

## CSE 403 Lecture 8

### Risk assessment

## Lecture goals

- Understand risk management and assessment techniques
- Guarding against failure to meet delivery deadline, resource constraints, or quality threshold for a software project

## Classic Mistakes

- McConnell, *Rapid Development*
  - People related mistakes
  - Process related mistakes
  - Product related mistakes
  - Technology related mistakes

## Process issues (high level)

- Accurate planning
  - Realistic scheduling
  - Contingency planning
- Paying attention to all stages of product development

## Process related mistakes

- Optimistic schedules
- Insufficient risk management
- Contractor failure
- Insufficient planning
- Abandonment of planning under pressure
- Wasted time in "fuzzy front end"
- Shortchanged upstream activities
- Inadequate design
- Shortchanged QA
- Insufficient management controls
- Premature convergence
- Omitting necessary tasks from estimates
- Planning to catch up later
- Code-like-hell programming

## Product related mistakes

- Requirements gold-plating
- Feature creep
- Developer gold-plating
- Push-me, pull-me negotiation
  - Adding new tasks when schedule slips
- Research-oriented development

## Technology related mistakes

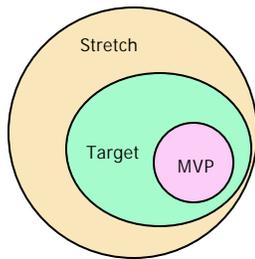
- Silver-bullet syndrome
- Overestimating savings from new tools or methods
- Switching tools in the middle of a project
- Lack of automated source-code control
- **Developing with inconsistent versions of software**

## Questions

- To what extent are these problems specific to software projects?
- Are there characteristics of software projects that make them more likely to occur?
- Why do people make the same dumb mistakes over and over again?

## Product delivery

- Minimum Viable Product (MVP)
  - Won't ship without meeting this
- Target delivery
- Stretch goals



## Other things that go wrong

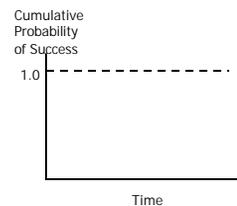
- Even if you don't make mistakes, problems arise
- Availability of people
- Dependency on external technology
  - Integration
  - Functionality
  - Instability
  - Delivery
- External constraints
- Acts of management

## Risk

- Exposure to the chance of injury or loss
- In the software project context:
  - Risks in activities
    - Exceeding expected resource requirements for activity
    - Not being able to complete activity (so that enclosing activities exceed resource requirements)

## Probability distributions on completion time

- Write the data entry module
- Hire a tester
- Fix a simple bug
  - Menu item not checked when operation selected
- Fix a challenging bug
  - Intermittent problem with thread logic



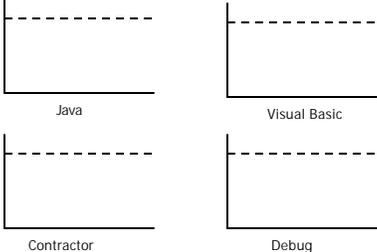


### Sketch completion time graphs for the following

- Write a tic-tac-toe program in Java
- Write a tic-tac-toe program in Visual Basic
- Hire a contractor and have the contractor write a tic-tac-toe program in Visual Basic
- Debug some else's tic-tac-toe program



### Completion graphs




### Sources of Risk I

- Development risks
  - Code harder to develop than thought
  - Learning curve on new facilities
  - Expected facilities not available
  - Need to iterate on requirements / design
  - Performance Issues
  - Trigger other bugs



### Sources of Risk II

- Integration risks
  - Parts don't fit together
  - Integration reveals bugs
  - Integration reveals design errors
  - Need to rewrite code after integration
  - Code left out



### Sources of Risk III

- Testing risks
  - Bugs will be found
  - Bugs won't be found
  - Complexity of testing matrix
  - Deployment beyond development machines
  - Difficulties in test automation and test tools
  - UI and Workflow feedback



### Sources of Risk IV

- Deployment Risks
  - Packaging distributable
  - Rights and licensing of components
  - Legal signoff
  - Marketing signoff
  - Systems configuration



## Risk Analysis

- Identify top schedule risks to project
- Risk exposure
  - Size of Loss X Probability of Loss
  - (Better might be Expected loss)
  - Guess two numbers and multiply them
  - Value in *attempting* to quantify
- Prioritize based upon exposure



## What to do with risk analysis

- Avoid the risk
- Transfer risk off the critical path
- Buy information
  - Bring in outside help
  - Prototype
- Publicize risk
  - The sky is falling
- Schedule to accommodate some risk
- Monitor risks as project progresses