The Role of Software Engineering

Brief overview of relationship of SE to managing DSD risks

Outline

• Role of software engineering
  – Purpose of software engineering
  – Software engineering areas addressing DSD risks
• Teams and Project Planning
• To Do List

Objectives

• What distinguishes DSD
  “The key phenomenon of DSD is coordination over distance.” – J. Herbsleb (2007)

• Overall goal is to produce the right software on time with available resources
• Implies the need to coordinate the work among distributed team members such that
  – Every task gets done
  – Tasks are assigned to the right people
  – Tasks are done in a right order
  – Work loads are balanced

Purpose of SW Engineering

• Defn: The purpose of software engineering is to gain and maintain intellectual and managerial control over the products and processes of software development
  – Intellectual control means that we are able make rational technical choices based on an understanding of the downstream effects of those choices
  – Managerial control means we are able to make rational choices about development resources (budget, schedule, personnel) to deliver software on time and within budget
Meaning of Intellectual Control

- Software development progresses through a sequence of decisions
  - Requirements decisions
  - Architectural design decisions
  - Detail design decisions, coding decisions, etc.
- Decision order matters
  - Ending up with the desired properties requires making the right decisions in the right order
  - E.g., cannot add properties like security late in the game (Windows vs. OSX)
- Being in intellectual control means we systematically build the right software:
  - Decide in advance the behavioral and quality requirements
  - Proceed systematically to produce a system meeting those requirements
  - In practice, a feedback-control loop

Meaning of Managerial Control

- Managerial control means we are able to make rational choices about development resources
  - Real projects have finite set of resources: time, man hours, skills, money
  - Must choose where, when, and how much of each resource is allocated
- Being in control means we can:
  - Effectively estimate the level of resources needed
  - Effectively allocate resources to tasks over time
  - Adjust resource allocation as needed
  - Such that the right software is delivered on time and within budget

DSD Risk Drivers

- Risk
  - Defn: event or situation leading to a loss of control
  - Risk Mitigation: strategies for avoiding risks or minimizing their effects
- Example: losing traction due to icy road conditions in winter mountains
  - Mitigations: snow tires, studded tires, chains
  - Avoidance: drive around, wait, fly
- DSD communication difficulties and context differences lead to coordination and control problems
- What kinds of risks does this present?

Examples of Risks?

Goal: keep business goals and system capabilities in synch

What kinds of risks are aggravated by DSD?
Example: Requirements Goals and Risks

- Requirements goals: negotiate and agree on a common set of requirements among distributed stakeholders
- Requirements risks in DSD
  - Many different stakeholders with different goals and different understanding
  - Risk of building the wrong system
  - Risk of requirements changing during development
- Risk mitigation strategies
  - Methods for eliciting requirements, negotiating among stakeholders where conflicts exist
  - Methods for communicating and validating requirements with stakeholders
  - Explicit processes for managing change throughout development

How should components be distributed among teams?

Goal: distribution of work on components requiring least inter-team communication

Q: What properties should the components have?

Example: Design Risks and Goals

- Relationship between system design and communication overhead
  - Must decompose the system into work assignments for distributed teams
  - System components depend on one another
- Risk: greater dependencies require more communication and coordination
- Risk mitigation
  - Decompose the system into work assignments that are as independent as possible
  - Work can proceed concurrently and independently
  - Little need for communication
  - Little need for one team to wait for another

Summary

- DSD presents certain specific problems that make it difficult to control software development
- Purpose of software engineering is to provide technology and techniques for maintaining control
- We will cover SE theory and methodology that address key DSD problems
  - Quality Assurance: testing and reviews
  - Architectural design
  - Interface specification
Project Considerations

- Have ~20 UO students, 12 PKU students
- PKU students are proposing to develop Windows 8 Phone apps, but can negotiate
  - C#, Xna, Silverlight
- Different from the past where we have pre-defined the application
  - Opportunity: opportunity to choose application, define requirements as a team
  - Risk: need to converge quickly to allow sufficient development time
- Interests? Skills available? Team constraints?

Example Project from 2011: Notable Faces Service (NFS)

Proposal

- We propose a system to recognize and retrieve information about individuals from their pictures.
- To keep the system portable and appealing to the on-the-go user
  - The system is implemented on a smartphone device such as the iPhone and/or Android
  - The picture may be taken then and there
  - May supply context-relevant information (allow different data-bases to be accessed)

Basic Application

1. Take a picture of John Smith
2. The system searches a pre-established data-base of faces
3. If recognized, retrieves relevant information
**Architectural Concept**

- Take advantage of existing web services
- Support a family of possible applications

**Anatomy of a Good Project**

- Supports independent development
  - Clear division into work assignments
  - Few dependencies
- Complex enough to support expansion/contraction (useful subsets, increments)
- Complex enough to require some effort in all the parts of development (requirements, design, testing, etc.)
- Simple enough to finish in ten weeks

**To Do**

- First team meeting (this week)
  - Meet with UO team members
  - Set up meeting with PKU team members
  - Invite PKU team members to assembla
  - Create team page on assembla space
- UO liaison contact PKU liaison
- Discuss possible project

**Questions?**