Software Risk Analysis and Management

Bojan Cukic
1.304.293.0405
cukic@csee.wvu.edu

What Is Risk Analysis?

• Steps that help manage uncertainty.
• A risk is a potential problem.
  – It might happen.
• Identify it.
• Assess its occurrence probability.
• Estimate its impact.
• Establish contingency plans, should it occur.
Why Is It Important?

• Be prepared!
• Because, if things can go wrong, they will.
  – Simplified: Things will go wrong.
• Understanding risks and taking proactive measures to avoid and manage them is the KEY element of good software engineering.

Steps?

• Risk identification.
• Analyze occurrence likelihood and impact/damage.
• Rank risks by probability and/or impact.
• Develop a plan to manage high probability/impact risks.
• Product?
  – Risk mitigation, monitoring and management plan (RMMM).
Fundamentals

• Can we, by changing our actions today, create an opportunity for a better situation tomorrow?
• Risk mitigation always involves a change.
• Risk involves a choice, which brings uncertainty
  – But the risk itself is very certain!
• On a software project, choose the “right risks” to tackle.

Strategies

• Reactive.
  – Indiana Jones:”Don’t worry, I’ll think of something.”
  – Not every software engineer is Indiana Jones.
  – If reactive, set resources aside for firefighters.
• Proactive strategies based on risk identification, ranking and contingency planning.
Software Risks

- Involve uncertainty and loss.
- Project risks: threaten project plans.
  - Schedule, cost, personnel, resource, customer, requirements, complexity, size, structural uncertainty.
- Technical risks: quality and timeliness.
  - Design, implementation, interface, verification, maintenance problems.
  - (B)leading edge technologies.

Software Risks (2)

- Business risks.
  - Does anyone want the product?
  - Does the product fit business strategy?
  - Do salespeople understand the product?
  - Loosing support of senior management.
  - Loosing budgetary of personnel commitment.
Risk Management Paradigm

- Identify
- Plan
- Track
- Control

Risk Identification

- Generic vs. product specific risks.
- Risk checklists.
  - Product size, business impact, customer characteristics, process definition, development environment, technology to be built, staff size and experience.
Assessing Overall Software Risk

• Questions ordered by relative importance.
  – Have top software and customer managers formally committed to support the project?
  – Are end-users enthusiastic?
  – Are requirements fully understood?
  – Have customers participated in requirements definition?
  – Do end-users have realistic expectations?
  – Does SE team have the right mix of skills?

Risk Components and Drivers

• Components.
  – Performance risks, meeting requirements and fitting intended use.
  – Cost risks
  – Support risk.
  – Schedule risk.
Risk Projection

- The same as risk estimation.
- Rates each risk in two ways:
  - Likelihood (probability) that the risk is real.
  - Consequences of the problem, should it occur.
- Procedure.
  - Establish the scale for likelihood, delineate consequences, estimate the impact, note the overall accuracy of estimation.
Risk Impact

- Consequences determined by the nature, the scope and the timing of the risk occurrence.
- Nature: the problems likely to occur.
- Scope: severity and the overall distribution of the effect.
- Timing: how long will the impact be felt.
Risk assessment

• Define risk reference level.
  – Performance, cost, support, schedule.
  – Some degradation is acceptable, but how much?
  – Define the referent point, or the break point.
Risk Mitigation

- Avoidance, monitoring, management and contingency planning.
  - Meet with current staff.
  - Mitigate causes for the problem (personnel turnover, for example).
  - Disperse information about development activities.
  - Conduct peer reviews.
  - Assign a back-up staff member(s)
Assess Performance

Risks

Identify critical use cases

Select key performance scenarios

Establish performance objectives

Construct performance models

Add software resource requirements

Add computer resource requirements

Evaluate performance model(s)

V&V of performance models

[feasible] [infeasible]

Modify/create scenarios

Modify product concept

Revise performance objectives

Success

Risk information sheet

<table>
<thead>
<tr>
<th>Risk ID: P02-4-32</th>
<th>Date: 5/9/02</th>
<th>Prob.: 80%</th>
<th>Impact: high</th>
</tr>
</thead>
</table>

Description:
Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.

Refinement/context:
Subcondition 1: Certain reusable components were developed by a third party with no knowledge of internal design standards.
Subcondition 2: The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components.
Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.

Mitigation/monitoring:
1. Contact third party to determine conformance with design standards.
2. Press for interface standards completion; consider component structure when deciding on interface protocol.
3. Check to determine number of components in subcondition 3 category; check to determine if language support can be acquired.

Management/contingency plan/trigger:
RE computed to be $20,000. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly.
Trigger: Mitigation steps unproductive as of 7/1/02

Current status:
5/12/02: Mitigation steps initiated.

Originator: D. Gagne

Assigned: B. Laster