COGS 121
HCI Programming Studio
Spring 2016

Instructor: Nadir Weibel
Website: cogs121.ucsd.edu

Amy Rae Fox  Jesse Qin  Jasmine Roberts  Andrew Du  Brian Soe  Jacob Browne  Jingchun Zhou
Project Management

what it is (and isn’t!)
understanding the SDLC
making a productive team
agile vs. waterfall methods
the role of design
tips and tools
What is a Project?
a temporary endeavor undertaken to create a unique product, service or result
What is Project Management?
The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.
Project Management

• a discipline and profession

• a set of practices that anyone can (and should!) apply

• governed by the Project Management Institute, which sets standards and administers certifications

• knowledge codified into the PMBOK: Project Management Book of Knowledge
Project Management

• management of resources & constraints to meet a goal as efficiently as possible
  – Resources: Time, money, people, equipment
  – Constraints: preceding task completion
• a science or an art?
  – Science: based on statistical means & norms
  – Art: based on intuition into human behavior

*Balancing the known and unknown in a chaotic and risk-filled environment in order to achieve a more predictable result*
The Emergence of Standardized PM

- Organized as a formal Discipline in 1917
- Henry Gantt introduced standardized PM tools
- Gantt Chart – visual tracking of tasks and resources, including relationships between tasks
- Created out of need and frustration as industrialization became ever more complex
- Little change to PM for another 40 years
Project Management

5 Processes
- Initiating
- Planning
- Executing
- Monitoring and Controlling
- Closing

10 Knowledge Areas
- Scope
- Time
- Cost
- Quality
- Human Resources
- Communication
- Risk
- Procurement
- Stakeholders
- Integration
What’s most relevant to us?

- develop project plan
- manage project work
- collect requirements & define scope
- perform quality assurance
- manage project team

Table 3-1. Project Management Process Group and Knowledge Area Mapping

A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fifth Edition. ©2013 Project Management Institute, Inc. All rights reserved.
Project Management in Software Development

Managing the SDLC  
*Systems Development Lifecycle*

“Waterfall”

- Plan
- Analyze
- Design
- Build
- Test
- Deliver
Enter agile...
The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over **processes and tools**
- **Working software** over **comprehensive documentation**
- **Customer collaboration** over **contract negotiation**
- **Responding to change** over **following a plan**

That is, while there is value in the items on the **right**, we value the items on the **left** more.

Manifesto for Agile Software Development, 2001
Agile Project Management

• Premise: software projects are unpredictable and market uncertainty is going to drive change.

• Requirements will need to change over the life of the project, and the more uncertain the project is, the more the organization should plan to adapt.

• Project requirements should be written as thin vertical slices of the overall system and constructed in such a way that they are mostly independent, which allows them to be prioritized and implemented in any order.
Agile Project Management

**IS**
- A philosophy
- A set of practices and methods
- Responsive to change
- Focused on development

**IS NOT**
- Better for all projects
- An excuse to not plan
- Build first, design later
- An excuse to not document

[LEAN Software Development](#)
[Scrum](#)
[XP Extreme Programming](#)
and others…
“Agile”

Plan → Analyze → Plan → Design → Build → Test → Deliver

“Waterfall”

Plan → Analyze → Design → Build → Test → Deliver
Concepts

Requirements the “needs” the system needs to fulfill (functional, non-functional)

Designs decisions on how to meet requirements

Activities units of work

Features implementation of designs to meet requirements

Milestones checkpoints in a process

Deliverables tangible outcomes delivered to the customer
Tools

Gantt Chart: A bar chart. While visually appealing on a task/duration basis, it is limited because it does not show task or resource relationships well. Strength: easy to maintain and read.
Tools

Network Diagram: A wire diagram, Also known as a PERT network diagram. A diagram that shows tasks and their relationships. Limited because it shows only task relationships. Strength: easy to read task relationships.

![Network Diagram Image]
Work with the client to determine the project needs & constraints (ANALYZE)
Define project milestones and deliverables (PLAN)

while project has not been completed or cancelled (EXECUTE)
  Draw up project schedule
  Initiate activities according to schedule
  Wait ( for a while )
  Review project progress
  Revise estimates of project parameters
  Update the project schedule
  Re-negotiate project constraints and deliverables
if ( problems arise ) then
  Initiate technical review and possible revision
end if
end loop
Close project (DELIVER)
An example... 

Writing a research paper
**Requirements:** defined by the assignment, and my expectations
—> I will list these in a **requirements backlog**

**Designs** decisions on how to meet requirements
—> I will make a **proposal** so I can remember and get approval from my professor

**Activities** work I need to do to meet the requirements
—> I will organize the work into logical units — a **work breakdown structure (WBS)**

**Features** the contents and qualities of the paper
—> I will actually **do the work** :-)

**Milestones** the checkpoints to keep myself on track
—> I’ll define these in the **WBS** and schedule them in the **project plan**

**Deliverables** tangible outcomes delivered to the customer
—> I’ll define these in the **WBS** and schedule them in the **project plan**
1: Requirements Definition

\{product goals\}

- 20 pages
- Double spaced
- On a topic addressing a question of the effectiveness of agile and waterfall methods
- Includes a literature review
- Includes a proposal for a research study
- Includes hypotheses & expected results
- IEEE citation format
- Reference at least 10 peer-reviewed papers
2: Work Breakdown Structure

{logical units of work to accomplish goals}

1. Planning
   A. Pick topic & research question
   B. Brainstorm potential research studies
   C. Make list of papers to read
   D. Document A-C in a proposal
   E. Discuss proposal with professor

2. Researching
   F. Read research papers
   G. Document key ideas

3. Writing
   H. Outline paper
   I. Write first draft
   J. Discuss draft with professor

4. Editing & Polishing
   K. Revise draft
   L. Check references and citation format
   M. Check length and formatting
   N. Proofread
   O. Submit paper

milestone

deliverable

milestone

deliverable

milestone

deliverable
3: Project Plan ... part of it

activity network diagram to find activity dependencies
3: Project Plan ... part of it

gantt chart to estimate time and schedule with dependencies

can also reflect assigned “resources” (people), and to see their work allocations (how many hours they are assigned)
Project Management Tools

- Trello
- Basecamp
- Jira
- Asana
- Github + ZenHub
- Tom’s Planner
- Ganttter
- Github + Zenhub
Trello

- Highly visual
- Flat and simple
- Boards, Lists and Cards
- mobile apps available
Basecamp

- More structure and PM tools
- to-dos with deadlines
- project templates
- time tracking
- invoicing tools
- file backup a
- synchronizing tools
- software development tools.
- Mobile

https://basecamp.com
Jira

- Project Tracking
- Agile, Development support
- Project Planning, Issue Tracking
- Code Integration
- Mobile
- Connect to LDAP and Active Directory
- Bug Tracking
- Git Integration
- 1000's of Add-ons
- OnDemand or Hosted
- Free for Open source projects
- eMail Notifications
Gantter

- Web-based, can add as chrome extension
- EASY to create Work Breakdown Structure
- Gantt chart with multiple people and external resources
- Google Drive integration
- Export capabilities
- FREE
Github + Zenhub

- Github for free code repository and source control
- Zen hub (FREE Chrome extension) adds kanban-style boards features to Github issues
Cloud-based “connectors”

- Zapier: https://zapier.com/
- Cloudwork: https://cloudwork.com/
- IFTTT: If this then that: https://ifttt.com/
Amy’s Personal Recommendation

**Trello**
For capturing requirements and sorting them into priorities

**Gantter**
Turning requirements into a WBS and scheduling w/ dependencies

**Github + Zenhub**
Source control + feature tracking linked to commits

*FREE cloud based enterprise project scheduling tool*
Amy’s Advice

- Use deliverable and planning templates
- Use a system like Google Docs or Slack to record & document your team meetings, with special attention to **action items** and **decisions**
- Take the time to make a Gantt chart, and keep it updated based on your progress
- Keep your project plan fairly simple (not too detailed at the activity level), but set clear milestones (with dates!) and stick to them
- Learn the lingo... it may help you get a job!
References


Want to learn more about PM practices, certification, templates? See Amy at office hours - before class on Thursdays
1. Customer satisfaction by early and continuous delivery of valuable software
2. Welcome changing requirements, even in late development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly