Learning Resource
On
Software Project Management

SPM_Unit-4
Risk Management

Prepared By:

Kunal Anand

Assistant Professor, SCE

KIIT, DU, Bhubaneswar-24

Talkflow

- Introduction to Risk
- Risk categories
- Risk Management
  - Risk Identification
  - Risk Analysis
  - Risk Planning
  - Risk Monitoring
- Using PERT in risk management
- Monte Carlo simulation
- Critical Chain approach

Some definitions 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': PM-BOK

- In simple terms, risk is any adverse situation that can hamper the software development activity. Risks relate to **possible future** problems, not current ones.

- They involve a possible **cause and its effect(s)** e.g. developer leaves -> task delayed

Categories of Risk

Three main categories of risks:

- **Project risks:** risks concern varies forms of budgetary, schedule, personnel, resource, and customer-related problems. An important project risk is schedule slippage.

- **Technical risks:** risks concern potential design, implementation, interfacing, testing, and maintenance problems. Most technical risks occur due to the development team's insufficient knowledge about the project.

- **Business risks:** risks include risks of building an excellent product that no one wants, losing budgetary or personnel commitments, etc.

Risk Management

The planning for risk includes these steps:

- **Risk identification** – what risks might there be?
- **Risk analysis and prioritization** – which are the most serious risks?
- **Risk planning** – what are we going to do about them?
- **Risk monitoring** – what is the current state of the risk?

Risk Identification

Approaches to identify risks include:

- **Use of checklists** – usually based on the experience of past projects

- **Brainstorming** – getting knowledgeable stakeholders together to pool concerns

- **Causal mapping** – identifying possible chains of cause and effect

Boehm's top 10 development risks

Risk	*Risk reduction techniques*
Personnel shortfalls	Staffing with top talent; job matching; teambuilding; training and career development; early scheduling of key personnel
Unrealistic time and cost estimates	Multiple estimation techniques; design to cost; incremental development; recording and analysis of past projects; standardization of methods
Developing the wrong software functions	Improved software evaluation; formal specification methods; user surveys; prototyping; early user manuals
Developing the wrong user interface	Prototyping; task analysis; user involvement

Boehm's top ten risk (contd..)

Gold plating (inclusion of features that in fact are unnecessary and which end up never actually being used.)	Requirements scrubbing, prototyping, design to cost
Late changes to requirements	Change control, incremental development
Shortfalls in externally supplied components	Benchmarking, inspections, formal specifications, contractual agreements, quality controls
Shortfalls in externally performed tasks	Quality assurance procedures, competitive design etc
Real time performance problems	Simulation, prototyping, tuning
Development technically too difficult	Technical analysis, cost-benefit analysis, prototyping , training

Risk Prioritization

- **Risk exposure (RE)**

  = (potential damage) x (probability of occurrence)

- *Ideally*
  - **Potential damage**: a money value e.g. a flood would cause Rs. 0.5 millions of damage
  - **Probability** 0.00 (absolutely no chance) to 1.00 (absolutely certain) e.g. 0.01 (one in hundred chance)
- **Risk probability: qualitative descriptors**

Probability level	***Range***
High	Greater than 50% chance of happening
Significant	30-50% chance of happening
Moderate	10-29% chance of happening
Low	Less than 10% chance of happening

Qualitative descriptors of impact on cost and associated range values

Impact level	*Range*
High	Greater than 30% above budgeted expenditure
Significant	20 to 29% above budgeted expenditure
Moderate	10 to 19% above budgeted expenditure
Low	Within 10% of budgeted expenditure.

Risk Planning

- Different risks require different containment procedures.

- Three main strategies to plan for risk containment:
  - **Avoid the risk:** discussing with the customer to change the requirements to reduce the scope of the work, giving incentives to the engineers to avoid the risk of manpower turnover, etc.
  - **Transfer the risk:** Involves getting the risky component developed by a third party, buying insurance cover, etc.
  - **Risk reduction:** Planning ways to contain the damage due to a risk. For example, if there is risk that some key personnel might leave, new recruitment may be planned.

Risk Reduction Leverage (RRL)

- **Risk Reduction Leverage (RRL)** can be defined as the variation in risk exposure divided by the amount of reducing the risk.
- Mathematically, Risk Reduction Leverage can be represented as below:
  - **$RRL = (RE_{before} - RE_{after}) / (cost\ of\ reduction)$**
  - **RE_{before}** is risk exposure before risk reduction e.g. 1% chance of a fire causing £200k damage
  - **RE_{after}** is risk exposure after risk reduction e.g. fire alarm costing Rs. 500 reduces probability of fire damage to 0.5%
- If the **RRL < 1**, it means that the cost of the risk reduction activity outweighs the probable gain from implementing the action i.e. one should opt for a risk reduction solution only when the **RRL > 1**.
- **Example:** RRL = (1% of Rs. 200k)-(0.5% of Rs. 200k)/Rs. 500 = 2

  RRL > 1.00 therefore worth doing

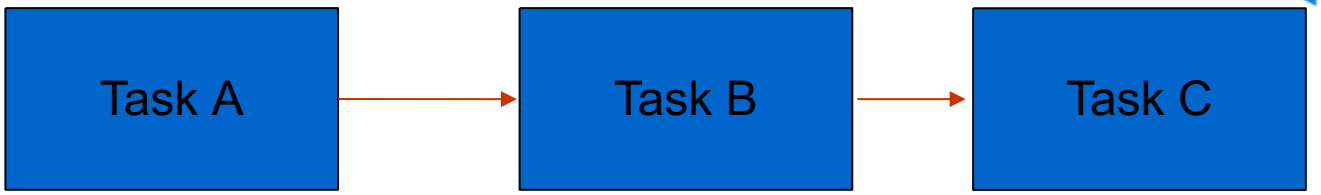
Using PERT to evaluate the effects of uncertainty

Three estimates are produced for each activity

- **Most likely time (m):** the time taken in most general scenario
- **Optimistic time (a):** the time taken in the best case scenario
- **Pessimistic (b):** the time taken in worst case scenario
- **Expected Time (t_e)** = (a + 4m +b) / 6
- **Activity Standard Deviation (S)** = (b-a) / 6
- **Standard Deviation of the Project**

$$\surd \textit{sum of square of standard deviation of critical activitites}$$

A chain of activities



Task	a	m	b	t_e	s
A	10	12	16	?	?
B	8	10	14	?	?
C	20	24	38	?	?

What would be the expected duration of the chain **A** + **B** + **C**?

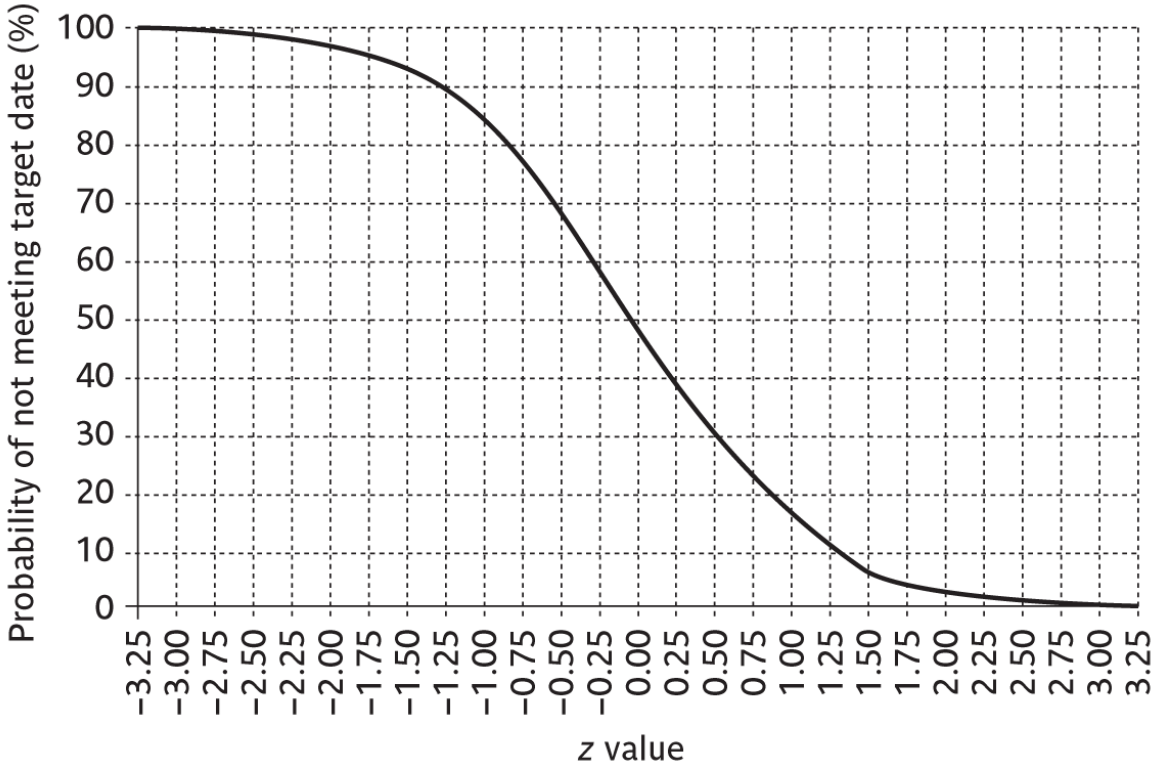
Answer: 12.66 + 10.33 + 25.66 i.e. 48.65

What would be the standard deviation for **A** + **B**+ **C**?

Answer: square root of ($1^2 + 1^2 + 3^2$) i.e. 3.32

Assessing the likelihood of meeting a target

- Say the target for completing A+B+C was 52 days (T)
- Calculate the z value; $\mathbf{z = (T - t_e)/s}$
- In this example, **z = (52-48.33)/3.32 i.e. 1.01**
- Look up in table of z values – see next overhead



Monte Carlo Simulation

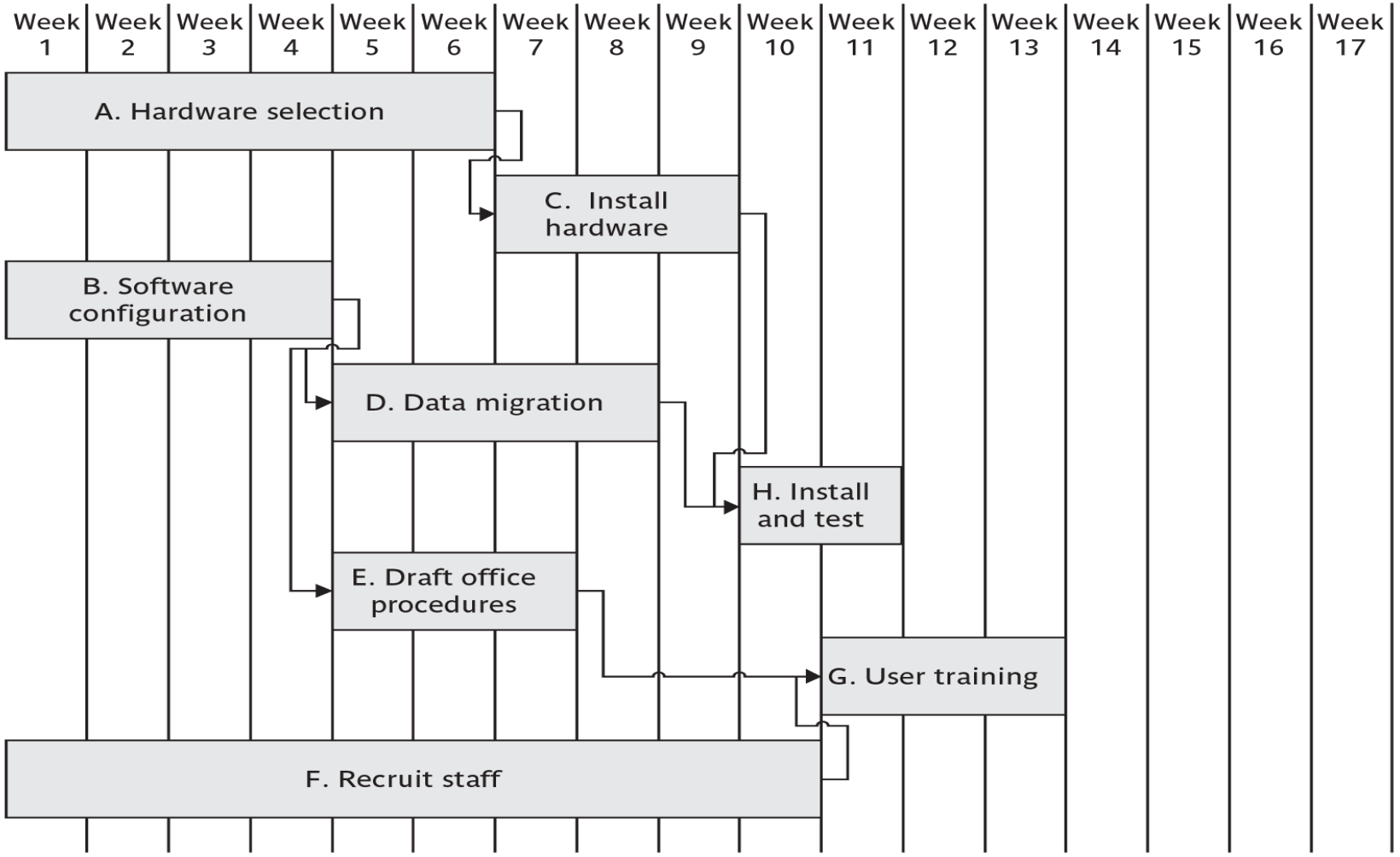
- **Monte Carlo Simulation** is a quantitative risk analysis technique used in software project management to estimate project timelines, costs, and risks by running multiple simulations based on probability distributions.
- It helps project managers understand potential outcomes and uncertainties.
- **Key Concepts:**
  - **Probability-Based Estimation:** Instead of using a single-point estimate, the simulation uses multiple possible values (optimistic, most likely, pessimistic).
  - **Random Sampling:** Runs thousands of iterations, each time selecting random values within the estimated range.
  - **Statistical Analysis:** Provides insights into the likelihood of meeting deadlines, budget constraints, and project risks.
  - **Visualization:** Generates probability distributions, histograms, and cumulative probability graphs to help decision-making.

Steps in Monte Carlo Simulation

- **Define Project Variables:** Identify uncertain factors like task duration, cost, or resource availability.
- **Assign Probability Distributions:** Use distributions like Normal, Triangular, or Beta for estimations (e.g., PERT).
- **Run Simulations:** Generate thousands of random scenarios using software tools (e.g., @RISK, Primavera Risk Analysis, or Excel).
- **Analyze Results:** Examine probability curves to determine risks and possible completion dates.
- **Make Informed Decisions:** Adjust resource allocation, buffer time, or budgets based on risk exposure.

Critical chain concept

Traditional planning approach



Critical chain approach

- One problem with estimates of task duration:
  - Estimators add a safety zone to estimate to take account of possible difficulties
  - Developers work to the estimate + safety zone, so time is lost
  - No advantage is taken of opportunities where tasks can finish early – and provide a buffer for later activities
- One answer to this:
  - Ask the estimators for two estimates
    - **Most likely duration:** 50% chance of meeting this
    - **Comfort zone:** additional time needed to have 95% chance
  - Schedule all activities using most likely values and starting all activities on latest start dates

Most likely and Comfort Zone estimates

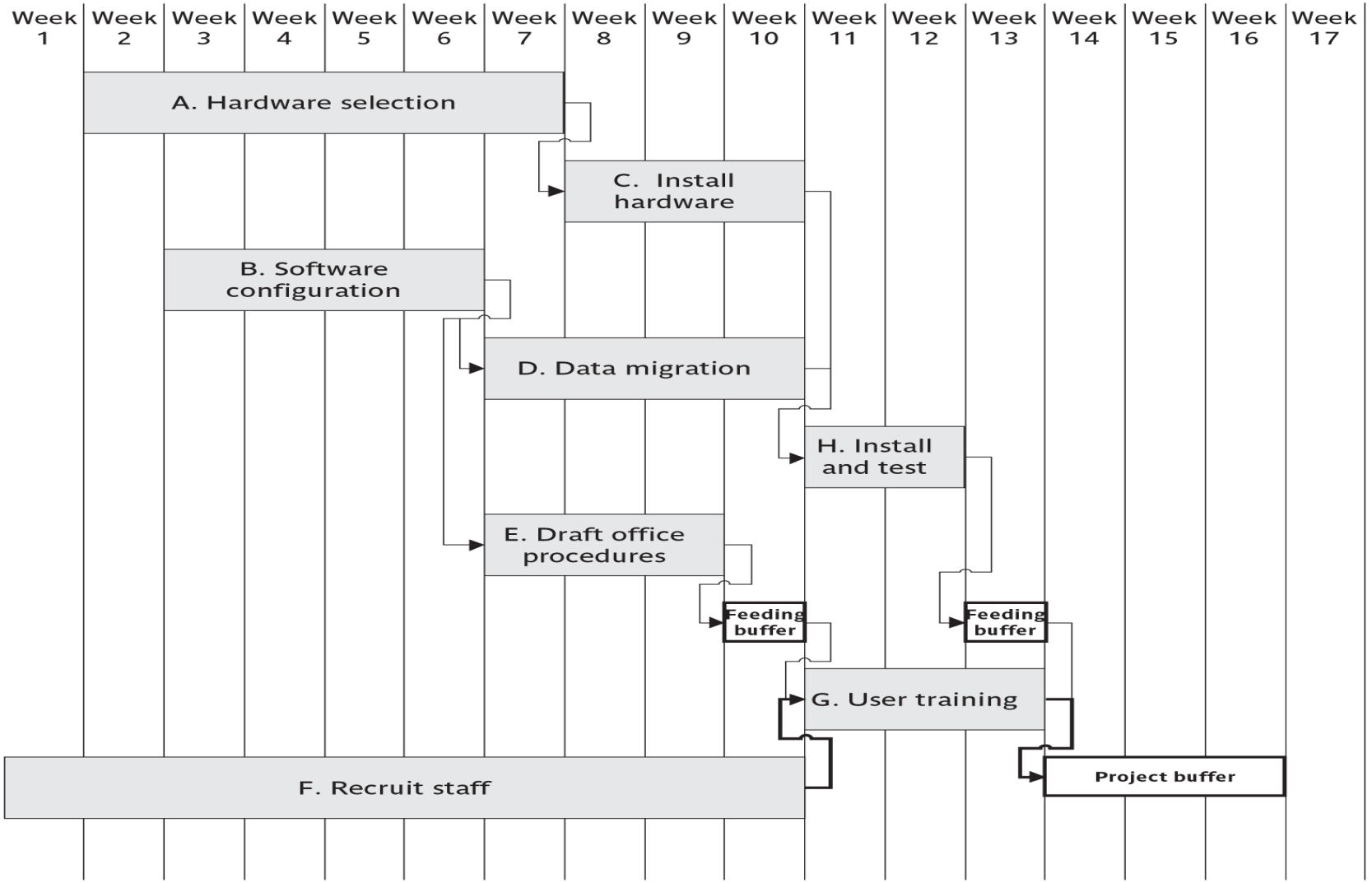
Activity	Most likely	Plus comfort zone	Comfort zone
A	6	8	2
B	4	5	1
C	3	3	0
D	4	5	1
E	3	4	1
F	10	15	5
G	3	4	1
H	2	2.5	0.5

TABLE 7.8 Most likely and comfort zone estimates (days)

Critical chain - continued

3. Identify the **critical chain** – same as critical path but resource constraints also taken into account
4. Put a **project buffer** at the end of the critical chain with duration **50% of sum of comfort zones of the activities** on the **critical chain**.
5. Where subsidiary chains of activities **feed into critical chain**, add **feeding buffer** with duration **50% of sum of comfort zones of activities** in the **feeding chain**
6. Where there are **parallel chains**, take the **longest and sum those activities**

Plan employing critical chain concepts



Executing the critical chain-based plan

- No **chain** of tasks is started earlier than scheduled, but once it has started is finished as soon as possible
- This means the activity following the current one starts as soon as the current one is completed, even if this is early – the relay race principle
- Buffers are divided into three zones:
  - Green: the first 33%. No action required
  - Amber : the next 33%. Plan is formulated
  - Red : last 33%. Plan is executed.