SSW-540: Fundamentals of Software Engineering

Software Project and Risk Management

Roberta (Robbie) Cohen, Ph.D.
Industry Professor
School of Systems and Enterprises

Software project management

- Project management is needed because software development is always subject to budget and schedule constraints
- Concerned with activities involved in ensuring that software is delivered on time and on schedule and in accordance with the requirements of the organizations developing and procuring 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 management distinctions

Managing software projects is different than classic project management because:

- The product is intangible.

  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 software projects. 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 software project management

- Company size
- Software customers
- Software size
- Software type
- Organizational culture
- Software development processes
- These factors mean that project managers in different organizations may work in quite different ways.

Universal project 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.

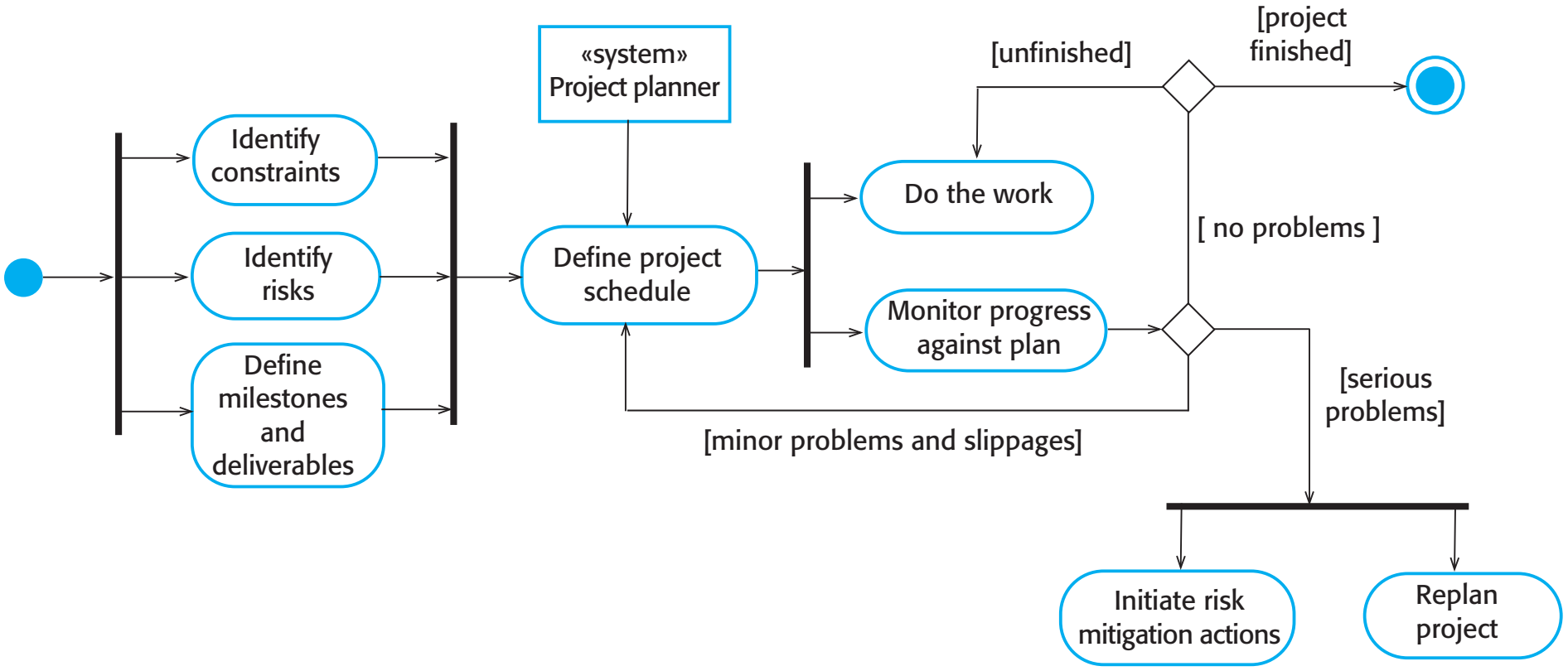
Not-so-universal, but common, activities

- 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.

Project planning

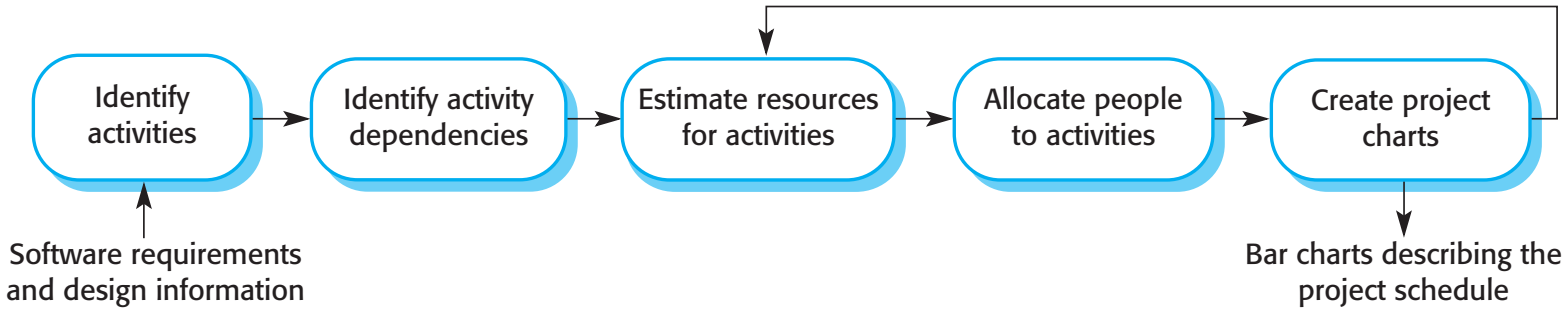
- Project planning involves breaking down the work into parts and assigning these to project team members, anticipating problems that might arise and preparing tentative solutions to those problems.
- The project plan, which is created at the start of a plan-driven project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.
- A startup plan, supporting decisions about budget and staffing, is needed even for agile development to allow resources to be allocated to the project.
- The project plan should be regularly amended as the project progresses, as you know more about the software and its development.
- The project schedule, cost-estimate and risks have to be regularly revised.

The project planning process for plan-driven development

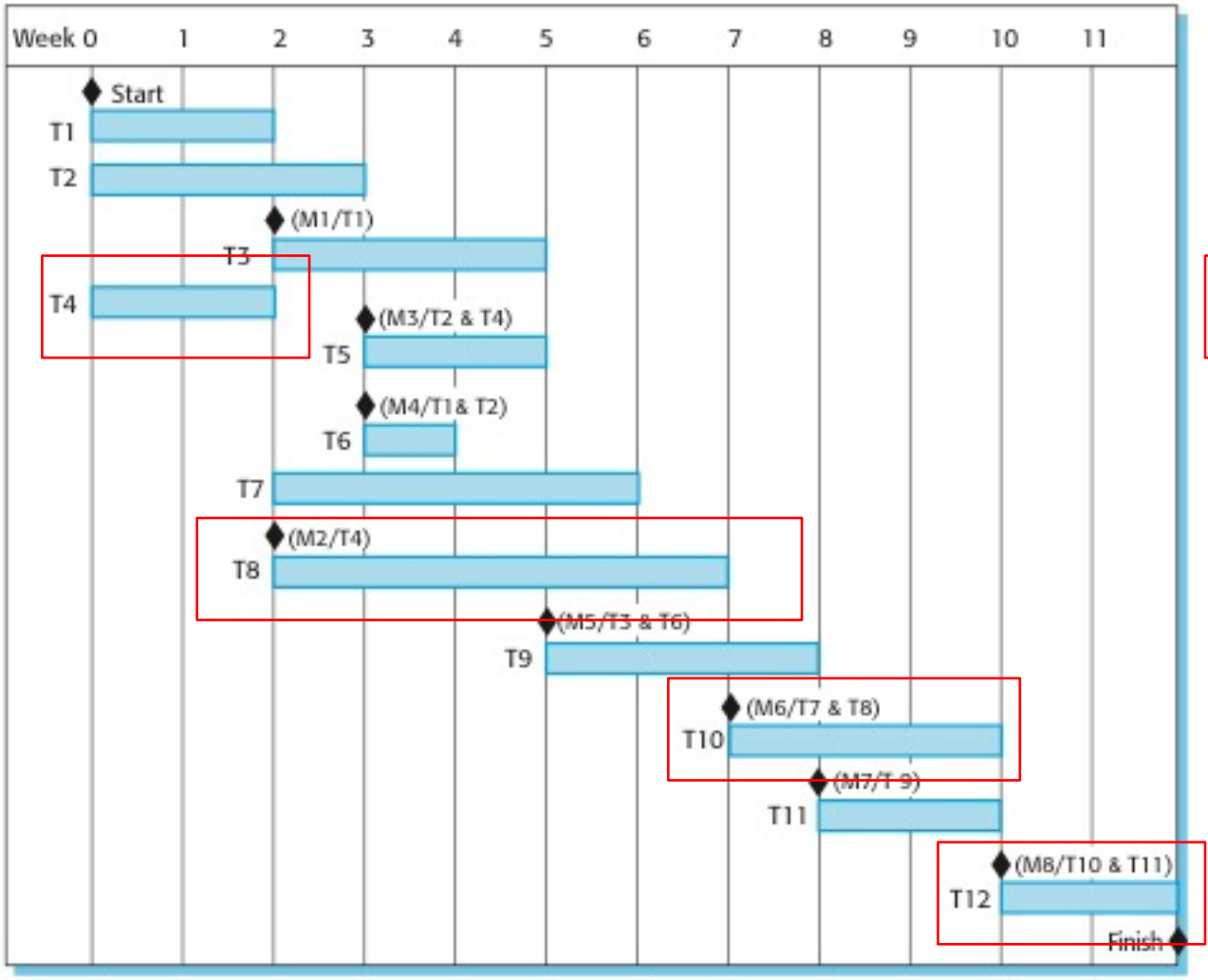


Project scheduling

- Project scheduling is the process of deciding how the work in a project will be organized as separate tasks, and when and how these tasks will be executed.
- You estimate the calendar time needed to complete each task, the effort required and who will work on the tasks that have been identified.
- You also estimate the resources needed to complete each task, such as the server time, the time required on specialized hardware, such as a simulator, and/or what the travel budget needs to be.

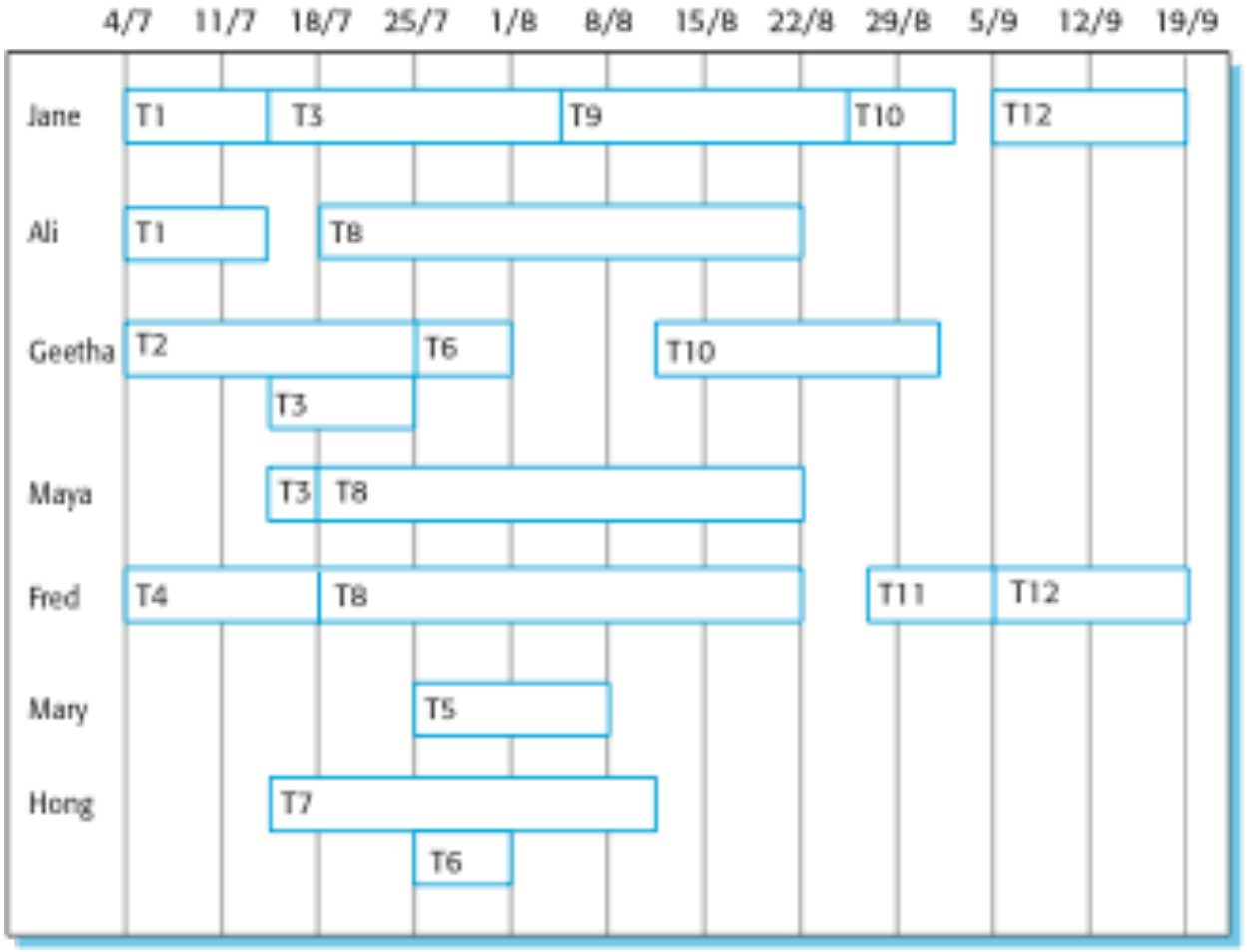


Activity bar chart (Gantt Chart)



Critical path

Staff allocation chart



Scheduling problems

- Estimating the difficulty of problems and hence the cost of developing a software solution is hard.
- Productivity is not proportional to the number of people working on a task.
  - Individual productivity declines as group size grows.
  - Bell Labs estimated the optimal size of software development teams to be 5-6.
- Adding people to a late project makes it later.
  - This is called **Brooks' Law**, first cited by Fred Brooks Jr. in his essay collection *The Mythical Man-Month*.
  - Projects lag when new staff arrives because of training (ramp-up) needs, additional communication overheads and the limited divisibility of tasks.
- The unexpected always happens. Always allow contingency in planning.

Agile planning

- Agile methods of software development are iterative approaches where the software is developed and delivered to customers in increments.
- Unlike plan-driven approaches, Agile teams spread the planning process throughout the development cycle.
  - The decision on what to include in an increment depends on progress and on the customer's priorities.
- The customer's priorities and requirements change so it makes sense to have a flexible plan that can accommodate these changes.
- Stages of Agile planning:
  - Release planning, which looks ahead for several months and decides on the features that should be included in a release of a system.
  - Iteration planning, which has a shorter term outlook, and focuses on planning the next increment of a system. This is typically 2-4 weeks of work for the team.

Agile planning plusses and minuses

- Benefits of this approach:
  - Agile planning works well with small, stable development teams that can get together and discuss the stories to be implemented.
  - The whole team gets an overview of the tasks to be completed in an iteration.
  - Developers have a sense of ownership in these tasks and this is likely to motivate them to complete the task.
- Agile planning difficulties
  - Agile planning is reliant on customer involvement and availability.
  - Some customers may be more familiar with traditional project plans and may find it difficult to engage in an agile planning process.
  - When teams must be large, or when team membership changes frequently, it is not practical for everyone to be involved in the collaborative planning that is essential for agile project management.

Risk management

- Risk management is concerned with identifying risks and drawing up plans to minimize their effect on a project.
- Software risk management is important because of the inherent uncertainties in software development.

  See chart on next slide!

- You must anticipate risks, understand the impact of these risks on the

  Project - impacting schedule or resources

  Product - affecting the quality or performance of the software

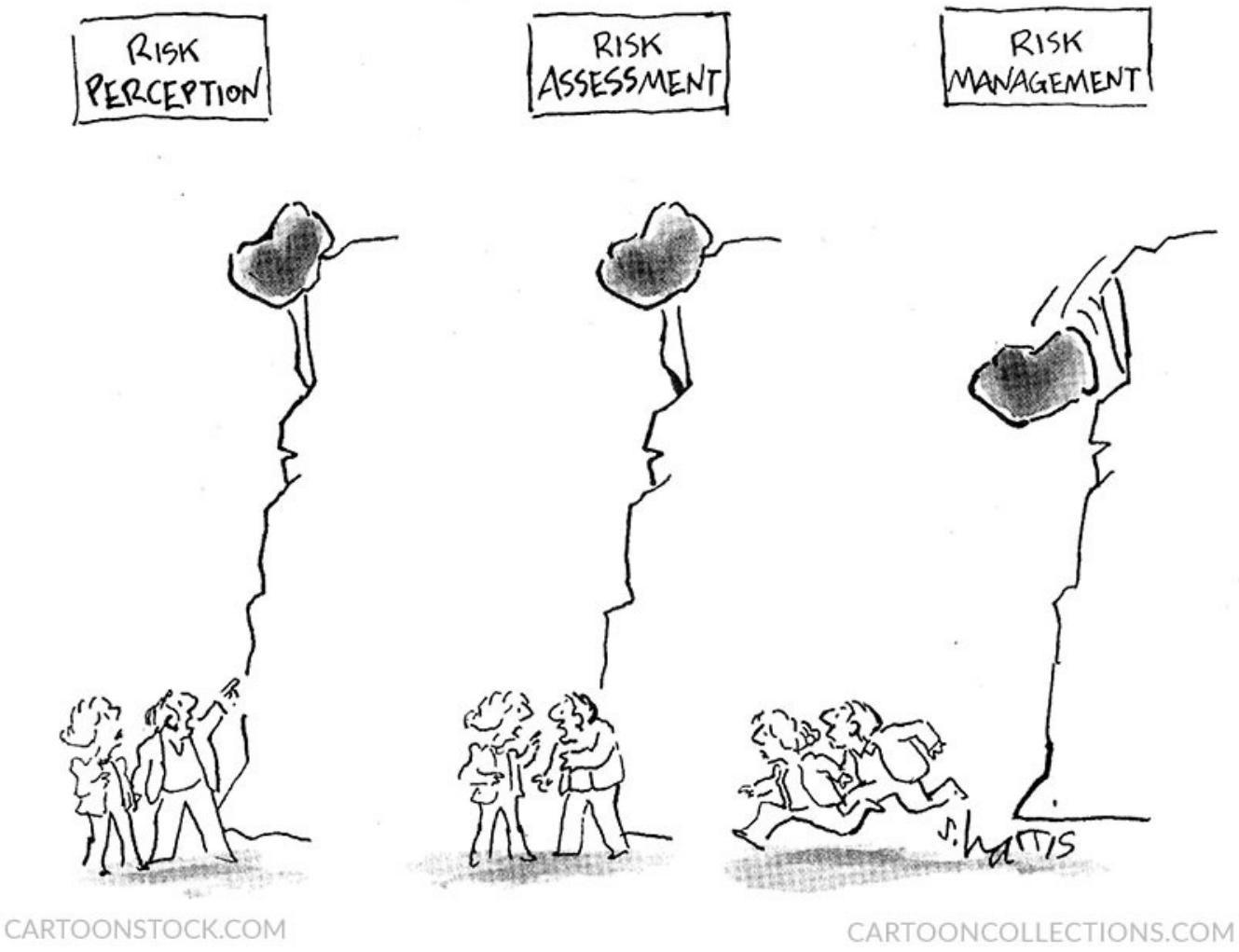
  Business - affecting the organization developing or procuring the software

- And take steps to avoid these risks.

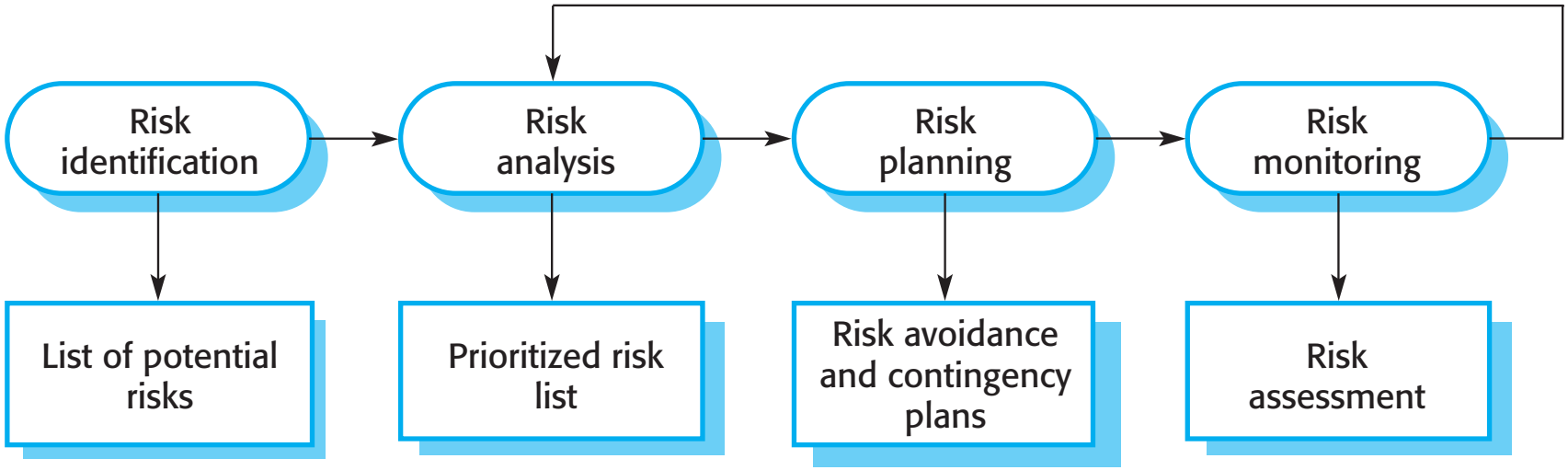
Examples of risks

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.

Dealing with Risk



The risk management process



Identify project, product and business risks

Assess likelihood & consequences

Create plans to avoid or minimize

Monitor the risks throughout

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
  - Technology risks.
  - Organizational risks.
  - People risks.
  - Requirements risks.
  - Estimation 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.
- These analyses require detailed knowledge about the project, the process, the development team, and the organization.

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, cont.

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

Quantification of risks

- Risk exposure is the product of risk probability and risk impact, the latter usually estimated in dollars (RE = Probability x Cost). E.g.,
  - The impact of a software defect found in released software can be measured in terms of cost to repair, cost to reputation, cost to customer, etc.
  - The probability of an expensive defect being found might be estimated based on history, code complexity, or even “gut feeling”.
- The US Air Force uses a matrix and categories of probability and impact:

Probability \ Impact	Very Low	Low	Medium	High	Very High
Very High	5	7	9	11	13
High	4	6	8	10	12
Medium	3	5	7	9	11
Low	2	4	6	8	10
Very Low	1	3	5	7	9

Why quantify risks?

- Quantification helps set priorities. Greater risk **exposure** should mean higher priority!
- Quantification can help understand leveraging of risk.
  - We can often spend to reduce risk; leverage is the original exposure minus the new exposure, divided by the amount spent or (RE1 – RE2)/AmtSpent
  - Example: Losing my star developer, Alex, during a critical development period would set my project back several weeks. Each week costs me $100,000 in loaded wages so 2½ weeks delay will cost $250,000. Alex is prone to sickness so the probability of losing her/him at some point in the 6-week project is 50%. My exposure (RE1) is $125,000.
  - Should I spend $30,000 to have a personal physician tend to Alex during the project if that would reduce the probability of losing him/her to 20%?
- RE2 = (250,000 * 0.20) + (.80 * 30,000) = 74,000
- Leverage = (125,000 – 74,000)/30,000 = 1.7
- Any leverage over 1 is worth considering!

This term is money spent even though the risk doesn't happen; it's added to the exposure.

Another risk leverage calculation example

There is a 0.5% probability that a latent defect will execute and lead to a failure, and that failure would cost the customer **$100,000**

So, **RE** = (0.005) * (100,000) = **$500**

We could hold a design review that will cost **$100** in professional time and halve the number of latent faults

The resulting (new) risk exposure is

NewRE = ((0.005/2) * (100,000)) +
((0.9975) * $100) = ~**$350**

The risk reduction leverage (RRL) is (RE-NewRE)/cost of risk reduction

RRL = (500-350)/100 = **1.5**

Any risk reduction leverage equal to or greater than 1 is worth serious consideration.

What if the design review had cost $200?

Your Turn: Get out your pencils!

- Your $25,000 software project must be delivered by its due date or you lose the contract.
- With 5 weeks left until the due date, you realize that the probability of delivering the entire project on time is only 50%.
- You can negotiate a reduction in deliverables that would take $5000 off of price but increase your probability of on time delivery to 80%.
- Is it worth it?

- Initial risk is 0.5 x 25000 = 12,500
- This leverage costs 5,000 so new risk is (0.20 x 20000) + (0.80 x 5000) = 8,000
- Leverage = (12,500 – 8,000)/5,000 = 0.9

Risk planning

- Consider each risk and develop a strategy to manage that risk.
- Avoidance strategies
  - The probability that the risk will arise is reduced;
- Minimization strategies
  - The impact of the risk on the project or product will be reduced;
- Contingency plans
  - If the risk arises, contingency plans are plans to deal with that risk;

What-if questions

- What if several engineers are ill at the same time?
- What if an economic downturn leads to budget cuts of 20% for the project?
- What if the performance of open-source software is inadequate and the only expert on that open source software leaves?
- What if the company that supplies and maintains software components goes out of business?
- What if the customer fails to deliver the revised requirements as predicted?
- What if a world-wide pandemic forces the entire development team to work at home?

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 components.
Staff illness	Reorganize team so that there is more overlap of work so people understand each other's jobs.
Defective components	Replace potentially defective components with purchased components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact; maximize information hiding in the design.

More strategies to help manage risk

Risk	Strategy
Organizational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying components; investigate use of a program generator.
Unrealistic schedule	Negotiate reducing deliveries with the customer.

Risk monitoring

- Assess each 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.

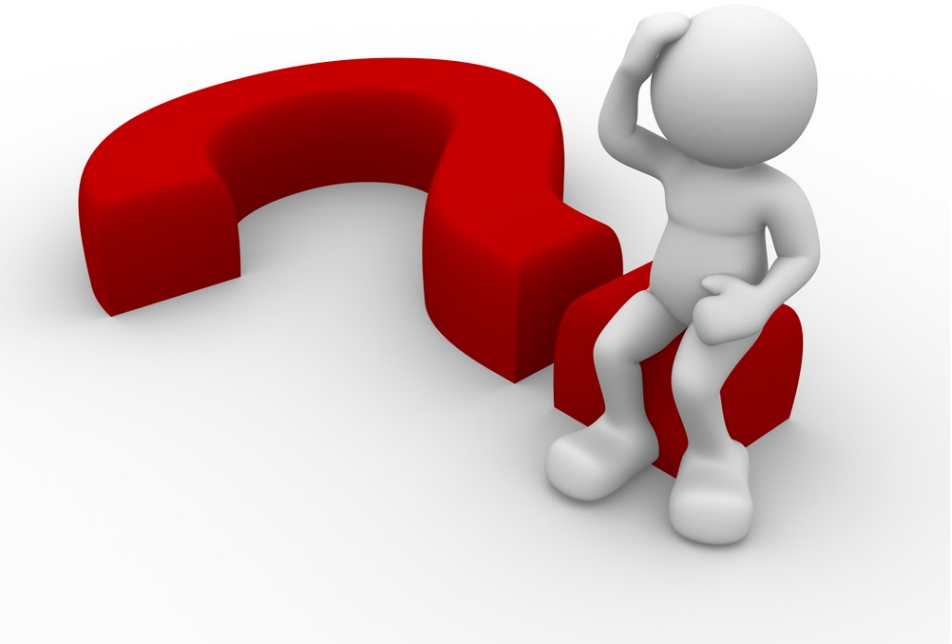
Risk indicators

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.

Summarizing dealing with risks

- Fundamental strategies for dealing with risks fall into three categories:
  - Avoidance strategies to reduce the probability that the risk will arise.
  - Minimization strategies to reduce the impact of the risk on the project or product.
  - Contingency plans to deal with that risk when it does occur.
- Many software development organizations set up financial reserves to cover risk exposure during development.
  - Such funding can be drawn upon for extra purchases and/or hiring.
  - These monies are released back into the budget once the threat of the risk has passed. (Works like warranties.)

Questions?



Risk Identification Group Exercise

Software development risks are found in 3 areas:

- Business (Process)
- Project
- Product

For the development of software for a medical device such as the insulin pump described in the Sommerville text, what might be the **two** (2) most significant risks **in each** of these 3 areas. What could eliminate or at least mitigate each risk? Are the mitigations you suggest likely to provide cost-justified *leverage*?