Sprint Release & Next Sprint Requirements List

Purpose: Develop working software for a Sprint's worth of requirements (e.g., Sprint 1 Release) and create a list of requirements for the next Sprint (e.g., Sprint 2 Requirements List).

Due Dates:

- Sprint 1 Release & Sprint 2 Requirements List - 11:59 PM, Sunday, October 27, 2024
- Sprint 2 Release & Sprint 3 Requirements List - 11:59 PM, Sunday, November 10, 2024
- Sprint 3 Release & Final Sprint Requirements List - 11:59 PM, Sunday, November 24, 2024

Deliverables (submitted to Canvas):

1. Github link for this Sprint Release, e.g., Sprint 1 (50% of grade – team based)
2. Updated Requirements Stack spreadsheet (5% of grade – team based)
3. Requirements Artifacts for the next Sprint, e.g., Sprint 2 (20% of grade – team based)
4. Team Peer Evaluations (25% of grade – individual based)

Steps:

1. Develop working software that meets the requirements specified in the Requirements List and Artifacts.
    - The software must be adequately commented with:
        - Prologue Comments
            - Name of code artifact
            - Brief description of what the code does
            - Programmer's name
            - Date the code was created
            - Dates the code was revised
                - Brief description of each revision & author
            - Preconditions
                - Acceptable and unacceptable input values or types, and their meanings
            - Postconditions
                - Return values or types, and their meanings
            - Error and exception condition values or types that can occur, and their meanings
            - Side effects
            - Invariants
            - Any known faults
        - Comments summarizing major blocks of code
        - Comments on every line
    - Comments for "4GLs" (e.g., scripts, frameworks, graphically generated code) should be sufficient enough that another programmer familiar, but not expert, in the 4GL can understand what is going on.

2. Submit the Github URL for your Sprint Release to Canvas.
    - All teams must have a code freeze as of the due date and time.
    - Only one team member needs to submit the URL.
    - The timestamp will be judged by the final commit on your master branch of the team's Github repository.
    - You must demo your code during your team meeting with your GTA based on that master branch as of the code freeze.
3. Update your Requirements Stack spreadsheet as necessary.
    - Make any corrections you received points off on last time.
    - You may add, delete, and split Requirements and update columns as you see fit.
    - Update which requirements go in which Sprint.
    - Your grade for the next Sprint Release will be based on the ones you identify for it, so consider these carefully.
    - The grade for your next Sprint Release grade will be determined partially by the number of requirements you actually complete AND the number you think you will complete in comparison with the other teams your GTA is responsible for.
    - Teams that successfully complete more requirements than other teams will get a better grade for the next Sprint Release.
4. Create and submit a Requirements Artifact for each requirement in the next Sprint Release.
    - Each Requirements Artifact should contain enough detail that your GTA can determine if the requirement is met in the next Sprint Release.
5. Planning (i.e., updating Requirements List and creating Requirements Artifacts) vs. developing
    - Agile best practices say you should spend about 1 hour of planning for every week of a Sprint.
    - However, most sources say teams do not spend enough time on planning.
    - The 581 sprints are 2 weeks, so you should spend about 2 hours planning each sprint.
    - My suggestion:
        - Spend about 1 hour as a team identifying the requirements for the next Sprint Release:
            - Pick the number of Story Points you think you can do in a Sprint; you should get better at this as you do more Sprints together as a team.
            - Then select the number of highest priority requirements which have a total number of Story Points that you think you can do in a Sprint.
        - Divide up the requirements between the team members to create requirements artifacts for each requirement (about 1 hour per team member).
        - Reconvene for a short 15-minute meeting to review the requirements artifacts and decide how you are going to divide up the development.
        - Discord, email, etc. can facilitate this process, but face-to-face meetings (either in-person or on-line) are essential to successful project planning and speed up the planning process.
    - If it takes you longer than this, then you probably are not doing Just-In-Time design and you are wasting time you could be coding, commenting, and testing your software.
6. Submit a Team Peer Evaluation form

- This part will be based on the Team Peer Evaluation forms as described in the first lecture.
- You may receive more than 25 points if your teammates think you did more than your fair share of the work.
- If you do not turn in a Team Peer Evaluation form or turn it in late:
  - You will still get a Team Peer Evaluation score based on your teammate's evaluation.
  - You will lose 25 points, which could result in a negative score.
  - The $10,000 bonus you were supposed to divide between your teammates will be divided equally among your teammates.

Grading Rubrics:

Sprint Release (50% of grade – team based)				
Measure	Points	Grading Level		
		Exceeds Expectations (90-100%)	Meets Expectations (80-89%)	Unsatisfactory (0-79%)
Requirements met based on Requirements Artifact (i.e., requirement delivered)	12.5	Met all requirements and exceeded some	Met all requirements	Did not meet all requirements
Overall quality of Sprint Release in comparison to GTA's other teams*	12.5	Base on degree of difficulty and requirements met, project is in upper 25% of the GTA's teams	Base on degree of difficulty and requirements met, project is in middle 50% of the GTA's teams	Base on degree of difficulty and requirements met, project is in lower 25% of GTA's teams
Comments	12.5	Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line; comments for "4GLs" are sufficient enough that another programmer familiar, but not expert, in the 4GL can understand what is going on.	Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line; comments for 4GLs are insufficient.	Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line; there are no comments for 4GLs.
Ratio of Story Points Planned vs. Story Points Delivered	12.5	More than or equal to the average for all of the GTA's teams	Less than the average, but more than 25% of the average	Less than 25% of the average
*Not all projects are equal. This measure is to help teams with difficult projects and high aspirations that don't quite meet all their requirements, but still produce a Sprint Release that exceeds expectations.				

Requirements Stack Spreadsheet (5% of grade – team based)				
Content	Points	Grading Level		
		Exceeds Expectations (90-100%)	Meets Expectations (80-89%)	Unsatisfactory (0-79%)
Spreadsheet format	5	100% if specified 5 columns are present and in correct order; requirement IDs are sequentially numbered; description of requirements is present in all rows and not duplicated; the story point numbers are valid (i.e., 1, 2, 3, 5, 8, 13, or 21); the priority is present in all rows; and a Sprint No. is present in all rows. 0% if not		

Requirements Artifacts (20% of grade – team based)				
Content	Points	Grading Level		
		Exceeds Expectations (90-100%)	Meets Expectations (80-89%)	Unsatisfactory (0-79%)
Artifacts	20	All of the artifacts are of sufficient detail that the GTA can tell if the requirement is delivered in next Sprint Release or not; some of the artifacts are the best ones from all of the GTA's teams.	All of the artifacts are of sufficient detail that the GTA can tell if the requirement is delivered in next Sprint Release or not.	Some of the artifacts do not have enough detail for the GTA to tell if the requirement is delivered in next Sprint Release or not, or some are missing.