Risk

BRIAN LAMARCHE

COMPUTER SCIENCE 323 SOFTWARE DESIGN

Project Unknowns

- These unknowns, called risks, could potentially lead to an undesirable outcome
  - The system does not work

- Project unknowns are not always technical
  - “We are not an effective development team”

- We try to map out our project needs and resources accordingly

- This helps us plan and execute

Risks

- Actually, in risk management risk is a quantifiable metric defined as the probability that an event (problem or source) will occur that will lead to an unwanted outcome.

Types of Risk

- Schedule
  - Cannot complete all requirements in allotted time
- Budget
  - Money is reallocated.
- Operation
  - Improper resource allocation
- Programmatic
  - Customer changes or changes requirements
- Technical

Schedule

- Improper planning
- Improper resource allocation
- Unexpected project scope
  - Fire Drills

Budget

- Budget risks are often linked to scheduling problems
- Running out of funds due to improper planning
- Funding changes during project
- Unexpected technology purchases
  - Licenses
  - New hardware

Operation

- Improperly allocating resources
- Poor planning and prioritization
- Poor execution of process
- Insufficient training
- Poor use of tools

Programmatic

- Customer changes requirements
- Customer changes completely
- Marketing steers scope of project another way

Technical

- Technology components are difficult to integrate
- Technologies are out-dated
- Technologies are immature
- System is too complex
- Deployment of technologies is not well understood or executed

Our Project

- In this course we will not have budget risks
- Programmatic risks will be minimal
  - Customer could change their mind on the target system requirements
  - Competition rules could change slightly

Risk Management

- The ISO 31000 defines principles and guidelines for risk management.
- We will not cover all of these details, however, we will explore the basic principles and then apply this understanding to known software engineering processes

Steps

- Here are the main points:
  - Identify and characterize unknowns
  - Assess the degree
  - Define stakeholders and constraints
  - Prioritize
  - Create risk mitigation framework
  - Mitigate

Identify Risk

- Analysis:
  - Source Based
    - The camera module – component based
  - Problem Based
    - Camera does not initialize – event based
- Both approaches allow us to identify new risks that we had not previously considered
- The goal is to dig into both our organization and our technical requirements and determine possible problems

Identify Risk

- Checklists are a nice way to identify specific problems if we have a well established domain
- e.g. Having a responsive user interface:
  - Requires asynchronous programming
    - Requires multi-threaded support
  - Does language support multi-threading
  - Does GUI framework support display of multiple image formats

Assessment

- Each risk should be assessed by a degree, or measure of how it will impact the project.
- Degrees are not priorities
- You need an assessment to help you prioritize your path forward

Example Assessments

- Extreme
  - Will affect the outcome, emphasis must address this or will not pursue
- High
  - Will affect the outcome, should focus efforts to mitigate as much as possible.
- Normal
  - Should address and seen as not to burden the project
- Low
  - Could affect some things, but effect is minimal.
- Trivial
  - May not affect outcome of software

Assessment to include possibility of occurence

- When assessing you should consider the rate of possible occurrence
  - The camera drops it's feed
  - The missile launcher stops responding
  - The internet goes out and we cannot get data from the server

Stakeholders & Constraints

- Make sure you relate the risks to your stakeholders and constraints
- Stakeholders have interests
- Risks can be bound by constraints
  - Environmental factors may limit or amplify a problem or risk
  - e.g. The cross wind from the hallway may knock our missiles off course

Stakeholders

- Define who your stakeholders are
  - Not necessarily your user
  - These are people that have interest in your project
  - People that drive your project requirements
  - Have negative or positive influence in the completion of the project
- The last part is key
  - e.g. stakeholders can withdraw funding

Stakeholders

- Because stakeholders are usually drivers of your project you need to consider their interests
- So define them!
  - This helps you prioritize your risk mitigation strategy

Stakeholders

- Because stakeholders are usually drivers of your project you need to consider their interests
- So define them!
  - This helps you prioritize your risk mitigation strategy

Prioritize

- Once we know:
  - What the risk is
  - Its degree of difficulty
  - Connection to stakeholders
  - Operational constraints
- We can start to prioritize

What you need to prioritize

- Build a matrix or cards with the name of the risk, it's degree, connection to stakeholders
- Define a priority rating system
  - 1-5, where 1 is the highest
  - But put this into context
    - 1 = Highest and needs to be addressed now
    - 2 = High and should be addressed next
    - ...
    - 5 = Will not attempt

Passing Priority

- Work with your team to determine these priorities
- Use simple dispute resolution techniques to “pass” or resolve a priority
  - Fist to Five
    - Simple way to get consensus
    - http://freechild.org/Firestarter/Fist2Five.htm

Fist to Five Example

- **Fist**
  - A no vote - a way to block consensus. I need to talk more on the proposal and require changes for it to pass.
- **1 Finger**
  - I still need to discuss certain issues and suggest changes that should be made.
- **2 Fingers**
  - I am more comfortable with the proposal but would like to discuss some minor issues.
- **3 Fingers**
  - I’m not in total agreement but feel comfortable to let this decision or a proposal pass without further discussion.
- **4 Fingers**
  - I think it’s a good idea/decision and will work for it.
- **5 Fingers**
  - It’s a great idea and I will be one of the leaders in implementing it.

Resolution and Mitigation

- By this time risks are defined, assessed, and prioritized
- Now we need to start resolving:
  - Avoid – we won't do this
  - Reduce – we are going to break this down
    - Sub-components may not be as risky as the component as a whole
  - Share – outsource or transfer
  - Retain – accept, plan and budget

Mitigation Plan

- Develop the specific plan, task, story, etc to resolving the risk
  - Once accepted you should be able to tie it to a development effort
- We will build a prototype of the missile launcher control program that will fire a missile and move the turret.
- Risk mitigation can help you understand WHAT you need to check for to say it's done, i.e. acceptance criteria

End product

- Essentially at the end you have a matrix:
  - Risk
  - Degree
  - Priority
  - Resolution
  - Mitigation

Conclusion

- Without understanding what we don't know, we will never understand what we do.
- Risk assessment gives us an idea of how to proceed, what processes we might choose, and how to organize ourselves.
- Be flexible, apply the principles, and be honest while you are doing this
- Keep in mind, that we still have not defined tasks!