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:

19 Nov 2019 - Carried forward Intro Slide and Logo requirements from earlier milestone changes.

02 Dec 2019 - Corrected Checklist item for Design Artifacts to match para 3.f

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
- Intro Slide
- Concise Project Overview & Logo
- System Status (% completion of your Starting Backlog)
- Functional Requirement's Trace Table
- Design Artifacts (for all completed acceptance test cases)
- Release Burndown Chart
- Customer meetings summaries and action items with lead mid for each item id'd
- ScrumMaster's Plan (for next sprint)
- Discussion of worst Coupling
- Discussion of worst Cohesion
- Code walkthrough
- McCabe's metric
- Screen shots of system passing each test case the Customer has signed off on for this milestone
- Paper copy of presentation ready to turn in at start of period
- System integrated and ready to demo all test cases Customer has signed off on to date
- Email presentation slides to instructor with subj: Group X, Milestone Z as the subject line

Milestone Deliverables. Paper copies of your slides must be turned in to your instructor *prior* to beginning your oral presentation.

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

2. **Administration.** Same requirement as Milestone 1.

3. **Project Management**.

   a. **Intro Slide.** Same requirements as Milestone 1.

   b. **Concise Project Overview & Logo**. Same requirements as Milestone 2.

   c. **Targeted Requirements.** 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.

   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.
         1. 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.
         2. 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. **Functional Requirements Trace Table**. Present your entire Functional Requirements Trace Table modified as follows:
      i. **Effort Value**: Add an Effort Value column to your trace table giving the effort

valuation that your team has assigned to each acceptance test plan test case. (See table below).

ii. **Meeting Progress Requirements**. As per Agile Development, working (and tested) software that meets the customer's needs is the primary measure of progress. As such, there is a penalty for falling short of the required percentage of backlog completion for any given milestone.
   1. For example, if 60% of the Backlog effort is required to be complete but the team can only demonstrate 55% as being signed off by their Customer as well as demo'd to their instructor when asked, (60-55)/60 = 8.3 points will be deducted from the milestone grade.
   2. These points cannot be recovered even if you later pass the test cases in question to your Customer and instructor's satisfaction.

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 abnormal uses of the system.	**Effort Value**	**Testing Status** – note that each test case has its own status field.	**Customer's Initials -** only for test cases demo'd by the team for which the customer is 100% satisfied (to include non-functional aspects such as the way a user interface looks, etc).
Login/Password GUI: 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 attempts login with the wrong password, and is prevented from logging in.~~ User attempts login with the wrong password, and is prevented from logging in after 3 attempts. (13 Oct: Customer requested that the number of invalid login attempts be limited to 3). (abnormal scenario) 1.2 User with correct password is able to successfully log in to system and is given the User Role associated with the login/password pair. (normal scenario) 1.3 User forgets their password. System emails a temporary password which system requires the user to change at the first login in which the temporary password is used. (abnormal scenario)	1.1 1 1.2 1 1.3 3	1.1 *In progress* (we thought this one was done, but then the customer modified the requirement and told us to lock the user out after 3 tries). 1.2 Completed –IC470 Mile 4 (customer must initial agreeing that they are satisfied with the test case demonstration) 1.3 future milestone (build 2).	1.1 _____ 1.2 WTD 1.3 _____

Figure 1. Functional Requirements Trace Table

f. **Design Artifacts**. Present all of the design artifacts for the *completed* (signed off on by your Customer) acceptance test cases of your project to date, highlighting the parts of the design added or modified for this milestone. Note that neither High-Level Diagrams such as used in your proposal nor UML Use Case diagrams are sufficient for your system design (instead, see the Design Artifacts section below).
   i. You may use any of the artifacts described in "Design Artifacts" available from the course's Resources page.
   ii. Include a title for each design artifact that also identifies the type of design artifact used. Example: "Figure 1. Pseudocode for the FuzzySmart Search Algorithm"
   iii. Include a one or two paragraph discussion of *how* the acceptance test cases completed for this milestone map to your design artifacts.
   iv. **Traceability**. All of your acceptance test cases (except preliminary steps – see IC470 Milestone 3) must be traceable to portions of your design by the time the acceptance test cases are demo'd at a milestone delivery. You should complete these design activities *before* you implement a solution.

g. **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.

h. **Customer Meeting Summaries.** Same requirements as Milestone 1.

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

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 countdown timer elapses, player loses their turn.	Add new requirement to Functional Requirements Trace Table. Adjust Burndown chart to include replotting of Ideal Work remaining line. Adjust Starting 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:**

- Every team member (including the ScrumMaster) must be assigned as a lead developer for at least one acceptance test case in the milestone sprint.
- Team members may need to be assigned as the lead developer to more than one test case (especially ones with low expected "effort" values).
- 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.

4. **Modeling.** N/A for this milestone.

5. **Coding (Implementation)**. Analyze the working portions of the software that <u>your team</u> wrote from the following perspectives:

   a. **Lowest (Worst) Cohesion.** Identify the module in your current implementation that exhibits the worst level of cohesion in your system. Present:
      i. the source code for the module (using a font size large enough to be read),
      ii. explain what the level of cohesion is (give the name, not the number), how you arrived at this evaluation, and
      iii. your rationale for allowing this module to remain in your system. Discuss what alternatives are possible that would improve the level of cohesion.

   b. **Tightest (Worst) Coupling.** Identify the modules in your current implementation that exhibit the worst level of coupling in your system. Present:
      i. the source code for the modules involved (using a font size large enough to be read),
      ii. explain what the level of coupling is (give the name, not the number) and how you arrived at this evaluation, and
      iii. your rationale for allowing the coupling to remain in your system. Discuss what alternatives are possible that would improve the level of coupling.

   c. **Complexity.** Select the **most complex method** (from a McCabe's Metric viewpoint) from your current implementation. With this method as your focus, include the following in your presentation:
      i. **Code Walkthrough**: Conduct a code walkthrough. Make sure that the font size, color, etc allows your code to be readable from the back of the room.
         1. The code you present is to be well commented (such that a person unfamiliar with the source code can follow the control flow logic, etc).
         2. Your code must use descriptive variable naming conventions (ie., numPilots is a descriptive variable name for holding the number of pilots, whereas np or x1 would not be considered suitable).
      ii. **McCabe's Complexity Metric.**
         1. Compute and present McCabe's complexity metric for the method you selected for your Code Walkthrough above.
         2. Present a graph showing the edges and nodes in the module, and specifically how

the M value was computed (ie., clearly indicate the #edges, #nodes, etc).

6. **Testing.** Same requirements as Milestone 4.

Notes:

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