

# Risk Management

Module Code: ELEE1149

Module Name: Software Engineering

Credits: 15

Module Leader: Seb Blair BEng(H) PGCAP MIET MIHEEM FHEA

# Software project management

- Concerned with activities involved in ensuring that software is delivered on time and on schedule and in accordance with the requirements of the organisations developing and buying the software.
- Project management is needed because software development is always subject to budget and schedule constraints that are set by the organisation developing the software

# Success Criteria

- Deliver the software to the customer at the agreed time.
- Keep overall costs within budget.
- Deliver software that meets the customer's expectations.
- Maintain a coherent and well-functioning development team.

# Software engineering is different from other types of engineering

- The product is immaterial
  - Software cannot be seen or touched. Software project managers cannot see progress by simply looking at the artefact that is being constructed.
- Many software projects are 'one-off' projects
  - Large software projects are usually different in some ways from previous projects as every environment where software is developed is, in some ways, different from all others. Even managers who have lots of previous experience may find it difficult to anticipate problems.
- Software processes are variable and organization specific
  - We still cannot reliably predict when a particular software process is likely to lead to development problems.

# Factors influencing project management

These factors mean that project managers in different organizations may work in quite different ways.

- Company size – small or large companies
- Software customers – internal, external, governmental agency
- Software size – small systems – small teams, large systems – can be geographically distributed
- Software type – consumer or safety critical
- Organizational culture – encouraging individual or group focused approach
- Software development processes – agile – lightweight management, others – more formal

# Universal management activities

- Project planning
  - Project managers are responsible for planning, estimating and scheduling project development and assigning people to tasks.
- Risk management
  - Project managers assess the risks that may affect a project, monitor these risks and take action when problems arise.
- People management
  - Project managers have to choose people for their team and establish ways of working that leads to effective team performance.

# Universal management activities

- Reporting
  - Project managers are usually responsible for reporting on the progress of a project to customers and to the managers of the company developing the software.
- Proposal writing
  - The first stage in a software project may involve writing a proposal to win a contract to carry out an item of work. The proposal describes the objectives of the project and how it will be carried out.

# Risk Management

All previous content has associated risk attached to them...

So what does risk management involve?



# Risk Management

- One of the most important jobs of a project manager
- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- Software risk management is important because of the inherent uncertainties in software development.
  - These uncertainties stem from loosely defined requirements, requirements changes due to changes in customer needs, difficulties in estimating the time and resources required for software development, and differences in individual skills.
- You have to anticipate risks, understand the impact of these risks on the project, the product and the business and take steps to avoid these risks.

# Risk classification

There are two dimensions of risk classification

 center

# What is affected by the risk

- **Project risks**
  - affect schedule or resources;
- **Product risks**
  - affect the quality or performance of the software being developed;
- **Business risks**
  - affect the organisation developing or buying the software.

# Examples of project, product, and business risks

Risk	Affects	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organizational management with different priorities.
Hardware unavailability	Project	Hardware that is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule.
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool underperformance	Product	CASE tools, which support the project, do not perform as anticipated.
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

# Segway : CASE Tooling

## Computer Aided Software Engineering

- Diagramming Tools
- Computer Display and Report Generators
- Analysis Tools
- Central Repository
- Documentation Generators
- Code Generators

# The risk management process

- *Risk identification*
  - Identify project, product and business risks;
- *Risk analysis*
  - Assess the likelihood and consequences of these risks;
- *Risk planning*
  - Draw up plans to avoid or minimise the effects of the risk;
- *Risk monitoring*
  - monitor the risks throughout the project;

# The risk management process



# Risk identification

- May be a team activity or based on the individual project manager's experience
- A checklist of common risks may be used to identify risks in a project
  - Estimation risks.
  - Organizational risks.
  - People risks.
  - Requirements risks.
  - Technology risks.
  - Tools risks.



# Examples of different risk types

Risk type	Possible risks
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.
Organizational	The organization is restructured so that different management are responsible for the project. Organizational financial problems force reductions in the project budget.
People	It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Requirements	Changes to requirements that require major design rework are proposed. Customers fail to understand the impact of requirements changes.
Technology	The database used in the system cannot process as many transactions per second as expected. Reusable software components contain defects that mean they cannot be reused as planned.
Tools	The code generated by software code generation tools is inefficient. Software tools cannot work together in an integrated way.

# Risk analysis

- Assess probability and seriousness of each risk.
- Probability may be very low, low, moderate, high or very high.
- Risk consequences might be catastrophic, serious, tolerable or insignificant.

# Risk types and examples

Risk	Probability	Effects
Organizational financial problems force reductions in the project budget.	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project	High	Catastrophic
Key staff are ill at critical times in the project	Moderate	Serious
Faults in reusable software components have to be repaired before these components are reused	Moderate	Serious
Changes to requirements that require major design rework are proposed	Moderate	Serious
The organization is restructured so that different management are responsible for the project	High	Serious
The database used in the system cannot process as many transactions per second as expected	Moderate	Serious

# Risk types and examples

Risk	Probability	Effects
The time required to develop the software is underestimated	High	Serious
Software tools cannot be integrated	High	Tolerable
Customers fail to understand the impact of requirements changes	Moderate	Tolerable
Required training for staff is not available	Moderate	Tolerable
The rate of defect repair is underestimated	Moderate	Tolerable
The size of the software is underestimated	High	Tolerable
Code generated by code generation tools is inefficient	Moderate	Insignificant

# Risk Planning

- Consider each risk and develop a strategy to manage that risk. The project manager has to think of actions that need to be taken to minimize the disruption to the project if a specified risk occurs
- *Avoidance strategies*
  - The probability that the risk will arise is reduced, example – strategy with dealing with defective components in the table that follows
- *Minimization strategies*
  - The impact of the risk on the project or product will be reduced, example – the strategy for staff illness in the table that follows
- *Contingency plans*
  - If the risk arises, contingency plans are plans to deal with that risk, example – the strategy for organisational financial problems in the table that follows

# Strategies to help manage risk

Risk	Strategy
Organizational financial problems, e.g. not enough money to finish the project	
Recruitment problems, e.g. not able to recruit the appropriate people to work on the project	
Staff illness	
Defective components	
Requirements changes	

# Strategies to help manage risk

Risk	Strategy
Organizational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business and presenting reasons why cuts to the project budget would not be cost-effective.
Recruitment problems	Alert customer to potential difficulties and the possibility of delays; investigate buying-in components.
Staff illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact; maximize information hiding in the design.

# Risk Monitoring

- Assess each of the identified risks regularly to decide whether or not it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Each key risk should be discussed at management progress meetings.



# Risk Indicators

Risk type	Potential indicators
Estimation	Failure to meet agreed schedule; failure to clear reported defects.
Organizational	Organizational gossip; lack of action by senior management.
People	Poor staff morale; poor relationships amongst team members; high staff turnover.
Requirements	Many requirements change requests; customer complaints.
Technology	Late delivery of hardware or support software; many reported technology problems.
Tools	Reluctance by team members to use tools; complaints about CASE tools; demands for higher-powered workstations.

# Using RAG/BRAG in Risk Management

- **RAG (Red, Amber, Green):**
  - Red: Critical risks requiring immediate attention.
  - Amber: Risks needing monitoring and potential mitigation.
  - Green: Risks under control or with acceptable impact.
- **BRAG (Blue, Red, Amber, Green):**
  - Blue: Potential opportunities (optional).
  - Red: Critical risks.
  - Amber: Risks requiring attention.
  - Green: Risks under control or resolved.

## BRAG Risk Management Status

**Blue**

Potential  
Opportunities

**Red**

Critical Risks

**Amber**

Risks  
Requiring  
Attention

**Green**

Risks Under  
Control or  
Resolved

# RAG Risk Management Matrix

	Red	Amber	Green
Category 1	Critical Risks	Risks Needing Attention	Risks Under Control or Resolved
Category 2	More Critical Risks	Risks Requiring Monitoring	Risks Under Control or with Acceptable Impact
Category 3	Yet Another Critical Risk	Risks to Address Soon	Risks Under Control or Resolved

# Implementing RAG/BRAG in Software Projects

- **Assigning Colors:**
  - Criteria for assigning Red, Amber, Green (or Blue) to risks.
  - Consistent application across the project.
- **Regular Updates:**
  - Periodic reviews and updates of risk status.
  - Real-time adjustments based on project dynamics.

# Benefits of RAG/BRAG in Software Risk Management

- **Clear Communication:**
  - Simple visual indicators enhance communication.
  - Stakeholders quickly grasp the project's risk status.
- **Proactive Decision-Making:**
  - Enables proactive decision-making.
  - Focuses attention on critical risks for timely resolution.
- **Continuous Improvement:**
  - Facilitates continuous improvement.
  - Lessons learned from past projects inform future risk management strategies.

# RAG for Risk Matrix

# RAG for Risk Matrix

Date Released	Risk Description	Likelihood of the Risk Occurring	Impact if the Risk occurs	Severity	Owner	Mitigating Actions
6/19/2023	This is the sample text. that you can edit.	Medium	High	High	Project Manager	This is the sample text. that you can edit.
6/19/2023	This is the sample text. that you can edit.	Low	High	High	Project Manager	This is the sample text. that you can edit.
6/19/2023	This is the sample text. that you can edit.	Low	Medium	Medium	Project Manager	This is the sample text. that you can edit.
6/19/2023	This is the sample text. that you can edit.	Low	Medium	Medium	Project Manager	This is the sample text. that you can edit.