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
• Effective meetings revisited
• Teams and Project Planning
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 (e.g., tradeoffs, priority)
  - Architectural design decisions
  - Detail design decisions, coding decisions, etc.
- Earlier decisions affect the difficulty of later decisions
  - 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 demonstrates this vs. OSX (Unix private address space)
- 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, people, 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
  - Effectively adjust resource allocation on the fly
  - 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: project strategies for avoiding 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?
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 requirements change throughout development
    - Negotiating, communicating to distributed stakeholders

How should components be distributed among teams?

Goal: distribution of work on components requiring least inter-team communication
Q: Properties should the components have?

© S. Faulk 2012
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
- Software architecture goals
  - 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
- How do we design the software architecture to have these properties?

Verification and Validation

- How could V&V activities help mitigate DSD risks?
- What might this suggest about how these activities should be organized?
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 technique that address key DSD problems
  – Quality Assurance: testing and reviews
  – Architectural design

Effective Meetings

A Cross-Cultural View
Notes on effective meetings

• Only hold meetings if necessary
  – “Necessary” means that the most cost effective way to accomplish a goal is by meeting
• Have a goal, and a plan (agenda)
  – Clear meeting objectives
  – Known to all in advance (i.e., distribute via email)
• Plan to goal:
  – Participants - Invite only the necessary people
  – Schedule
  – Intended outcome
• Prepare
  – Cost of wasted time = Time x people x hourly cost
  – Cost of individual prep time is much less

Notes on effective meetings (2)

• Start on time, end on time
• Write down and disseminate the results
  – Leaves an audit trail of decisions
  – Makes people feel included
  – Limits the number of (informational) invitees
• End with concrete, specific action items
  – What must be done
  – Who should do it
  – What the follow-up is
• Reflect items in the schedule and developer logs
Cross-Cultural View

• From what (or whose) perspective would this kind of meeting be “effective?”
  – What does it assume about the purpose of a meeting?
  – What does it imply about culture?
  – What might we want to do differently in a cross-cultural meeting?
• “In other countries, entire meetings are devoted to establishing relationships, without conducting the core of the task at all.” [Olson & Olson]

Distributed Team Meetings

• What issues are likely to arise in distributed team meetings?
  – Time and schedule (e.g., exams) differences
  – Language? Which communication medium works best?
  – Decision making: what can we expect in terms of people speaking up and speaking their minds?
    • They said “yes” so we have an agreement, right?
    • No one objected so we can proceed.
• What might we do to mitigate help ensure effective communication?
To Do

• First team meeting (this week if possible)
  – Kean: have four teams, should be ready to go.
  – PKU: will speak to professor this evening
    • Hopefully will have contact information this evening

• UO liaison contact PKU liaison

Questions?
Analogy to Driving a Car

- Driving a car
  - In control: decide in advance where we want to go, how long, it should take, how much fuel it will take
  - Out of control: end up at a different destination, takes twice as long as expected, uses twice as much fuel, etc.

- Managing a software project
  - In control: decide in advance what capabilities and properties the software will have, how long it will take, how much effort
  - Out of control: software is delivered with less or wrong capabilities, delivered late, over budget

- Many software projects are out of control in this sense
  - Perfect control is not possible
  - Requires constant feedback and correction