IC470 – Software Engineering

Due date: As per the course syllabus

Milestone 5 – Testing: Progress Demo and Code Analysis (30% of Backlog Complete as per your Release Burndown Chart)

Due date: At the start of class as per the syllabus

Changes: none (yet)

See the course web page to determine during which period your Milestone will be presented (all team members in that section will participate in the presentation). Team members that are not in the same section as the scheduled milestone presentation will earn the same grade as the one earned during the presentation period, and are expected to contribute their fair share to the development of the presentation's materials.

Note: Some important points are highlighted below in blue.

Milestone _____ Team _____
(the above filled in by the team)

Deliverables checklist (see below for more info on each of these items). Be prepared to present the below, in order, during your milestone delivery.

- — Checklist (a copy of this deliverables checklist sheet)
- — Customer's Evaluation Cover Sheet completed by Customer (and Tech Advisor), or, copies of emails showing your attempts to contact them at least two days prior
- — Concise Project Overview (discuss any changes)
- — High-level diagram (highlight today's focus)
- — Customer meetings summaries and action items with lead mid for each item id'd
- — System Status (% completion of your Starting Backlog)
- — Release Burndown Chart
- — ScrumMaster's Plan (for next sprint)
- — Link to full Functional Requirements Trace Table (shared with all IC470 instructors)
- — Project Demo (3 parts):
  - Requirements mapping
  - Design Artifact
  - Live Demo
- — Back up Demo: Video demo/screen shots of system passing each test case the Customer has signed off on for this milestone (only use if you have problems during your live demo)
- — Email a pdf (named teamX_mileZ.pdf) of the milestone presentation slides to the instructor, with the subject line: team X, mile Z. All things you include in the presentation must be in this single pdf file (insert screengrabs, etc if necessary)

30% Complete: Target acceptance test cases for this milestone sufficient to meet being *at least* at 30% Backlog Complete with regard to your Release Burndown Chart. Customer involvement alert => Meet with your Customer and determine which test cases you will focus on for this milestone.

Milestone Deliverables (paper copies turned in to your instructor *prior* to beginning your oral presentation)**:**

0. **Arrange to meet with your Customer**. Same requirements as Milestone 1

1. **Admin.** Customer's Evaluation Cover Sheet and Milestone Lead (same requirements as Milestone 1.)

2. **Proj Mgnt** (Project Management)

    a. **Concise Project Overview.** Discuss any changes since the last milestone.

    b. High-level diagram. Include a high-level diagram, and specifically point out the portions of your project that are the focus of today's presentation.

    c. **Customer Meeting Summaries. Customer Involvement Alert**. Remember to:
        i. Include a summary of what was discussed at each Customer meeting to include a list of all action items,
        ii. Identify which team member is the lead on getting each action item resolved.

    d. **System Status.** Present your current system status in the form of the % completion of your Starting Backlog. Include an unreduced fraction showing the total of the effort values of each backlog that can be demo'd as complete (and that your Customer has signed off on) as of this milestone as the numerator, and the total effort of the Starting Backlog as the denominator. For example **=> System Status** (% of Starting Backlog Complete): 23/83 = 27.7%
        i. For example **=> System Status** (% of Starting Backlog Complete): 23/83 = 27.7%
        ii. This example shows that the effort values of the acceptance test plan test cases for the entire project (ie., the Starting Backlog) sum to 83, and that the team is able to actively demonstrate (you may only include those that the Customer has signed off on) passing test cases that sum to an effort level of 23. Give this number both as the unreduced fraction as well as the percentage.
        iii. **Shortfall Penalty (up to 10%).** As per the Agile Manifesto, working (tested and able to be demonstrated) software that meets

the customer's needs is the primary measure of progress. As such, there is a penalty (up to 10% per milestone) for falling short of the required percentage of functional requirements completion for any given milestone.

i. For example, if 60% of the functional requirements are required to be complete for a milestone but the customer only concurs on 55% of them being 'Completed' to their satisfaction, ((60-55)/60)*100 = 8.3 points will be deducted from the milestone grade.

ii. These points cannot be recovered even if you later pass the test cases in question to your customer's satisfaction.

iv. Ahead of Schedule Extra Credit (up to 10%). Extra credit (up to 10%) will be earned by teams that are ahead of the progress requirements for this milestone using the same formula as the shortfall penalty.

e. **Release Burndown Chart**. Present your Release Burndown Chart with your Actual Work Remaining Line updated to include those acceptance test plan test cases (Customer involvement alert =>) which your Customer has signed off on as being completed to their satisfaction.

f. **ScrumMaster's Plan.** Present your ScrumMaster's Plan (see Tables 1..5 in Appendix A below) for the sprint to the next progress demo milestone.

i. Assume that the next progress demo milestone is Milestone 7 in IC480 and requires 50% of the Starting Backlog to be complete by the beginning of the 3rd week of the IC480 semester.

ii. Replace "X" in the tables with the milestone number of the *next* milestone delivery (so, 7 in this case). The filled out tables comprise your ScrumMaster's Plan for the sprint to the next milestone delivery.

g. **Link to full Functional Requirements Trace Table**. Instead of presenting your entire trace table, provide a link to a Google drive spreadsheet with your trace table, and share this link with all the IC470 instructors. Be prepared to bring up and discuss your trace table if asked to do so by your instructor. Modify (but do not present) your entire Functional Requirements Trace Table as follows (and as shown in Figure 1 of Appendix A below).

i. **Effort Value**: Add an Effort Value column to your trace table giving the effort value that your team has assigned to each acceptance test plan test case. (See table below).

3. **Testing.**

a. **Project Demo:** For your project demo, interleave the following three

steps by 1) showing the trace table row with the test case and your Customer's initials, 2) step-by-step walk through of the test case using the relevant design artifact, and 3) live demo'ing your system passing the test case. Then repeat these steps for the next test case. The following further discusses each of these steps:

i. **Requirements mapping**. Bring up a copy of JUST the relevant portion of your Functional Requirements Trace Table (ie. the rows corresponding to a test case your Customer signed off on for this milestone). You may put in your Customer's initials electronically for the test case once they have signed off on it, or have otherwise agreed that the test case has been completed. Have documentation (physical signature, email, etc) available if your instructor asks to see it.

ii. **Relevant Design Artifacts**. Present the design artifact, to include a title and the category (data flow diagram, decision tree, etc), corresponding to the test case. Your focus is to give your audience the context, from a design perspective, of the requirements and test cases they are about to see demo'd. In your design artifact presentation:

    1. Go through a step-by-step scenario corresponding to the acceptance test cases using your design artifact that clearly shows how your system design fulfills the test case.

    2. Give an explanation for any test case signed off on by your Customer for which you do not have a design artifact. Warning: this should be the case only for extremely simple to implement portions of your project, all complex parts of your project must have a design artifact.

    3. The goal of each Agile sprint is to develop the design artifacts corresponding to the requirements first, and then implement the test cases. Avoid the trap of doing this the other way around, as you'll miss the implementation benefits of having a well thought out design.

iii. **Live Demo of test cases**: Clearly show, by running your code, how each of your acceptance test cases that your Customer agrees you have completed is met by your system in operation.

    1. Use input that you know will work, but be prepared to enter alternate input as directed by your instructor.

    2. Refer back to your trace table as you demo each test case

b. **Backup Demo**: Prepare videos of your system passing each test case you plan to show in your Live Demo.

i. Use these videos as a backup demo only if you have trouble with the live demo.

ii. If a video demo is not a viable back up option, include slides in your presentation that show static screen shots of your system in operation and use these as your back up to demo your test cases if you have problems with the live demo.

Appendix A: Figures and tables discussed in the milestone

Functional Requirement	**Set of Acceptance Test Plan test cases -** in total, these test cases must demonstrate that the Functional Requirement in question has been met, and must include (and include an indication of) both normal and exception uses of the system.	**Build**	**Use Cases**	**Design Artifacts** – include the category of the design artifact. If none, explain why (See 3.a.ii below)	**Effort Value**	**Testing Status & Milestone #** – note that each test case has its own status field. If status is 'completed,' include the milestone # in which that the Customer signed off on the test case	**Customer's Initials** only for test cases demo'd by the team for which the customer is 100% satisfied.
<u>Login/Password GUI</u>: Each user must have their own login and password pair that sets their User Role within the system. Primary: MIDN J. Gish Backup: MIDN W.T. Door	1.1 User with correct login/password attempts to login. **Expected result -** > User is able to login, the correct User Role is associated with the login/password pair. (*normal*) 1.2 User attempts login with the wrong password. **Expected result -** > ~~User is prevented from logging in.~~ <u>User is prevented from logging in after 3 failed attempts. (13 Oct: Customer requested that the number of invalid login attempts be limited to 3</u>	Build 1	UC1: Assign User Role	User Role Assignment - UML Sequence Diagram	1 1	1.1 *Completed* –IC470 Mile 4 (customer must initial in the column to the right agreeing that they are satisfied with the test case demonstration) 1.2 *In progress.* We thought this one was done, but the customer modified the requirement and told us to lock the user out after 3 failed login attempts.	1.1 <u>WTD</u> 1.2 <u>_____</u>

	(*exception*) 1.3 User indicates they forgot their password. **Expected result** -> System emails a temporary password which user is required to change upon logging in. (*exception*)				3	1.3 *In progress.* Have to push this to a later build since we haven't yet figured out how to send an automatic email. Anticipate completion by Build 2.	1.3 _____

Figure 1. Functional Requirements Trace Table

ScrumMaster's Plan for Milestone "X" **ScrumMaster** ___Gish_____

Lead Developers*	**Sprint Backlog** (test cases being worked on)	**Expected "effort" value** for each test case
W.T. Door	1.1 Forgotten password and 1.7 FAQ page	2 and 3
J. Gish and S. Sam	3.2 Terrain map display - realistic shadow growth and motion at dusk.	6
J. Mid	5.7 Save gameplay to file for subsequent restart.	4

Table 1. Lead Developers and Sprint Backlog for Milestone "X"

Lead for addressing Unplanned Requirements/Test Cases	**Description** of Unplanned Requirements/Test Cases	**Steps** to address Unplanned Requirements/Test Cases
J. Gish	Customer wants to add a new functional requirement that shows a 30 second countdown timer. After the	Add new requirement to Functional Requirements Trace Table. Adjust Burndown chart to include replotting of Ideal Work remaining line. Adjust Starting

	countdown timer elapses, player loses their turn.	Backlog to include effort values related to new functional requirement.

Table 2. Unplanned Requirements/Test Cases

Projected Burndown at next milestone (projection must account for the impact of any Unplanned Requirements/Test Cases)	**Projected Burndown: Ahead, On, or Behind Schedule** at the next milestone delivery. Identify amount for ahead or behind schedule projections.
25%	Behind by 5%

Table 3. Burndown Projection for Milestone "X"

Lead for item	**Description of non-test case items**
W.T. Door	Arrange Customer meetings.
J. Gish	Prepare Action Items list emerging from Customer meetings, and ID who has the lead for each action item
J. Mid	Prepare Milestone Delivery PowerPoint

Table 4. Additional (non-test case) Items for Milestone "X"

Lead for resolving impediment	**Description** of impediment	**Steps** to resolve impediment
W.T. Door	AttackGoat character's movements on the game board are jerky.	Figure out how GPU can be allocated to speed up AttackGoat character's refresh rate to produce a smoother display on the game board.

Table 5. Remaining Known Impediments to date (Cumulative, not milestone-specific)

*Notes on Lead Developers.

a) Every team member (including the ScrumMaster) must be assigned as a lead developer for at least one acceptance test case in the milestone sprint.
b) Team members may need to be assigned as the lead developer to more than one test case (especially ones with low expected "effort" values).
c) You may have up to two team members assigned as the lead developer(s) to each acceptance test case in the milestone sprint (especially ones with high expected "effort' values). If more than two developers are needed, refactor the test case as needed, to include updated effort values.

Notes:

a. Each team is to be fully ready to go at the beginning of the presentation period to include handing in a paper copy of all slides, source code, and GUI screen shots used in the presentation/software demonstration as well as the documentation. Also, each team is to turn in a copy of the oral-presentation grading sheet (available from the course web page), with your team members' names filled in, at the *start* of the period *prior* to beginning your oral presentation.

b. Any team not ready to hand in their paper copies of the above, or to deliver their presentation/demonstration when called upon, will have 10 points deducted from their presentation grade and will go to the end of the presentation cycle for that day. Presentations not delivered during class on the due date will earn a grade of zero, but will still have to be completed and turned in to receive a passing grade for the course.

c. Each team member must participate in all portions of the term project, including *each* oral presentation.