Topics

✓ Introduction

✓ Software Engineering Models

✓ BPM, BRMS, SOA

✓ Software Requirements Engineering

✓ Software Analysis Models

➢ Software Project Management

7. Software Design Concepts, Principles and Models
8. Software Coding Practices
9. Software Testing Techniques
10. Software Quality Assurance
11. Emerging Trends in Software Engineering

Software Project Management

- ➢ Software Pricing
- Effort Estimation Methods
- Software Project Management: Planning and Scheduling
- Risk Analysis and Mitigation

Software pricing

- ✧ Estimates are made to discover the cost, to the developer, of producing a software system.
  - ▪ You take into account, hardware, software, travel, training and effort costs.

- ✧ There is not a simple relationship between the development cost and the price charged to the customer.

- ✧ Broader organisational, economic, political and business considerations influence the price charged.

Factors affecting software pricing

Factor	Description
Market opportunity	A development organization may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the organization the opportunity to make a greater profit later. The experience gained may also help it develop new products.
Cost estimate uncertainty	If an organization is unsure of its cost estimate, it may increase its price by a contingency over and above its normal profit.
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.

Factors affecting software pricing

Factor	Description
Requirements volatility	If the requirements are likely to change, an organization may lower its price to win a contract. After the contract is awarded, high prices can be charged for changes to the requirements.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a smaller than normal profit or break even than to go out of business. Cash flow is more important than profit in difficult economic times.

Software Project Management

✓ Software Pricing

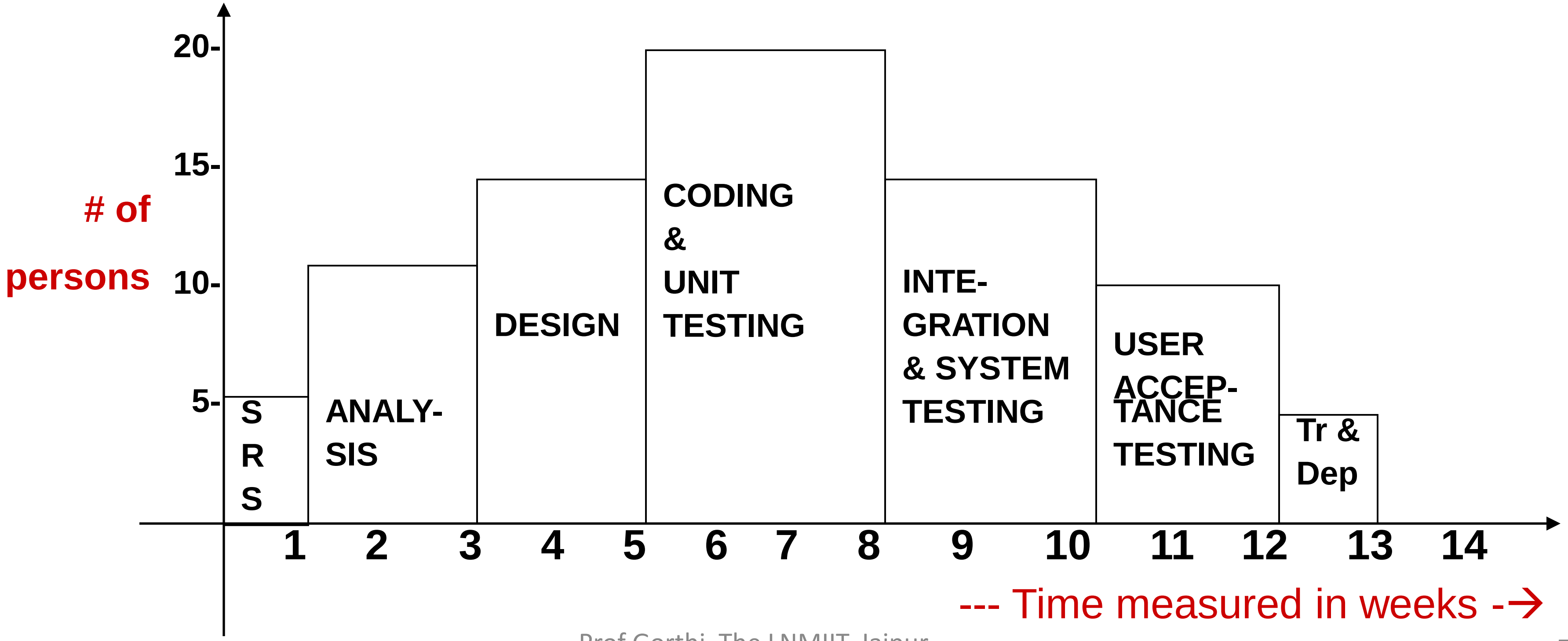
➢ Effort Estimation Methods

- Software Project Management: Planning and Scheduling

Software Project Effort Estimation

What is meant by software project effort? How is it measured?

- Software project effort is the integral of the graph of the people worked on a project since its inception till it is deployed



Software Project Effort Estimation

What is meant by software project effort estimation?

- It is to estimate the effort in person-weeks (person-months) of completing the phases of,
  - Elaboration,
  - Construction and
  - Transition

  using the SRS and Analysis Models

- Would the number of pages of SRS or UCADS give you an estimate? NO
- Function-Point / COCOMO models are better models

Function Point Analysis

What is Function Point Analysis (FPA)?

- It is designed to estimate and measure the time, and thereby the cost, of developing new software applications and maintaining existing software applications.
- It is also useful in comparing and highlighting opportunities for productivity improvements in software development.
- It was developed by A.J. Albrecht of the IBM Corporation in the early 1980s.

Function Point Analysis

How is Function Point Analysis done?

Using the SRS and Analysis Models, the following software functions are measured (counted):

- Inputs
- Outputs
- Data Stores
- Processing Inquires
- Processing Updates
- External Interfaces

Function Point Analysis

How is Function Point Analysis done?

Whether the following software functions are simple, average or complex are counted:

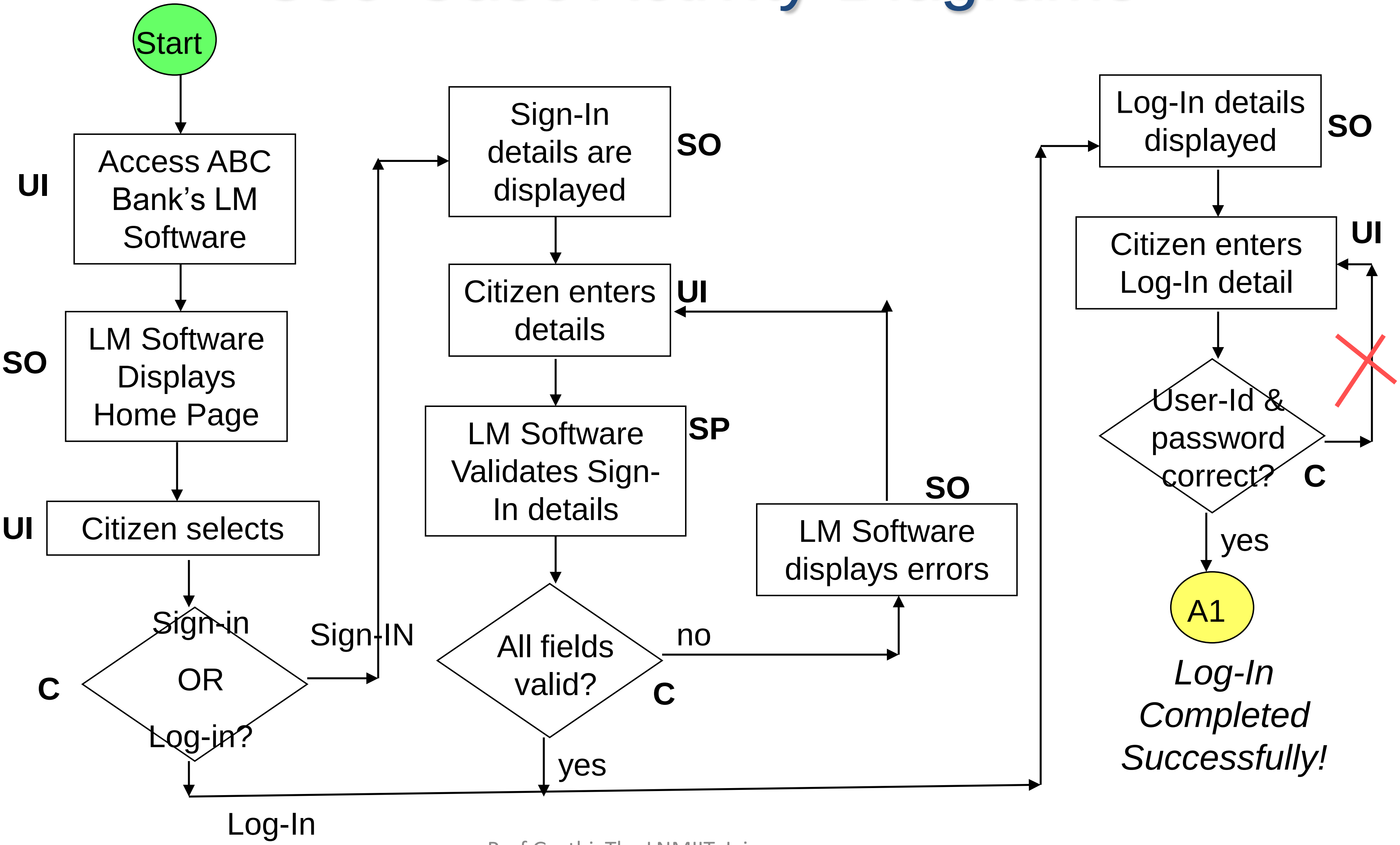
Inputs	# of S	# of A	# of C
Outputs	# of S	# of A	# of C
Data Stores	# of S	# of A	# of C
Processing Inquires	# of S	# of A	# of C
Processing Updates	# of S	# of A	# of C
External Interfaces	# of S	# of A	# of C

Function Point Analysis

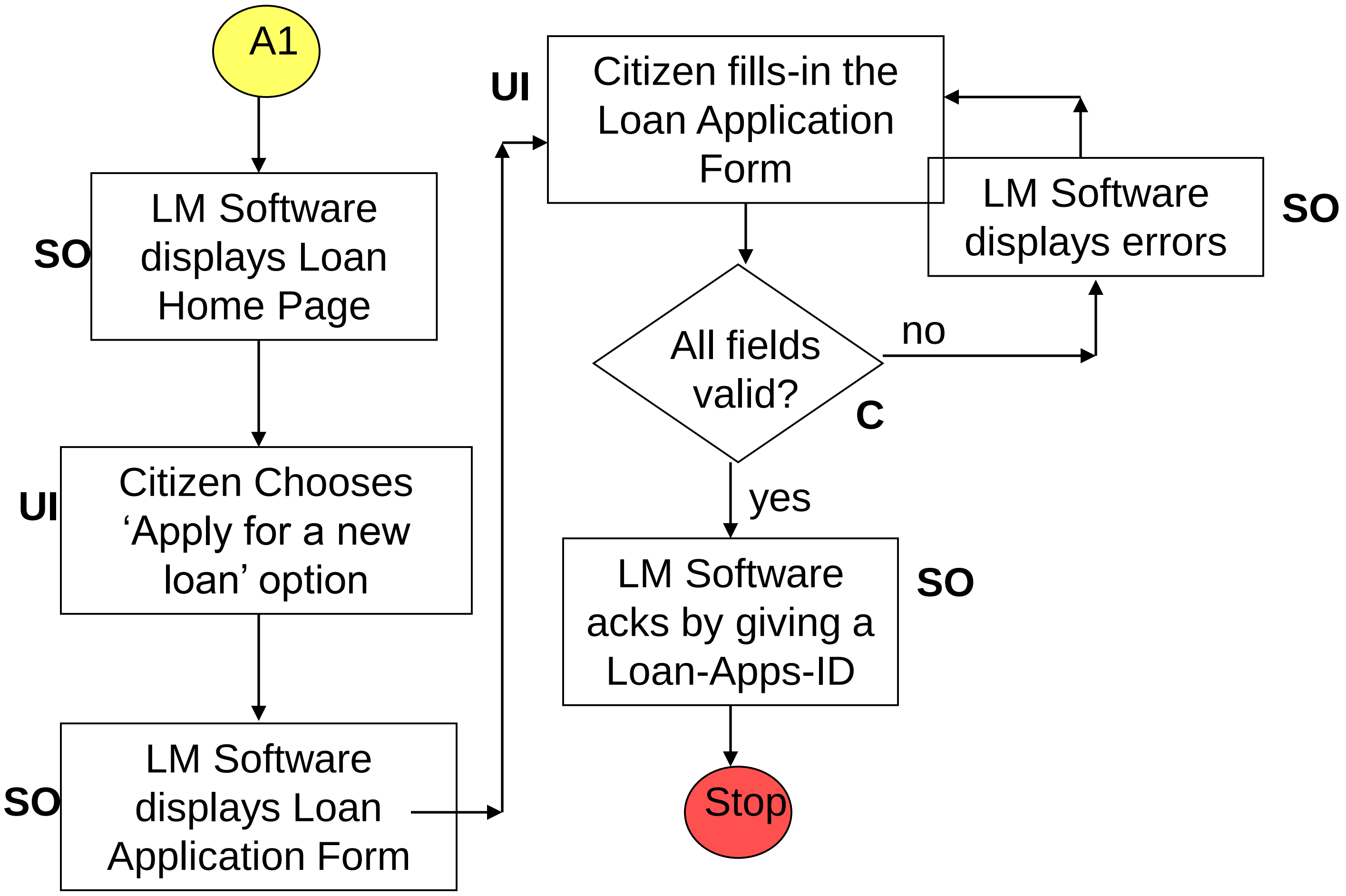
How is Function Point Analysis done?

```
Int S-UI, A-UI, C-UI, S-SO, A-SO, C-SO, S-DS, A-DS, C-DS, S-PI, A-PI, C-PI, S-PU, A-PU, C-PU, S-EI, A-EI, C-EI = 0;
{for each Object $O_J$ in the TBD software package do
 {for each Use-Case $UC_k$ do
   {for each Triggering-Action $TA_L$ do
     {for each Unit-of-Behavior $UoB_M$ do
       if  the UI is simple then +S-UI else if average then +A-UI else +C-UI;
       if  the SO is simple then +S-SO else if average then +A-SO else +C-SO;
       if  the DS is simple then +S-DS else if average then +A-DS else +C-DS;
       if  the PI is simple then +S-PI else if average then +A-PI else +C-PI;
       if  the PU is simple then +S-PU else if average then +A-PU else +C-PU;
       if  the EI is simple then +S-EI else if average then +A-EI else +C-EI;
     }; }; }; };
```

Elaborating Use Cases as Use-Case Activity Diagrams


Start
UI
Access ABC Bank's LM Software
SO
LM Software Displays Home Page
UI
Citizen selects
C
Sign-in OR Log-in?
Sign-IN
Log-In
Sign-In details are displayed
SO
Citizen enters details
UI
LM Software Validates Sign-In details
SP
All fields valid?
C
no
yes
SO
LM Software displays errors
Log-In details displayed
SO
Citizen enters Log-In detail
UI
User-Id & password correct?
C
yes
A1
Log-In Completed Successfully!

Elaborating Use Cases as Use-Case Activity Diagrams


A1
SO
LM Software displays Loan Home Page
UI
Citizen Chooses ‘Apply for a new loan’ option
SO
LM Software displays Loan Application Form
UI
Citizen fills-in the Loan Application Form
All fields valid?
C
no
LM Software displays errors
SO
yes
LM Software acks by giving a Loan-Apps-ID
SO
Stop

Function Point Analysis

How is Function Point Analysis done?

For the Loan-Management software package, the following are the counts:

• Inputs	3	4	1
• Outputs	2	6	2
• Data Stores	0	2	2
• Processing Inquires	2	8	4
• Processing Updates	2	8	4
• External Interfaces	0	0	1

Function Point Analysis

These function-point counts are then weighed (multiplied) by their degree of complexity:

	WEIGHTS		
	Simple	Average	Complex
Inputs	2	4	6
Outputs	3	5	7
Data Stores	5	10	15
Processing Inquires	2	4	8
Processing Updates	4	8	12
External Interfaces	4	6	8

Function Point Analysis

For the LM example:

Inputs

3 simple	X 2	= 06
4 average	X 4	= 16
1 complex	X 6	= 06

O***utputs***

2 simple	X 3	= 06
6 average	X 5	= 30
2 complex	X 7	= 14

Data Stores

2 average	X 10	= 20
2 complex	X 15	= 30

Processing Inquiries

2 simple	X 2	= 04
8 average	X 4	= 32
4 complex	X 8	= 32

Processing Updates

2 simple	X 4	= 08
8 average	X 8	= 64
4 complex	X 12	= 48

External Interfaces

1 complex	X 8	= 32

Unadjusted function points 348

Function Point Analysis

In addition to these individually weighted function points, there are global factors that affect the project and/or software as a whole.

There are 14 factors that affect the size of the project effort, and each is ranked from “0”- no influence to “5”- essential.

The following are some examples of these factors:

- Is the software to be run as a web-based apps?
- Is high performance critical?
- Is the software to be used in multiple sites?
- Is the code designed to be reusable?
- Is the processing to be distributed? and so forth . . .

This adjustment influence (AI) can range from 0 (14 * 0) to 70 (14 * 5)

Function Point Analysis

Using the above mentioned adjustment influence, we calculate the Complexity-Adjustment-Factor (CAF) as

CAF = 0.65 + 0.01 * AI

When AI is the least (AI = 0), CAF will be 0.65

At the other extreme, when AI is 70, CAF will be 1.35;

Using, CAF, Adjusted FP is calculated as,

Adj-FP = Unadj-FP * CAF

Notice that, AI can swing Adj-FP from -35% to + 35%

Function Point Analysis

Continuing with our Loan Management example, the following adjustment influence factors are found:

- the software to be run as a web-based apps: 5
- high performance critical: 3
- the software to be used in multiple sites: 1
- the code designed to be reusable: 3
- the processing to be distributed: 0

and so forth . . .

CAF = 0.65 + 0.01 * 55 = 1.2

Adj-FP = 348 * 1.2 = approx 418

Function Point Analysis

But how long will the project take and how much will it cost?

- As previously measured, software professionals in our organization perform at an average of 10 function points per month. Thus . . .

  418 Adj-FP divided by 10 = ~42 person-months

- If the average software professional is paid Rs 2L per month (including benefits and overheads), then the personnel cost of the project will be . . .

  42 person-months X Rs 2L = Rs 84L

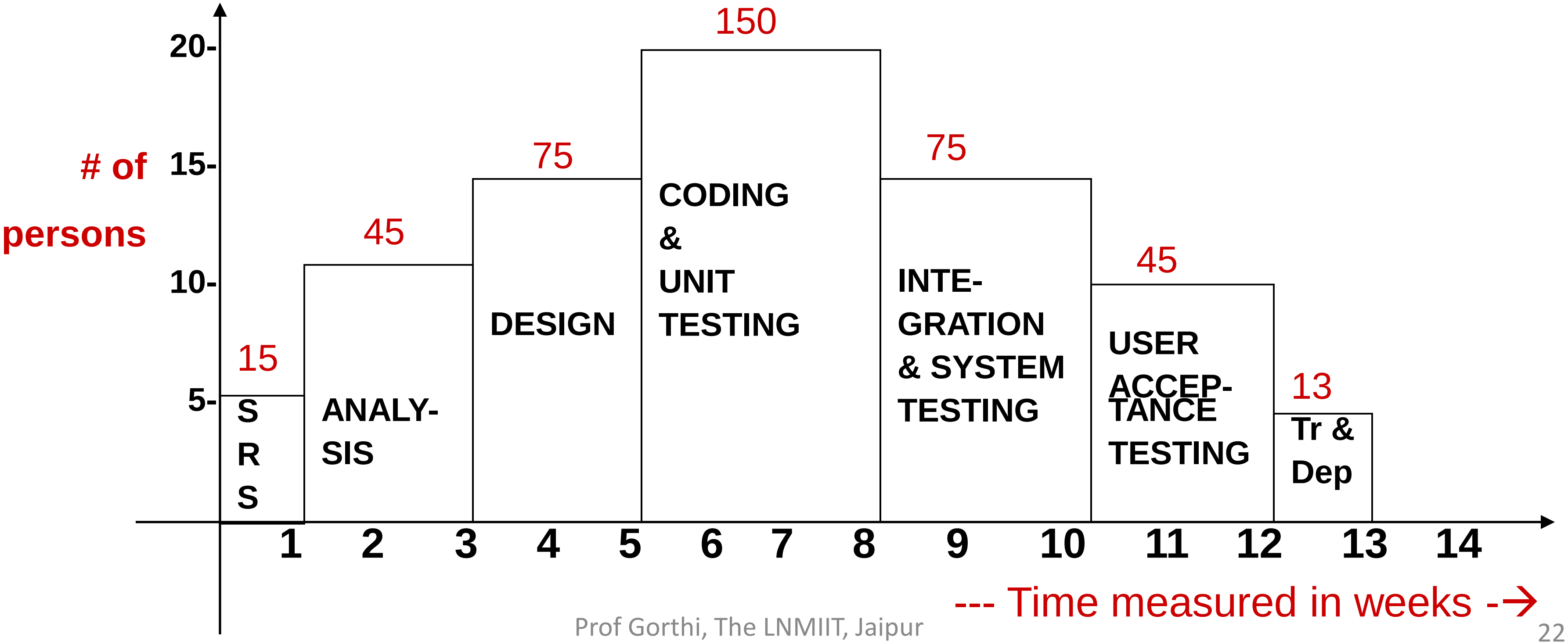
Software Project Effort Estimation

How are 420 FP distributed?

SRS: 3%; Analysis: 12%; design: 18%; Coding & Unit Testing: 35%;

Integration & System Testing: 18% User Acceptance Testing: 12%;

Training & Deployment: 2%



Function Point Analysis

FP analysis enables an organization to compare productivity of,

- One programming language over another
- One development platform over another
- One development methodology over another
- Before-and-after gains in investing in software engineers' training
- And so forth . . .

Function Point Analysis

But there are problems and criticisms:

- Function point counts are affected by past experiences, accuracy of data kept
- Difficult to apply to massively distributed development centers
- The validity of the weights that Albrecht established – and the consistency of their application – has been challenged
- Different companies will calculate function points slightly different, making inter-company comparisons questionable

Software Project Management

- ✓ Software Pricing
- ✓ Effort Estimation Methods
- ➢ Software Project Management: Planning and Scheduling

Ref: www.cse.iitb.ac.in/~cs682/reading/5-ProjPlanning-Jalote.**ppt**

Software Project Management

- Goal: To meet commitments on cost, schedule, quality

- Worldwide - many projects fail
  - one-third are runaways with cost or schedule or quality overrun of more than 125%

Software Project Management

- Major reasons for project runaways
  - unclear objectives
  - bad planning
  - no project management methodology
  - new technology
  - insufficient staff
  - new to software engineering
- All of these relate to project management
- Effective project management is key to successfully executing a project

Software Project Management

The Need . . .

- Better predictability leading to commitments that can be met
- Lower cost through reduced rework, better resource mgmt, better planning,..
- Improved quality through proper quality planning and control
- Better control through change control, monitoring etc.

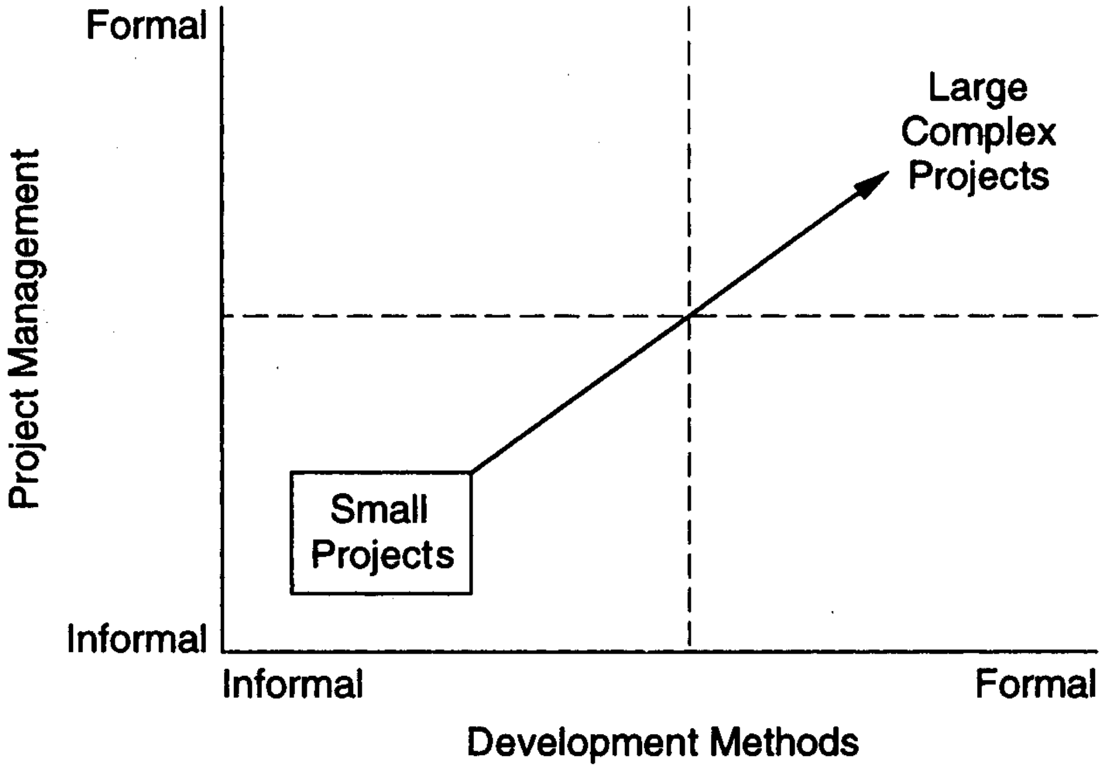
Software Project Management

The Need . . .

- Better visibility into project health and state, leading to timely intervention
- Better handling of risks, reducing the chances of failure
- All this leads to higher customer satisfaction
- And self and organization improvement

Software Project Management

The Problems of Scaling . . .



Software Project Management

Key Tasks of Planning:

- Project organization
- Risk analysis
- Hardware and software resource requirements
- Work breakdown
- Project schedule
- Monitoring and reporting mechanisms

Software Project Management

Project Scheduling:

- A project schedule is at two levels:
  - overall schedule and
  - detailed schedule

- Overall schedule comprises of major milestones and final date

- Detailed schedule is the assignment of lowest level tasks to resources

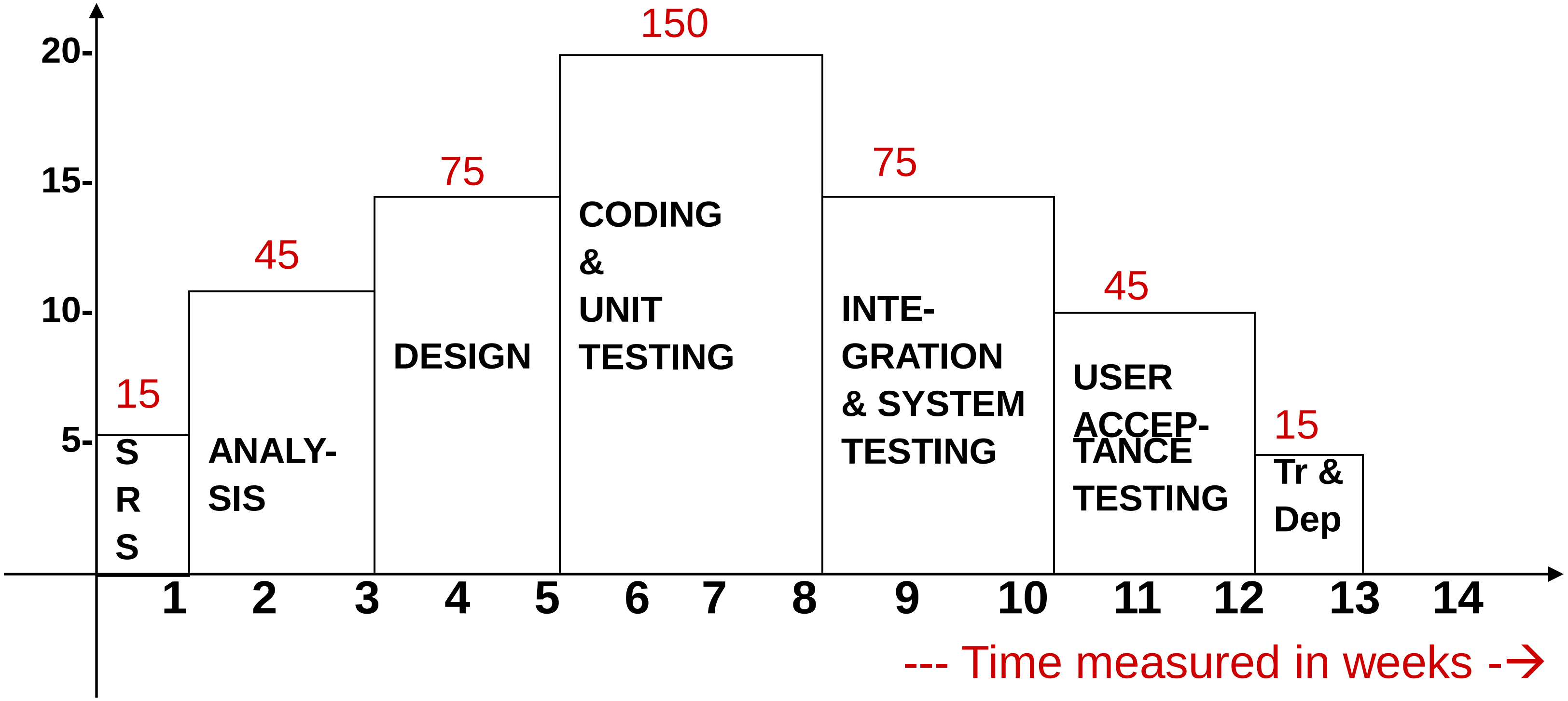
Software Project Management

- **Overall Schedule:**

#	Task	Dur. (days)	Work (p-days)	Start Date	End Date
1	Training	50	24	5/4	6/22
2	Knowledge Sharing	105	49	5/8	8/20
3	Inception and Elaboration (Iteration 1)	122	20	6/2	9/30
4	Construction (Iteration 1)	80	400	6/2	8/20
5	Transition (Iteration 1)	25	50	8/21	9/14

Software Project Management

- **Overall Schedule:**



--- Time measured in weeks -→

Software Project Management

Project Scheduling: Tasks, Durations and Dependencies

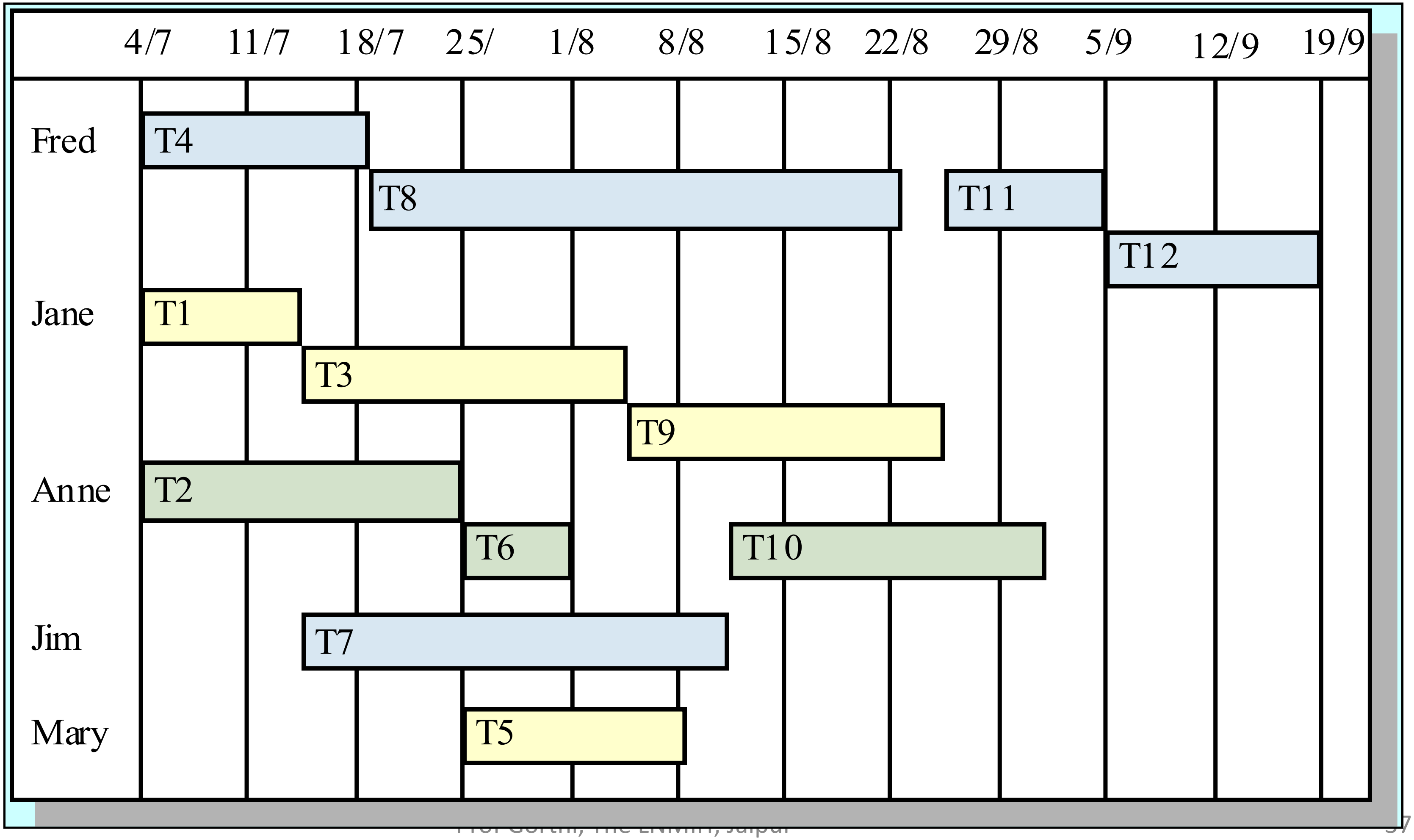
Task	Effort (person-days)	Duration (days)	Dependencies
T1	15	10	
T2	8	15	
T3	20	15	T1
T4	5	10	
T5	5	10	T2, T4
T6	10	5	T1, T2
T7	25	20	T1
T8	75	25	T4
T9	10	15	T3, T6
T10	20	15	T7, T8
T11	10	10	T9
T12	20	10	T10, T11

Project Roles

- Project Manager
- Analyst
- Designer
- Document Editor
- Configuration Manager
- Team Lead
- Programmer
- Integration Tester
- Trainer
- System Administrator
- Risk Auditor
- Quality Assurance Manager
- System Test Planner
- System Tester
- Non-Functional Test Planner
- Non-Functional Tester

Software Project Management

- **Project Scheduling: Staff Allocation Chart**



Risk management

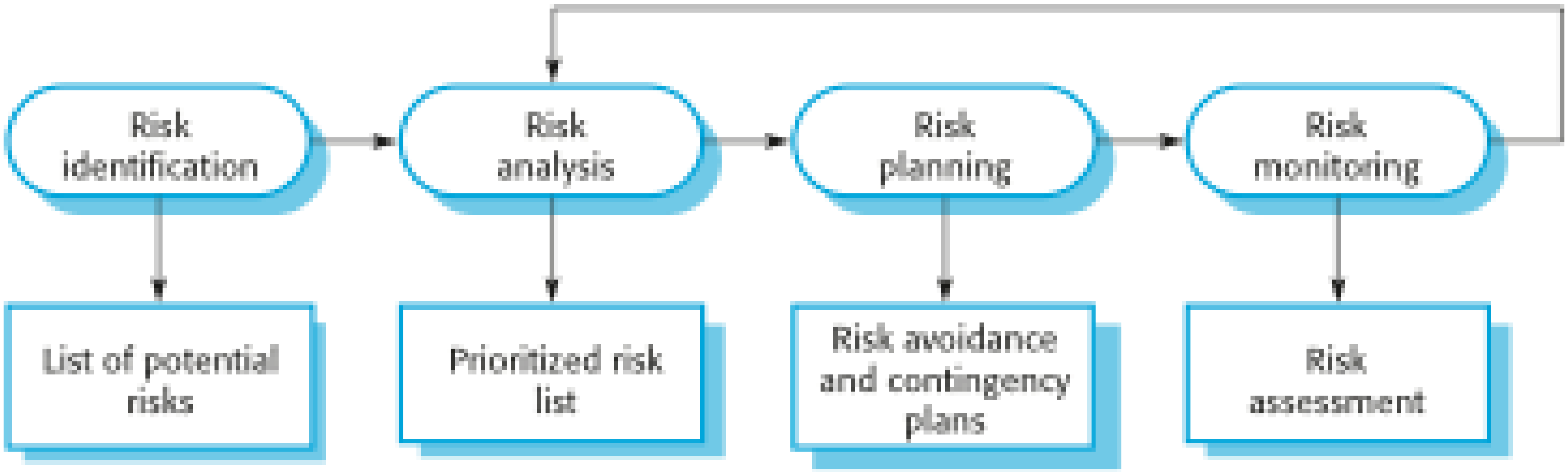
- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.

- A risk is a probability that some adverse circumstance will occur
  - 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 procuring the software.

The risk management process

- ✧ Risk identification
  - Identify project, product and business risks;
- ✧ Risk analysis
  - Assess the likelihood and consequences of these risks;
- ✧ Risk mitigation 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



Examples of different risk types

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. (1) Reusable software components contain defects that mean they cannot be reused as planned. (2)
People	It is impossible to recruit staff with the skills required. (3) Key staff are ill and unavailable at critical times. (4) Required training for staff is not available. (5)
Organizational	The organization is restructured so that different management are responsible for the project. (6) Organizational financial problems force reductions in the project budget. (7)
Tools	The code generated by software code generation tools is inefficient. (8) Software tools cannot work together in an integrated way. (9)
Requirements	Changes to requirements that require major design rework are proposed. (10) Customers fail to understand the impact of requirements changes. (11)
Estimation	The time required to develop the software is underestimated. (12) The rate of defect repair is underestimated. (13) The size of the software is underestimated. (14)

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 (7).	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project (3).	High	Catastrophic
Key staff are ill at critical times in the project (4).	Moderate	Serious
Faults in reusable software components have to be repaired before these components are reused. (2).	Moderate	Serious
Changes to requirements that require major design rework are proposed (10).	Moderate	Serious
The organization is restructured so that different management are responsible for the project (6).	High	Serious
The database used in the system cannot process as many transactions per second as expected (1).	Moderate	Serious

Risk types and examples

Risk	Probability	Effects
The time required to develop the software is underestimated (12).	High	Serious
Software tools cannot be integrated (9).	High	Tolerable
Customers fail to understand the impact of requirements changes (11).	Moderate	Tolerable
Required training for staff is not available (5).	Moderate	Tolerable
The rate of defect repair is underestimated (13).	Moderate	Tolerable
The size of the software is underestimated (14).	High	Tolerable
Code generated by code generation tools is inefficient (8).	Moderate	Insignificant

Risk mitigation planning

- Consider each risk and develop a strategy to manage that risk.

- Avoidance strategies
  - The probability that the risk will arise is reduced;

- Minimisation 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;

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.

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-in components; investigate use of a program generator.

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

People management factors

- Consistency
  - Team members should all be treated in a comparable way without favourites or discrimination.
- Respect
  - Different team members have different skills and these differences should be respected.
- Inclusion
  - Involve all team members and make sure that people’s views are considered.
- Honesty
  - You should always be honest about what is going well and what is going badly in a project.

Personality types

✧ Task-oriented.

- The motivation for doing the work is the work itself;

✧ Self-oriented.

- The work is a means to an end which is the achievement of individual goals - e.g. to get rich, to play tennis, to travel etc.;

✧ Interaction-oriented

- The principal motivation is the presence and actions of co-workers. People go to work because they like to interact with team members.

Teamwork

- Most software engineering is a group activity
  - The development schedule for most non-trivial software projects is such that they cannot be completed by one person working alone.
- A good group is cohesive and has a team spirit. The people involved are motivated by the success of the group as well as by their own personal goals.
- Group interaction is a key determinant of group performance.
- Flexibility in group composition is limited
  - Managers must do the best they can with available people.

Software Project Management

- ✓ Software Pricing
- ✓ Effort Estimation Methods
- ✓ Planning and Scheduling a Software Project

Key Points:

1. Pricing depends upon multiple factors such as the market conditions, product uniqueness, etc.
2. Function Points and COCOMO are popular software project effort estimation models
3. Focus of Software Project Management is to plan, schedule monitor, control software life cycle activities so as to meet the cost and time targets
4. Identify, analyze, mitigate and monitor risks