Software Project Management

-or-

there's more to developing apps than coding

thomas.crichlow@duke.edu

Digital Projects Program Coordinator
Perkins Library
What will I cover tonight?

• Requirements
• Project Planning/Management
• Look at a real project from Perkins Library
• Discussion
Where to start?

• Start with a concept:
  – client says they want you to do something
  – or, you think of something cool you want to do
• Is it Feasible?
  – Benefits and risks
  – Compare to other options
  – Resources available
    • people | technical skills | time
    • software | hardware
  – Will it really solve the problem?
Concept+\((\text{feasible=yes})\neq \text{product}\)

- What's needed now?
  - Requirements = details about what you will build
  - Clear plan = Who is doing what and when
  - Careful management of the plan = keeping things on track
Requirements

- Identify stakeholders
- Determine needs
- Analyze the requirements
- Record the requirements
  - Measurable goals
- Communicate
- Revisit and Revise
Requirements - Example

• Let's look at one tool for collecting requirements:
  – See “SRS-Software Requirements Specification Template.doc” for more information
  – This is from RedRockResearch.com
    • http://www.redrockresearch.com/frmUtilities.aspx
Project Planning / Management

- Team = Who
- Milestones = What
- Timeline = When
- Communication = Essential
- Leader = Coordinates items listed above
The Team

• Skills
• Personalities
• Availability vs. other duties
• Roles
• Location
Skill sets and personalities

• Consider skill sets and personalities when selecting and working with teams
• What do people already know?
• What can people learn?
• Ability to work together
• Ability to work directly with a client
Milestones

• What are your deliverables?
• Smaller = Better
• Plan backwards: End ---> Beginning
• Identify pre-requisites/dependencies
Timelines

• When things get done
• Plan backwards: End ---&gt; Beginning
• Estimate of time needed
• Check your estimates with someone with more experience
Communication

• Know thyself, understand your communication style
  – Consider how others listen to you
  – Modify your style to meet the needs of your audience
  – Myers-Briggs can help you understand

• Listen
  – To your boss, team members, clients
  – You can learn from them
Let’s be clear

• Don’t leave people in the dark
• Communicate clearly
• Communicate often
• Pause before clicking “submit”
• Edit your message for clarity
• Be succinct
• Don’t make your audience struggle to decipher your message
Team leadership Tips

• Share direction, give permission, provide support. Commit.
• Understand the range of outcomes. Seek the one you want
• Be interested in being wrong.
• If everyone agrees, go back to the drawing board.
• Don't mitigate risk. Mitigate the cost of failure

Managing time

• Delegate
• 30 minute meetings
• Still shifting gears?
• Address your time sinks
Manage up / Manage clients

• You aren't limited to managing members of your team
  – You can manage up -- to your boss
  – You can manage clients

• Be a mentor. Be clear. Be respectful.
Examples of Project Planning/Management

• Two common approaches to managing software projects are:
  – Waterfall
    • A top-down style derived from older, manufacturing-based approaches.
  – Agile
    • An approach that uses self-organizing and cross-functional teams
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.

  – http://agilemanifesto.org/
  – See also: http://agilemanifesto.org/principles.html
Scrum: one method of Agile Development

- **People**: Scrum Master, Team, Product Owner, Stakeholders, Managers
- **Product backlog**: what will be built sorted by importance
- **Sprint backlog**: to do list for current sprint
- **Burn down**: list or chart of remaining work
- **Meetings**:
  - Daily scrum
  - Sprint planning, review, and retrospective
Scrum-related example

• An example of a tool that you can use with a Scrum-related development effort:
  – See “Agile - Project Release Plan Template.doc” for more information.
  – This is from RedRockResearch.com
    • http://www.redrockresearch.com/frmUtilities.aspx
Real Life Example

• Tripod2: project to build a web interface to enable library patrons to easily discover, understand, and use digitized objects from library collections. It is built on three main components:
  – Django-based interface
  – Django-based Metadata Webservices layer
  – Solr-based index

• See Tripod-wiki.pdf for more information.
Discussion

• What do you want to know more about?
• What do you want to talk about?