COSC 499: Capstone Software Project Course

INTRO TO TEAMS AND PROJECT OVERVIEW

ANNOUNCEMENT

- Survey is available on Canvas.
- Complete the survey before next Monday at 12 pm.
- Teams will be announced on Wednesday

Agenda

- Project overview (Sep-Dec)
- Meetings
- Team roles
- Weekly Contributions

Project Overview

- September – December
  - **Week 2** → Teams-client introductions.
  - **Week 3** → Start meetings with clients, set up your project in GitHub, basic understanding of the project, defining features
  - **Week 4** → Agree on features for each milestone, identify possible features, develop testing strategy, identify the tech stack you plan to use.
  - **Week 5** → Data flow diagrams, prepare req video presentation
  - **Week 6** → Finalize requirements report, incorporate feedback from Week 5
  - **Week 7, 8** → Work on your user tasks
  - **Week 9** → create heuristic evaluation
  - **Week 11, 12** → Peer testing and video demo (prototype)
  - **Week 13** → Prepare for next term

Meetings

- Meetings with the client (weekly)
  - Need to have an agenda (short headings): Should be sent out to everyone before the meeting. Agree on the agenda with other teams.
  - To prioritize features, identify acceptance criteria, and communicate team progress.
  - Need to take minutes
  - You will create detailed notes
  - You will identify action items.
    - Each item must have a name and due date
      - e.g. Joe will ask client for account name by Sep 10
  - All of this should be committed in your repo.
- Meetings with your team (weekly)
- Meetings with me (weekly)

Meetings

- Meetings with the client (weekly)
- Meetings with your team (weekly)
  - All member should attend.
  - Need to take minutes
  - Decide what you hope to accomplish and who will be assigned which task.
    - 2-3 issues per team member
  - There should be a retrospective on the previous meeting
  - Arrange the task on a board: To do, in progress, and completed.
  - Tasks are broken down and have a success criteria assigned
  - Should be submitted in your repo
- Meetings with me (weekly)

Meetings

- Meetings with the client (weekly)
- Meetings with your team (weekly)
- Meetings with me (weekly)
  - Discuss on project progress
  - Discuss on individual progress
  - I will use the meeting to assess your progress

TEAM ROLES

There are many roles in a software project. We will see four common roles which will be taken on by your team.

- CLIENT LAISON
- TECHNICAL LEADER / DevOps
- PROJECT/SCRUM MANAGER
- QA LEADER

Each team member should assume one of these roles for the duration of the project. The responsibilities of QA leader should be distributed among all members.

TEAM ROLES: CLIENT LAISON

Your job is to manage all the communication with the client. You will be the expert on the product and the client's needs. You will help clarify project requirements. If you are unsure / unclear about what the client would want, you should communicate with the client until you become sure.

Some responsibilities:

- Setup weekly meetings with clients
- Taking minutes
- Introduce your team in the first meeting with client
- Submit weekly report meeting with client

TEAM ROLES: TECHNICAL LEADER/DevOps

Your job is to make sure that the technical requirements are being met. You are also responsible for ensuring that defined tests are executed, including executing any manual tests or setting up CI to execute automated tests.

Some responsibilities:

- Setup the Git repository with the correct settings for pull requests, code reviews, etc
- Integration help with merging PRs, if required
- unit testing for your project

TEAM ROLES: PROJECT/SCRUM MANAGER

Your job our job is to make sure that the project is on track and that the process is being followed by everybody. You should make sure that the participants have a voice.

Some responsibilities:

- Set up weekly meetings with clients and with your team
- Introduce your team in the firs meeting with client
- Submit weekly report meeting with client

TEAM ROLES: QA LEADER

Your job is to define quality standards which your project must meet. You will help to identify areas which needed to be better tested, brainstorm ways to test, and help to refine documentation through review.

Some responsibilities:

- Ensuring the acceptance criteria of the user stories
- Ensuring the validation techniques of the user stories
- Ensuring that any required documentation is completed

Weekly Contributions

Objective reporting with subjective judgment of overall progress:

- Objective reporting: gitstats
- Each team submits gitstats report by Friday (almost) every week.
- Graded on Canvas by next Friday
- Class time: ask questions, discuss project progress

Calculating Weekly Contributions

- Based on report, your score is:

IndivWeeklyScore= Punctuality + Professionalism + Preparedness
+ **IndivWeeklyContr**

IndivWeeklyContr= **TeamProgress** * **IndivEffort**

Calculating Weekly Contributions

- Based on report, your score is:

Out of 3

Out of 10 IndivWeeklyScore= Punctuality + Professionalism + Preparedness
+ **IndivWeeklyContr**

Out of 7

IndivWeeklyContr= TeamProgress * IndivEffort

Calculating IndivEffort

- **IndivEffort** is a measure of how much work you did individually over the past week, relative to how much the entire team did.

- **IndivEffort** = weighted sum of 4 types of work done (normalized by the respective quantities done by the team)
  - Number of completed pull requests
  - Number of commits on completed pull requests
  - Number of changes on completed pull requests
  - Number of comments on completed pull requests

Calculating IndivEffort

$$\text{effort} = 5 * f(p_i, \bar{p}) + 5 * f(c_i, \bar{c}) + 3 * f(h_i, \bar{h}) + 2 * f(o_i, \bar{o})$$

- p is the number of completed pull requests contributed to
- c is the number of commits on completed pull requests
- h is the number of changes (line additions or subtractions) on completed pull requests
- o is the number of comments on completed pull requests
- i indicates the amount completed by the individual

Calculating IndivEffort

- Each element is calculated as a piecewise linear function of the expected number for a week. The expected contribution per week is:

$$f(x, \bar{x}) = min(\frac{x}{\bar{x}}, 1) + 0.25 * max(min(\frac{x}{\bar{x}}, 2) - 1, 0)$$

- Create 4 commits (merge commits are excluded)
- Generate 200 changed lines of code (excluded merge commits)
- Contribute to 4 pull requests (ideally 2 as a coder and 2 as reviewer)
- Create 4 comments on pull requests.

Calculating IndivEffort

- Normalized to highest **IndivEffort** mark for the team
  - E.g. If you get the highest **IndivEffort** mark in your team, then you will get 15/15 = 100%
  - E.g. .Suppose you get **IndivEffort** mark of 1, and the highest is 15, then you will get 1/15 = 6.67%
- Score ∈ [0,100]

Calculating TeamProgress

- **TeamProgress** is a measure of how much work the entire team did, relative to how much work we expect the team to have done in a week
  - Amount of work expected is a function of teamsize
- **TeamProgress** = min(**workDone**/**workExpected**, 1)
  - **workDone** is the sum of 3 types of work:
    - Number of tasks completed by the team
    - Number of exploration tasks completed by the team
    - Number of chores completed by the team
  - **workExpected** is the sum of:
    - 2 of any work * number of team members of the above work

Calculating TeamProgress

- Based on in-class meeting, the ratio of **TeamProgress** could increase to 1.0 if I believe there are **real** obstacles why everyone is behind

- Note that you will not get a score more than one for completing more than the expected number of tasks. It is better to work consistently to promote team sustainability.

- Score $\in [0,1]$

NEXT WEEK:

- GitHub
  - GitHub for software development
  - GitHub Exercise

- Contact client (after Wednesday)
  - Introduce yourself and your teammates
  - Set up a weekly client meeting that works for all 3 teams on the project
    - Have as many team members attend as possible
    - The client liaison is required to attend all client meetings

TIME TO COMPLETE THE SURVEY