Software Project Management

CITS1220 Software Engineering
Lecture Overview

- What is software project management?
- Software development lifecycles
- People: Skills and responsibilities
- Heads up: difficulties and risks in SW PM
Project management is

- the discipline of defining and achieving **targets** while optimizing the use of **resources** over the course of a **project** (a set of activities of finite duration)
Project management activities

- Deciding what needs to be done
- Estimating costs
- Ensuring there are suitable people to undertake the project
- Defining responsibilities
- Scheduling
- Making arrangements for the work
- Doing the work!
Some important terms

- **Process (n)**
  - a series of actions or operations designed to achieve an end

- **Product (n)**
  - something produced by a natural or artificial process
  - a saleable or marketable commodity

- **Lifecycle (n)**
  - the series of stages in form and functional activity through which an organism passes during its lifetime
Lecture Overview

- What is software project management?
- **Software development lifecycles**
- People: Skills and responsibilities
- Heads up: difficulties and risks in SW PM
Software Development Lifecycle (SDLC)

- A model for the process of developing SW
- Three standard models (you should know)
  - opportunistic (aka chaotic)
  - waterfall
  - iterative (agile)
- All SDLCs involve steps for requirements, design, implementation and quality assurance
Chaotic (opportunistic) SDLC

First Prototype → Modify Until Satisfied → Think of Idea for Improvement

First Prototype → Modify Until Satisfied

First Prototype → Think of Idea for Improvement
Chaotic approach (not recommended)

- Does not acknowledge the importance of working out the requirements and the design before implementing a system
- Since there are no plans, there is nothing to aim towards, or measure against
- No explicit recognition of the need for systematic testing or other forms of quality assurance.
- Leads to … very high cost of developing and maintaining software with the opportunistic approach
Waterfall Model
Waterfall model limitations

- The model implies that you should attempt to complete a given stage before moving on to the next stage
  - Does not account for the fact that requirements constantly change.
  - It also means that customers cannot use anything until the entire system is complete.
- The model makes no allowances for prototyping.
- It implies that you can get the requirements right by simply writing them down and reviewing them.
- The model implies that once the product is finished, everything else is maintenance.
Agile Programming [Cockburn et al 2002]

“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.”

Source: http://www.agilealliance.org/home
Agile

- Active stakeholder participation
- Obtain funding and support
- Start building the team
- Initial requirements modeling
- Initial architecture modeling
- Setup environment

- Active stakeholder participation
- Collaborative development
- Model storming
- Test driven design (TDD)
- Confirmatory testing
- Investigative testing
- Evolve documentation
- Internally deploy software

- Active stakeholder participation
- Final system testing
- Final acceptance testing
- Finalize documentation
- Pilot test the release
- Train end users
- Train production staff
- Deploy system into production

- Operate system
- Support system
- Identify defects and enhancements

Start work on release N+1

Copyright 2006 Scott W. Ambler
Underlying Assumptions for Agile SW Development

- “Different projects need different processes or methodologies”
- “Focussing on skills, communication and community allows the project to be more effective and more agile than focussing on process”

Agile Development Techniques

- **Test Driven Development** – because each iteration must deliver working software it must work.
- **Short iterations** – from 2 weeks to 2 months.
- Strong **customer** involvement
- **Adaptable process** allowing for requirements to change
Lecture Overview

- What is software project management?
- Software development lifecycles
- **People: Skills and responsibilities**
- Heads up: difficulties and risks in SW PM
Building SE Teams

- Software engineering is a human process.
- Choosing appropriate people for a team, and assigning roles and responsibilities to the team members is, therefore, an important project management skill
Motivation for Groups

Why do we need groups in software engineering?

- most products are too large for 1 person
- one person can not be proficient in all the information technology options that exist today
What is the best team size?

- **3 members**
  - ✓ everyone has a chance to speak
  - ✗ one member can dominate
  - ✗ too many hats for each member

- **4 members**
  - ✓ works even if someone drops out
  - ✗ ties when voting (ancient Greek prob)
5 or 6 members is ideal

- face to face meetings good
- diverse perspectives for creative thinking
- specialisation of one role per member

7 or 8 members

- longer meetings
- coalitions and competition can arise
Roles

- Role: a set of responsibilities in the project assigned to a person or team
- One person can fill one or more roles
  - Architect
  - Project manager
  - Configuration management and build specialist
  - User interface specialist
  - Technology specialist
  - Hardware and third-party software specialist
  - User documentation specialist
  - Tester
Identifying Individuals’ Skills

- Application domain skills
- Communication skills
  - e.g. negotiation, communication with non-experts
- Technical skills
  - knowledge of specific technology and tools, programming skills, design risk assessment skills, ability to find workarounds
- Quality skills
  - attention to detail, boundary case identification, follow procedures
- Management skills
  - assess personal skills, motivate, feedback, assess project risks
Personality: Belbin Team Roles

- action-oriented roles
  - Shaper, Implementer, and Completer Finisher

- people-oriented roles
  - Co-ordinator, Teamworker and Resource Investigator

- cerebral roles
  - Plant, Monitor Evaluator and Specialist

- http://www.belbin.com/
Lego Project: practical SE teamwork

- Agree your goals for the project
- Identify each team member’s strengths and work preferences
- Agree team rules (time, contact, respect)

- Identify tasks and responsibilities
- Work in short iterations (agree milestones for 1 week’s time)
Lecture Overview

- What is software project management?
- Software development lifecycles
- People: Skills and responsibilities
- Heads up: difficulties and risks in SW project management
Some Anecdotal Observations about new SE Graduates

- Graduates have high technical competence *but*,
  - Are not good team players,
  - Have very little feel or interest for company goals,
  - Tend to be over-confident, over-optimistic as regards project effort estimation and late in project delivery,
  - Tend to adopt new technologies recklessly, with little or no risk and cost assessment,
  - Prefer to start, or restart, development efforts from scratch, especially if they had not participated in the earlier development efforts
Difficulties and Risks in Project Management (1)

It is very difficult to measure progress and meet deadlines

- Improve your cost estimation skills so as to account for the kinds of problems that may occur.
- Develop a closer relationship with other members of the team.
- Be realistic in initial requirements gathering, and follow an iterative approach.
Difficulties and Risks in Project Management (2)

It is difficult to deal with lack of human resources or technology needed to successfully run a project.

- When determining the requirements and the project plan, take into consideration the resources available.
- If you cannot find skilled people or suitable technology then you must limit the scope of your project.
Difficulties and Risks in Project Management (3)

Communicating effectively is hard

- Take courses in communication, both written and oral.
- Learn how to run effective meetings.
- Review what information everybody should have, and make sure they have it.
- Make sure that project information is readily available.
- Use ‘groupware’ technology to help people exchange the information they need to know.
Difficulties and Risks in Project Management (4)

It is hard to obtain agreement and commitment from others

- Take courses in negotiating skills and leadership.
- Ensure that everybody understands
  - The position of everybody else.
  - The costs and benefits of each alternative.
  - The rationale behind any compromises.
- Ensure that everybody’s proposed responsibility is clearly expressed.
- Listen to everybody’s opinion, but take assertive action, when needed, to ensure progress occurs.