagree with his/her ideas	5	4	3	2	1
Invites views or opinions from group members not actively participating in the discussion	5	4	3	2	1

Based on your responses to the above questions, assign an overall rating you think is the fairest assessment of the team member's contribution to the overall project. Use the following scale:

Excellent	Very Good	Satisfactory	Average	Marginal	Deficient	Unsatis-factory	Superficial	No Show
100%	95%	85%	75%	50%	25%	15%	5%	0%

Please use the following descriptions of each level in the overall rating as guidance.

Rating	Score	Description
Excellent	100%	Consistently carried more than his/her fair share of the workload. They rarely relied on others and were often an individual on whom others relied.
Very Good	95%	Consistently did what he/she was supposed to do, very well prepared and cooperative. Demonstrated a high degree of skill. Made a few minor mistakes. Occasionally relied on others, but consistently contributed to the success of their peers.
Satisfactory	85%	Usually did what he/she was supposed to do, acceptably prepared and cooperative. Demonstrated reasonable competence. They sometimes made mistakes, but recovered from them, learned from them, and did not make the same mistake again. Relied on others at the beginning, but by the end of the project were contributing as an equal.
Average	75%	Often did what he/she was supposed to do, minimally prepared and cooperative. Did not volunteer for additional activities or to assist their team members who might have been over-committed.
Marginal	50%	Sometimes failed to show up or complete assignments, rarely prepared. Did not demonstrate competence. Made frequent mistakes, sometimes learning from them, sometimes not. Relied heavily on others at the beginning but were able to gradually improve their own skills although they never contributed an equal share to the success of the project.
Deficient	25%	Often failed to show up or complete assignments, rarely prepared. They often made mistakes but did not learn from them. They relied heavily on others and did not try to improve their own skills.
Unsatisfactory	15%	Consistently failed to show up or complete assignments, unprepared. They were disruptive to the team. They made mistakes and failed to learn from them assuming they even tried at all. They relied heavily on others and made no attempt to improve their own skills.
Superficial	5%	Practically no participation. Shows up at the last minute looking to help in the hopes of improving their peer evaluation score.
No Show	0%	No participation at all.

Frequently Asked Questions

I've had some recurring questions about the group project for the class. I wanted to pull together some of these questions and my responses to them.

Q: I'm not a strong coder so I'm not sure how much help I'll be as a developer. Can I still succeed in this class?

A: Recognize that not everyone has the same skill set, or even the same goals for the class. If you're not a strong coder, you'll need to be realistic during sprint planning when you commit to the tasks you'll be able to complete. You can also look for other ways to contribute, for example as a tester or a user experience designer. Perhaps you'd prefer to play the role of the project manager or Scrum Master. There are many ways to contribute, but each team member must play some kind of technical role, so try to zero in on what your strengths are and look for ways to stretch yourself.

Q: Why should my grade depend on other people who might not have my work ethic?

A: This one comes up during every group project and it reflects an arrogance that has no place in either the classroom or the workplace. Why <u>shouldn't</u> your grade depend on other people? Contrary to the narrative many of us have grown up with, your success in most areas of endeavor is not due solely to your own efforts. Each of us constantly receives help from others. A wise person recognizes that and does not take more credit than is their due. An even wiser person will help others to be their best. This argument, however, is one of the reasons that you complete a peer evaluation at the end of the term. That's your best means of holding your group mates accountable to their commitments.

Q: Why is so much of my grade dependent on my peers' evaluations of me?

A: Because how you conduct yourself on any project will have a significant impact on your team's ability to deliver quality software. It will also have a significant impact on your career. In the professional world you're going to receive feedback at least once a year (to which salary adjustments and bonuses are often tied) and in some organizations it's much more frequent than that. If you're used to such feedback, then this won't cause you any grief. If you're not used to such feedback, then this is a good chance for you to experience it. So, if you're a conscientious, well-meaning participant in your team's effort to complete the project, you'll likely do well. If you're an overbearing, colossal, PITA with whom no one wants to work, you're going to have a rough time, not just in class, but in life.