**Tracking and Control Mechanisms:**

*Version Control:*

We will use Git for version control. We have a central repository which all the code and documentation is to be checked in and out of. Git is a type of distributed version control, which means that in the configuration we’re using each computer has a full backup of all revision history and code history. GitHub is hosting our code and documentation publically.

*Feature Tracking:*

We will be using Pivotal Tracker, by Pivotal Labs, to manage our sprints. Pivotal Tracker is a feature/ticket/bug-tracking web application built specifically for teams using agile project management methods. We are also using Pivotal as a means to communicate with our client: if he thinks of features outside of client meetings, he can immediately post a feature in Pivotal Tracker’s icebox. We can do the same. During meetings, we will review current items, mark items completed as necessary, then move items from the icebox to the next sprint’s list of items to work on. Finally, we’ll add an estimate of the rough amount of time a feature will take, on a scale of [0,1,2,3,5,8].

Changes will be handled by uploading requests to Pivotal Tracker, then discussing them during our weekly meetings. We believe strongly that all changes have a cost, and therefore that they should be tracked just as all other features are tracked.

Once we finish the basic port, our software will have ample flexibility in accommodating additional features. We will discuss specific features to implement as Winter quarter begins and as we finalize the port. Both we and our client will add features to our backlog as we think of them. During meetings, we will prioritize features and discuss what might be suitable for implementation during the upcoming week. This flexible cycle allows us to avoid locking ourselves into specific design decisions until they actually need to be made.

Both we and our clients will want to receive extensive feedback from in-field tests before we prioritize certain additional features to add. Our scope isn’t completely defined, so we’ll be able to put much more detail into this as the course progresses.

**Basic Schedule:**

**FALL**

**Week 7:** Project Plan

Our project plan consists

**Week 8:** Finish requirements and problem statement documents.

**Week 9:** Investigate and understand algorithms being used by the current system. Finish initial architecture document.

**Week 10:** Finalize all fall documents for Dr. Bohner. This includes a metrics document, a requirements document, and an initial architecture document.

**Week 10:** Verify resources are set up to begin iOS development.

**WINTER**

**Week 1:** New team member joins. Brief him on the project and get his computer up to speed.

**Week 2:** Technical prototype of a port to iOS. Allow dragging of points and basic statistics.

**Week 3:** Revised architecture / design document. Include initial version of developer’s guide.

**Week 4:** Bring image recognition onboard. Test port for functionality in the field.

**Week 5:** Investigate additions for iOS. Finish debugging port if necessary.

**Week 5:** Design Review Meeting

**Week 5**: Midterm Documents Due

**Week 9:** Investigate basic iOS additions

**Week 10:** Final Winter Documents Due. This includes an architecture document and a developer’s guide. Get feedback for basic iOS additions.

**Week 10:** End of Term Expo

**SPRING**

**Week 1:** Prototype basic iOS additions, continue works

**Week 2:** Midterm Documents including support documents and maintenance plan. Get feedback for upgraded iOS additions.

**Week 3:** Document & System Audit

**Week 4:** Deliver final documents, if at all possible. This is right before Spring Break: we feel that it’s wise to schedule the project to “end” before Spring Break in case we encounter unforeseen issues.

**Week 5:** Deliver upgraded iOS additions to client.

**Week 6:** Perform maintenance and verify that everything is in order. Deliver additional small improvements as needed.

**Week 7:** Deliver final system. This means that the code will be in a maintainable and well-documented state in addition to working, if it wasn’t already in this state.

**Week 10:** Deliver final documents, if at all possible. This is right before Spring Break: we feel that it’s wise to schedule the project to “end” before Spring Break in case we encounter unforeseen issues.

**Week 10:** Project Retrospectives