Identifying and Mitigating Risks

Why do projects fail?

80 % of commercial projects fail.

90 % of open source projects fail.

Why do projects fail?

- We need to ask these important questions:
  - What kind of failure was it?
    - e.g. incomplete, unreliable, off-schedule/budget
  - Who was responsible?
  - What happened or did not happen?
  - Which process(es) broke down?
  - What module(s)/feature(s) failed?

Project Management Phases

1. Defining/Requirements	Built the wrong thing
2. Risks	Ignored potential pitfalls
3. Planning & Scheduling	Overly optimistic schedules
4. Launching	Poor team dynamics
5. Monitoring & Controlling	Poor development practices
6. Closing	No customer acceptance

Who is responsible?

- Developers?
  - Poor design
  - Uncommitted or de-motivated developers
  - Silver bullet syndrome
  - Lack of source control, abandoned planning
- Client/Upper management?
  - Feature creep
  - Unrealistic schedules
  - Unrealistic expectations
  - Incorrect requirements
- Project Manager?
  - Poor planning
  - Insufficient risk management
  - Insufficient quality assurance

Case Study 1: ATCSoft

- The ATCSoft project was launched, and steady progress was made
- When the team set out to integrate the ATCSoft application with existing RADAR equipment, however, they hit a snag
  - The team members could not figure out how to integrate the systems
    - The RADAR system did not have appropriate physical connections, nor was there an appropriate driver for the interconnection
  - The project manager had to hire engineering consultants to work out the integration details
  - This diversion took 2 months and cost a significant amount of money (additional labour, consultancy fees, and business value)

What happened?

- A *technical problem* ended up stalling development
  - This could easily have become a complete disaster, if it were not possible to integrate the systems

Case Study 2: PathFinder 2.0

- The PathFinder project was started in August
  - Rory T. was the star developer
    - His knowledge of neural nets inspired him to suggest that a neural net implementation of a PathFinder would be a good idea
    - He wrote up much of the foundation code for the neural network
    - Initial tests showed a definite improvement in the PathFinder performance, and more realistic, human-like, decisions
  - Despite the large salary increase he was offered, Rory took a good offer with another firm
    - When some tests showed that the neural net was not fast enough to make real-time decisions, the team had no immediate answers
    - A bug was found that sent the avatars wandering aimlessly around the maze in a circuit, when certain rare conditions were present
    - Again, the team had no idea how to approach the problem

What happened?

- A *personnel problem* was at fault
  - The team's over-reliance on a single person was their downfall
  - Taking him out of the equation stalled development

In both cases, the team neglected their risks

- It is critical for a project team to understand and plan for risks

Risk Mitigation

- In ATCSoft project: the team should have investigated the integration of various systems at the start of the project
  - Given adequate time, the integration could have been worked out before it was needed

- In the PathFinder project
  - Rory could have thoroughly documented the neural network code as it was developed
  - He could have had seminars for team members, explaining the concepts of neural networks
  - Understanding neural networks, the team would have a better chance of carrying on without Rory

Risk Assessment

- The following describes the risk assessment process:

1. Identifying risks
2. Estimating a risk's cost/effects
3. Estimating a risk's likelihood
4. Identifying alternatives
5. Evaluating/comparing alternatives

- Once risks are assessed, a project manager should plan for them

Risk Identification

- The first step in risk analysis is to identify the project's risks
  - Each project has its own set of unique risks
- Identifying risks seems like a dark art
  - How do you identify something that could potentially be hidden until it is too late?
  - Risk identification can be made easier using categories of risk
    - This leverages the knowledge of many project managers who have experienced risks

Categories of Risk

- Technical risks (related to using a particular technology)
  - Performance
  - Reliability
  - Availability
  - Complexity
- Project management risks
  - Poor resource allocation
  - Poor planning
  - Poor prioritization
- Organizational risks
  - Lack of support or resources
  - Inadequate or inefficient management
  - Interference from other projects & management agendas

Categories of Risk

- Constraint risks
  - Deadlines
  - Resources
- Business risks
  - Marketability
  - Timing
  - Vendor delays
  - Economic conditions
- External risks
  - Changing laws and regulations
  - Dependence upon suppliers and contractors

When do we identify risks?

- Identify risks before the planning phase
  - Some risks may be difficult to spot when looking at requirements at a high level
- Identify risks after the planning phase
  - It is useful to know risks before the planning phase, so that extra time can be dedicated to their mitigation

- A good compromise is to perform risk identification during the planning phase:
  - After creating the work breakdown structure
  - Before creating the schedule

Common Risks

- Feature creep
  - New features are frequently added after development has started
- Implementation gold-plating
  - Developers are working on the perfect implementation
- Inadequate design
  - Too little attention has been paid to design
- Overly optimistic schedules
  - Management pushed schedules down, rather than schedules work their way upward from developers
- Poor motivation/weak personnel
  - Developers are working at a less-than-optimal pace
- Silver-bullet syndrome
  - A trendy technology was expected to produce the equivalent to 10,000 lines of code in only 50 lines of code
- Contractor failure
  - A contractor lacked expertise/commitment needed to do the job on schedule

Estimating Risk Costs & Effects

- Estimating the costs & effects of a risk is dependent upon the risk
  - e.g. A project using a new technology might realize that the technology is inadequate or unreliable
    - Now, the application must be retrofitted to another (trusted) technology
    - Much of the software may need to be replaced
    - The cost in this case is the cost of developing the obsolete components
    - In addition, there may be hidden costs due to delays (such as customer confidence or personnel availability)

Estimating Risk Costs & Effects

- Estimating the costs & effects of a risk is dependent upon the risk
  - e.g. In some projects there is a risk that a key developer will leave the project
    - If the key developer leaves, what will it take to replace her?
    - Given market conditions, you might estimate a replacement in 2 months
    - Some project deliverables might be delayed by up to that amount in her absence
    - Also, you may have to consider signing bonuses, relocation expenses, travel expenses, and other hiring costs
    - It depends on the project whether or not these costs are considered high

Estimating Risk Likelihood

- Like risk cost, risk likelihood also depends on the risk
  - The likelihood that a technology will fail can usually be estimated accurately, e.g.
    - Based on performance in similar projects
    - Based on performance in “well-known” projects
    - Based on available expertise
  - Other types of risks e.g. likelihood of a person leaving a project may be harder to quantify
    - One possibility is to ask

Identifying Alternatives

- C++ or Java?
- If Sarah leaves, who can replace her?

Evaluating & Comparing Alternatives

- Let us examine alternatives to Sarah:
  - Gerard: Has leadership, but lacks the technological expertise
    - Gerard is a *take charge* kind of person
    - He is also a *get it done* kind of person
    - However, he is not familiar with XML and many other technologies we plan to use
  - Helen: Knows some of the technology, but is very inexperienced
    - Helen knows XML and a few other technologies we plan to use
    - However, Helen is just starting her career
    - She has difficulty being assertive and taking charge
    - She doesn't command respect from her colleagues
    - Her development itself is slow