

CS 121 Project Timetable

Fall 2001

General Note

- The following table outlines certain checkpoints.
- It does not mean that all work toward the checkpoint is done only in the week between it and the previous checkpoint.
- In other words, **plan ahead.**

As soon as possible

- Decide on a name for your team. Otherwise your name will be the name of your project.
- Decide upon your project and inform me.
- Negotiate specifics with your client.
- Discuss the proposal draft with me.

General Requirements

- All work will be in machine-readable form (no hand-written documents, please).
- You **must** use a version control system (RCS, CVS, etc.) for source and documents and use *make* or equivalent. You will be checked for compliance.
- Contributions to code and documentation should be noted with the individual contributor's names on each document or section.

Mon. 10/29

- Your team's proposal should be complete.
- It will be presented in class.
- The proposal should describe:
 - The product
 - The client
 - The intended users and platform
 - General requirements (not necessarily detailed use-cases)
 - Your team organization
 - Life cycle model you will be using
 - General idea of approach (not detailed schedule)

Establish a web page (Let me know the URL)

- Here you will *maintain* (i.e. keep up-to-date) various written items for the project:
 - Proposal
 - Meeting log
 - Change log (for requirements, plan, etc.)
 - SRS
 - Project plan, schedule, work breakdown structure
 - Design document
 - Traceability matrices
 - Testing plan, with automation
 - Acceptance test

Mon. 11/5

- Your requirements analysis should be complete. You should have a first draft of the schedule and design.
- Present analysis in class.
- The analysis should be self-contained and include:
 - Review of product description and intended users
 - Firm requirements, including detailed use-cases and clarifying scenarios
 - *Draft* of a detailed development schedule, from design through validation and testing
 - *Draft* of work breakdown structure
- All of this must be on your web page.

Mon. 11/12

- Your design should be complete. You may have done some exploratory prototyping.
- Present design in class.
- The design should:
 - Relate to the SRS, especially the use cases.
 - Use UML (class diagrams at a minimum; other diagrams as deemed useful)
 - Take eventual testing into consideration.
- Present implementation work breakdown structure and schedule (with staff assignments and loading diagram).
- All material should be on your web page.

11/13 to 12/9

- You have this time for development work and coding.
- There will be other assignments in this interval, including review presentations.
- Your web page must be kept up to date.

Mon. 12/10

- A good portion of your development work should be done.
- Your team should conduct a **code inspection, including a walkthrough** that goes into depth in one or two key components.
- General coding guidelines are on the next page.
- Also describe how you are going to **test** the product. Prefer automated tests to manual ones.

General Coding Guidelines

- Code for readability (bracketing, identifiers, etc.)
- Comment (purpose, pre- and post-conditions, using a documentation system such as doxygen or javadoc)
- Exploit relevant resources (STL, etc.)
- Document any design patterns used
- Use no “magic” constants
- Avoid cut & paste coding
- Memory management (use consistent style, no space leaks, no hard-wired arrays, etc.)
- Justify globals & static (where need is not obvious)

Mon. 12/19 (Final Presentation)

- This is during the time the final exam is scheduled, 8:00 a.m. - noon.
- We can shift to an evening time if everyone agrees.
- Present overview of the project, actual development compared with plan, shared wisdom, etc.
- Demonstrate of your product
- You have up to 45 mins. to present, but try to keep it to 30 mins.
- All work is due on the web at this time.