

# Semester Project

Tom Kelliher, CS 245

Sept. 6, 2006

## 1 Administrivia

### Announcements

### Assignment

Read Ch. 2. Online essay.

### From Last Time

Why software engineering?

### Outline

1. Teams.
2. Deliverable due dates.
3. Operational aspects.
4. Comments.
5. Initial team meeting.

## Coming Up

Overview of object-oriented paradigm.

## 2 Teams

1. Team Cardinal: Chris, Christine, Justin, Noah.
2. Team BlueJay: Caren, Dan, Dave, Evan, Jon.

Determine your own group name (eight characters or fewer) and let me know. I won't create your Wiki until you inform me of your team name.

## 3 Deliverable Due Dates

Chapter 6 contains a complete example. All deliverables have equal weight, with the exception of source code, which is weighted six times as much as any other deliverable.

1. 9/15: Refined requirements specification (3.4–3.9).
2. 9/20: Scenarios (2.2).
3. 9/17: Primary class list (3.10).
4. 9/27: Class diagrams (3.10).
5. 9/27: Use case diagrams (3.10).
6. 9/29: Structured walk-through (in class) (3.8).
7. 10/4: Object diagrams (5.3).
8. 10/4: Revised class diagrams (5.2, 5.3).
9. 10/11: User interface mock-ups.

10. 10/4: State machines (4.7).
11. 10/11: Collaboration diagrams (5.3).
12. 10/18: Sequence diagrams (5.3).
13. 10/18: Object diagrams (5.3).
14. 10/25: Refined class diagrams (5.2, 5.3).
15. 10/25: Class skeletons (5.2).
16. 10/27: Informal walk-through/design refinement (in class).
17. 10/25: Implementation plan (7.4, 7.5).
18. 11/20: Source code.
19. 11/1: Test plan (8.4, 8.7).
20. 11/27: Test analysis report (8.5).
21. 12/6: System Integration (7.4).
22. 12/8: **Code Freeze** (no more work permitted).
23. Final: System delivery and demo.

### **3.1 Form of Deliverables**

Pages, documents left in team Wiki.

I also expect each of you to create and maintain a work journal in your team's Wiki.

Spelling and grammar matter a great deal. If necessary, use the Writing Center.

## 4 Operational Aspects

1. Don't let technical or personnel problems fester — keep me in the loop.
2. Be conscientious about due dates — there are no back-ups for anyone on a team.
3. If you aren't willing to work as hard as others, be up front about it. The team will have to adjust. I think it would be fair to adjust grades as a result. What do you think?
4. Meet regularly with your team — good communication is a key.
5. Always create an agenda for every team meeting — be as effective as possible.
6. Rotate responsibility for chairing team meetings — give everyone a chance to manage; don't let one person dominate.
7. Technologies for use:
  - (a) Phoenix for server and client development (Java). Individual, group accounts.
  - (b) Wiki for deliverables. Automatic versioning of everything except attachments (?).
  - (c) Subversion repository for code files. Also keep versions of documents here?
  - (d) Eclipse on phoenix for Java development. Subclipse plug-in for access to Subversion repository.
  - (e) Visual Paradigm for UML for diagram production.
8. New technologies ripe for developing small-scale prototyping: Java applications (server), threads (client and server), networking (client and server).

## 5 Comments

- Keep in mind that you will be producing a client/server system. (Feedback from a previous semester.)

- The functional requirements speak for themselves.
- Expect some ambiguities.
- I will serve as the “domain expert” — resolve ambiguities with me.

## 6 Initial Team Meeting

Things to address now or at next meeting:

1. Establish common meeting time.
2. Assign meeting chair and rotation mechanism. Chair responsible for agenda.
3. Assign meeting secretary and rotation mechanism. Secretary responsible for recording and distributing minutes.
4. Discuss tool preferences and areas of expertise.
5. Discuss how to share work products.