Risk Management

Strategies for Making Decisions

17 November 2009

Dr. Joan A. Smith
Emory University

Part 1: General Risk Management

- Risks are typically multi-dimensional
- Specific interactions may impact risk assessment
  - Relationship between deadline and investment cost
  - Relationships between several risk factors
- Risks are not limited to time and money
  - Project Risks (threaten the project plan)
  - Technical risks (quality, schedule)
  - Business Risks (market, sales, budget, strategic position)
  - Personal Risks (finances, family, health)
  - Product-Specific Risks and Generic Risks

Some Risk Interaction Examples

- Time <--> Money <--> Competition <--> Delivery Failure:

  Ready for Xmas Season <--> $95K manpower cost <--> Estimated 50% loss in total sales

- Social Impact <--> Operational Need <--> Development Cost:

  Family Medical Bills <--> Current Operational Margins <--> Investment cost for equipment

- Performance <--> Cost <--> Support <--> Schedule

- Scope <--> Staffing <--> Customer Expectations

Assessing Risk

- Identify Each Risk Element
  - Generic (e.g., by categories as discussed on previous slide)
  - Product-Specific (e.g., level of expertise required vs expertise at hand)
  - Known (based on formal evaluation, expert information)
  - Unknown (past experience + best guess at what *might* go wrong)
  - ✓ **Create a check-list of the risks**
- Quantify
  - Establish a scale that reflects the risk probability (Ex: 1-low to 5-high)
  - Specify consequences
  - Estimate impact on project and/or product
  - Assign Level-of-Confidence for each quantified risk
  - Have cost data ready for risk exposure estimation
- Assess
  - Estimate probability each will occur
  - Estimate impact of each risk on overall success
  - Estimate impact of various risk combinations on success
    - Some combinations are deadlier than others
  - Sort the risks by probability and impact

Risk Exposure

- Estimate Risk Exposure (RE) for each risk/combination
- R.E. = Risk Probability X Cost
- Costs are based on known factors:
  - Average time to code solution X average cost/developer hour
  - Current per-DVD distribution costs
  - Per-GB bandwidth cost from network provider
  - Etc….
- Costs can reflect money lost or price to fix problem
- Costs are usually best-estimate, rarely absolute price
- Cf. slides 17-23 from Chapter 28 (publisher's slides)

True Tales

- 3 true stories (companies and people disguised)
  - 3 brothers build a software business & sell for over $50M
  - 4 friends build a software business and sell for $5M
  - 2 brothers build a software-driven manufacturing improvement device (still in business)
- Examined from the following risk points:
  - Product Size
  - Business Impact
  - Customer Risk
  - Technology Risk
  - Staff/People Risk
  - Financial Risks
- Different paradigms but similar opportunity-drivers
  - Emerging/maturing technology + immediate, dramatic cost-savings for customers + unique market position

Part 2: Software Development Risk Management

- Factors similar to general risk management
- Proactive and Reactive Strategies
  - Figure out what **can** go wrong and a reasonable solution/prevention
  - Figure out what **did** go wrong and how to recover
- For each factor:
  - Estimate probability
  - calculate cost
  - assess risk
- Monitor status of each
  - Note adverse changes
  - Add new/unexpected factors as they arise
  - Regularly re-evaluate risk exposure

Classic Factors Relating to Programming

- Schedule
  - Time to market is ***everything***
  - Plan for shortest path to good-enough release
- Requirements Creep
  - Possible to do <> Should be done
  - Income generation from good-enough product on the shelves can finance development of more features for next release
- Tool Chain
  - Should provide a clear advantage (Ex: faster; easier; well-understood; easy to hire)
  - Should meet all development needs (Ex: special libraries; debug tools; testing tools; platform compatibility)
  - Should support a test-based methodology
- Code Complexity
  - Product components can be classified by easy, moderate, hard
  - *Do the hard stuff first!*
  - Do the easy stuff last
  - Double-check your component classification (did you classify each correctly?)

SW Project Management

- Common methods established for managing software projects
- Once a team grows beyond a few people, some formalized management will help keep project on track
- The larger the team, the more critical the PM tools will be to success
- Example of tool set that operates like many of those used in big business:

http://www.pragmaticsw.com/Movies.asp?Topic=OverviewSoftwarePlanner