Project Planning Exercise

JD Kilgallin

CPSC:480

09/21/22

"Give me six hours to chop down a tree and I will spend the first four sharpening the axe."
–Abraham Lincoln

Learning Objectives

- Git merge conflict resolution
- Analyzing requirements
- Modeling requirements
- Developing components of a product specification
- Identifying and estimating tasks

- Project 2 assigned

Planning Poker

- Method for producing consensus estimates of effort needed for tasks.
- Team meets at the beginning of a product iteration or development sprint with a list of pre-defined tasks to be estimated.
- Each developer has a set of cards with numbers on one side (the "face").
  - The set of numbers varies by team, as do the units (hours, days, generic "points").
  - Keyfactor uses points in multiples of Fibonacci numbers: ½, 1, 2, 3, 5, 8, 13, 20, 30.
- For each task to be estimated, the team:
  - Reads the task to ensure everyone understands the task to be done.
  - Has each member select a card from their "deck" for their individual estimate.
  - Places cards face down on a table, then flips them when everyone has played.
  - Discusses any major differences to identify causes of disagreement (may redo round)
  - Takes the mean or median as the consensus estimate for that task.
  - Tries to break down any tasks that don't fit in a reasonable amount of effort.

Project 2

- Select a software project to plan and design
- Define and prioritize requirements (due Monday, Sept 26)
- Model requirements
- Build a requirements specification (due Sunday, Oct 2)
- Create tasks from requirements
- Estimate effort required for tasks
- Create high-level architecture and product design
- Plan and start sprint
- Report on project and product (due Sunday, Oct 16)

Software Project Constraints

- You must be able to define at least ten functional requirements of the software. If a project has substantially more than ten functional requirements, consider narrowing the scope to focus on a sub-portion of the proposed software that could still have value.
- The software must be constructable (so no software for time travel)
- Plan to use this product for projects 2, 3, and 4; while change is possible, it may significantly increase the amount of work that needs to be done.
- You will not need to construct a complete, working product during this class; you should, however, expect to contribute *some* code toward development of this product.
- A project that you *are* able to complete as a team during the semester will provide the most value to you in terms of both portfolio content and software engineering skills mastery.

Defining a project

- Each team member should propose one software project, including the project goal, scope, and definition, along with ten functional requirements presented as user stories. Use the exercise 2 assignment as a template.
- This may be a new (greenfield) software project, a significant enhancement to a previous project, or a contribution to open-source software.
- As a team, select one of the proposed projects to move forward with.
- You may modify the proposed functional requirements at this point, and must add at least 5 non-functional requirements.
- Prioritize all requirements from highest to lowest. Also select a team name.
- Submit proposed team name, project, and list of prioritized requirements through Brightspace for approval by **Monday, Sept 26, 11:59 PM**.
- I will create a GitHub repo for the team and provide access to team members to submit the rest of the work.

Checkpoint Oct 2

- Submission through GitHub. At this point, GitHub repo should contain:
  - Each student's alternative proposal that was considered.
  - Source files for all diagrams.
  - Project specification document.
  - At least 2 commits or pull requests from each team member (e.g. proposal + scenario model)
  - A release containing the pdf version of the specification.
- Checkpoint grades will be 80% shared and 20% individual, as follows:
  - 10% - Original project proposal and requirements from each user (individual)
  - 10% - Sequence diagram contribution & 2 commits/pull requests from each user (individual)
  - 05% - Proposal submission and prioritized requirements
  - 10% - Class diagram
  - 20% - Sequence diagrams
  - 10% - State diagram(s)
  - 30% - Specification
  - 05% - Pdf release and directions followed

Proposed Successful Project Schedule

	9/19 Read project assignment	**9/20 Submit team requests**	9/21 Exchange contact info	9/22 Complete exercise 2	9/23 Develop idea and requirements	9/24 Meet; Select project for proposal
9/25 Adjust and prioritize requirements	**9/26 Makeup any delay & submit**	9/27 Review feedback & GitHub	9/28 Complete exercise 3	9/29 Develop class model	9/30 Develop scenario models	10/1 Meet; finish spec sections remaining
10/2 Makeup any delay & submit	10/3 Study for midterm	10/4 Study for midterm	10/5 Review feedback; assign task definition	10/6 Develop tasks for requirements	10/7 Draft product design	10/8 Meet; estimate tasks & review design
10/9 Finalize design; assign tasks	10/10 Review tasks; enter comments	10/11 Standup meeting	10/12 Make progress on tasks	10/13 Complete anything remaining	**10/14 Meet; Review; early submission**	10/15 Makeup for any delay
10/16 Makeup for any delay; submit	**10/17 Fill out team survey**	10/18 Review feedback	**10/19 Begin project 3**			

Exercise

- Four groups use back row and second row on both edges of the room.
  - Turn the chairs from the next row around so everyone can cluster.
- Move front tables and aisle chairs up to give enough space for the last two teams.
- Take a set of index cards each for planning poker
- Exchange contact info for group project
- Decide which version of the exercise to do
- Complete exercise

Teams

Beary Boys
Brandon Kiser
Andy Mee
Andy Markland
Jordan McKee
Nicholas Limbach

Common thread: Security
Andrew Santa
Nathan Brannon
Ashton Carruthers
Ken Mills
Zifeng Yuan

Spinning Dog Cube
Jenice Mario
Amid Babaev
Anthony Lupica
Nick Reichlin
Angelo Indre

Common thread: C++/IoT focus
Shubh Patel
Ja'den Martin
Avery Kuhn-Brooks
Matt Dudek
Kevin Nguyen

Common thread: Game dev
Kent Hoang
Maddie Zakham
Riley Holschuh
Jason Kotowski
Desmond Vang

Common thread: Web dev
Mitchell Ruple
JorNyece Cox
Brad Sandorf
Brandon Kulmala
Isabel Beam