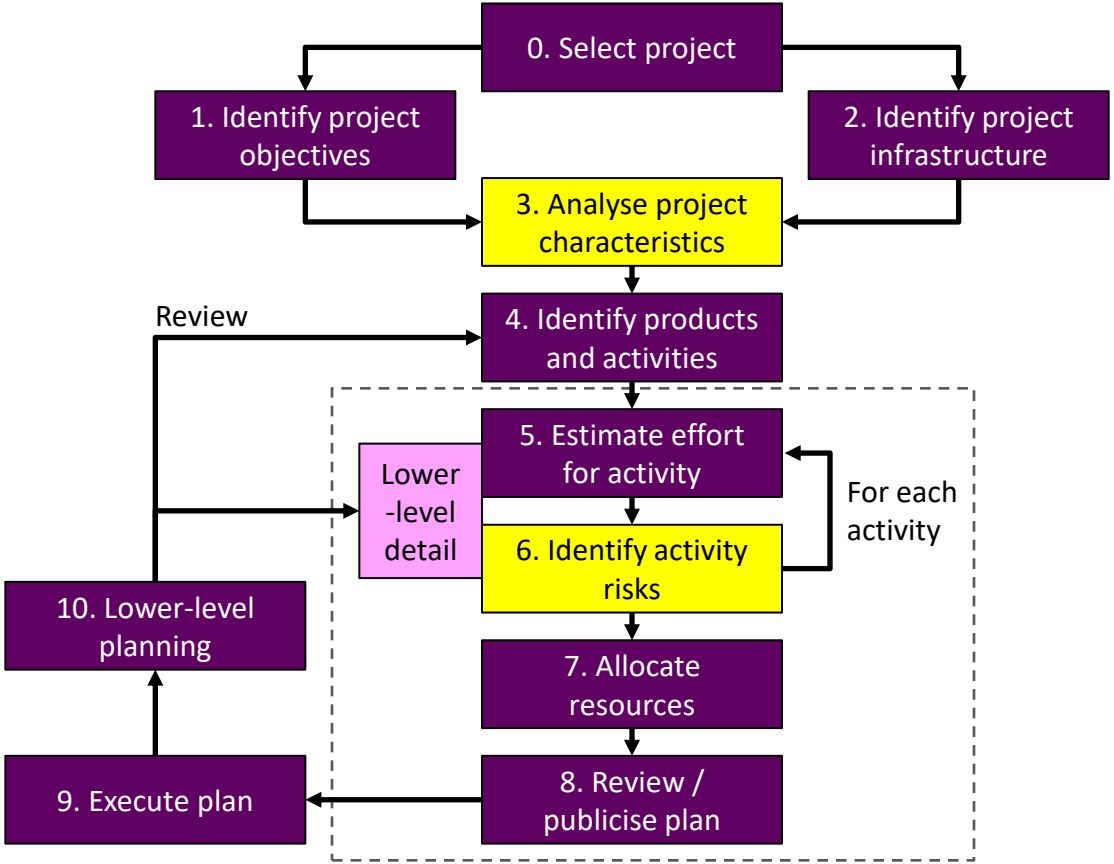
Aston University
BIRMINGHAM UK

Software Project Management

Unit 5: Risk Management

Thais Webber
Richard Lee

According to Step Wise



Unit Objectives

- Define risk and risk management in the context of software projects
- Categorise software project risks based on their impact and probability
- Understand risk management steps including risk identification, risk analysis, risk planning and risk monitoring

Outline

- **Introduction**
- Categories of risk
- Dealing with risk in software projects
- Evaluating risks to the schedule using PERT

Definition of Risk

- The chance of exposure to the adverse consequences of future events (PRINCE2)
- An uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives (US Project Management Body of Knowledge standard – PM-BOK)
- Risks can be negative or positive
- Risks relate to possible future problems, not current ones
- Risks involve a possible cause and its effect(s)

Negative Risk

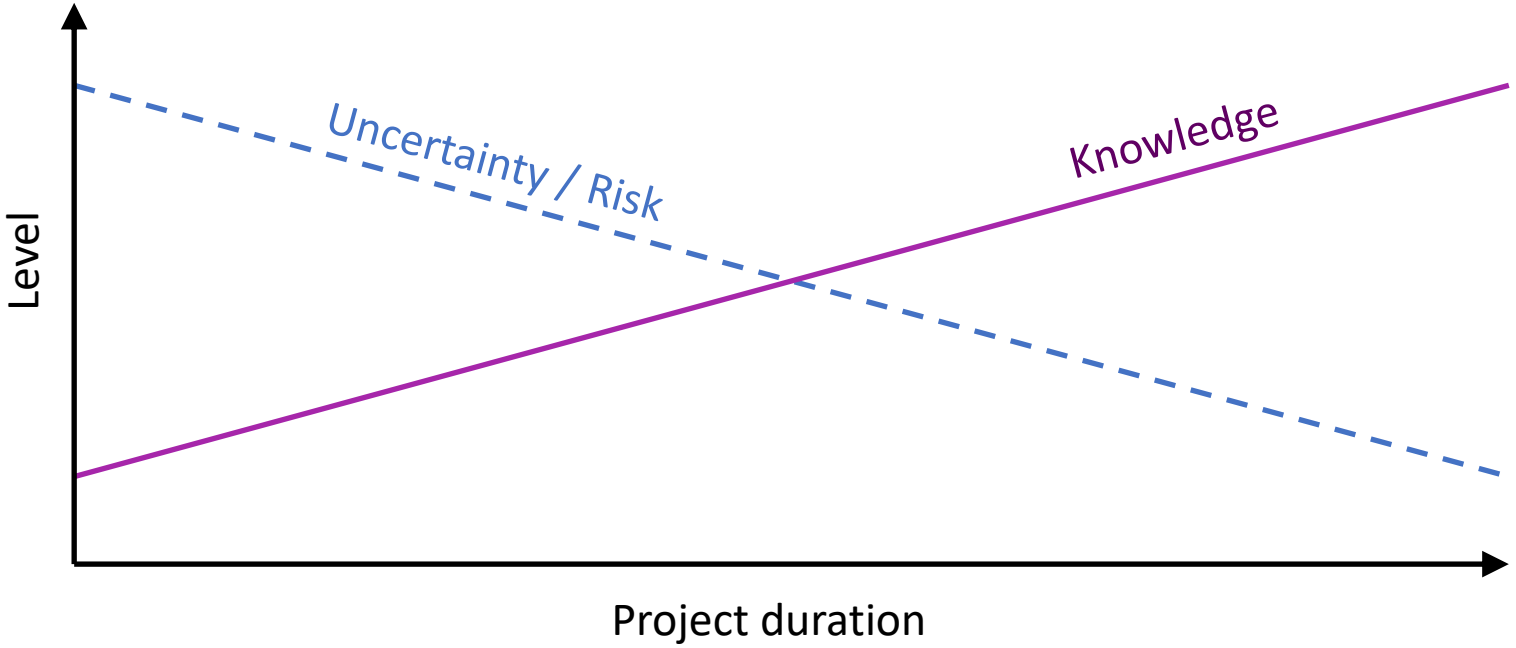
- Negative risk involves understanding potential problems that might occur in the project and how they might impede project success

- Example : Key staff are ill at critical times in the project

Positive Risk

- Positive risks *can* result in good things happening; sometimes called opportunities
- Positive risk management is an investment
- Example : a new technology developed saving you time if released

Risk changes as the project progresses



Why is risk planning not widely used?

- Lack of awareness of the approach
- Unwillingness to spend additional time and resources on risk management
- Development managers may want projects to go ahead and do not want project sponsors to be deterred by consideration of possible failure
- If successful, you might not experience any tangible benefit, in spite of there being a cost associated with its use (which is actually a risk)

Project Risk Management

- Project Risk Management includes the **processes** concerned with identifying, analysing, and responding to project risk

- It aims at:
  - maximizing the results of positive events
  - minimizing the consequences of adverse events

Project Risk Management

- Project Risk Management includes the **processes** concerned with identifying, analysing, and responding to project risk

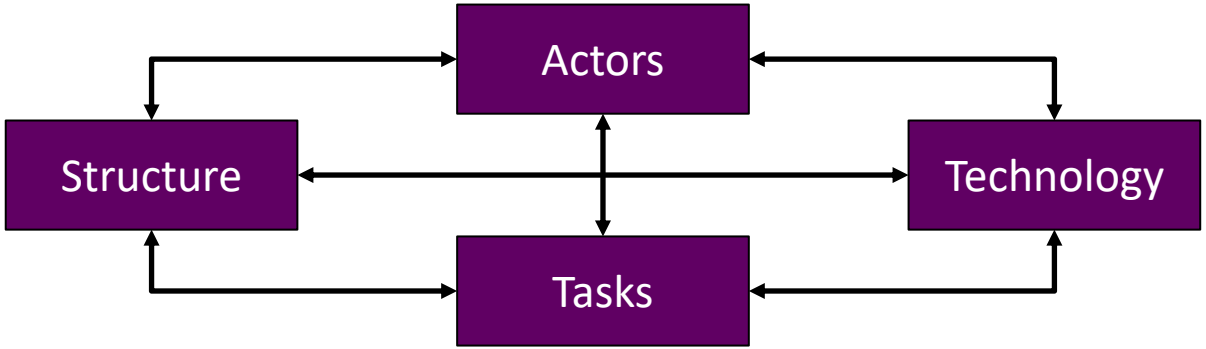
- It aims at:
  - maximizing the results of positive events
  - minimizing the consequences of adverse events

- Risks factors:
  - Project size
  - Complexity
  - Speed of implementation

Outline

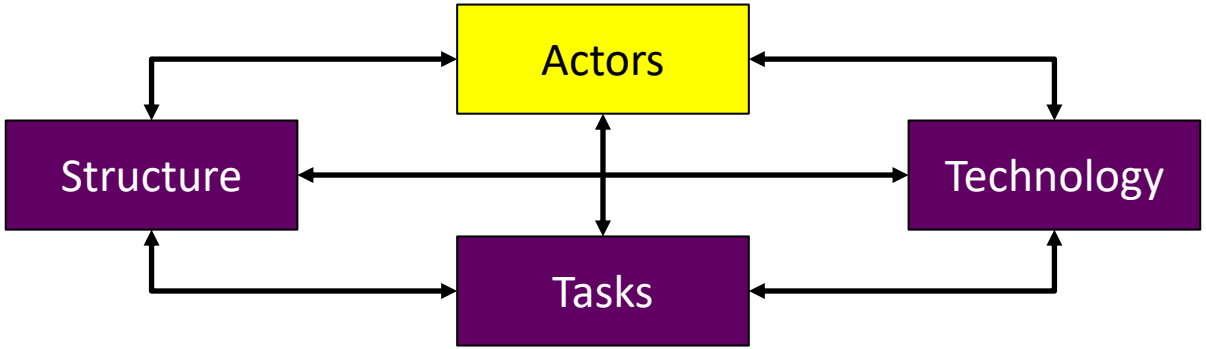
- Introduction
- **Categories of risk**
- Dealing with risk in software projects
- Evaluating risks to the schedule using PERT

Categories of Risk



Based on a socio-technical model of system development (Lyytinen, Mathiassen & Ropponen, 1998)

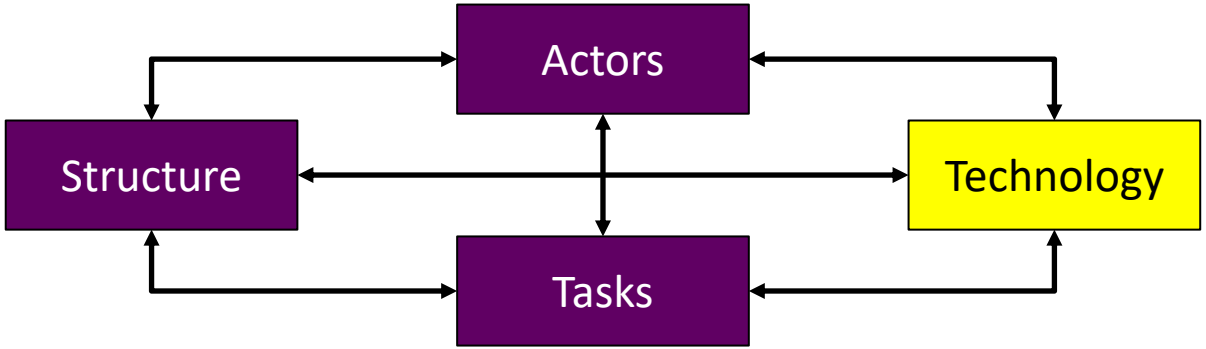
Categories of Risk



Based on a socio-technical model of system development (Lyytinen, Mathiassen & Ropponen, 1998)

- **Actors** include people, all those involved in the project - developers, users, managers, etc.
- For example, high turnover/staff leaving leads to loss of information that is importance to the project

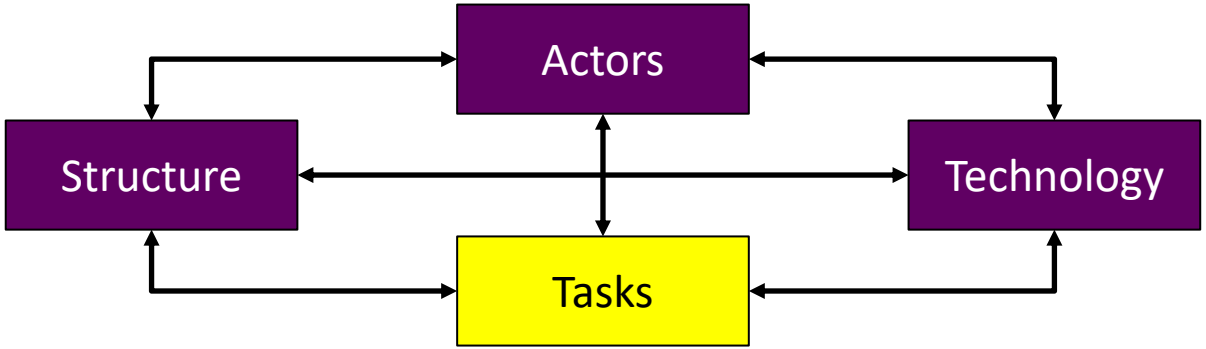
Categories of Risk



Based on a socio-technical model of system development (Lyytinen, Mathiassen & Ropponen, 1998)

- **Technology** relates to development tools and techniques used to implement the project and to technology embedded in the project deliverables

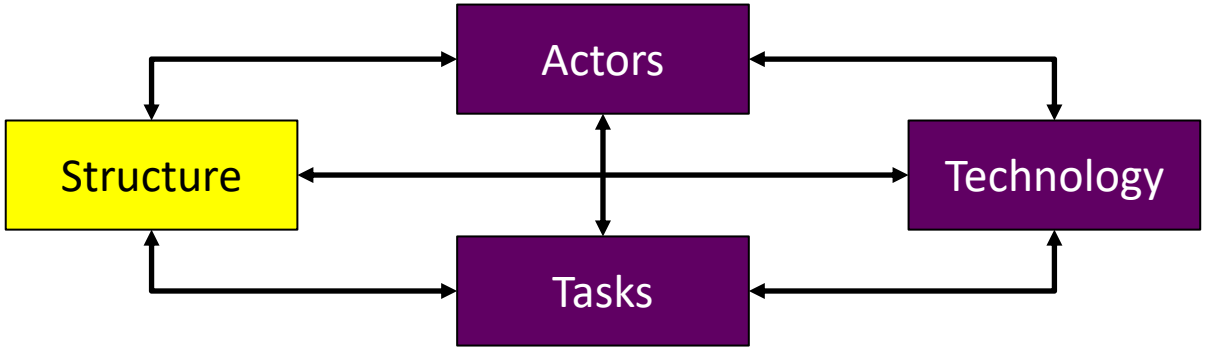
Categories of Risk



Based on a socio-technical model of system development (Lyytnien, Mathiassen & Ropponen, 1998)

- **Tasks** represent the work to be carried out
- A typical risk is that the amount of effort needed to carry out the task is underestimated, so work is not completed by the deadline

Categories of Risk



Based on a socio-technical model of system development (Lyytinen, Mathiassen & Ropponen, 1998)

- **Structure** covers management procedures
- For example, a group assigned a project task is not informed of the assignment because it is not part of the project communication network, so task completion is delayed

Outline

- Introduction
- Categories of risk
- **Dealing with risk in software projects**
- Evaluating risks to the schedule using PERT

A Risk Management Framework

The planning for risk includes four steps:

1. **Risk identification** – what risks might we find?

2. **Risk analysis and prioritisation** – which are the most serious risks?
   - Quantitative risk analysis
   - Qualitative risk analysis

3. **Risk planning** – what are we going to do about them?

4. **Risk monitoring** – what is the current state of the risk?

Risk Identification

The most serious effects of risk are:

- Failure to keep within the **cost** estimate
- Failure to achieve the required completion **date (schedule)**
- Failure to achieve the required **quality** or operational requirements

Boehm's Top Ten Risks (1/2)

	Risk Item	Risk Management Technique
1	**Personnel shortfalls**	Staffing with top talent, job matching, team building, key personnel agreements, cross training.
2	**Unrealistic schedules and budgets**	Detailed multisource cost and schedule estimation, design to cost, incremental development, software reuse, requirements scrubbing.
3	**Developing the wrong functions & properties**	Organisation analysis, mission analysis, operations-concept formulation, user surveys and user participation, prototyping, early users' manuals, off-nominal performance analysis, quality-factor analysis.
4	**Developing the wrong user interface**	Prototyping, scenarios, task analysis, user participation.
5	**Gold-plating (superfluous features)**	Requirements scrubbing, prototyping, cost-benefit analysis, designing to cost.

Boehm's Top Ten Risks (2/2)

	Risk Item	Risk Management Technique
6	**Continuing stream of requirements changes**	High change threshold, incremental development (deferring changes to later increments).
7	**Shortfalls in externally-furnished components**	Benchmarking, inspections, reference checking, compatibility analysis.
8	**Shortfalls in externally-performed tasks**	Reference checking, pre-award audits, award-fee contracts, competitive design or prototyping, team-building.
9	**Real-time performance shortfalls**	Simulation, benchmarking, modelling, prototyping, instrumentation, tuning.
10	**Straining computer science capabilities**	Technical analysis, cost-benefit analysis, prototyping, reference checking.

Quantitative Risk Analysis

- Numeric estimate of the overall risk on project objectives

- Used for projects:
  - that require a contingency reserve for schedule and budget
  - that are large and complex, involving go/no-go decisions
  - for which senior management wants more detail

- Techniques
  - Three-point estimate
  - Expected Monetary Value (EMV) or Risk Exposure
  - Program Evaluation and Review Technique (PERT)

Quantitative Risk Analysis

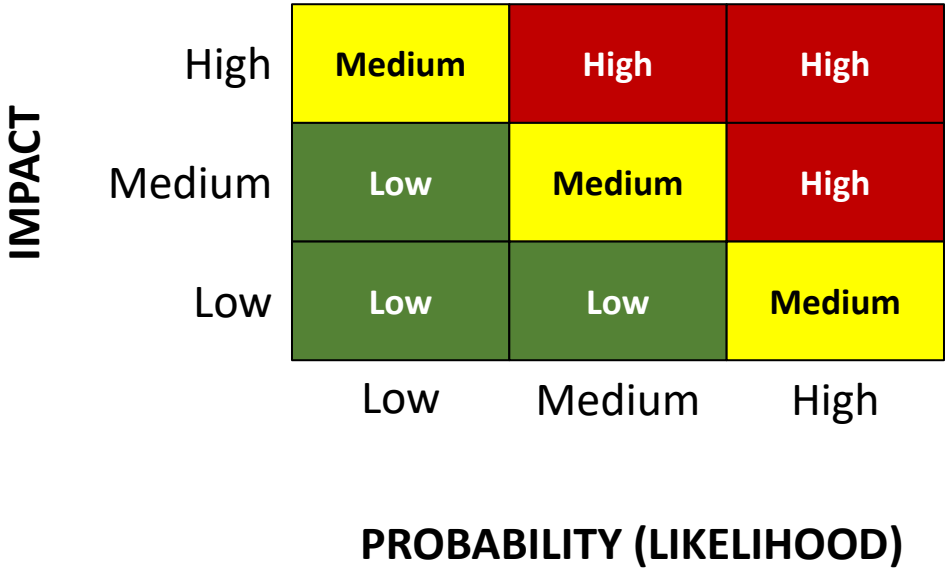
- A key quantitative indicator in risk analysis and prioritisation is **probability**
  - Lowest probability is 0 (absolutely no chance - 0%)
  - Highest probability is 1 (absolutely certain - 100%)
  - In practice, risks tend to be one extreme or the other very rarely
- Another quantitative indicator is the **cost** of the risk occurring:
  - Money
  - Time
  - A quality metric, such as defects per 1,000 lines of code, increasing
- The **threat** can be readily quantified by multiplying these
  - Probability x impact

Qualitative Risk Analysis

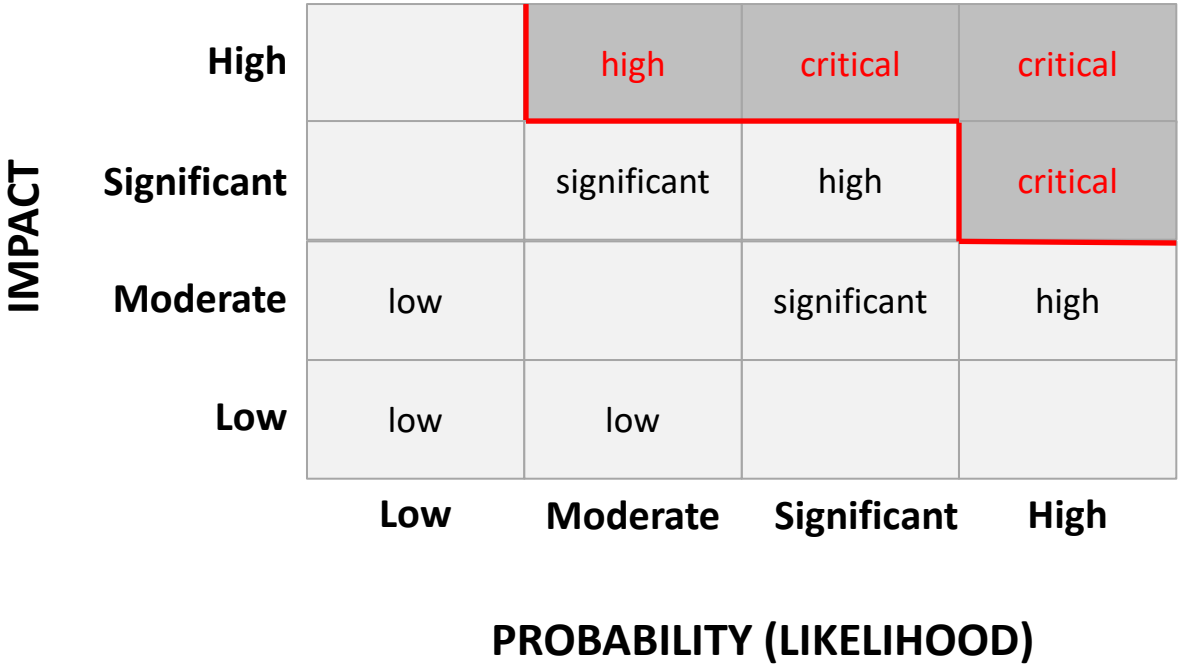
- Prioritisation of risks uses a pre-defined rating scale, which considers **likelihood** and **impact**
  - **More subjective evaluation of both probability and impact**
  - Quick and easy to perform
  - No special software or tools are required

Probability-impact matrix

Risk matrix



Probability-impact matrix



- Risk exposure cannot be calculated
- An area (in the top-right) is selected as containing tasks that require risk planning

How to deal with risks (5 ways)

- **Risk acceptance** – when the cost of avoiding the risk is (estimated to be) greater than the actual cost of the damage that might be inflicted

- **Risk avoidance** – avoid the cause associated with the risk
  - e.g., buying an off-the-shelf application avoids the risk associated with developing the application (such as poor estimates of effort)

- **Risk reduction** – actions are taken to reduce the likelihood of the risk
  - e.g., prototyping ought to reduce the risk of incorrect requirements; or offer generous bonuses on project completion to reduce likelihood that staff may leave during project

How to deal with risks (5 ways)

- **Risk transfer** – the risk is transferred to another person or organisation
  - e.g., the risk of incorrect development estimates can be transferred by negotiating a fixed price contract with an external software supplier

- **Risk mitigation/contingency measures** – actions are taken to reduce the impact if the risk does occur
  - need to monitor progress of project activities to identify occurrence of causes of risk
  - e.g., backups can be taken to allow rapid recovery in the case of data corruption

Cost-effectiveness of risk actions

- Addressing a risk usually has a cost associated with it, and the cost-effectiveness can be measured:

$$\mathbf{leverage = (RE_{before} - RE_{after}) / cost}$$

- **leverage: how much have we reduced risk** (should be > 1.00)
  - RE_{before}: the risk exposure cost before the intervention
  - RE_{after}: the risk exposure cost after the intervention
  - cost: usually money, but not necessarily

Cost-effectiveness of risk actions

Example:

- 1% chance of a fire causing £200,000 damage
- Fire alarm costing £500 reduces probability of fire to 0.5%
- Cost-effectiveness of this risk reduction action?

  - leverage = $(RE_{before} - RE_{after}) / cost$

Cost-effectiveness of risk actions

Example (answer):

- 1% chance of a fire causing £200,000 damage
  - RE_{before} = 0.01 * 200,000 = 2,000
- Fire alarm costing £500 reduces probability of fire to 0.5%
  - RE_{after} = 0.005 * 200,000 = 1,000
  - cost = £500
- Cost-effectiveness:
  - leverage = (RE_{before} - RE_{after}) / cost
  - leverage = (2,000 - 1,000) / 500 = **2.00** **Buy the fire alarm!**

Risk Management Plan

Risk Description	Likelihood (low/med/high)	Impact (low/med/high)	Severity	Owner	Mitigation strategies
Poorly-defined project purpose	Medium	High	High	Project manager	Complete a business case...

Outline

- Introduction
- Categories of risk
- Dealing with risk in software projects
- **Evaluating risks to the schedule using PERT**

PERT

- Program Evaluation and Review Technique
- Used to determine the **expected time for project activities** based on a combination of optimistic, pessimistic and expected durations
- Can be used to calculate the probability of project overrun
- Requires an understanding of critical path analysis

What do we need?

- For the whole project:
  - A list of activities
  - Predecessor/dependency relationships
  - Critical path

- For each activity (three-point estimate):
  - Likely duration
  - Optimistic duration
  - Pessimistic duration

What do we need?

Task	Description	Predecessors	Duration Estimates		
			Optimistic	Pessimistic	Likely
A	Requirements	N/A	6	12	9
B	Software	A	5	12	8
C	Hardware	A	7	13	9
D	Training	A	12	18	14
E	Implementation	B, C	3	7	5
F	Testing	E	4	9	5

Optimistic time (the best-case scenario, where everything goes as planned)
Likely time (the most realistic time, based on experience or best estimate)
Pessimistic time (the worst-case scenario, where things go wrong and delays occur)

What do we calculate?

						1	2
Task	Description	Predecessors	Duration Estimates			Expected Duration	Variance
			Optimistic	Pessimistic	Likely		
A	Requirements	N/A	6	12	9	9.00	1.00
B	Software	A	5	12	8	8.17	1.36
C	Hardware	A	7	13	9	9.33	1.00
D	Training	A	12	18	14	14.33	1.00
E	Implementation	B, C	3	7	5	5.00	0.44
F	Testing	E	4	9	5	5.50	0.69

(For each task, new columns…)

1. We calculate the mean of (optimistic), (pessimistic) and (4 * likely) to get the **expected duration.** e.g. for A, (6 + 12 + 9 + 9 + 9 + 9) / 6 = 9.00

$$\text{Expected duration} = \frac{(\text{optimistic}) + (\text{pessimistic}) + (4 * \text{likely})}{6}$$

What do we calculate?

						1	2
Task	Description	Predecessors	Duration Estimates			Expected Duration	Variance
			Optimistic	Pessimistic	Likely		
A	Requirements	N/A	6	12	9	9.00	1.00
B	Software	A	5	12	8	8.17	1.36
C	Hardware	A	7	13	9	9.33	1.00
D	Training	A	12	18	14	14.33	1.00
E	Implementation	B, C	3	7	5	5.00	0.44
F	Testing	E	4	9	5	5.50	0.69

(For each task, new columns…)

1. We calculate the mean of (optimistic), (pessimistic) and (4 * likely) to get the **expected duration.** e.g. for A, (6 + 12 + 9 + 9 + 9 + 9) / 6 = 9.00
2. For the variance (a measure of how much these values vary), we calculate ((pessimistic - optimistic) / 6) ^ 2

e.g. for A, ((12-6) / 6) ^ 2 = 1.00

What do we calculate?

Task	Description	Predecessors	Duration Estimates			Expected Duration	Variance
			Optimistic	Pessimistic	Likely		
A	Requirements	N/A	6	12	9	9.00	1.00
B	Software	A	5	12	8	8.17	1.36
C	Hardware	A	7	13	9	9.33	1.00
D	Training	A	12	18	14	14.33	1.00
E	Implementation	B, C	3	7	5	5.00	0.44
F	Testing	E	4	9	5	5.50	0.69
						Sum CP: 28.33	3.14

- We need to know the critical path, which can be determined using an **activity network**
- For the sake of simplicity, the critical path is highlighted here in yellow
- Only these items feed into subsequent calculations
  - The sum of expected duration CP and the sum of variances CP)

What do we calculate?

1	2	3	4	5
Total Duration	Total Variance	Target Date	Z-Score	Probability of Target
28.83	3.14	28	-0.470	31.90%
28.83	3.14	29	0.094	53.75%
28.83	3.14	30	0.659	74.49%
28.83	3.14	31	1.223	88.93%
28.83	3.14	32	1.787	96.31%
28.83	3.14	33	2.352	99.07%

1. Sum of all expected durations on the critical path
2. Sum of all variance values on the critical path

All rows now will take these same values into account

What do we calculate?

1	2	3	4	5
Total Duration	Total Variance	Target Date	Z-Score	Probability of Target
28.83	3.14	28	-0.470	31.90%
28.83	3.14	29	0.094	53.75%
28.83	3.14	30	0.659	74.49%
28.83	3.14	31	1.223	88.93%
28.83	3.14	32	1.787	96.31%
28.83	3.14	33	2.352	99.07%

1. Sum of all expected durations on the critical path
2. Sum of all variance values on the critical path
3. The number of days we've told our client this will take (target date)

What do we calculate?

1	2	3	4	5
Total Duration	Total Variance	Target Date	Z-Score	Probability of Target
28.83	3.14	28	-0.470	31.90%
28.83	3.14	29	0.094	53.75%
28.83	3.14	30	0.659	74.49%
28.83	3.14	31	1.223	88.93%
28.83	3.14	32	1.787	96.31%
28.83	3.14	33	2.352	99.07%

1. Sum of all expected durations on the critical path
2. Sum of all variance values on the critical path
3. The number of days we've told our client this will take (target date)
4. Z-score: a measure of the difference between columns 1 and 3:
   - (Column 3 - Column 1) / Square Root (Column 2)
   - (Target Date – Total Duration) / (Standard Deviation)

What do we calculate?

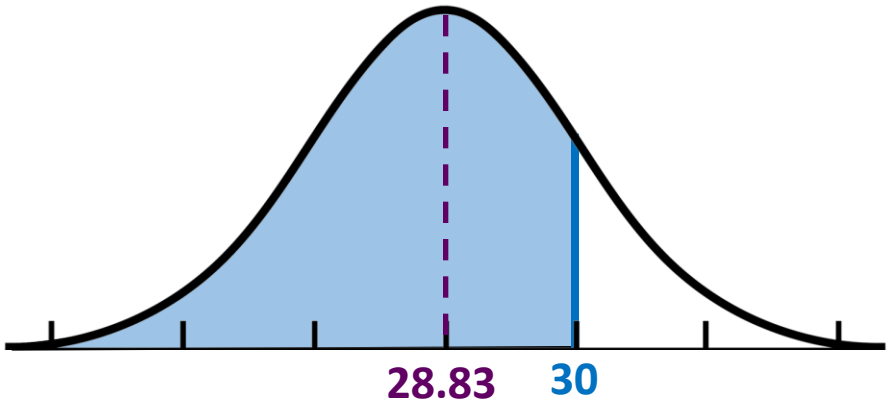
1	2	3	4	5
Total Duration	Total Variance	Target Date	Z-Score	Probability of Target
28.83	3.14	28	-0.470	31.90%
28.83	3.14	29	0.094	53.75%
28.83	3.14	30	0.659	74.49%
28.83	3.14	31	1.223	88.93%
28.83	3.14	32	1.787	96.31%
28.83	3.14	33	2.352	99.07%

1. Sum of all expected durations on the critical path
2. Sum of all variance values on the critical path
3. The number of days we've told our client this will take (target date)
4. Z-score: a measure of the difference between columns 1 and 3:
   - (Column 3 - Column 1) / Square Root (Column 2)
   - (Target Date – Total Duration) / (Standard Deviation)
5. The probability that we will meet the target date (using Z-Score)

Normal Distribution

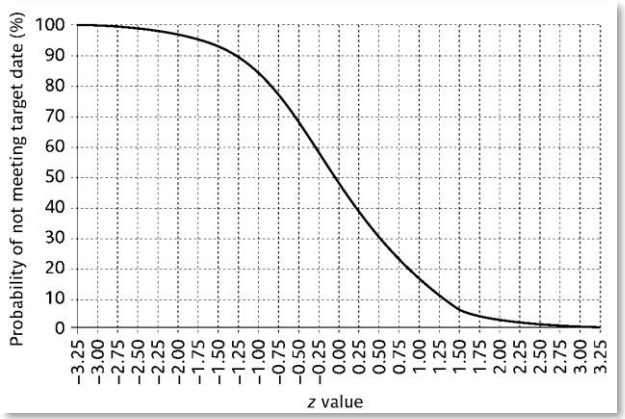
Total Duration	Total Variance	Target Date	Z-Score	Probability of Target
28.83	3.14	28	-0.470	31.90%
28.83	3.14	29	0.094	53.75%
28.83	3.14	30	0.659	74.49%
28.83	3.14	31	1.223	88.93%
28.83	3.14	32	1.787	96.31%
28.83	3.14	33	2.352	99.07%

- These calculations assume a normal distribution of potential duration; the likeliest duration is the interval in which sits 28.83 (e.g., weeks), and “longer is as likely as shorter”



What do we calculate?

Total Duration	Total Variance	Target Date	Z-Score	Probability of Target
28.83	3.14	28	-0.470	31.90%
28.83	3.14	29	0.094	53.75%
28.83	3.14	30	0.659	74.49%
28.83	3.14	31	1.223	88.93%
28.83	3.14	32	1.787	96.31%
28.83	3.14	33	2.352	99.07%



Z-table

0.659 ➡

Prob. (%) meeting target

z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767
2.0	.9772	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9864	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916

Summary of Key Points

- Risks represent potential future problems or opportunities, and link causes to their effects
- Risks in software projects are related to actors, technology, structure, tasks (or a combination thereof)
- Identified risks needs to be analysed in terms of impact and likelihood
- High-impact and/or high-likelihood risks may need to be addressed by eliminating or reducing their likelihood or impact
- Risks to the schedule can be analysed using PERT

References (further reading)

- Boehm, B. (1989) 'Software risk management', *European Software Engineering Conference, Berlin*, vol. 387. https://doi.org/10.1007/3-540-51635-2_29

- Hughes, B. & Cotterell, M., (2009) *Software Project Management*, 5th ed., London: McGraw-Hill. (Chapter on Risks - pp.162-191)

- Lyytinen, K., Mathiassen, L. & Ropponen, J. (1998) 'Attention Shaping and Software Risk – A Categorical Analysis of Four Classic Risk Management Approaches', *Information Systems Research*, 9(3), pp.233-255. https://pubsonline.informs.org/doi/pdf/10.1287/isre.9.3.233

Aston University
BIRMINGHAM UK

Software Project Management

Unit 5: Risk Management

Thais Webber
Richard Lee