

ECE 1140 Deliverables

1. **Work Package 1 – Answers the Question “What are we building?”**
 - a. SRS – Software Requirements Specification follows IEEE 830 format.
 - b. System Architecture – Create block diagram of the architecture with information flow and users that includes all the major modules. Include the Interface Dictionary with inputs, outputs from all sub-modules with description and units of each I/O. This was done in Lab 2
 - c. Personas – description of a user. Use the template provided in Lecture 3 UX Design slides and as done in Lab 3.
 - d. UX design – the wireframes generated in Balsamiq
 - e. Defect Tracking Policy & Workflow – the workflow of defects to be used in your group. It should include the states a defect may have and who is responsible in each of the states, dates of transitions as well as the defect creator.
 - f. Coding Standard – document the standard will you follow for writing code. It should include comments, class names, variable names, etc. A new standard does not need to be created. Use one with which you are familiar.
 - g. Schedule – the Gantt schedule with each of the major task shown, estimated time to complete a task, when the task will start and stop, dependency to other tasks, milestones and the person assigned to the task.
 - h. Risk Assessment – this is a risk assessment of completing the project. It is NOT a risk assessment of the safety of the train control system. Consider the issues that may occur with tools, people, technology and budget, the likelihood of the event occurring, the severity of the event and the mitigation strategy should the event occur.
2. **Work Package 2 – Answers the Question “How are we building the system?”**
 - a. User Stories – written as done in Lab 5
 - b. Software Design Description (SDD) – documented per IEEE 1016. This should include:
 - i. The system Architecture – this is an update from Work Package 1 along with a system level sequence diagram.
 - ii. Key system wide concept of the train control system such as route, authority, distribution of wayside controllers, simulation timing, vitality and wayside language that drive the system software design. What is the design of each of these in the system?
 - iii. UML diagrams for the software design (Use Case, Class, Sequence & Deployment diagrams at a minimum).
 - c. Test Plan – documented per IEEE 829. Should include the specific test to be done meaning what are the specific inputs to be used for each test, what are the specific outputs expected. There should be a template for the outcome of the test, who performed the test and when the test was performed that will be completed during system testing.
 - d. Scrum Boards – use Jira to plan iterations for the project. Going forward either Gantt or Jira can be used (both need not be updated – only one.)
3. **Work Package 3 – The system implementation**
 - a. Project Presentation – Presentation of the working system as a group.
 - b. Code Base – turn in zipped up source code.
 - c. Installation Guide – document how your project or sub-module should be installed such that it can be run.
 - d. Configuration Management – use of Github should be documented and access to your project made available.

- e. Executed Test Plans – execute the test plan written in Work Package 2.
 - f. Defect Reports – provide a sampling of the defects throughout the lifecycle. Track and present the number of bugs / week in the last 8 weeks of the project.
 - g. Video of Final Project – create a video with voice over demonstrating each of the features required in the project.
 - h. Work Schedule Update – Gantt or Jira should be used to track progress and effort of the team project including number of hours spend on each task.
-
- 4. Iteration #1: UX Demonstration – present your UX design for each module and the Persona associated with your module.
 - 5. Design Review – Present a summary of your design. This should include the system Architecture, key system wide design elements of the train control system such as route, authority, distribution of wayside controllers, hardware/software interface and wayside controller language that drive the software design. UML diagrams (Use Case, Class and Sequence diagrams).
 - 6. Iteration #2: Sub-System Demonstration – demonstrate the functionality of your module through a test UI where inputs are provided by the user and outputs are shown. If you find it easier to implement a stub(s) of the sub-system(s) with which you communicate, this is also acceptable. As a first iteration, use the Blue Line. The sub-system functionality should include the full User Interface, Test interface, key functionality and as many of the minor functions implemented as possible.
 - 7. Iteration #3: System Demonstration – at a minimum all the key functionality of each module should be complete, and all the modules should communicate through the defined interfaces. Key focus should be on the Use Case: Dispatcher dispatches a train to Dormont station for arrival at 3 pm. As much of the functionality that can be complete should be complete.
 - 8. Iteration #4: System Demonstration – present the full system working and demonstrate each of the requirements being fulfilled.