

Project Description

Objectives

- Work in a group of size 4 or 5 people
- Learn how to communicate and how to work with others, resolve conflicts
- Make use of the methods and tools discussed in the course
 - Gain experience in some Agile processes
 - Standup meetings
 - Demo'ing to a client (e.g., end of Sprint demo)
 - Retrospective
- Produce a large semester-long software product to add to your portfolio
- Have a project on Github to reference in resume/interviews

Purpose

The purpose of this project is to provide the basis for applying what you are learning in class to a software project as you learn. Group work is an essential part of computer science and help develop communication and teamwork skills.

Group Project Description

All students in Software Engineering Methods and Tools are required to complete a group project. Each group will comprise 4 or 5 people. Time will be provided during class to discuss projects. Teams will be assigned based on schedules. Teams must submit a project proposal, which must meet certain minimum standards (see Minimum Project Standards below), and have that proposal approved by the instructor.

Rules:

- Team must consist of 4 or 5 team members.
- Teams will be formed with the help of catme.org.
- Each of you must commit frequently to Github throughout the semester.
- If you use pair programming, then when you commit to the code base you must have the name of your pair in the commit message to receive credit towards participating frequently to the repository.
- Peer Evaluation - putting all the same #'s in everything does not count as filling out a peer evaluation. Be sure to fill out the explanation of the peer evaluation on how your team did overall or issues with your fellow team members.
- If it is found that you personally did not do much on the project, this *can* retroactively affect your individual final overall project grade.
- Submission issues:
 - Submission is to your GitHub account.
 - If a team member submits your submission late, the whole team gets the late penalty. It is your whole team's responsibility to ensure you work together and coordinate submissions.
 - If a team member submits the wrong file or other problems with submission, this is the whole team's responsibility and the whole team is affected.

- If you are worried, then make a group rule that you always submit together, and/or the day before the due date.
- No submissions will be accepted via email. You must submit online including during the late period.
- The focus is on the *methods* and *tools* and less focus on fully completing what you set out to do. However, you still need to clearly demonstrate that you did enough work for 4 or 5 people working on a project.
- You will need to use GitHub (either github.com or for private repo use our department's GitLab (<https://git.cs.colorado.edu/>) or ask instructor for permission for something else.

Default Project Ideas

- There are some example projects on the website including past project ideas.
- Get your group involved in an open-source project.
- The department has some Raspberry Pi's and Arduino's available for projects. Ask Andy: andy.sayler@colorado.edu
- Some non-profit ideas
 - Tutoring/education site
 - Charity auction site
 - Helper website for disaster sites (find food, housing, etc) e.g., Katrina, Nepal earthquake, etc.
 - Android app to help mentor kids in Africa
 - Android app to help connect homeless to one-day job opportunities
- Website to display hard-to-understand data in an easier way (see <http://D3js.org>)
- Participate in a contest (<http://gocode.colorado.gov/> I know a guy who's group won \$25k from this!)
- Implement your own idea!
- Past projects:
 - Gitlytics: <https://www.youtube.com/watch?v=kKCORas1Tk0>
 - Monopoly: https://www.youtube.com/watch?v=P_kmAqMB3XU

Logistics

Project Part 1 - Proposal Submission, Repository, Project Tracker, Requirements

A huge part of software development goes into planning, whether planning up front as in Waterfall or planning every sprint in Agile. Planning consists of the requirements for a project and project tracking software to keep track as you go (among other things). You may opt for a Waterfall or Agile (or mash-up) approach for the project (you can change your methodology during the project as you wish).

- Create a group GitHub project and ensure all group members are contributors.
- Add your team information in the Google doc on the website under "Your Projects".

You need to submit a .PDF of your Project part 1 on GitHub which must be named:

ProjectName_Part1.pdf and have the following information:

- **Who:** List of people on the team: First name and Last name
- **Title:** Title of the project



- **Description:** Short Description of the project (help explain to us what you are doing)
- **Vision statement:** (what you would tell potential customers) Be sure to follow the guidelines from lecture!
- **Motivation:** for working on this particular project (Why would you develop this?)
- **Risks:** to project completion
(these could include:
 - Working environment or language new to some team members,
 - No prior experience working with the people on the team,
 - Lack of availability of some needed resource, etc.)
 - **Mitigation strategy** for dealing with the risk
- **VCS:**
 - Name of the version control software repository you will use (GitHub).
 - Link to your repository. Make sure instructor & TAs have access.
 - *Repository must contain your Part 1*
- **List of requirements:** for the project
 - 3 Separate tables for the requirements:
 - User Requirements
 - Functional Requirements
 - Non-Functional Requirements
 - Each requirement must have a unique ID number
 - Written in the Agile format (“As a [user], I want to [do X] so that I [can do Y]”)
 - Each requirement must have a size. You have two options for sizing:
 - Agile sizing – story points
 - Time estimate to complete

All stories should be sized to be completed in less than 8 hours (approximately, and something equivalent to a day or less in Agile story points). If your story is bigger than that, then break it up into smaller stories.
 - [optional] you can add priority (Critical, High, Med, Low, Nice-to-have)
 - [optional] you can add topic/area (e.g., Login, Profile, DB, etc.)
 - [optional] you can add user type (e.g., Admin, end user, potential customer,...)

Example table:

User Requirements				
ID	Description	Agile Sizing	Priority	
US-01	As a system admin, I want to see a dashboard of weekly usage by user type so that I can monitor the usefulness of the system.	2	Medium	
US-03	As a customer service rep, I want to be notified of failures so that I can respond to all users within an hour.	5	High	

- **Methodology:** Which methodology you plan to follow, though it may change: Waterfall, iterfall, Agile, mash-up of ...
- **Project Tracking software:**
 - Name of the software you will use (Trello is most common)
 - Link to project tracking software: make sure instructor and TAs have access

- *Be sure to continue to use this throughout the semester! Even though these are small projects, the point is to practice the methodologies and tools used in industry!*
- **Project plan:** created from your Project Tracking software. Copy-paste the plan (or screenshot of) from your project tracking software into the .pdf document.
- One person submit to Moodle a pdf file named: **ProjectTitle_Part_1.pdf** where *ProjectTitle* is the name of your project. (do not zip the file)
- **Extra Credit:**
For 1 point of extra credit towards your grade, you can add a section on market analysis to your submission. There is a lot to do for this:
 - Market analysis is done when starting your own business, buying a business, or to understand your market better as a product owner/manager/etc. to fully understand how your product or service compares to other similar products or services already on the market. This is required for anyone entering TechStars or other startup incubators, and if you go to your local SBDC (Small Business Development Center, which usually has mostly retiree volunteers who have successfully started or ran businesses who offer free advice *if you bring in a Business Plan*).
 - You will need to fill out ALL of the following template:

Market Analysis

<< This section covers market research and competitor analysis. It must demonstrate that there is a viable market and that you can beat the competition in the market for sales. >>

Target Market

<< The market to which you are planning to sell the product or service. Analyze the segments of this market as follows:

- Size of each market segment
- Is the segment growing or declining?
- Characteristics of potential customers in each segment >>

Profile of Competitors

<<Analysis of your competitors in the market:

- What are the competing products and services?
- Profile of key players (company size, turnover, profitability etc) and their market share
- Advantages and disadvantages of the competitors' offerings >>

Competitive Advantage

<< This is your assessment of why potential customers will choose to buy your product in place of those profiled above. Advantages may include:

- Unique features
- Price
- New technologies or systems
- Better value to customers in terms of efficiency or ROI or cost/benefit ratios
- Greater compatibility with existing systems
- Include any independent validation or case studies >>



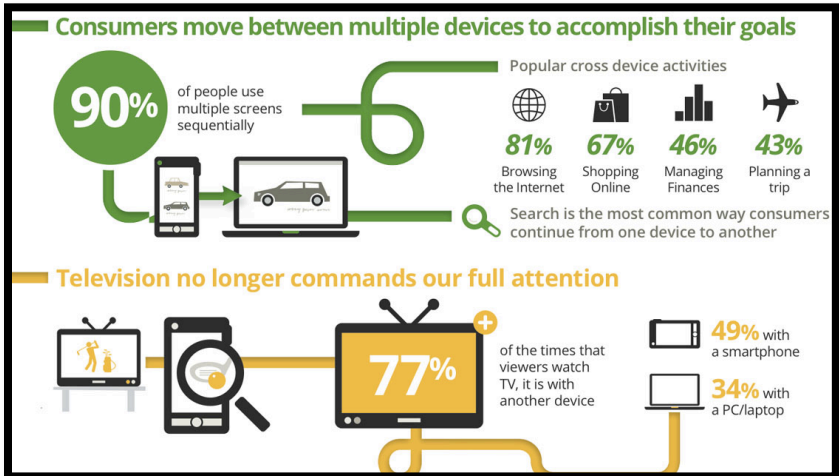
- **Extra Credit:**

You can get an additional 1 point extra credit if you create an infographic detailing this market analysis information as well (*you must do the market analysis extra credit before you can get credit for doing an infographic*).

An infographic is not a bar or pie chart!

If you steal something from the internet this would violate the honor policy!

- Example infographic



Project Part 2 – Agile Experience: *Stand-up Meeting, Demo to Client, Retrospective*

This is a 30-minute group interview grading meeting to be scheduled online. All group members must attend for credit. If you cannot find a time that works for everyone, you will need to email the instructor before the sign-up deadline to organize a different time.

ONE member of the team must sign up online for this meeting on Moodle. When you attend, you must present a piece of paper as you enter the meeting with your team name and a list of all members on the team (first and last name). There is nothing to submit online for this part of the project.

Standup

The first 5 minutes only will be the standup (you need to stand-up for the standup meeting). The purpose is to experience an Agile-format daily stand-up meeting. In industry we usually take turns answer the following 3 questions:

1. What did you do yesterday?
2. What are you doing today?
3. Are there any obstacles stopping you from doing your tasks?

We will modify this slightly to be “last week” and “next week” since you are probably not working on the project every day.

1. What did you do last week?
2. What are you doing next week?
3. Are there any obstacles stopping you from doing your tasks?

These questions are not asked during the meeting. Someone just starts:

“Yesterday I worked on setting up the database so we can add the tables. Today I will be working on creating the tables we need. No obstacles.” Then the next person goes.

Obstacles are specifically things that that *someone else needs to finish* as they are preventing you from completing the tasks currently assigned to you. Therefore, “I don’t know Angular.js so I need to learn it” is not an obstacle because you can fix that. “I need the DBA to create the tables so I can start adding the data into the database” is an obstacle.

If you are late or miss the standup (the first 5 minutes of your scheduled meeting), you will get a zero for the standup portion. You can still get credit for the demo and retro.

Demo

The objectives for the demo are:

- * Prepare and present a quick end-of-Sprint demo to the client.
- * Present using remote meeting software (commonly used if you work-from-home and to communicate with off-site clients).
- * Participate in a team Agile Sprint Retrospective.

The first 5-10 minutes is the end-of-Sprint demo. Create a presentation slide-show covering what you have accomplished so far (just 1-2 slides listing/describing/pictures of new features) and then do a demo of the features that currently work (hopefully you have something!). Do NOT show code or the database – pretend your client is not a technical person!

Your whole group needs to be present in person for this demo. To experience what a remote session is like, you will create a <http://join.me> account and invite the grader to practice connecting a remote session. Join.me has a limit to the number of users logged in (free version) so only the TA and 1 or 2 team members will log in to Join.me. The rest of the group will look at one of their team's screens.

Retrospective

Once you finish your sprint demo to the client, you will start the end-of-Sprint retrospective. Read up on the techniques described here:

<https://www.scrumalliance.org/community/articles/2014/april/key-elements-of-sprint-retrospective>

Your goal will be to have a set of team rules to use moving forward of things that worked well and what you want to improve on. Some examples may be: all team members need to use Google Hangout to easily notify the rest of the team if they are stuck or issues that arise; if a team member is more than 5 minutes late for a meeting, they have to bring cookies to the next meeting.

Step 1: Either hang a post-it on the wall or write on the whiteboard/chalk board for each category:

- What went well (keep doing),
- What didn't work (stop doing), and
- How we can improve (start doing).

Step 2: For the first 5 minutes each member will individually and quietly write out items for each category on post-it notes provided by the TAs (one note per post-it).

Step 3: When time is up, silently have everyone post up their note in the appropriate category.

Step 4: Now divide your team into two groups.

Step 5: One group will analyze the Keep Doing and Start Doing, and the other group will analyze the Stop Doing. The analysis should be to cluster similar themes together within a category.

Step 6: Each group reports on their findings for each category in order: Stop Doing, Keep Doing, Start Doing.

Step 7: Whole team makes decisions about how to make use of this information, what they may change moving forward. For example, you may come up with a decision that "When a team meeting is agreed upon, all members will show up on time." or "We get into these heated arguments, if someone calls 'time-out' then we will stop and focus on listing the pros and cons of each side." The idea at each Sprint retrospective is to help the team improve as a cohesive unit moving forward. This will totally depend on the team dynamics and every team ends up with different dynamics and different rules/guidelines.



Project Part 3 – Testing

The purpose is to create two types of tests for your project:

- Automated unit tests
- User Acceptance Test plans

1. Automated test cases

- Provide link to the tool you use to automate testing, or explain how to run the automated test cases, or schedule time with the TAs to demonstrate your automated tests.
- Provide a copy of the output showing the results of the automated test cases running.

2. User Acceptance Test plans

The purpose of these tests is to have a formatted plan that you could provide to users to go through the steps in using your application and report whether it was successful or not.

- Provide at least 3 test cases formatted similar to the following:

Test Case Template						
Project Name:						
Test Case ID: Fun_10		Test Designed by: <Name>				
Test Priority (Low/Medium/High): Med		Test Designed date: <Date>				
Module Name: Google login screen		Test Executed by: <Name>				
Test Title: Verify login with valid username and password		Test Execution date: <Date>				
Description: Test the Google login page						
Pre-conditions: User has valid username and password						
Dependencies:						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= sample@gmail.com	User should be able to login	User is navigated to	Pass	
2	Provide valid username	Password: 1234		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					
Post-conditions: User is validated with database and successfully login to account. The account session details are logged in database.						

One member of your team must submit to GitHub a .PDF with the following (in this order):

- **Who:** List of people on the team
- **Title:** of project
- **Vision:** from Project Part 1 Proposal
- **Automated Tests:** Explanation and screenshot (see above)
- **User Acceptance Tests:** Copy of at least 3 UAT plans
- **VCS :** Link to your VCS
- One person submit a pdf file named: **ProjectTitle_Part_3.pdf** where *ProjectTitle* is the name of your project. (do not zip your file)

Project Part 4 – **Individual interview**

This part is an individual interview meeting so we can learn about your own contributions to the project. This is required and failure to attend could get you a zero for your final project grade.

Project Part 5 - **Presentation**

The purpose of the presentations is to practice presentation skills, inspire your colleagues, and discuss the challenges you faced and how you got through them.

Group Project Presentation Description

Your presentation requires all members to be present and standing up front. If you are not present for the presentation you will get a zero for the presentation portion of the project. Each team member must discuss something about the project during the presentation.

You should cover the following items in whatever creative way you wish

- Title of the project
- Names of each person in the group
- All the tools your group used
 - Name of the tool, logo, and purpose (e.g., Project Tracking, VCS)
 - Your group's rating on how useful/good this tool/methodology was (ranked 1..5 where 5 stars is best and 1 star is useless)
 - Methodologies
 - Iterative, Waterfall, Agile, TDD, pair programming, peer code reviews, other...
 - Expected tools
 - Project Tracker
 - VCS Repository
 - Database
 - Testing
 - Auto-documenter
 - Deployment environment
 - (Optional) Additional tools you may have used
 - IDEs (e.g., Eclipse, Code::Blocks, xCode)
 - Frameworks (e.g., Laravel, Yii, Android Studio)
 - Hardware (e.g., Raspberry Pi, Arduino)
- Challenges you encountered, and how you overcame them and how it may have affected your original project plans.
- Demo your project

You only have around 5-7 minutes (exact times announced in lecture) to present! (including setup time). Feel free to practice ensuring your computer hooks up to the projector the week or so beforehand. (*Hint: to mirror displays on a mac, press cntrl-f1*).

Hint: Because you have a limited amount of time to present, make use of images – remember, a picture says a thousand words! Designing good pictures/infographics can really enhance a presentation! You can show a picture of your repository or project tracker, etc. to show how you made use of the tools. Be sure to use a minimum of 20-point font for all text so everyone can read it.

Presentations should appear as if created by a single person but should clearly demonstrate that all team members were involved. A group or company should always present a unified front to a client/customer/etc. Therefore be sure there is consistency throughout your presentation (theme, colors, fonts, grammar, tense [past/present], etc).

Presentations will be during the last week of classes or so.

Please respect each other: On presentation days, please do not enter or leave the lecture room while a group is in the middle of presenting. Please wait until they finish and the next group starts setting up so as not to disrupt their presentation.

One member of your team must submit to GitHub the following:

- Copy of the presentation
- Submit a pdf file named: **ProjectTitle_Part_5.pdf** or **.ppt** where *ProjectTitle* is the name of your project.

Project Part 6 - **Final Submission**

This last part will be graded based on the amount of effort, overall project, and use of tools and methodologies throughout the semester.

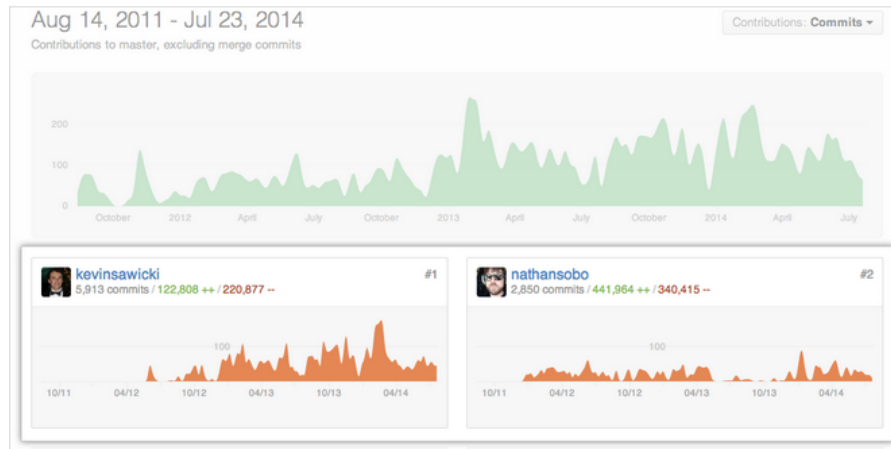
You must deploy your project and store your project source code, test cases, and documents in Github. To submit your project, one member of your team must submit a pdf with the names of everyone in your group and a link to these materials and access permission. The pdf document must have the following in the order provided as a *list or table* (not an essay!):

(Links should be click-able from the .PDF document)

- **Title:** of the project
- **Who:** Names of each person in the group
- **Project Tracker:**
 - Link to your Project Tracker *(Liz and TAs have access)*
 - Screenshot showing your project in your project tracker
- **Video:** 5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.
- **VCS:** Link to your VCS Repository *(Liz and TAs have access)*

We will check to ensure the following are stored in your VCS Repository:

 - Source code *(throughout the semester)*
 - Test cases
 - Auto-documenter documents
 - Video of demo
 - README in GitHub explaining to others what your project is about.
 - Project Part 5 document named **ProjectTitle_Part6.pdf**
- **Screenshot** of each member's contributions throughout the semester from Github:
e.g.,



- **Deployment:** Link to deployment environment *or explain how to access/run*.
- **Auto-doc:** The purpose is to run an auto-documenter on your code base. While Doxygen is very common (and Javadoc is used for Java), you may need (or want) to use another documenter. Make sure you choose a documenter that produces both a pdf and an html-based website.
 - List which program you did
 - Link to GitHub where the auto-generated docs are

Note: if you ran the program without adding any comments, you will receive a zero for the Auto-doc part. The purpose is to learn how to write the comments to properly document your code that are picked up by an auto-documenter. We will spot check a portion of your code to ensure it is documented properly.

Be sure to:

- Ensure Liz and the TAs all have access to all links for grading
- Tag your repo with " Final Submission" (make sure to push your tag to your repo)
- Include a README in your repo:
 - Describe repo organization
 - Describe where to find and/or how to build the docs
 - Describe how to build/run/test/etc code
 - If using a CI system, provide link to the CI status page

Project Part 6 - Peer Evaluation + Project Reflection

This portion of the project must be completed individually. The first part is to fill out an evaluation of your peers on the project. The second part is to answer a common potential interview question about working in teams.

All members of the group must submit individually to Moodle

- Submit the peer evaluation as a spreadsheet (not a pdf) as **Lastname_Firstname.xls**
- Note: putting all the same # in everything or per person or the like does not count as filling out a peer evaluation and will receive a zero.
- Be sure to fill out the explanation of the peer evaluation on how your team did overall or issues with your fellow team members.

- In the spreadsheet, write a short paragraph answering one of these following potential interview questions based on your experiences this semester on the team project. Grading will be based on the quality of your answer and how well you handle team dynamics. BE HONEST. An interviewer can always figure out if you are lying, and once that happens they will question everything else you say and have on your resume. The goal of this exercise is to prep you for interviewing.
 - a. Give an example of when you worked in a team and there was conflict or disagreement. How did you handle it? Did you reach a consensus? What would you do differently next time?
 - b. Give an example of when you worked in a team and one member always dominated the meetings and discussions. What did you do to help the rest of the team become more involved? What would you do differently next time?
 - c. Give an example of when you worked in a team and one member refused to do any work. How did you handle the situation? What would you do differently next time?

PROJECT GRADING

Total project grade is 100 pts. Distributed as follows:

25 pts for Part 1 Project Proposal, Repository, Requirements & Project Tracker + 1 or 2 XC

5 pts for Part 2 Stand-up Meeting

5 pts for Part 2 Demo to client

10 pts for Part 2 Retrospective

10 pts for Part 3 Testing

15 pts for Part 5 Presentation

25 pts for Part 6 Final Project submission

5 pts for Part 7 Team Dynamics interview question

Peer Evaluation (incorporated as factor of final individual project grade)

Part 4 Individual interview incorporated into final project individual grade

Overall effort (worthy of the size of your team) will also affect the grade.

If you do not participate throughout in the project then you cannot pass the course. Your level of participation within your group will also affect your final total project grade.

Individual project grade = (sum project parts) * overall effort * individual participation