

## 2 Project Plan

### 2.1 Project Management/Tracking Procedures

Which of agile, waterfall or waterfall+agile project management style are you adopting. Justify it with respect to the project goals.

**We plan on using the agile methodology. One of our main goals is to continuously roll out homework assignments with randomized parameters and overall migrate these assignments from static/on paper to fully be on PrairieLearn. Using sprints can help us stay on track with creating homeworks and focusing on the questions that will have randomized parameters. So each sprint will be dedicated to a homework assignment and testing. Towards the end, the focus will be on hosting our own PrairieLearn server on ISU dedicated servers.**

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

- **GitLab boards**
- **Git**
- **Weekly meetings with our client**
- **WSR (weekly service review) documents to track progress up to that point and get everyone same page**

### 2.2 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project.

**Sprint 1: Get used to VM's and using PrairieLearn**

**Sprint 2: Get GitLab connected to prarielearn and make sure everybody was connected**

**Sprint 3: Begin creating homework 1 with help from TA and Professor**

**Sprint 4: Homework 1**

**Sprint 5: Homework 2**

**Sprint 6: Homework 3**

**Sprint 7: Homework 4**

**Sprint 8: Homework 5**

**Sprint 9: Homework 6**

**Sprint 10: Homework 7**

**Sprint 11: Homework 8**

**Sprint 12: Create users locally / Create user privilege tiers**

**Sprint 13: Get PL Production on ISU servers / Connect PL to Canvas**

**Sprint 14: Testing**

## 2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

What are some key milestones in your proposed project? It may be helpful to develop these milestones for each task and subtask from 2.2. How do you measure progress on a given task? These metrics, preferably quantifiable, should be developed for each task. The milestones should be stated in terms of these metrics: Machine learning algorithm XYZ will classify with 80% accuracy; the pattern recognition logic on FPGA will recognize a pattern every 1 ms (at 1K patterns/sec throughput). ML accuracy target might go up to 90% from 80%.

In an agile development process, these milestones can be refined with successive iterations/sprints (perhaps a subset of your requirements applicable to those sprint).

### Milestones:

#### 1. Get Prairie Learn running locally on each group members computer

##### 1.1: Milestone Subtasks for each group member

1.1.1: Install docker

1.1.2: Pull PL in a docker container

1.1.3: run PL locally

1.1.4: connect PL locally

1.1.5: Pull gitlab course project in container

1.1.6: Mount gitlab course project in container

1.1.7: Run PL with course project

1.1.8: Create a temporary HW

1.1.9: create temporary questions

##### 1.2: Progress Evaluation Criteria

1.2.1: percentage of group members with temporary questions

##### 1.3: Completion Criteria

1.3.1: All group members have implement a running local PL with temporary HW that has temporary questions

#### 2. Home Work 1

##### 2.1: Milestone Subtasks

2.1.1: Create hw on PL course page

2.1.2: Professor approvaes each Hw question prototype based on the given course assignment question

2.1.3: Implement question with verification

##### 2.2: Progress Evaluation Criteria

2.2.1: percentage of questions implemented from course assignment

### **2.3: Completion Criteria**

**2.3.1: Demo each Hw question to professor**

**2.3.2: Professor accepts each question**

**2.3.3: Professor has no additional requests**

## **3. Home Work 1**

### **3.1: Milestone Subtasks**

**3.1.1: Create hw on PL course page**

**3.1.2: Professor approves each Hw question prototype based on the given course assignment question**

**3.1.3: Implement question with verification**

### **3.2: Progress Evaluation Criteria**

**3.2.1: percentage of questions implemented from course assignment**

### **3.3: Completion Criteria**

**3.3.1: Demo each Hw question to professor**

**3.3.2: Professor accepts each question**

**3.3.3: Professor has no additional requests**

## **4. Home Work 2**

### **4.1: Milestone Subtasks**

**4.1.1: Create hw on PL course page**

**4.1.2: Professor approves each Hw question prototype based on the given course assignment question**

**4.1.3: Implement question with verification**

### **4.2: Progress Evaluation Criteria**

**4.2.1: percentage of questions implemented from course assignment**

### **4.3: Completion Criteria**

**4.3.1: Demo each Hw question to professor**

**4.3.2: Professor accepts each question**

**4.3.3: Professor has no additional requests**

**5. Home Work 3**

**5.1: Milestone Subtasks**

**5.1.1: Create hw on PL course page**

**5.1.2: Professor approves each Hw question prototype based on the given course assignment question**

**5.1.3: Implement question with verification**

**5.2: Progress Evaluation Criteria**

**5.2.1: percentage of questions implemented from course assignment**

**5.3: Completion Criteria**

**5.3.1: Demo each Hw question to professor**

**5.3.2: Professor accepts each question**

**5.3.3: Professor has no additional requests**

**6. Home Work 4**

**6.1: Milestone Subtasks**

**6.1.1: Create hw on PL course page**

**6.1.2: Professor approves each Hw question prototype based on the given course assignment question**

**6.1.3: Implement question with verification**

**6.2: Progress Evaluation Criteria**

**6.2.1: percentage of questions implemented from course assignment**

**6.3: Completion Criteria**

**6.3.1: Demo each Hw question to professor**

**6.3.2: Professor accepts each question**

**6.3.3: Professor has no additional requests**

**7. Home Work 6**

**7.1: Milestone Subtasks**

**7.1.1: Create hw on PL course page**

**7.1.2: Professor approves each Hw question prototype based on the given course assignment question**

**7.1.3: Implement question with verification**

## **7.2: Progress Evaluation Criteria**

**7.2.1: percentage of questions implemented from course assignment**

## **7.3: Completion Criteria**

**7.3.1: Demo each Hw question to professor**

**7.3.2: Professor accepts each question**

**7.3.3: Professor has no additional requests**

## **8. Home Work 7**

### **8.1: Milestone Subtasks**

**8.1.1: Create hw on PL course page**

**8.1.2: Professor approves each Hw question prototype based on the given course assignment question**

**8.1.3: Implement question with verification**

### **8.2: Progress Evaluation Criteria**

**8.2.1: percentage of questions implemented from course assignment**

### **8.3: Completion Criteria**

**8.3.1: Demo each Hw question to professor**

**8.3.2: Professor accepts each question**

**8.3.3: Professor has no additional requests**

## **9. Home Work 8**

### **9.1: Milestone Subtasks**

**9.1.1: Create hw on PL course page**

**9.1.2: Professor approves each Hw question prototype based on the given course assignment question**

**9.1.3: Implement question with verification**

### **9.2: Progress Evaluation Criteria**

**9.2.1: percentage of questions implemented from course assignment**

### **9.3: Completion Criteria**

**9.3.1: Demo each Hw question to professor**

**9.3.2: Professor accepts each question**

**9.3.3: Professor has no additional requests**

**10. Create user accounts locally**

**10.1: Milestone Subtasks**

**10.1.1: Run PL on production locally**

**10.1.2: Create new user account**

**10.1.3: Login into User account**

**10.2: Progress Evaluation Criteria**

**10.2.1: percentage of steps completed towards running PL in production locally**

**10.3: Completion Criteria**

**10.3.1: Demo to professor**

**11. Have user privilege tiers**

**11.1: Milestone Subtasks**

**11.1.1: Set different permissions for created users**

**11.1.2: Set permission requirements for course editing**

**11.1.3: Test user permissions**

**11.2: Progress Evaluation Criteria**

**11.2.1: percentage permission implemented and tested**

**11.3: Completion Criteria**

**11.3.1: Demo to professor**

**12. Have PL output users grades on assignments**

**12.1: Milestone Subtasks**

**12.1.1: Output grade for each student user based on selected HW**

**12.2: Progress Evaluation Criteria**

**12.2.1: percentage of questions properly graded**

**12.3: Completion Criteria**

**12.3.1: Demo to professor**

**13. Professor can modify HWs**

**13.1: Milestone Subtasks**

**13.1.1: Admin users can edit each: hw, questions, due dates, grading, access and create new question and hws**

**13.2: Progress Evaluation Criteria**

**13.2.1: percentage of edittable option implemented**

**13.3: Completion Criteria**

**13.3.1: Demo to professor**

**14. Have production PL on ISU servers**

**14.1: Milestone Subtasks**

**14.1.1: Set up ISU server**

**14.1.2: Run PL on ISU in Production mode**

**14.2: Progress Evaluation Criteria**

**14.2.1: percentage of previous milestone functional**

**14.2.2: percentage of steps completed for running Production PL on ISU server**

**14.3: Completion Criteria**

**14.3.1: Demo to professor**

**15. Connect Canvas with PL on ISU servers**

**15.1: Milestone Subtasks**

**15.1.1: Get access to Canvas API**

**15.1.2: User accounts are connected to canvas**

**15.1.3: User grades are updated via canvas**

**15.2: Progress Evaluation Criteria**

**15.2.1: percentage of sub task completed**

**15.3: Completion Criteria**

**15.3.1: Demo to professor**

## 2.4 Project Timeline/Schedule

- A realistic, well-planned schedule is an essential component of every well-planned project
- Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity
- A detailed schedule is needed as a part of the plan:
  - Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar (including both 491 and 492 semesters). The Gantt chart shall be referenced and summarized in the text.
  - Annotate the Gantt chart with when each project deliverable will be delivered
- Project schedule/Gantt chart can be adapted to Agile or Waterfall development model. For agile, a sprint schedule with specific technical milestones/requirements/targets will work.

Chart link:

<https://docs.google.com/spreadsheets/d/1lyzIXPBDKoeGpdNPpuI7LrRjG8i2VHRceaZuIXnzNwI/edit?usp=sharing>

## 2.5 Risks And Risk Management/Mitigation

Consider for each task what risks exist (certain performance target may not be met; certain tool may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

Agile projects can associate risks and risk mitigation with each sprint.

Probabilities are as of when we figure out user roles and permissions (we don't have different permissions for student or admin accounts currently)

**Students may be able to cheat on the questions, Probability: <5%**

**Student information (grades) may be accessible to other students, Probability: <5%**

**Students may have admin rights, Probability: <5%**

**(as long as the server and PrairieLearn service is secure these won't happen)**

**Not many risks since we are running this on an already created software**

## 2.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.



	hours
Creating homeworks	8(?) homeworks * ~5 questions * 2 hrs = 80 hrs
Connecting to Canvas	10(?) hrs (getting permission + actually connecting)
Account creation/server hosting	5(?) hrs (likely easy just need to figure out how)

## 2.7 Other Resource Requirements

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

- **PrairieLearn Docs**
- **PrairieLearn Slack**
- **ISU dedicated server**
- **Canvas API**
- **TA/professor input**