



# Software Project Management

Durga Prasad Mohapatra

Professor

CSE Deptt.

NIT Rourkela



# **Risk management**

# Introduction

- Suppose, three experienced programmers were available for the coding of modules A, B, C and D of a project. However, two developers left for better-paid jobs, and so far only one replacement has been recruited, who happens to be a trainee. What to do?
- In another project - payroll implementation project, imagine that a payroll package has been purchased. However, a new requirement emerges that the payroll database should be accessed by a new application that calculates the staff costs for each course delivered by the college. Unfortunately, the purchased payroll application does not allow this access. What to do?

# Introduction


- In this chapter we consider whether the two project leaders could have foreseen that these problems were likely to occur and made plans to deal with them.
- In other words, could these problems have been identified as risks?

# Risk

- PM-BOK (Project Management Body of Knowledge) defines risk as 'an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives'.
- PRINCE2, the UK government sponsored project management standard, defines risk as “the chance of exposure to the adverse consequences of future events”.
- The two definitions differ, as the first includes situations where a future uncertainty actually works in our favour and presents us with an opportunity.

# Risk

- The key elements of a risk are as follows:
  - **It relates to the future:** Future is uncertain. Some things which seem obvious when a project is over, might not have been so obvious during planning, e.g. the costs were underestimated or a new technology was overly difficult to use, etc.
  - **It involves cause and effect:** For example, a 'cost over-run' might be identified as a risk, but 'cost over-run' describes some damage, but does not say what causes it. Example: an inaccurate estimate of effort, the use of untrained staff, or a poor specification etc. Both the cause (or hazard), such as 'inexperienced staff', and a particular type of negative outcome, such as 'lower productivity', should be defined for each risk.

- 
- Most of the techniques used to assure the quality of software, such as **reviews** and **testing**, are designed to reduce the risk of faults in project deliverables.
  - The key role of risk management is considering **uncertainty remaining after a plan has been formulated**. Every plan is based on assumptions and risk management tries to plan for and control the situations where those assumptions become incorrect.





# Categories of Risk

- **Risks:** Two types
  - ✓ **Project Risk:** related to threats to successful project execution.
  - ✓ **Business Risk:** related to factors threatening the benefits of the delivered project.  
In business case, main focus is on *Business Risk*.



# Categories of Risk

- An ICT project manager is normally given the objective of installing the required application by a specified deadline and within an agreed budget. Other objectives might be set, especially with regard to quality requirements.
- Project risks are those that could prevent the achievement of the objectives given to the project manager and the project team.

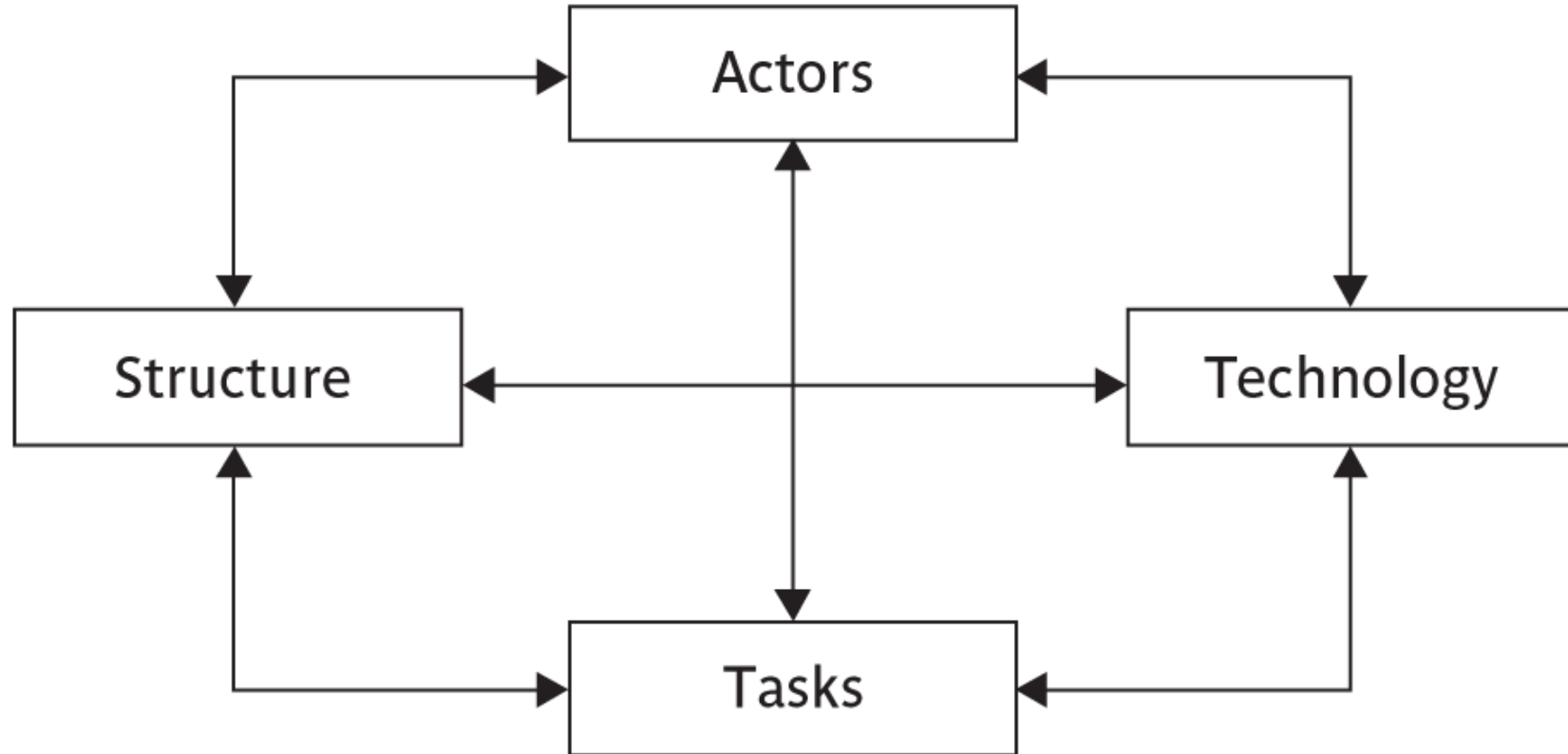
# Categories of Risk


- There could be risks that an application after successful implementation is a business failure, e.g. if an e-commerce site is established to sell a product, the site might be correctly implemented, but customers fail to use the site because of the uncompetitive prices demanded.
- Dealing with these **business risks** is likely to be outside the direct responsibilities of the application implementation team.
- However, the failure to meet any **project objective** could have a negative impact on the business case for the project. For example, an increase in development cost might mean that the income (or savings) generated by the delivered application no longer represents a good return on the increased investment.

# Categories of Risk

- Risks have been categorized in other ways. Kalle Lyytinen and his colleagues, for instance, have proposed a sociotechnical model of risk, a diagrammatic representation of which appears in next figure.
- **Actors** refers to all the people involved in the development of the application in question. A typical risk in this area is that **high staff turnover leads to expertise of value to the project being lost.**
- **Technology** encompasses both the technology used to implement the application and that embedded in the delivered products. Risks here could relate to the appropriateness of the technologies and to possible faults within them, especially if they are novel.

# Lyytinen-Mathiassen-Ropponen risk framework



- 
- **Structure** describes the management structures and systems, including those affecting planning and control. For example, the implementation might need user participation in some tasks, but the responsibility for managing the users' contribution might not be clearly allocated.
  - **Tasks** relates to the work planned. For instance, the complexity of the work might lead to delays because of the additional time required integrate the large number of components.
  - In the figure, all boxes are interlinked, e.g. risks often arise from the relationships between actors and technology. If a development technology is novel then the developers might not be experienced in its use and delay results. The novelty of the new technology is really a characteristic of the developers: once they are used to the technology, it is no longer 'novel'.



# Risk Management Approaches

- Risk management approaches can broadly be classified into **reactive** and **proactive approaches**. The latter Approach is much more effective in risk handling and, therefore, used wherever possible.



# Reactive approaches

- Reactive approaches take no action until an unfavourable event occurs.
- Once an unfavourable event occurs, these approaches try to contain the adverse effects associated with the risk and take steps to prevent future occurrence of the same risk events.
- An example of such a risk management strategy can be the Following.





# Example of reactive approach

- Consider a project in which the server hosting the project data crashes. Once this risk event has occurred, the team members may put best effort to recover the data and also initiate the practice of taking regular backups, so that in future such a risk event does not recur.
- It is similar to calling the emergency fire-fighting service once a fire has been noticed, & then installing fire-fighting equipment in all the rooms of the building to be able to instantly handle fire the next time it is noticed.
- It can be seen that the main objective of this is to minimize the damage due to the risk and take steps to **prevent future recurrence of the risk.**

# Proactive approaches

- The proactive approaches try to anticipate the possible risks that the project is susceptible to. After identifying the possible risks, actions are taken to **eliminate the risks**. If a risk cannot be avoided, these approaches suggest making plans to contain the effect of the risk.
- For example, if man power turnover is anticipated (i.e. some personnel may leave the project), then thorough documentation may be planned. Also, more than one developer may work on a work item and also some stand-by man power may be planned.
- Obviously, proactive approaches incur lower cost and time overruns when risk events occur and, therefore, are much more preferred by teams. However, when some risks cannot be anticipated, a reactive approach is usually followed,



# A Framework for Dealing with Risk

Planning for risk includes these steps:

- (i) Risk identification
- (ii) Risk analysis and prioritization
- (iii) Risk planning
- (iv) Risk monitoring

# Summary

- Discussed categories of risks.
- Discussed risk management approaches.



# References :

1. B. Hughes, M. Cotterell, R. Mall, *Software Project Management*, Sixth Edition, McGraw Hill Education (India) Pvt. Ltd., 2018.
2. R. Mall, *Fundamentals of Software Engineering*, Fifth Edition, PHI Learning Pvt. Ltd., 2018.



Thank you