Fall 2017 - CS 458 - Project Handout

This handout describes the semester project, including its milestones.

Important project goals

Important project goals include:

- maintaining an *evolving* collection of user stories, within a Product Backlog List (PBL), INCLUDING appropriate consultation with a "client"
    - it is expected that you have some user stories you have NOT reached as of the final project milestone -- the negotiations with regard to what should be added to the Sprint Backlog List (SBL) for each next sprint are an important aspect of Scrum/Extreme Programming/agile development that you are expected to be practicing as part of this project.
- working TOGETHER, as a TEAM, to implement **at least three working, deliverable iterations** of a system for your "client" that EACH implement a useful, operational subset of that "client"'s current version of those user stories
- a strong, concerted attempt at a **disciplined** approach to testing at VARIOUS levels
- using Scrum and Extreme Programming practices as discussed in class
- using and practicing software engineering practices and considerations

CS 458 User Story requirements

Remember that each user story should be:

- valuable to the customer
- small enough that several can be implemented in a sprint

AND, for CS 458 project purposes, each is expected to include:

- a client-understandable description of a desired capability in the form:
    - As a [Type-of-User],

      I want to [Do This]

      so that I can [Achieve That]
- an estimate of how long this will take to implement
- a list of acceptance criteria

Team Coding Standards

- Within your team's GitHub project repository, include a directory `coding-stds`
- With this directory, include one or more files stating/listing your team's agreed-upon coding standards.
- It is the team's choice to decide whether to have one coding standards file per language used in the project, or one file containing these coding standards for all languages used in the project.

Required Comments for Project Code

These guidelines should be followed in the project code submitted for each project sprint.

- Note that, AS MUCH AS POSSIBLE, you are expected to work in pairs for the programming in this Scrum/Extreme Programming/Agile team project; your project grade will be affected adversely if this is not the case.
  - Remember that pair programming is two people working at the same time on the same code (ideally on the same computer and keyboard, but if this is essentially virtually the case, that will be acceptable unless there are problems), with one typing, one saying what to type, and discussion taking place along the way, with roles and pairs changing frequently.
- Remember: careful refactoring is encouraged in Scrum/Extreme Programming/Agile programming!
- Each module of project code -- for example, each function, each procedure, each class, each method within a class -- is expected to have an opening comment block that follows the team's coding standards, and **in addition** also includes at least the following information:
  - what it expects/needs to work (parameters, other necessary inputs, and/or preconditions)
  - what it does (what it returns, what are its side-effects, what are its results, and/or postconditions)
  - the names of the **pair** or person who created it, and the date of the initial creation
  - for each modification: **also** include the names of the **pair** or person who modified it, and the date of the modification
- Not including the names of who created/worked on each module may adversely affect the project grade.
- And, it is a **serious breach of ethics** to include your name on code you did not work on!

Grading Breakdowns for the Milestones

Your project grade will be the sum of the following:

Max Pts Possible	Project part	Description
5	Milestone 1	Project "bids"
5	Milestone 2	Sprint 1 - Sprint Planning Meeting
10	Milestone 3	Completed Sprint 1
5	Milestone 4	Sprint 2 - Sprint Planning Meeting
15	Milestone 5	Completed Sprint 2
5	Milestone 6	Sprint 3 - Sprint Planning Meeting
25	Milestone 7	Completed Sprint 3
10	Milestone 8	Formal presentation/demonstration to class and to other observers
5	Milestone 9	Individual reflection paper
5	Participation 1	Based on 1st set of peer evaluations & instructor's evaluation of performance
5	Participation 2	Based on 2nd set of peer evaluations & instructor's evaluation of performance

Max Pts Possible	Project part	Description
5	Participation 3	Based on 3rd set of peer evaluations & instructor's evaluation of performance

(What if your team has additional sprints? Based on the team's situation, the grade for an extra-sprint-BEFORE-the-final-sprint could replace the grade for an earlier not-final-sprint with a lower grade, or the 65 max total points for sprints could be allocated differently amongst all of the sprints completed, with the caveat that the final sprint will be worth more points; see the instructor if your team is in this situation.)

The sum of the grades for the above project components will then be multiplied by 30% in computing the project portion of your CS 458 final semester grade.

The grades for Milestone 9 and Participation 1, 2, and 3 are individual grades, and may differ for each team member. Except for unusual circumstances, the grades for the other milestones will be the same for each member of a team (although I reserve the option of giving different members of a team different grades for *any* project part, if I feel this is appropriate).

Part of your Participation grade will be determined by your thoughtfully and thoroughly filling out several peer evaluations forms at times to be specified. I will consider your and your peers' comments on these forms, meeting minutes and other submitted files/documents/activities, and my own observations in determining each team member's Participation grade.

A Few Important Points re: Project Teams

last modified: 2017-09-21 (also posted on the public course web site, under "References")

- At ANY time during the semester, if any student or team has serious issues with a team member, please bring them to my attention as soon as possible.
- (slightly adapted from Cashman and Eschenbach's ENGR 111 Team Contract Guidelines): There will be **NO illegal activity** during any team meetings or team working sessions.
  - Illegal activity includes, but is **not** limited to, underage drinking and illicit drug use.
  - This rule **must** be enforced **regardless** of whether the meeting takes place on or off campus.
  - The team must notify me **immediately** (or by the next class meeting) if a violation of this rule occurs and the offending team member will be removed from the team and assigned a **0** for the entire course project.
- NOTE that it is each team member's responsibility to, if at all possible, contact all of the other team members if they are going to be late for or are going to miss a team meeting (whether that meeting is during scheduled CS 458 class time or outside of it).
  - The team should discuss and agree to the means for such notifications (e-mail? text? etc.)
- Here is good advice for teams, from http://www.ece.rutgers.edu/~marsic/Teaching/SE/projects.html:
  - "Saying that “nobody asked me to do this or that,” or, “I did everything that I was asked to do” is an **unacceptable** excuse.
  - Each team member should be **proactive** and not wait passively to be assigned responsibilities.
  - Do not ask others what should be done; rather, take initiative and suggest what should be done to make your project successful.

  - Take every opportunity to redistribute and/or rotate the responsibilities, make your personal suggestions be heard!
  - Many times defining the problem and determining what needs to be done is more difficult than actually doing it. Hence, problem defining and task assignment must be contributed to by all team members, rather than by the team leader alone."
- It will be each team member's responsibility to keep track of their contributions to the project. You will be expected to submit a list of these at some point (or points).
- Note that peer evaluations will be required at several points during the semester. Your thoughtful participation in these will also be part of your project grade.

Required Team Meeting Reports

- Note that, when class time is provided for team meetings, teams are expected to meet during **all** of the provided class time (members aren't to leave before class time is over, work on other course work, etc)
- **Except** for meetings that take place during class time, team meetings can be with members physically in the same place, or virtual (with the help of tools such as Google Hangouts, for example)
- A team's grade may be affected if teams do not meet regularly or if team meeting reports are not completed as specified below.
- For **each** team meeting, the team is expected to include the appropriate "official" team meeting report based on the type of meeting, meeting the specifications given below.
- For **each** sprint, within your project's GitHub project directory, create a directory `sprint-X` (where `X` is the number of this sprint).

  WITHIN each directory `sprint-X`, create a sub-directory `team-meetings`

  Place all team meeting reports for a sprint within the `team-meetings` sub-directory as noted below.

Sprint Planning Meeting

- This starts each sprint, and so you should have one Sprint Planning Meeting report for each project sprint
- In your sprint's `team-meetings` sub-directory, create a sub-directory `sprint-plan`, and put **all** of the pieces described below in this subdirectory
- Include, in an obviously-named file or directory, a **snapshot of your Product Backlog List** (PBL) as it ended up after its review during this sprint's Sprint Planning Meeting;
  - make sure it somehow includes an **ordered**, **complete** list of your user stories at this point
- Include, in an obviously-named file or directory, your **Sprint Backlog List** (SBL) as of the end of this sprint's Sprint Planning Meeting;
  - make sure it somehow includes an **ordered** list of those user stories you are planning to implement during this sprint
- Include a report with file name `sprint-plan.pdf` or `sprint-plan.txt`, which contains at least the following:
  - the Sprint Planning Meeting's **date**, **start time**, and **end time**

  - a list of the names of the **team members in attendance**
  - the **length** and **deadline** determined for this sprint
  - for *each* team member, **how many hours** that team member said they would be available during this sprint
  - the **total capacity**, then, for user stories during this sprint (in hours)
  - the **type, date, and time** for the planned **next** meeting
  - the statement that "The information in this `sprint-plan` directory is accurate, to the best of my knowledge", followed by each attending team member's signed or typed name

    [It will be considered a **serious breach of ethics** if this team meeting report is "signed" by a member who did not attend or who does not agree that the information in this directory is accurate, to the best of their knowledge.]

- Include a copy of something from the client (such as an e-mail, for example) saying that they have been consulted and are OK with this SBL's set of user stories for this sprint

Daily Scrum meetings

- These are not necessarily daily in this academic setting, but the more frequent they are during each sprint, the better. They should certainly occur multiple times per week during a sprint.
  - recall that these should be no longer than 15 minutes (hard limit) each
- So, you should have multiple Daily Scrum meeting reports per sprint.
- In your sprint's `team-meetings` sub-directory, create a sub-directory `daily-scrums`, and put all of the Daily Scrum meeting's reports in this subdirectory, each with the name `daily-scrum-YYYY-MM-DD.pdf` or `daily-scrum-YYYY-MM-DD.txt`
- Each Daily Scrum meeting's report should contain at least the following:
  - the Daily Scrum meeting's **date**, **start time**, and **end time**
  - a list of the names of the **team members in attendance**
  - for *each* attending team member, their THREE ITEM report:
    - what they DID since the last Daily Scrum,
    - what they are HOPING TO DO by the next Daily Scrum,
    - any BLOCKS in their way
  - the **type, date, and time** for the planned **next** meeting
  - the statement that "The information in this report is accurate, to the best of my knowledge", followed by each attending team member's signed or typed name

    [It will be considered a **serious breach of ethics** if this team meeting report is "signed" by a member who did not attend or who does not agree that the information in these is accurate, to the best of their knowledge.]

Sprint Review

- This occurs at the end of each sprint, when teams demonstrate what they have built during the sprint and get feedback from the client
- Include a report with file name `sprint-review.pdf` or `sprint-review.txt`, which contains at least the following:
  - the Sprint Review's **date**, **start time**, and **end time**
  - a list of the names of the **team members in attendance**
  - whether the client gave a **thumbs up** or **thumbs down** for this sprint's resulting iteration
  - also include any suggestions or critiques from the client
  - the statement that "The information in this report is accurate, to the best of my knowledge", followed by each attending team member's signed or typed name

    [It will be considered a **serious breach of ethics** if this team meeting report is "signed" by a member who did not attend or who does not agree that the information in these is accurate, to the best of their knowledge.]

Sprint Retrospective

- This occurs after the Sprint Review at the end of each sprint
- In your sprint's `team-meetings` sub-directory, create a sub-directory `sprint-retro`, and put **all** of the pieces described below in this subdirectory
- Include, in an obviously-named file or directory, your **Sprint Backlog List** (SBL) as of the end of this sprint's Sprint Retrospective
  - make sure it somehow indicates which user stories from the SBL were actually implemented during this sprint, and which were not implemented
- Include, in an obviously-named file or directory, some depiction of the Sprint Retrospective's list of what was good/pros from this sprint
- Include, in an obviously-named file or directory, some depiction of the Sprint Retrospective's list of what could have been better from this sprint
- Include, in an obviously-named file or directory, some depiction of the Sprint Retrospective's list of suggested actions to correct what didn't go so well
- Include a report with file name `sprint-retro.pdf` or `sprint-retro.txt`, which contains at least the following:
  - the Sprint Retrospective's **date**, **start time**, and **end time**
  - a list of the names of the **team members in attendance**
  - any other discussion, brainstorming, etc. that you wish to include in addition to the separate depictions of what was good, what could have been better, and what actions to correct what didn't go so well
  - the statement that "The information in this `sprint-retro` directory is accurate, to the best of my knowledge", followed by each attending team member's signed or typed name

    [It will be considered a **serious breach of ethics** if this team meeting report is "signed" by a member who did not attend or who does not agree that the information in this directory is accurate, to the best of their knowledge.]

Miscellaneous meetings

- The team may meet for other purposes during a sprint, perhaps to discuss issues that have arisen in more depth, perhaps to simply work together.
- A team might not have any miscellaneous meetings during a sprint; a team might have several such meetings.
- IF you have any miscellaneous meetings during a sprint:

  In your sprint's `team-meetings` sub-directory, create a sub-directory `misc-meetings`, and put all of the miscellaneous team meeting reports in this subdirectory, each with the name `misc-meet-YYYY-MM-DD.pdf` or `misc-meet-YYYY-MM-DD.txt`
- Each miscellaneous team meeting report should contain at least the following:
  - the meeting's **date**, **start time**, and **end time**
  - a list of the names of the **team members in attendance**
  - a **brief discussion/summary/list of topics** discussed or worked on
  - the **type, date, and time** for the planned **next** meeting
  - the statement that "The information in this report is accurate, to the best of my knowledge", followed by each attending team member's signed or typed name

    [It will be considered a **serious breach of ethics** if a team meeting report is "signed" by a member who did not attend or who does not agree that the information in that report is accurate, to the best of their knowledge.]

Milestone 1 - project "bids"

Deadline

Due by **11:59 pm** on **Friday, September 22**

(but earlier submissions *may* have the opportunity, *maybe*, to be approved on a "rolling" basis, so that teams that submit sooner *might* get approval sooner)

How to turn in

Make sure the pieces described below are in your team's private GitHub repository by the deadline; if you have them there sooner, please **E-MAIL** me, and **MAYBE** I can approve your proposed project more quickly!

What to turn in

- In your team repo, create a top-level directory named `proposal`
- In this directory, create a file named `proposal-bid.pdf` that contains:
  - your team name
  - your team members' names
  - this file's last modified date
  - **ONE or MORE** of the following, **IN ORDER of TEAM PREFERENCE**:

if you want to be considered for one of the "real" clients with a project whose description is given to the class

- (as I write this, there MAY be more than one of these, although only one is almost-definite)
- write why YOU should be selected to be the team to implement that project; make your case as to why your team would be the best choice.
- if there is more than one such client you would like to be considered for, give such a "bid" for each project

  (that is, I'll assume that the first description-why is for your first choice project, and if you give a second I'll assume it is for your second choice project, and so on)

if you have or find a client with a project you would like to do

- describe the project (1-2 paragraphs)
- give the name and e-mail of at least one person who is going to be your client representative; I am assuming you are making clear to the client(s) the restrictions as discussed in class
- (I may need to ask you additional questions to be able to decide whether your "bid" should be approved; you might be able to reassure me in advance by including a brief description of a *possible* small-enough-but-useful 1st iteration and a brief description of some of the many *possible* features that

might be able to be added in subsequent sprints)

if you have a project idea you are interested in (this can be in addition to the previous, in case your first choice "bid" is not approved)

- describe the project (1-2 paragraphs)
- do you already have a volunteer "client" or "clients" already? ...give their name(s) and e-mail(s)

  do you NOT, but want to be considered by one of our available client "volunteers"? ...state WHY one of them should want to be your project's "client" (and if not selected, the instructor will be figuring out which team will be which other team's "client"...)

Milestone 2 - Sprint 1 - Sprint Planning Meeting

deadline

This is due by 11:59 pm on Friday, October 6 **BUT SOONER WOULD BE BETTER!!!**

- **NOTE: keep in mind that you need to complete at least THREE sprints -- each resulting in a working system that does something valuable to the client -- by 11:59 pm on Friday of Week 15!**

how to "submit"

Make sure the milestone's pieces are in your team's private GitHub repository by the deadline.

what to submit

You are expected to have a directory in your GitHub private repository named `sprint-1`.

INCLUDE: a file `testing-plan-1.txt` or `testing-plan-1.pdf` within directory `sprint-1`, that describes how you plan to approach and attempt testing for Sprint 1.

- hint: this better not be empty!
- hint: think Extreme-Programming-attitude!
  - You'd better include HOW you will handle UNIT testing,
  - ...AND you'd better include HOW you will handle ACCEPTANCE testing

INCLUDE: you are expected to have, in `sprint-1/team-meetings/sprint-plan`, those pieces required from a Sprint Planning Meeting, as described in the section "**A Few Important Points re: Project Teams**" on pp. 4-5.

What needs to be in your team's GitHub repository for Milestone 2?

- your directories and files from Milestone 1
- your directory `coding-stds`, with contents as described in the section "Team Coding Standards" on p. 1
- in directory `sprint-1`:
  - a description of your testing plan for sprint 1 in `testing-plan-1.txt/.pdf`
  - sub-directory `team-meetings`, containing sub-directory `sprint-plan`, containing:
    - the snapshot of your resulting PBL
    - your resulting SBL
    - your file `sprint-plan.pdf`/`.txt`
    - confirmation that your client has seen and agrees with the SBL's set of user stories for this sprint

Milestone 3 - Completed Sprint 1

deadline

This is due by 11:59 pm on the date you gave in Milestone 2's `sprint-plan.pdf/.txt` file.

how to "submit"

Make sure the milestone's pieces are in your team's private GitHub repository by the deadline.

what to submit

INCLUDE: You need to include a copy of the code for your RUNNING, WORKING deliverable within your team's private GitHub repository.

Note that it needs to meet the following requirements:

- it must run/work!
- the "client" can run it!
- it does something (even if "small") the "client" considers to be useful!
- it (hopefully!) implements the user stories you said it would implement in the SBL included in `sprint-1/team-meetings/sprint-plan`, and this is backed up by acceptance testing
- its "modules" include unit testing (which I am assuming you wrote before writing the code)
- every "chunk" of code has opening comment blocks, including the names of the pairs who created/worked on each, as described in the section "Required Comments for Project Code" on p. 2

INCLUDE: in your `sprint-1` subdirectory, a file `README.txt` that contains:

- a CLEAR DESCRIPTION of what I need to do to run your deliverable, or see it running
- a list of the directories/files in your repository containing the code for this sprint, including a short description of each file listed

INCLUDE: in your `sprint-1` subdirectory, a file `testing-1.txt/.pdf` that describes how you ACTUALLY accomplished the unit and acceptance testing for this sprint

INCLUDE: in `sprint-1/team-meetings`, the reports for:

- Sprint 1's Daily Scrum meetings,
- any miscellaneous meetings,
- its Sprint Review, and
- its Sprint Retrospective,

...as described in the section "**A Few Important Points re: Project Teams**" on pp. 4-7

What needs to be in your team's GitHub repository for Milestone 3?

- your directories and files from previous milestones

- a copy of your code for your running/working Sprint 1 deliverable
- in directory `sprint-1`:
  - its previous contents
  - your file `README.txt` that describes your code for your completed Sprint 1
  - your file `testing-1.txt/.pdf` that describes how you actually tested your code for this completed sprint
  - in its sub-directory `team-meetings`, your additional reports for this sprint's Daily Scrum meetings, miscellaneous meetings, Sprint Review, and Sprint Retrospective

Milestone 4 - Sprint 2 - Sprint Planning Meeting

(note: also use these specifications if you have one or more additional sprints BEFORE your FINAL sprint)

deadline

This is due NO LATER THAN 11:59 pm on the date 1 week after your Sprint 1 deadline, **BUT SOONER WOULD BE BETTER!!!**

- **NOTE: keep in mind that you need to complete at least THREE sprints -- each resulting in a working system that does something valuable to the client -- by 11:59 pm on Friday of Week 15!**

how to "submit"

Make sure the milestone's pieces are in your team's private GitHub repository by the deadline.

what to submit

You are expected to have a directory in your GitHub private repository named `sprint-2`.

INCLUDE: a file `testing-plan-2.txt` or `testing-plan-2.pdf` within directory `sprint-2`, that describes how you plan to IMPROVE your testing for Sprint 2 (compared to Sprint 1)

- hint: this better not be empty!
- hint: think Extreme-Programming-attitude!
  - You'd better include HOW you will **improve** your UNIT testing,
  - ...AND you'd better include HOW you will **improve** your ACCEPTANCE testing
- (do you think no improvements are needed? Then **describe what went particularly well** with regard to **each** level of testing in Sprint 1, and note that you will be continuing to do that in Sprint 2.)

INCLUDE: you are expected to have, in `sprint-2/team-meetings/sprint-plan`, those pieces required from a Sprint Planning Meeting, as described in the section "**A Few Important Points re: Project Teams**" on pp. 4-5

What needs to be in your team's GitHub repository for Milestone 4?

- your directories and files from previous milestones
- in directory `sprint-2`:
  - a description of your testing plan for Sprint 2 in `testing-plan-2.txt/.pdf`
  - sub-directory `team-meetings`, containing sub-directory `sprint-plan`, containing:
    - the snapshot of your resulting PBL
    - your resulting SBL
    - your file `sprint-plan.pdf` /`.txt`
    - confirmation that your client has seen and agrees with the SBL's set of user stories for this sprint

Milestone 5 - Completed Sprint 2

(note: use these specifications if you have one or more additional sprints BEFORE your FINAL sprint)

deadline

This is due by 11:59 pm on the date you gave in Milestone 4's `sprint-plan.pdf/.txt` file.

how to "submit"

Make sure the milestone's pieces are in your team's private GitHub repository by the deadline.

what to submit

INCLUDE: You need to include a copy of the code for your RUNNING, WORKING deliverable within your team's private GitHub repository.

Note that it needs to meet the following requirements:

- it must run/work!
- the "client" can run it!
- it does something (even if "small") the "client" considers to be useful!
- it (hopefully!) implements the user stories you said it would implement in the SBL included in `sprint-2/team-meetings/sprint-plan`, and this is backed up by acceptance testing
- its "modules" include unit testing (which I am assuming you wrote **before** writing the code)
- every "chunk" of code has opening comment blocks, including the names of the pairs who created/worked on each, as described in the section "Required Comments for Project Code" on p. 2

INCLUDE: in your `sprint-2` subdirectory, a file `README.txt` that includes:

- a CLEAR DESCRIPTION of what I need to do to run your deliverable, or see it running
- a list of the directories/files in your repository containing the code for this sprint, including a short description of each file listed

INCLUDE: in your `sprint-2` subdirectory, a file `testing-2.txt/.pdf` that describes how you ACTUALLY accomplished the unit and acceptance testing for this sprint.

INCLUDE: in `sprint-2/team-meetings`, the reports for:

- Sprint 2's Daily Scrum meetings,
- any miscellaneous meetings,
- its Sprint Review, and
- its Sprint Retrospective,

...as described in the section "**A Few Important Points re: Project Teams**" on pp. 4-7

What needs to be in your team's GitHub repository for Milestone 5?

- your directories and files from previous milestones
- a copy of your code for your running/working Sprint 2 deliverable
- in directory `sprint-2`:
  - its previous contents
  - your file `README.txt` that describes your code for your completed Sprint 2
  - your file `testing-2.txt/.pdf` that describes how you actually tested your code for this completed sprint
  - in its sub-directory `team-meetings`, your additional reports for this sprint's Daily Scrum meetings, miscellaneous meetings, Sprint Review, and Sprint Retrospective

Milestone 6 - Sprint 3 (or *final* sprint) - Sprint Planning Meeting

deadline

This is due NO LATER THAN 11:59 pm on the date 1 week after your Sprint 2 deadline, **BUT SOONER WOULD BE BETTER!!!**

- **NOTE: keep in mind that you need to complete at least THREE sprints -- each resulting in a working system that does something valuable to the client -- by 11:59 pm on Friday of Week 15!**

how to "submit"

Make sure the milestone's pieces are in your team's private GitHub repository by the deadline.

what to submit

You are expected to have a directory in your GitHub private repository named `sprint-3`.

INCLUDE: a file `testing-plan-3.txt` or `testing-plan-3.pdf` within directory `sprint-3`, that describes how you plan to IMPROVE your testing for Sprint 3 (compared to Sprints 1 and 2)

- hint: this better not be empty!
- hint: think Extreme-Programming-attitude!
  - You'd better include HOW you will **improve** your UNIT testing,
  - ...AND you'd better include HOW you will **improve** your ACCEPTANCE testing
- (do you think no improvements are needed? Then **describe what went particularly well** with regard to **each** level of testing in Sprint 2, and note that you will be continuing to do that in Sprint 3.)

INCLUDE: you are expected to have, in `sprint-3/team-meetings/sprint-plan`, those pieces required from a Sprint Planning Meeting, as described in the section "**A Few Important Points re: Project Teams**" on pp. 4-5

What needs to be in your team's GitHub repository for Milestone 6?

- your directories and files from previous milestones
- in directory `sprint-3`:
  - a description of your testing plan for Sprint 3 in `testing-plan-3.txt/.pdf`
  - sub-directory `team-meetings`, containing sub-directory `sprint-plan`, containing:
    - the snapshot of your resulting PBL
    - your resulting SBL
    - your file `sprint-plan.pdf`/`.txt`
    - confirmation that your client has seen and agrees with the SBL's set of user stories for this sprint

Milestone 7 - Completed Sprint 3 (or *final* sprint)

deadline

This is due by 11:59 pm on Friday, December 8 (**unless** you indicated an **earlier** deadline in Milestone 6's `sprint-plan.pdf/.txt` file).

how to "submit"

Make sure the milestone's pieces are in your team's private GitHub repository by the deadline.

what to submit

INCLUDE: You need to include a copy of the code for your RUNNING, WORKING deliverable within your team's private GitHub repository.

Note that it needs to meet the following requirements:

- it must run/work!
- the "client" can run it!
- it does something (even if "small") the "client" considers to be useful!
- it (hopefully!) implements the user stories you said it would implement in the SBL included in `sprint-3/team-meetings/sprint-plan`, and this is backed up by acceptance testing
- its "modules" include unit testing (which I am assuming you wrote **before** writing the code)
- every "chunk" of code has opening comment blocks, including the names of the pairs who created/worked on each, as described in the section "Required Comments for Project Code" on p. 2

INCLUDE: in your `sprint-3` subdirectory, a file `README.txt` that includes:

- a CLEAR DESCRIPTION of what I need to do to run your deliverable, or see it running
- a list of the directories/files in your repository containing the code for this sprint, including a short description of each file listed

INCLUDE: in your `sprint-3` subdirectory, a file `testing-3.txt/.pdf` that describes how you ACTUALLY accomplished the unit and acceptance testing for this sprint.

INCLUDE: in `sprint-3/team-meetings`, the reports for:

- Sprint 3's Daily Scrum meetings,
- any miscellaneous meetings,
- its Sprint Review, and
- its Sprint Retrospective,

...as described in the section "**A Few Important Points re: Project Teams**" on pp. 4-7

IN ADDITION!!! in your `sprint-3` subdirectory:

- You will read about architecture views of a system in Jalote, Chapter 5 -- create and include an appropriate

Connector and Component (C&C) architecture diagram for your project's final sprint's version (in PDF format), in a file named `c-and-c-arch.pdf`.

  - Include a description of what you are trying to show here, and any additional information you think I should know about your C&C architecture diagram, in a file `c-and-c-arch-README.txt`.

- You will read about UML diagrams in Jalote, Chapter 6 -- select one type of **UML diagram** (DIFFERENT from your C&C architecture diagram, if you happened to depict it using UML!), and submit an instance of that diagram appropriate for your project's final sprint (in PDF format), in a file named `uml-ex.pdf`.
  - Include a discussion of why you chose this UML diagram type, a description of what you are trying to show here, and any additional information you think I should know about your UML diagram, in a file named `uml-ex-README.txt`.

What needs to be in your team's GitHub repository for Milestone 7?

- your directories and files from previous milestones
- a copy of your code for your running/working Sprint 3 deliverable
- in directory `sprint-3`:
  - its previous contents
  - your file `README.txt` that describes your code for your completed Sprint 3
  - your file `testing-3.txt/.pdf` that describes how you actually tested your code for this completed sprint
  - in its sub-directory `team-meetings`, your additional reports for this sprint's Daily Scrum meetings, miscellaneous meetings, Sprint Review, and Sprint Retrospective
  - a C&C architecture diagram in file `c-and-c-arch.pdf` and accompanying file `c-and-c-arch-README.txt`
  - a UML diagram (different from your C&C architecture diagram) in file `uml-ex.pdf` and accompanying file `uml-ex-README.txt`

Milestone 8 - Formal presentation/demonstration

A formal team presentation/demonstration of your final sprint's resulting project iteration will be given in class, during Week 15 of the semester; each team will present during its Week 15 lab section. Note that I hope to invite departmental faculty to attend these presentations if/as their schedules permit -- if you would like to invite anyone, particularly your "client(s)", please feel encouraged to do so!

To be presented:

during each team's lab section during Week 15, order probably to be determined by random drawing

(The exception would be if a team has non-CS-458 "client(s)" who would like to attend the presentation, and are only available during a particular time during the lab section -- such teams should let me know their "client"'s time constraints as soon as possible, and I will *try* to schedule such teams such that their "clients" can attend.)

Presentation purpose:

The purpose of this presentation is to:

- Demonstrate your final sprint's resulting project iteration in action -- to demonstrate its strengths, show off its capabilities!
- Add to your experience in giving a formal team presentation, especially one demonstrating software
- Give a final report to the instructor (and to audience members) about your final sprint's results

Turn in (BEFORE presenting):

- A typed outline of your team's presentation
- Print-outs of any slides, illustrations, etc. used in your presentation to the class that are not directly from the running final iteration
    - Black-and-white printouts with 4-6 slides per page are fine for meeting this requirement

Presentation guidelines:

- **Everyone** on the team should participate significantly and roughly equally in this presentation!
- The presentation should take 12-15 minutes (before the question-answer period).
    - It is expected that you will have planned, organized, practiced, etc. enough so that your presentation will be no shorter than 12 minutes, and no longer than 15 minutes.
    - At 15 minutes, your presentation will be stopped, so that there is time for all presentations.
    - If the presentation does take less than 12 minutes, or if it has to be stopped at 15 minutes because you have not yet finished, then there will be a **time penalty**.
    - A question-and-answer period of 3-5 minutes will follow the presentation.
- A formal **overview** bulleted-list of the presentation's main parts should **start** the presentation, and it should be both **displayed** and **verbally summarized** at the presentation's beginning.

- I hope that there will be guests from outside of the class at these final presentations; for them, as well as for formality, then, you should **briefly** introduce/describe your project, and then **briefly** discuss its "history";
  - You should **briefly** discuss interesting problems encountered, how you dealt with them, etc.
  - **Brief** discussions of general group-project issues your group encountered are also encouraged -- in fact, it would be remiss to omit them at this stage.
  - And, **brief** ties you have seen/experienced to general software engineering topics from class discussions and readings are **encouraged**.
  - You should definitely display and **briefly** discuss the Component and Connector architecture diagram of your final iteration as well as the UML diagram you chose to create related to your final iteration.
  - BUT **be careful not to take too long for this part**, because:
- ...Don't forget that the **MOST IMPORTANT purpose** of this presentation is to **demonstrate** your project final iteration "**LIVE**", actually showing it in operation; **this should be the HEART and the MAIN FOCUS of your presentation.**
  - Demonstrate its strengths -- show off its capabilities!
  - You are expected to demonstrate features "live" rather than to simply talk about them.
  - It is expected that you will carefully **plan** how you will demonstrate your project's capabilities (you know the order of demonstration of the different features, for example, and you have sample inputs/data already planned and ready to demonstrate;
    - ...you are **not** discussing within the team what to present next during the presentation, or what sample inputs/data to use!)
- After your live demonstration, you should include a **brief discussion of "future work"** -- what would you work on next, in the next sprint and/or in future sprints, if you had more time?
- Finally, be careful to close with a **definite conclusion**, briefly summarizing your accomplishments and/or lessons learned -- BUT end with a **positive** note, not an apology!
  - It should be clear to the audience that you have concluded your formal presentation -- (that is, I shouldn't have to guess whether you have finished or not... your presentation should not vaguely trail off, while you discuss: "Is that it? Yeah, that's it...")
- As implied above, it is expected that you will carefully plan, organize, and practice your presentation. E.g.,
  - Each presenter should avoid reading directly from notes and instead make eye contact with the audience (each should have practiced his/her part so that he/she can just glance at notes while presenting rather than reading them word-for-word)
  - The "segues" between team members need to be smooth and polished as well -- each member should introduce the next speaker and topic explicitly: "And now Sally will demonstrate the widget-adjusting portion of our application..."
  - One team member should operate (or "drive") the computer while another actually speaks/presents. This does require coordination and practice on the part of the team -- but the result looks more polished and professional.

    It should be planned in advance which team member is serving as "driver" when, and this "driver" should also know what to do next at any given point.

  - Each presenter should know what they plan to do next at any given point, as should the overall team;
    - (Avoiding having to discuss, during the presentation, what should be shown next -- everyone should already know that from having planned and practiced the presentation beforehand)

- You will use either the **BSS 313** lab projector; a laptop CAN be connected to these projectors, if you do not want to use the room's computer, BUT note that there is ONLY a VGA connector available (there is NO HDMI connector in BSS 313). It is the team's responsibility to make sure in advance that they can successfully project what they need to for this presentation; if you need to make special arrangements, do so well in advance of the presentation!

Milestone 9 - Individual reflection paper

Due:

Beginning of final exam period

Turn in:

Individually-written papers including:

- description of your favorite feature/aspect of your team's final sprint's resulting project iteration
- description of something that you learned (in general) from this project
- a description of tools/approaches your team used to communicate and collaborate on this project
  - include how well each tool/approach worked or did not work
  - how might you improve communication/collaboration on a future project that you work on?
- briefly discuss your experience using `git` and `GitHub` on this project, also including whether or not you now feel comfortable using it or not
- a description of how your team partitioned the work in the project
  - include whether this approach worked well or not
  - how might you improve work partitioning on a future project that you work on?
- a description of your team's approach to testing in the project
  - include whether this approach worked well or not
  - how you might improve testing on a future project that you work on?
- any other points, pertinent to course discussion, that you would like to make

These papers do not need to be long, but they do need to be taken seriously, and they need to discuss all of the areas listed above. Note that grammar and spelling will be taken into account -- as seniors presumably about to graduate, you should be able to write a technical reflection paper with proper mechanics!

The primary purpose of this milestone is to provide you with an opportunity for some final reflection on your project as the semester comes to an end.