

Software Development Risks

- *Situation involving exposure to danger –*
Apple dictionary
- McConnell
- Schedule, cost, product

Levels

- Reactive; crisis management
- Fix on failure
- Mitigation; plan for addressing
- Prevention
- Process improvement; root cause analysis

Risk Management

- Assessment
 1. Identification
 2. Analysis
 3. Prioritization
- Control
 4. Planning
 5. Resolution
 6. Monitoring

1. Risk Identification

- Familiarity with classic risks
- Familiarity with historical (organizational) risks
- Selection

Example: Schedule Risks

- Feature creep; gold plating
- Short change quality
- Overly optimistic quality assessment
- Inadequate design
- Reliance on silver bullets
- Research-oriented development
- Weak personnel
- Contractor failure
- Friction between customer and developer

Aside: Class Project Risks

- Team member dropping the course
- Team member not performing
- Not enough design
- Technology; not enough research

2. Risk Analysis

- Impact: Judgment of the effect that an unplanned event will have on a project if it does arise
- For example, number of staff months of delay
- Likelihood estimation
- Exposure = impact * likelihood

3. Risk Prioritization

- All risks have some probability of happening
- But for some, the probability may be too small to worry about
- So concentrate on the high-exposure risks
- That is, sort risks by exposure level and try to control the top ones

4. Risk Planning

- Plan for each risk
- That is, if it occurs what will be done to address it
 - Backup plan
- Extra resources
- Fall-back position
- *Etc.*

5. Risk Resolution

- Avoid activity
- Transfer risk
- Buy information
- Eliminate root cause
- Assume risk (do nothing)
- Publicize
- Contingency plans (previous slide)
- Remember (case base)

6. Risk Monitoring

- Top 10 risk list
- Interim postmortems
- Risk officer