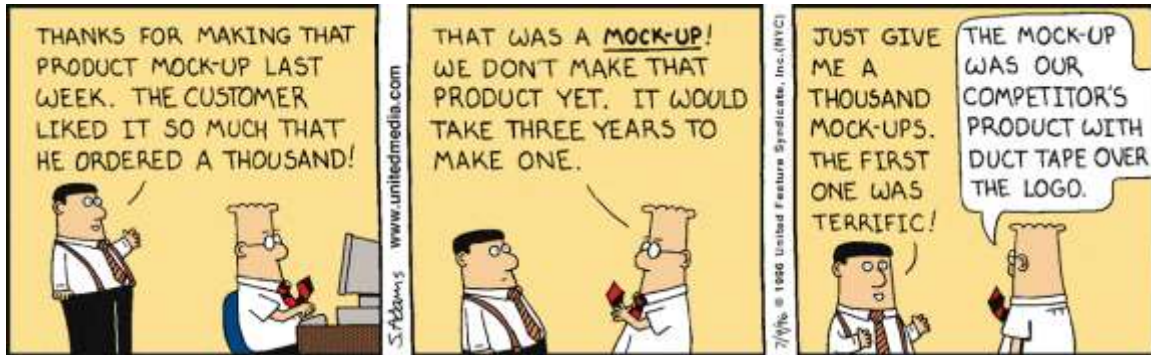


IC470 – Software Engineering

Milestone 2 – Targeted Object Oriented Analysis, Operational Prototype, Software Engineering Code of Ethics

Due date: As as per the syllabus.



Note: Teams are still not bound to what they present for this milestone. For this milestone you should present your best estimate of what your capstone will be based on what you know to date. However, you are free to change focus, Customer or even the entire capstone topic at any point until you submit Milestone 3 - Part I which is the formal proposal and which will be approved (or rejected) by a faculty committee. After Milestone 3, however, teams are committed to the capstone topic unless otherwise directed by your instructor.

Errata/Updates. Any errata or updates to this document will be dated and shown in **red** both as a summary below as well as in the contents of this milestone. Each team is responsible for checking (and delivering their milestone in compliance with) any changes indicated in this Errata/Updates section that are dated prior to the delivery.

Summary of Errata/Updates:

None(**yet**)



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 with checks showing completion of each item)
- Customer's Evaluation Cover Sheet completed by Customer (and Tech Advisor), or, copies of emails showing your attempts to contact them for this at least two days prior
- Project Overview, Logo, High-Level View
- Targeted Functional Requirement for this milestone and justification
- Functional Requirement's Trace Table (just the portion for your targeted functional requirement)
- Risk Management table
- Customer meetings summaries and action items with lead mid for each item id'd
- Project Planning Gantt Chart
- UML Use Case Diagram
- Noun extraction and resultant UML Class Diagram
- Operational Prototype
- Customer feedback on Operational Prototype
- Live demo of at least two acceptance test cases using your operational prototype
- Screen shots of system passing each test case (only use if you have problems during your demo)
- Software Engineering Code of Ethics presentation
- 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).
- If in-person presentation, paper copy of presentation ready to turn in at start of period

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

1. Admin (Administration). Same requirements as milestone 1.

2. Proj Mgmt (Project Management).

a. Project Overview.

1. Present a one paragraph overview of your project updated/expanded as needed to better describe your project as you delve deeper into your project's requirements. Include your team's logo on this slide.
 - i. Your team logo must be graphical, fun (but professional!) and have a tie-in with your project.
 - ii. If you need some help with your logo, jot some ideas down and head over to the Graphics Technology Lab on the first deck of Nimitz Library, Room 105 (x35856, mscgraphics@usna.edu) and contact the Graphics staff for assistance.
2. Present and discuss your High-Level View graphic of your proposed capstone system to include major software/hardware components of what your project will entail. Depending on your system, you may include the high-level view graphic on the overview slide, or, use a separate slide.
 - i. The High-Level View graphic can be a cartoonish image but must use appealing and appropriate graphics to represent systems (ie., a labeled cylinder that represents a database instead of just a box with the label "Database" on it).
 - ii. For example, Figure 1 shows a high-level view of the OCTOPUS Sea-Keeping System.
 - iii. If you need help with the graphical aspects of this, head over to the Graphics Technology Lab on the first deck of Nimitz Library, Room 105, call x35856 or email mscgraphics@usna.edu to contact the Graphics staff for assistance.

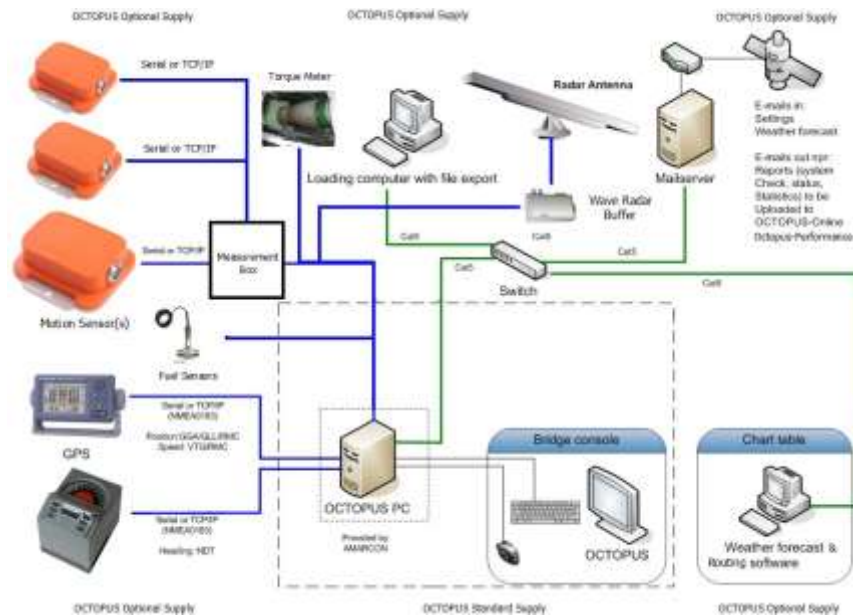


Figure 1. High Level View of the OCTOPUS Sea-Keeping System

- b. **Targeted Functional Requirement.** Present your targeted Functional Requirement. **Customer Involvement Alert:** Meet with your Customer and identify the Functional Requirement for your proposed system that presents your team the greatest risk (see Figure 3 below) and that can best be mitigated by your team developing a working prototype as part of this milestone.
 - i. This will be your Targeted Functional Requirement for the focus of the remainder of this part of the milestone.
 - ii. **Justification.** In your milestone presentation, provide a justification as to why you selected the particular acceptance test plan test case. Why is it so important to develop a working prototype for this test case?
 - iii. Teams picking easy to implement test cases, such as a simple login screen GUI, will be docked points for missing the intent of this portion of the milestone.
- c. **Functional Requirements Trace Table.** Present a Functional Requirements Trace Table (see Milestone 1 and Figure 2 below) that lists all the acceptance test cases related to just the Targeted Functional Requirement for this milestone. For each test case, identify the expected result that allows the test case to be objectively validated, and whether this is a *normal* or *exception* test case.
 - a. Include research or preliminary work you will need to do before you can start designing/implementing your capstone project as Functional Requirements (with a note that says the functional requirement is a “Preliminary Step”).
 - b. For example, foundational or exploratory activities, which we will term “preliminary steps,” such as first demonstrating that you can render a simple line, then rendering a rectangle (which must both be

accomplished as necessary prior steps leading to an eventual acceptance test case of rendering of a complex 3D shape of Bill the Goat) should be included as Functional Requirements with Acceptance Test Plan test cases, even if these preliminary steps won't directly end up in your final product.

Functional Requirement	Acceptance Test Plan test cases (Set of scenarios that, in total, show that the Functional Requirement has been met. Include all normal and exception uses of the system. Clearly distinguish between normal and exception test cases by identifying them as shown below.)
1. <u>Login/Password GUI</u> : Each user must have their own login/password pair that sets their User Role within the system.	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. (<i>normal</i>) 1.2 User attempts login with the wrong password. Expected result -> User is prevented from logging in. (<i>exception</i>) 1.3 User indicates they forgot their password. Expected result -> System emails a temporary password which user is required to change upon logging in. (<i>exception</i>) 1.4 ...
2. ...	2.1 ... 2.2 ...

Figure 2. Functional Requirements Trace Table

d. Risk Management. Present a table similar to Figure 3 that focuses on the risks associated with your Targeted Functional Requirement for this milestone.

- 1. Risk Analysis:** Model your Risk Analysis section after the approach taken by Boehm in "[A Spiral Model of Software Development and Enhancement](#)," *IEEE Computer* 21 (May 1988), pp. 61-72. As specifically as possible, identify at least two real risks that your team faces in the development of your operational prototype (discussed below), and consider ways for mitigating these risks that can be objectively validated. Avoid trivial risks such as "procrastination on the part of the team" and instead focus on the real risks involved in developing your project.
- 2. Risk Management Technique:** Summarize what you plan to do to resolve each risk.
- Figure 3 gives a sample Risk and a Risk Management Technique related to a Functional Requirement that passwords be encrypted.

Risk	Risk Management Technique
Unable to encrypt passwords that are embedded in Java non-serializable objects	Examine encryption techniques and either develop our own encryption algorithm or use an open source solution such as JASYPT (Java Simplified EncrYPTion)

Figure 3. Example Risk and Risk Management Technique

- e. **Customer Meeting Summaries.** Same requirement as Milestone 1.
 - f. **Project Planning Gantt Chart:** Same requirement as Milestone 1, but you will likely need to update your Gantt Chart as you get a better feel for what your project entails. Be sure to breakout the areas of your project that include significant/complex software development.
3. **Modeling.**
- a. **Object-Oriented Analysis** (focused just on your Targeted Functional Requirement).
 - 1. **Use Cases** (focused just on your Targeted Functional Requirement). The UML use case modeling technique aids in ensuring capture of the functional requirements of a system. Use case diagrams describe the interaction between actors (the initiators of the interactions) and the system itself.
 - i. Present use case diagrams for your system that encompass all scenarios of use (both normal and exception) related to just your Targeted Functional Requirement for services directly provided to the actors.
 - ii. Include a brief discussion of each scenario for every use case diagram. Include both normal and exception scenarios.
 - 2. **Class modeling** (focused just on your Targeted Functional Requirement). Give all stages of Noun Extraction for just the subset of your system pertaining to your Targeted Functional Requirement, including:
 - i. **Stage 1** (single sentence description of system that primarily focuses on your Targeted Functional Requirement),
 - ii. **Stage 2** (stage 1 expanded to a single paragraph, including any constraints),
 - iii. **Stage 3** (list all nouns from stage 2; cull these nouns by putting a line through all nouns outside the problem boundary, and circling all abstract nouns. All remaining nouns from Stage 3 of your noun extraction become candidate classes for your UML Class Diagram.

- iv. **UML Class Diagram.** Give the UML class diagram resulting from your Noun Extraction and updated from your examination of the use case scenarios. Include any abstract nouns as class attributes and describe (using correct UML notation) the relationships between the classes.

4. Coding (Implementation).

Operational Prototype - Implement an operational prototype (functioning software) that you can use to demonstrate your system passing at least two of the Acceptance Test Plan Test cases for your Targeted Functional Requirement (expand your scope to a second functional requirement if the one you selected only has one acceptance test case associated with it).

- a. Operational as used here means that the software actually works, and can be used to show (demo) your system passing the acceptance test cases for your Targeted Functional Requirement(s).
- b. **Stubbed-out Supporting Code:** If your targeted functional requirement A, in part relies on some other related functional requirement B that is not yet complete, you must provide a stubbed-out solution for B so that A can be demonstrated to be complete.
- c. For example, if functional requirement A requires that a sorted list of alpha codes is returned to it from a database, but the functional requirement (B) for the database is not yet complete, you must create a code stub in your database controller that simply mimics the necessary behavior of B by returning a static, sorted list of alpha codes in the proper format. This list of alpha codes is then used by the system to demonstrate attainment of functional requirement A.

5. Testing.

- a. **Acceptance Testing** - Test Plan Demonstration using your Operational Prototype: During your milestone delivery, give a live demonstration of how the behavior required by at least one acceptance test plan test case for your Targeted Functional Requirement is fulfilled by your operational prototype. Include a discussion of any Stubbed-out Supporting Code (see the Targeted Functional Requirement discussion in the Proj Mgmt section above) involved with your Targeted Functional Requirement.
- b. **Customer feedback: Customer Involvement Alert** => Prior to delivering this milestone, demo your operational Prototype and acceptance test case for your Customer. Present the feedback from your Customer regarding this demo to include whether your Customer agrees that you have demonstrated completion of the applicable acceptance test cases and any changes they want made.
- c. **Live Demonstration:** During your milestone presentation, clearly show, by running your code, how each of your acceptance test cases is met by your system in operation. As part of your live demonstration show and refer to the relevant portions of your Targeted Functional Requirements

Trace Table while demoing how your software passes the acceptance test case(s).

- d. **Static GUI Screenshots:** Include slides in your presentation that show static GUI screen shots or other evidence of your system in operation and passing of the applicable acceptance test cases. Only present them if you have difficulty running your live demonstration. Note that Control-Alt-PrintScreen will let you copy and paste the active window into PowerPoint.

6. Training.

Software Engineering Code of Ethics: As part of your presentation for this milestone, each team must include a concise presentation based in part on the article "[Software Engineering Code of Ethics is Approved](#)" *Communications of the ACM*, October 1999/Vol 42. No. 10. pp 102-107.

Your ethics presentation must include an example of the particular ethics principles identified for your team below **AND** a discussion of publicly available events (provide the citation for all events) that show a company, organization or individual that violated the Software Engineering Code of Ethics principles you present, and what happened as a result. If your team has been assigned more than one principal in the below, you are free to select your choice from among the assigned principals (you DO NOT have to do more than one principal).

1. Slide support: Use one slide to review your assigned ethics principle and no more than two slides to summarize the violation event you select.
2. Many online software engineering ethics sources exist as references, or you may use your own interpretation of current or past events.
3. You are free to use any properly cited website, paper or news article. Some good sources for ethical issues related to software engineering can be found at <http://onlineethics.org/> as well as at <http://computingcases.org/>
4. You must include a citation or web address for the violation-related-issue you present. For webpages: include the title of the webpage, address, as well as the most recent day you accessed the webpage for the purposes of this milestone.
5. Tailor your presentation of your team's assigned ethical principles to the violation that you cite.
6. See the Milestone Presentation Schedule from the course's [Resources](#) page to determine which particular **Ethics Focus** your team has been assigned to present based on the below table.

Ethics Focus	Principles from the Software Engineering Code of Ethics
A	Principle I. Public, or

	Principle II. Client and Employee
B	Principle III. Product, or Principle IV. Judgment
C	Principle V. Management, or Principle VI. Profession
D	Principle VII. Colleagues
E	Principle VIII. Self

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 and UI screen shots used in the presentation/software demonstration. 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. You will have a maximum of 15 minutes to complete your 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.