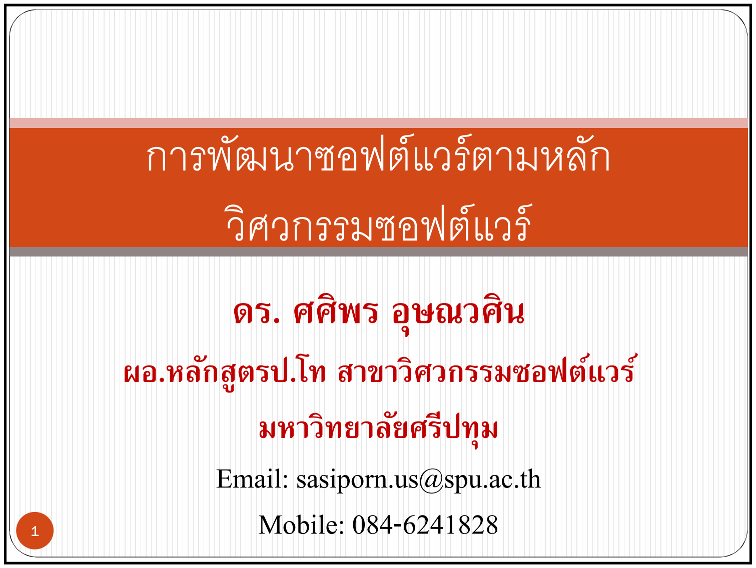


การพัฒนาซอฟต์แวร์ตามหลักวิศวกรรมซอฟต์แวร์

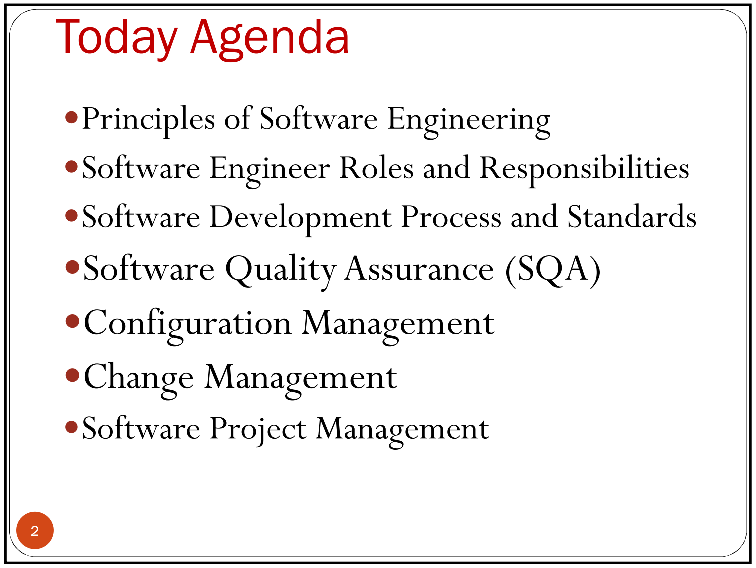
ดร. ศศิพร อุษณวศิน

ผอ.หลักสูตรป.โท สาขาวิศวกรรมซอฟต์แวร์

มหาวิทยาลัยศรีปทุม

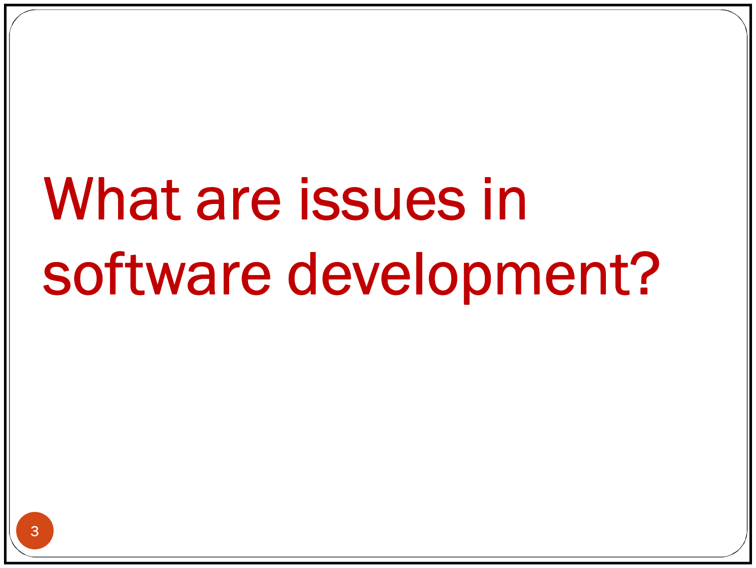
Email: sasiporn.us@spu.ac.th

Mobile: 084-6241828

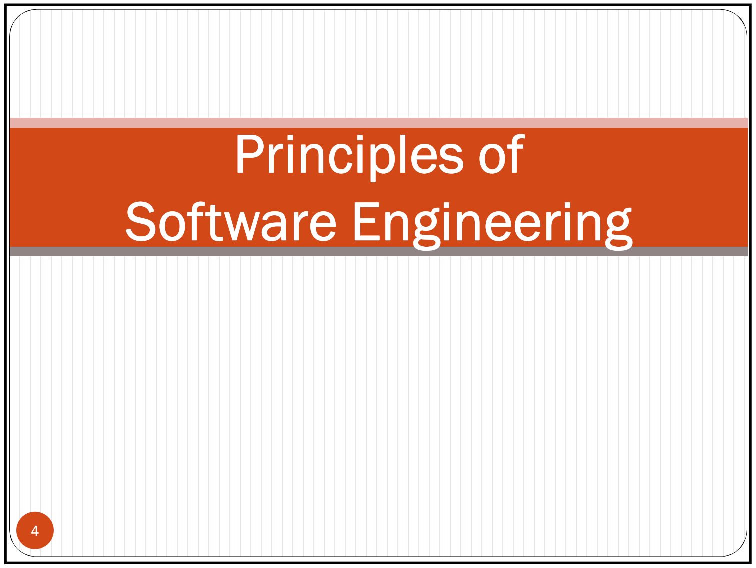


Today Agenda

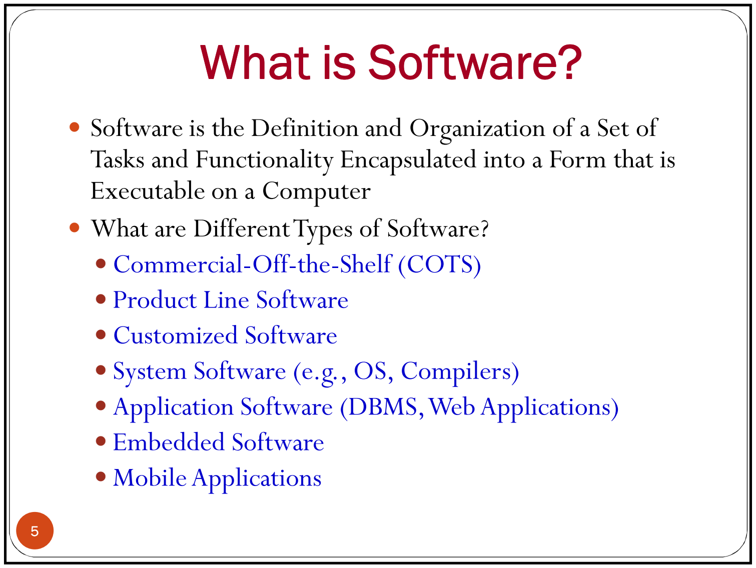
- Principles of Software Engineering
- Software Engineer Roles and Responsibilities
- Software Development Process and Standards
- Software Quality Assurance (SQA)
- Configuration Management
- Change Management
- Software Project Management



What are issues in software development?



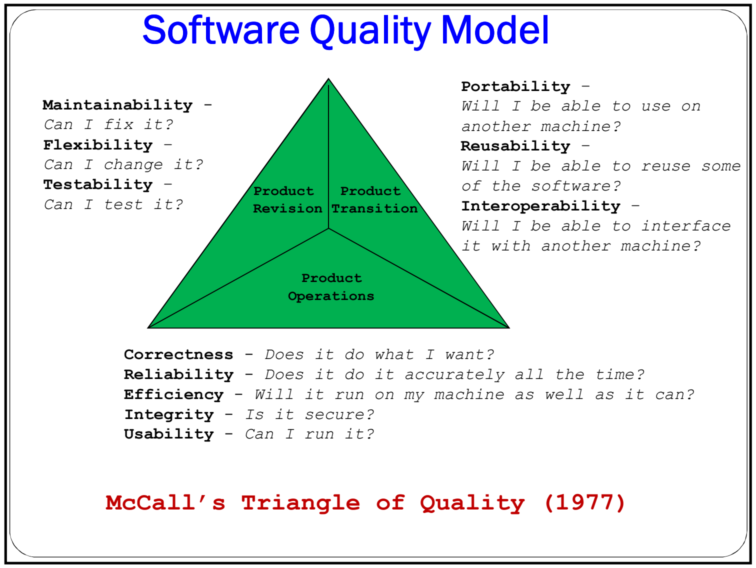
Principles of Software Engineering



What is Software?

- Software is the Definition and Organization of a Set of Tasks and Functionality Encapsulated into a Form that is Executable on a Computer
- What are Different Types of Software?
  - Commercial-Off-the-Shelf (COTS)
  - Product Line Software
  - Customized Software
  - System Software (e.g., OS, Compilers)
  - Application Software (DBMS, Web Applications)
  - Embedded Software
  - Mobile Applications

5



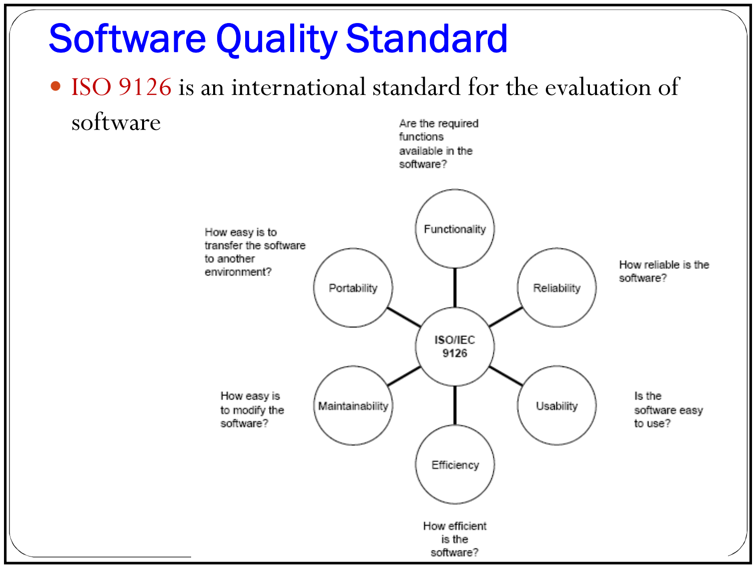
Software Quality Model

Maintainability –
Can I fix it?
Flexibility –
Can I change it?
Testability –
Can I test it?

Portability –
Will I be able to use on another machine?
Reusability –
Will I be able to reuse some of the software?
Interoperability –
Will I be able to interface it with another machine?

Correctness – *Does it do what I want?*
Reliability – *Does it do it accurately all the time?*
Efficiency – *Will it run on my machine as well as it can?*
Integrity – *Is it secure?*
Usability – *Can I run it?*

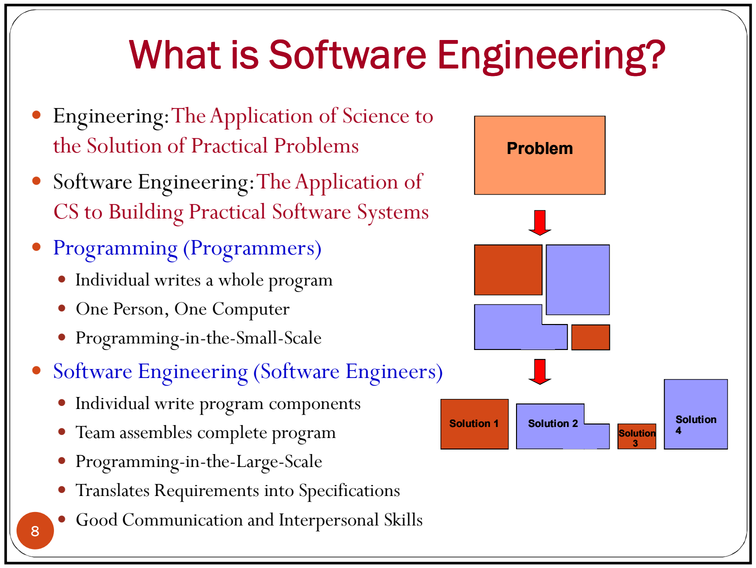
McCall's Triangle of Quality (1977)



Software Quality Standard

- ISO 9126 is an international standard for the evaluation of software

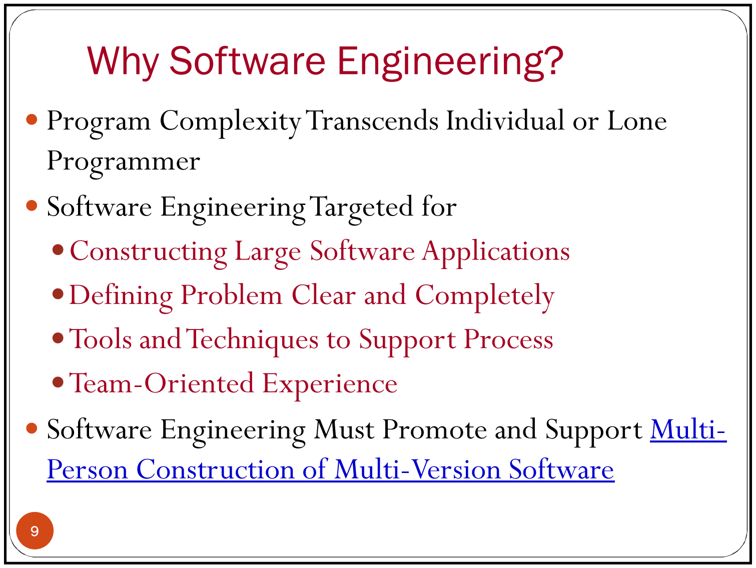
Are the required functions available in the software?
Functionality
How easy is to transfer the software to another environment?
Portability
Reliability
How reliable is the software?
ISO/IEC 9126
How easy is to modify the software?
Maintainability
Usability
Is the software easy to use?
Efficiency
How efficient is the software?



What is Software Engineering?

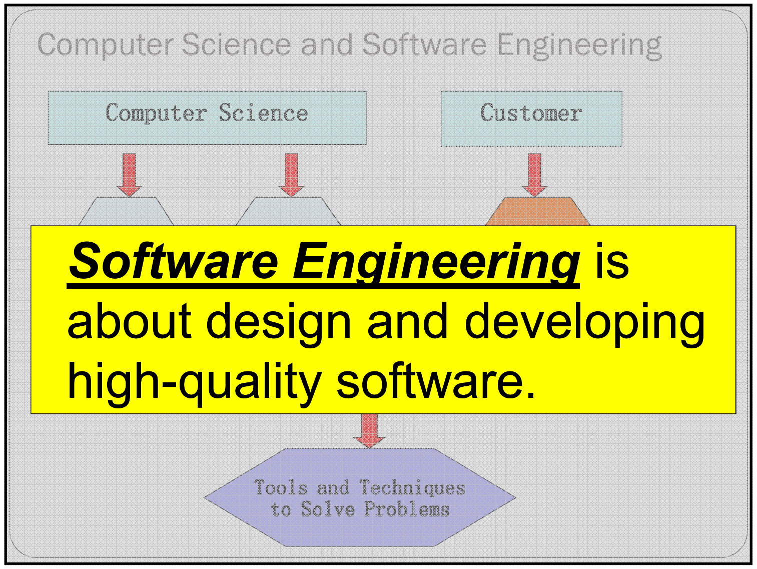
- Engineering: The Application of Science to the Solution of Practical Problems
- Software Engineering: The Application of CS to Building Practical Software Systems
- Programming (Programmers)
  - Individual writes a whole program
  - One Person, One Computer
  - Programming-in-the-Small-Scale
- Software Engineering (Software Engineers)
  - Individual write program components
  - Team assembles complete program
  - Programming-in-the-Large-Scale
  - Translates Requirements into Specifications
  - Good Communication and Interpersonal Skills

8



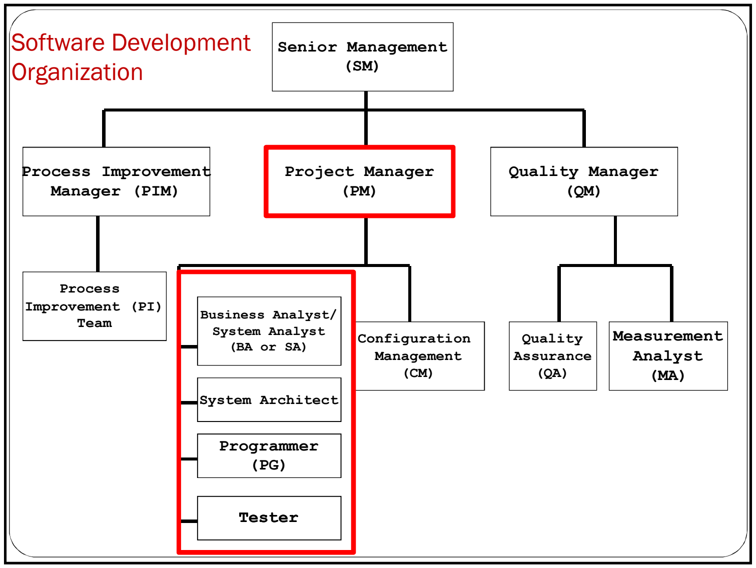
Why Software Engineering?

- Program Complexity Transcends Individual or Lone Programmer
- Software Engineering Targeted for
  - Constructing Large Software Applications
  - Defining Problem Clear and Completely
  - Tools and Techniques to Support Process
  - Team-Oriented Experience
- Software Engineering Must Promote and Support Multi-Person Construction of Multi-Version Software

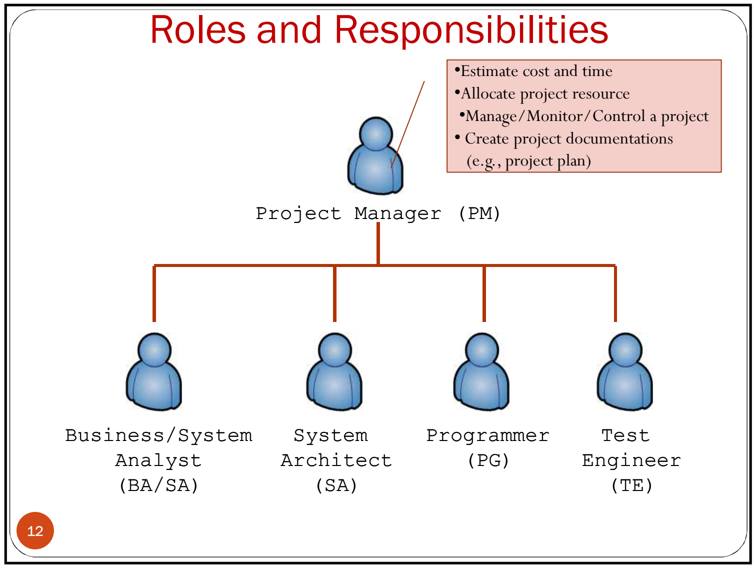


Computer Science and Software Engineering

Software Engineering is about design and developing high-quality software.



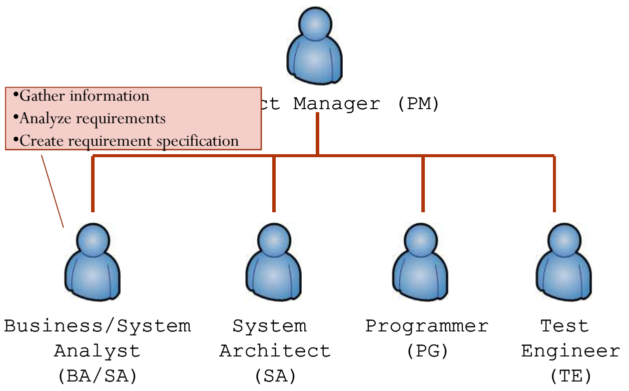
Software Development Organization



Roles and Responsibilities

- Estimate cost and time
- Allocate project resource
- Manage/Monitor/Control a project
- Create project documentations (e.g., project plan)

Roles and Responsibilities



- Gather information
- Analyze requirements
- Create requirement specification

ct Manager (PM)

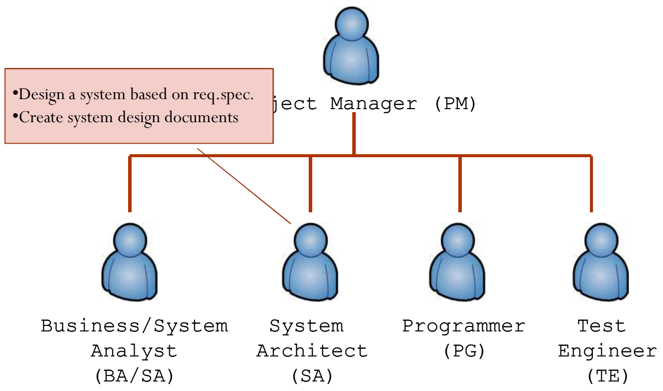
Business/System Analyst (BA/SA)

System Architect (SA)

Programmer (PG)

Test Engineer (TE)

Roles and Responsibilities



- Design a system based on req.spec.
- Create system design documents

ject Manager (PM)

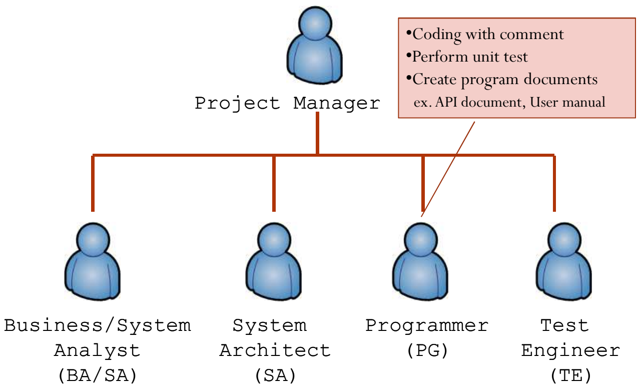
Business/System Analyst (BA/SA)

System Architect (SA)

Programmer (PG)

Test Engineer (TE)

Roles and Responsibilities



Project Manager

- Coding with comment
- Perform unit test
- Create program documents
  ex. API document, User manual

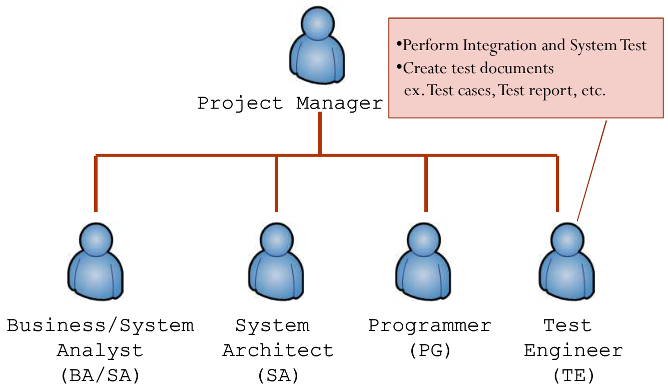
Business/System Analyst (BA/SA)

System Architect (SA)

Programmer (PG)

Test Engineer (TE)

Roles and Responsibilities



Project Manager

- Perform Integration and System Test
- Create test documents
  ex. Test cases, Test report, etc.

Business/System Analyst (BA/SA)

System Architect (SA)

Programmer (PG)

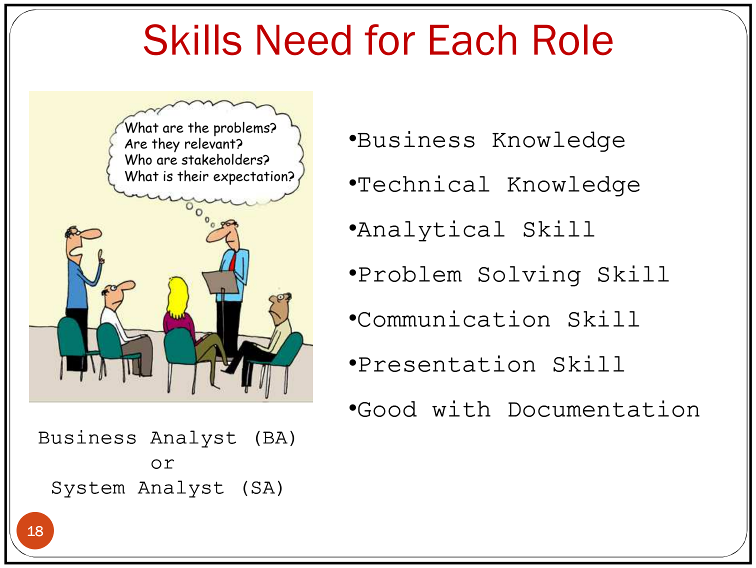
Test Engineer (TE)

Skills Need for Each Role

COST
SCHEDULE
PERFORMANCE

Project Manager (PM)

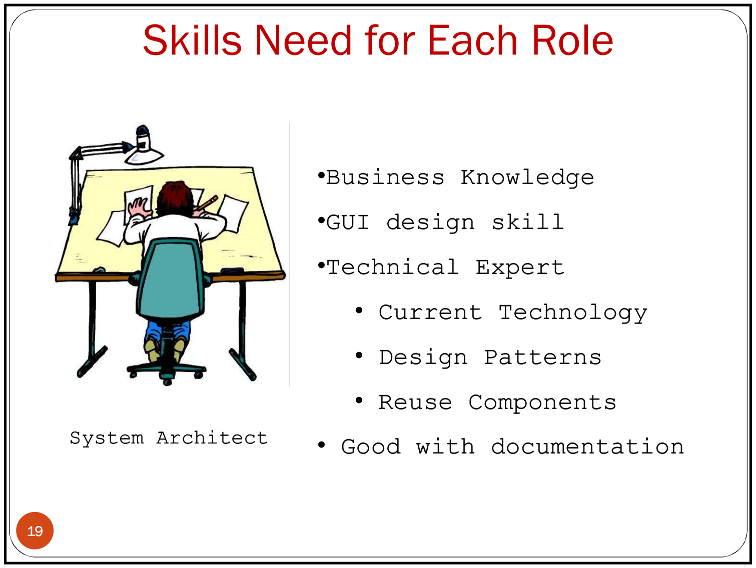
- Business Knowledge
- Technical Knowledge
- Management Skill
- Interpersonal Skill
- Analytical Skill
- Problem Solving Skill
- Presentation Skill
- Negotiation Skill
- Communication Skill



Skills Need for Each Role

Business Analyst (BA)
or
System Analyst (SA)

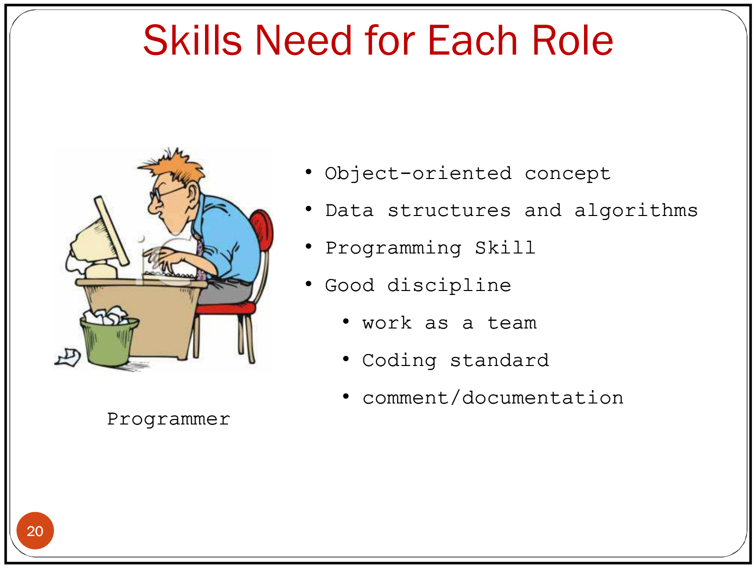
- Business Knowledge
- Technical Knowledge
- Analytical Skill
- Problem Solving Skill
- Communication Skill
- Presentation Skill
- Good with Documentation



Skills Need for Each Role

System Architect

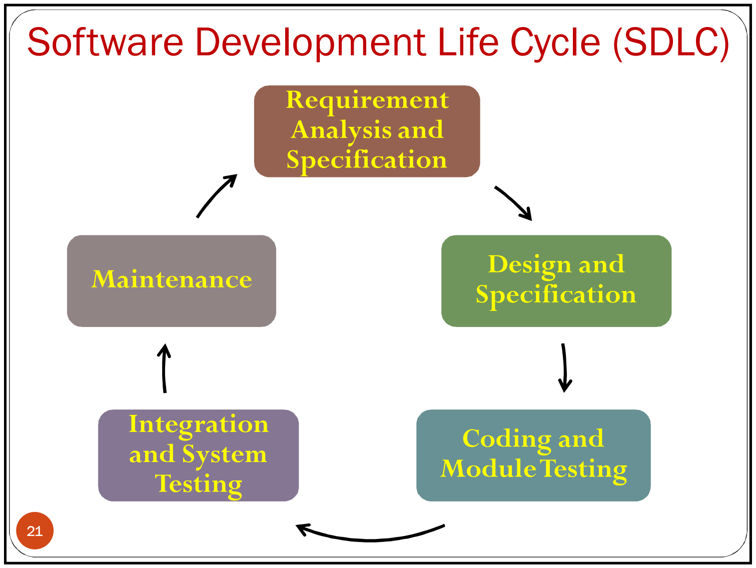
- Business Knowledge
- GUI design skill
- Technical Expert
  - Current Technology
  - Design Patterns
  - Reuse Components
- Good with documentation



Skills Need for Each Role

Programmer

- Object-oriented concept
- Data structures and algorithms
- Programming Skill
- Good discipline
  - work as a team
  - Coding standard
  - comment/documentation



Software Development Life Cycle (SDLC)

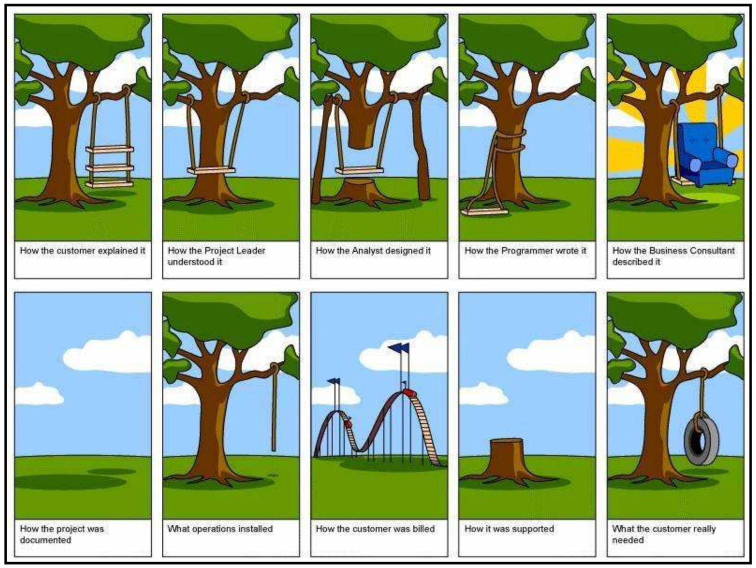
Requirement Analysis and Specification

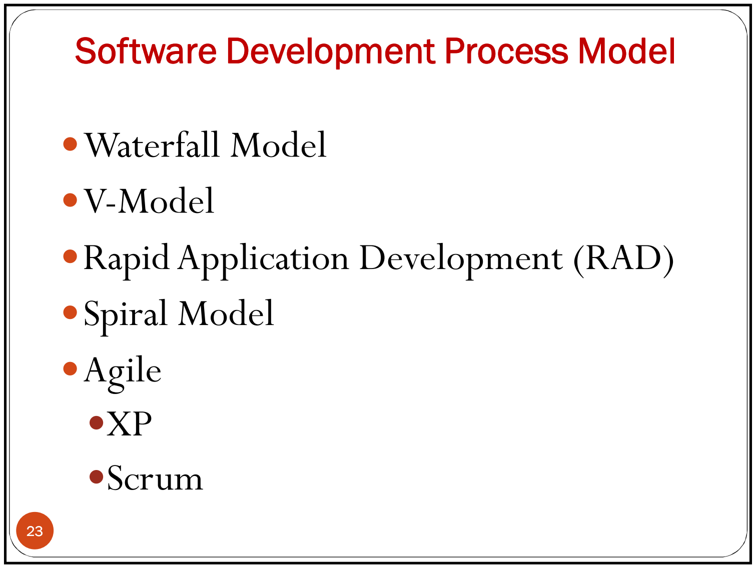
Design and Specification

Coding and Module Testing

Integration and System Testing

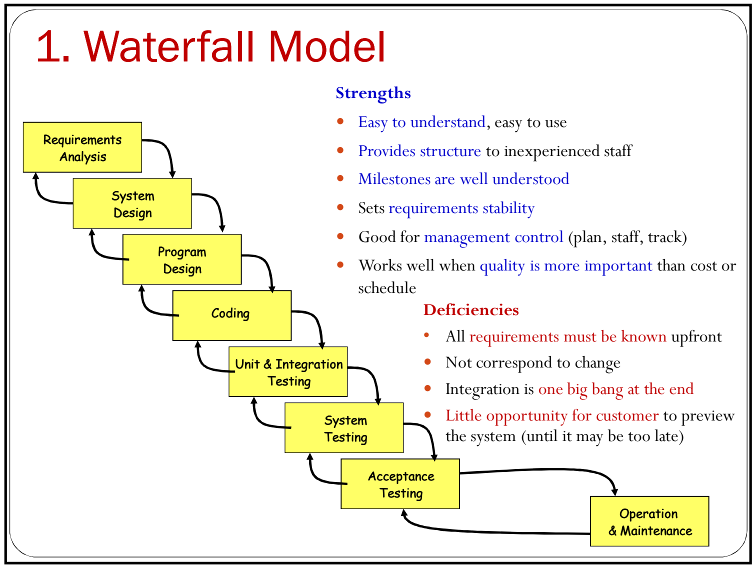
Maintenance


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed



Software Development Process Model

- Waterfall Model
- V-Model
- Rapid Application Development (RAD)
- Spiral Model
- Agile
  - XP
  - Scrum



1. Waterfall Model

Requirements Analysis

System Design

Program Design

Coding

Unit & Integration Testing

System Testing

Acceptance Testing

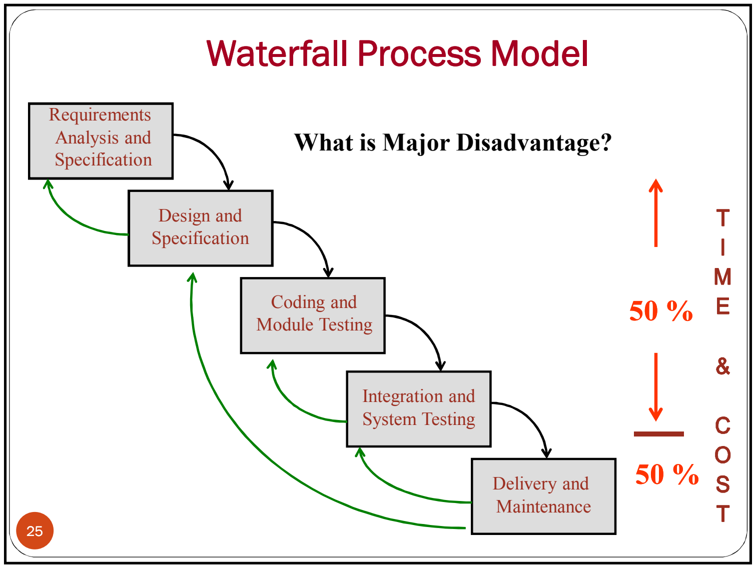
Operation & Maintenance

Strengths

- Easy to understand, easy to use
- Provides structure to inexperienced staff
- Milestones are well understood
- Sets requirements stability
- Good for management control (plan, staff, track)
- Works well when quality is more important than cost or schedule

Deficiencies

- All requirements must be known upfront
- Not correspond to change
- Integration is one big bang at the end
- Little opportunity for customer to preview the system (until it may be too late)



Waterfall Process Model

What is Major Disadvantage?

Requirements Analysis and Specification

Design and Specification

Coding and Module Testing

Integration and System Testing

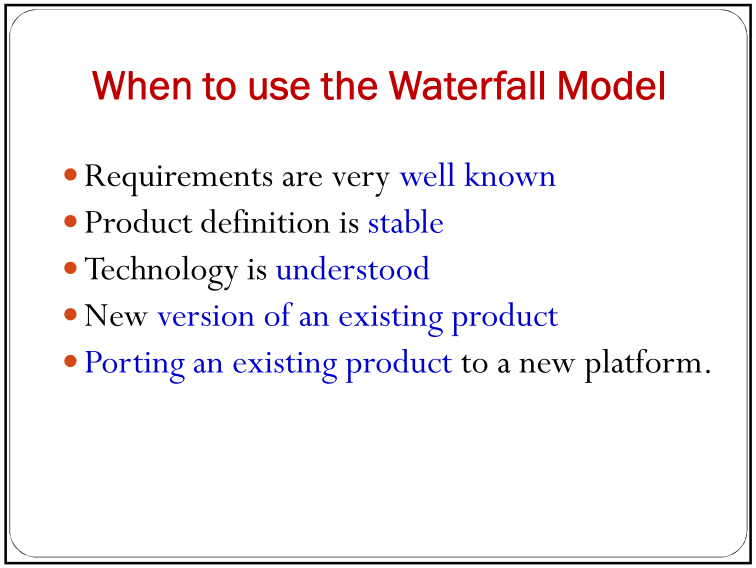
Delivery and Maintenance

50 %

50 %

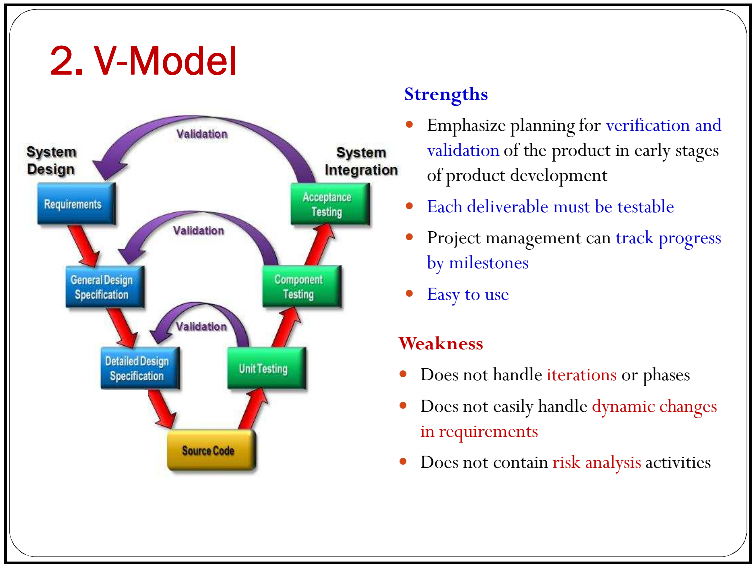
TIME & COST

25



When to use the Waterfall Model

- Requirements are very well known
- Product definition is stable
- Technology is understood
- New version of an existing product
- Porting an existing product to a new platform.



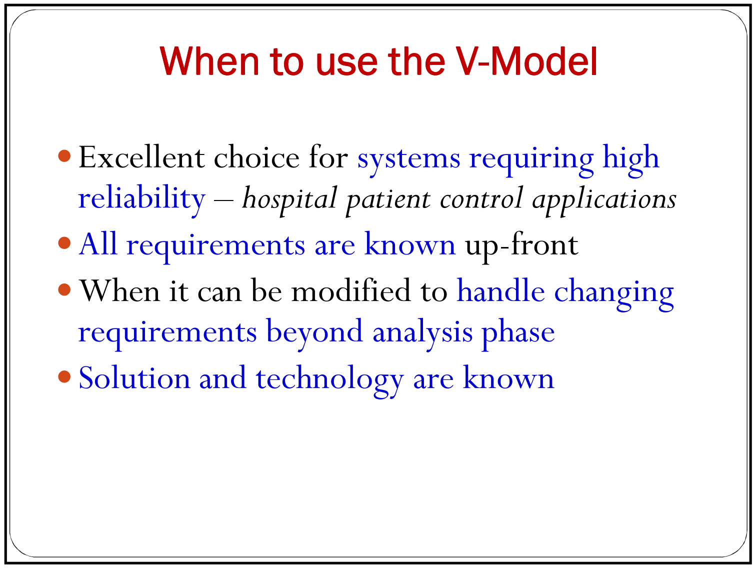
2. V-Model

Strengths

- Emphasize planning for verification and validation of the product in early stages of product development
- Each deliverable must be testable
- Project management can track progress by milestones
- Easy to use

Weakness

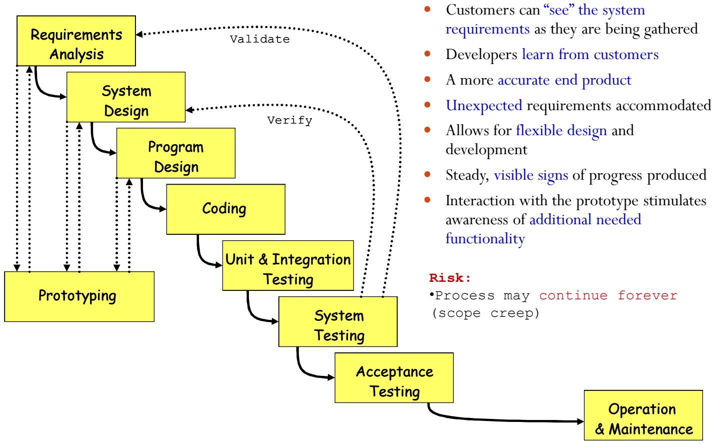
- Does not handle iterations or phases
- Does not easily handle dynamic changes in requirements
- Does not contain risk analysis activities



When to use the V-Model

- Excellent choice for systems requiring high reliability – *hospital patient control applications*
- All requirements are known up-front
- When it can be modified to handle changing requirements beyond analysis phase
- Solution and technology are known

3. Waterfall Model with Prototyping



Advantages

- Customers can "see" the system requirements as they are being gathered
- Developers learn from customers
- A more accurate end product
- Unexpected requirements accommodated
- Allows for flexible design and development
- Steady, visible signs of progress produced
- Interaction with the prototype stimulates awareness of additional needed functionality

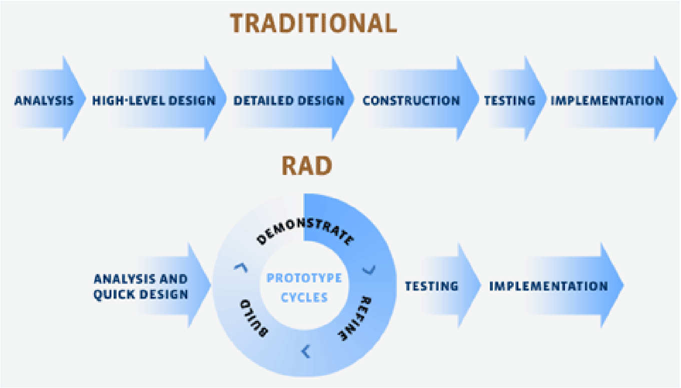
Risk:

•Process may continue forever (scope creep)

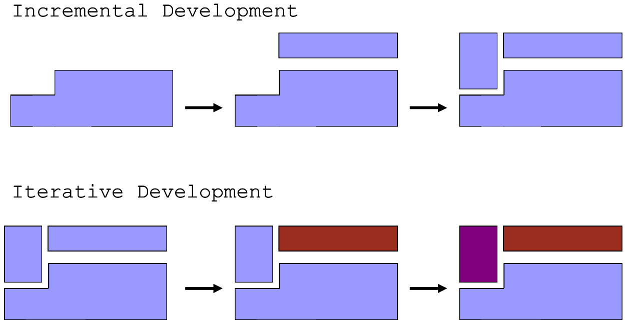
When to use Waterfall with Prototyping Model

- Requirements are unstable or have to be clarified
- As the requirements clarification stage of a waterfall model
- Develop user interfaces
- Short-lived demonstrations
- New, original development
- With the analysis and design portions of object-oriented development.

4. Rapid Application Model (RAD)



Incremental and Iterative Approach



RAD Strengths

- Reduced cycle time and improved productivity with fewer people means lower costs
- Time-box approach mitigates cost and schedule risk
- Customer involved throughout the complete cycle minimizes risk of not achieving customer satisfaction and business needs
- Focus moves from documentation to code (WYSIWYG).
- Uses modeling concepts to capture information about business, data, and processes.

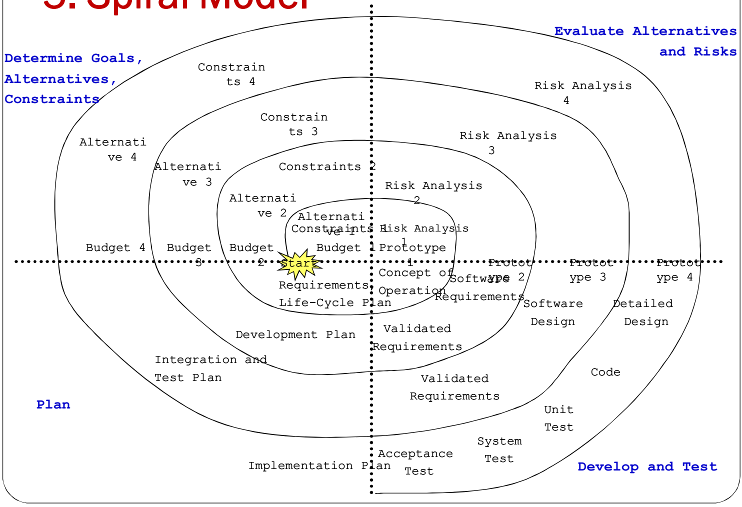
RAD Weaknesses

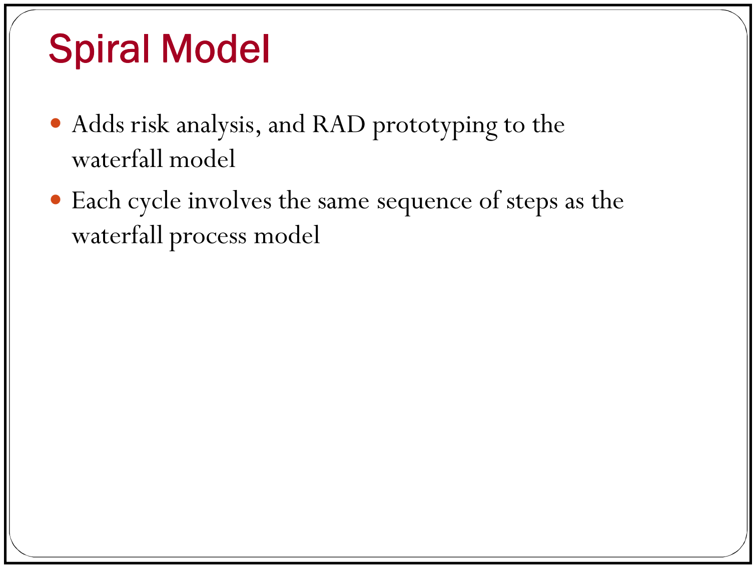
- Accelerated development process must give quick responses to the user
- Risk of never achieving closure
- Hard to use with legacy systems
- Requires a system that can be modularized
- Developers and customers must be committed to rapid-fire activities in an abbreviated time frame.

When to use RAD

- Reasonably well-known requirements
- User involved throughout the life cycle
- Project can be time-boxed
- Functionality delivered in increments
- Low technical risks
- System can be modularized

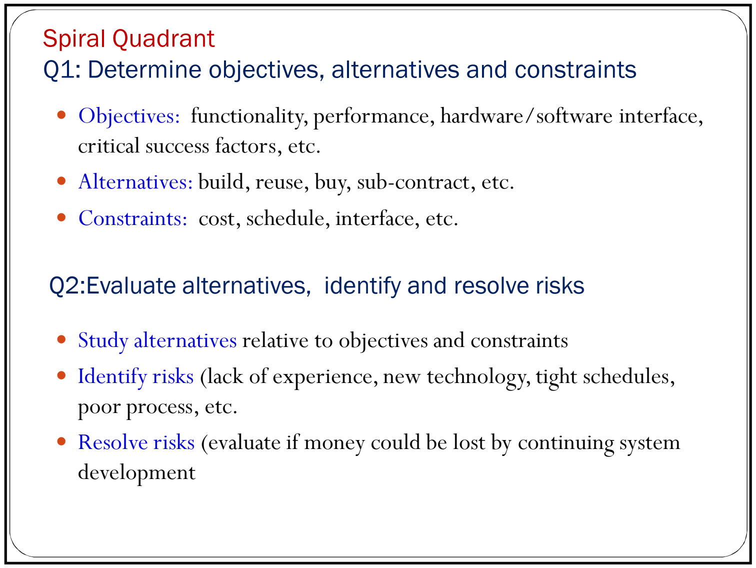
5. Spiral Model





Spiral Model

- Adds risk analysis, and RAD prototyping to the waterfall model
- Each cycle involves the same sequence of steps as the waterfall process model

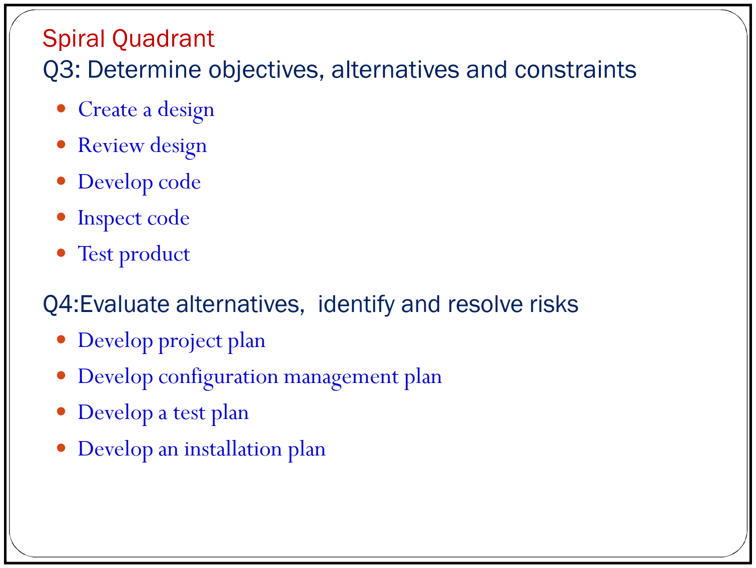


Spiral Quadrant
Q1: Determine objectives, alternatives and constraints

- Objectives: functionality, performance, hardware/software interface, critical success factors, etc.
- Alternatives: build, reuse, buy, sub-contract, etc.
- Constraints: cost, schedule, interface, etc.

Q2:Evaluate alternatives, identify and resolve risks

- Study alternatives relative to objectives and constraints
- Identify risks (lack of experience, new technology, tight schedules, poor process, etc.
- Resolve risks (evaluate if money could be lost by continuing system development

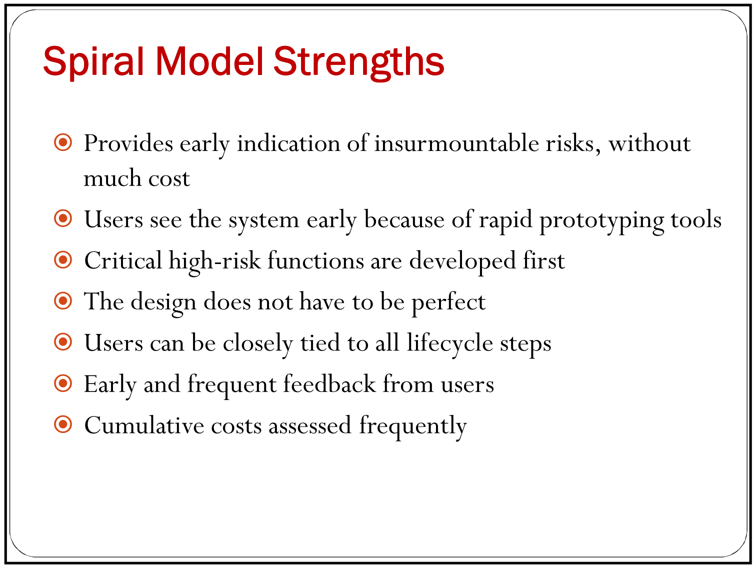


Spiral Quadrant
Q3: Determine objectives, alternatives and constraints

- Create a design
- Review design
- Develop code
- Inspect code
- Test product

Q4:Evaluate alternatives, identify and resolve risks

- Develop project plan
- Develop configuration management plan
- Develop a test plan
- Develop an installation plan



Spiral Model Strengths

- Provides early indication of insurmountable risks, without much cost
- Users see the system early because of rapid prototyping tools
- Critical high-risk functions are developed first
- The design does not have to be perfect
- Users can be closely tied to all lifecycle steps
- Early and frequent feedback from users
- Cumulative costs assessed frequently

Spiral Model Weaknesses

- Time spent for evaluating risks too large for small or low-risk projects
- Time spent planning, resetting objectives, doing risk analysis and prototyping may be excessive
- The model is complex
- Risk assessment expertise is required
- Spiral may continue indefinitely
- Developers must be reassigned during non-development phase activities
- May be hard to define objective, verifiable milestones that indicate readiness to proceed through the next iteration

When to use Spiral Model

- When creation of a prototype is appropriate
- When costs and risk evaluation is important
- For medium to high-risk projects
- Long-term project commitment unwise because of potential changes to economic priorities
- Users are unsure of their needs
- Requirements are complex
- New product line
- Significant changes are expected (research and exploration)

6. Agile Model

- Speed up or bypass one or more life cycle phases
- Usually less formal and reduced scope
- Used for time-critical applications
- Used in organizations that employ disciplined methods

Agile Manifesto (Agile Alliance 2001)

1. Prefer *face-to-face communication* (real time) rather than communication through written documents
2. Invest time in producing *working software* rather than in producing comprehensive documents
3. Focus on *customer collaboration* rather than contract negotiation
4. Concentrate on *responding to change* rather than on creating a plan

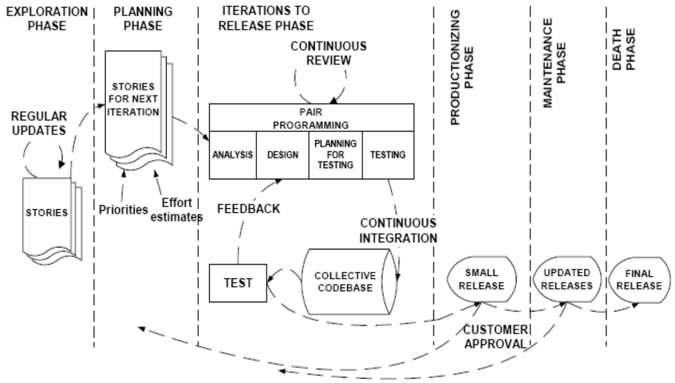
Some Agile Methods

- **eXtreme Programming (XP)**
- **Scrum**
- **Rational Unify Process (RUP)**

Agile Methods: Extreme Programming (XP)

- For small-to-medium-sized teams developing software with vague or rapidly changing requirements
- Coding is the key activity throughout a software project
- Communication among teammates is done with code
- Life cycle and behavior of complex objects defined in test cases – again in code

Life Cycle of Extreme Programming (XP)



XP Practices (1-6)

1. Planning game – determine scope of the next release by combining business priorities and technical estimates
2. Small releases – put a simple system into production, then release new versions in very short cycle
3. Metaphor – all development is guided by a simple shared story of how the whole system works
4. Simple design – system is designed as simply as possible (extra complexity removed as soon as found)
5. Testing – programmers continuously write unit tests; customers write tests for features
6. Refactoring – programmers continuously restructure the system without changing its behavior to remove duplication and simplify

XP Practices (7 – 12)

7. Pair-programming -- all production code is written with two programmers at one machine
8. Collective ownership – anyone can change any code anywhere in the system at any time.
9. Continuous integration – integrate and build the system many times a day – every time a task is completed.
10. 40-hour week – work no more than 40 hours a week as a rule
11. On-site customer – a user is on the team and available full-time to answer questions
12. Coding standards – programmers write all code in accordance with rules emphasizing communication through the code

XP is “extreme” because

Commonsense practices taken to extreme levels

- If code reviews are good, review code all the time (pair programming)
- If testing is good, everybody will test all the time
- If simplicity is good, keep the system in the simplest design that supports its current functionality. (simplest thing that works)
- If design is good, everybody will design daily (refactoring)
- If architecture is important, everybody will work at defining and refining the architecture (metaphor)
- If integration testing is important, build and integrate test several times a day (continuous integration)
- If short iterations are good, make iterations really, really short (hours rather than weeks)

Online references to XP at

- http://www.extremeprogramming.org/
- http://c2.com/cgi/wiki?ExtremeProgrammingRoadmap
- http://www.xprogramming.com/

Agile Methods: Scrum

A term “Scrum” is derived from a game of Rugby:

Getting out-of play ball back into the game with teamwork!.

Scrum - an agile process

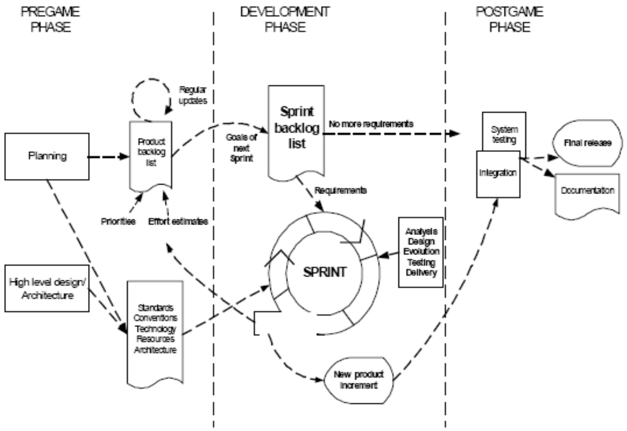
- SCRUM is an agile, lightweight process for *managing and controlling* software and product development in rapidly changing environment
  - Iterative and incremental process
  - Team-based approach
  - Developing systems/ products with rapidly changing requirements
  - Controls the chaos of conflicting interest and needs
  - Improve communication and maximize cooperation
  - Protecting the team form disruptions and impediments
  - A way to maximize productivity

52

Components of Scrum

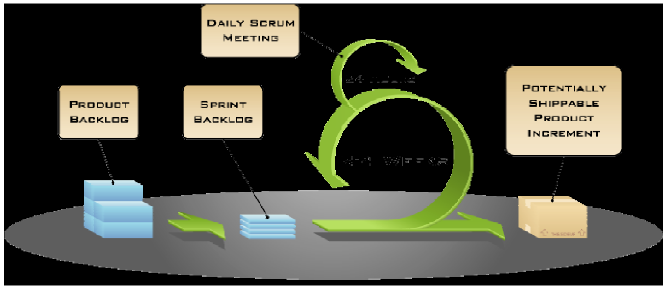
- Scrum Roles
  - Scrum Master: *managing the project and enacting scrum values and practices.*
  - Product Owner: *knows what needs to be build and in what sequence this should be done.*
  - Scrum Team: *5-10 full-time members, self-organizing, membership can be change only between prints.*
  - Customer
  - Management
- Process
  - Three Phases (Pre-Game, Development, Post-Game)
- Practices

Scrum Process



Scrum Practices

- Product Backlog
- Effort Estimation
- Sprint
- Sprint Planning Meeting
- Sprint Backlog
- Daily Scrum Meeting
- Sprint Review Meeting



Product Backlog

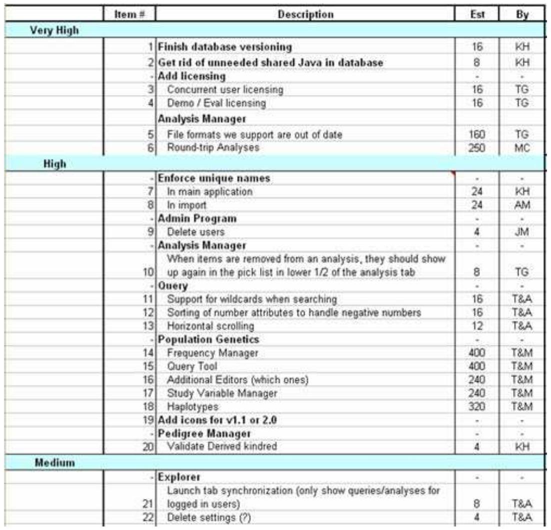
- Requirements for a system, expressed as a prioritized list of Backlog Items
- Is managed and owned by a Product Owner
- Spreadsheet (typically)
- Usually is created during the Sprint Planning Meeting
- Can be changed and re-prioritized before each PM

Estimation of Product Backlog Items

- Establishes team's velocity (how much Effort a Team can handle in one Sprint)
- Determining units of complexity.
  - Work days/work hours
- Methods of estimation:
  - Expert Review
  - Creating a Work Breakdown Structure (WBS)

57

Example of Product Backlog



	Item #	Description	Est	By
Very High				
	1	**Finish database versioning**	16	KH
	2	**Get rid of unneeded shared Java in database**	8	KH
	-	**Add licensing**	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
	-	**Enforce unique names**	-	-
	7	In main application	24	KH
	8	In import	24	AM
	-	**Admin Program**	-	-
	9	Delete users	4	JM
	-	**Analysis Manager**	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
	-	**Query**	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
	-	**Population Genetics**	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	**Add icons for v1.1 or 2.0**	-	-
	-	**Pedigree Manager**	-	-
	20	Validate Derived kindred	4	KH
Medium				
	-	**Explorer**	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

58

Sprint

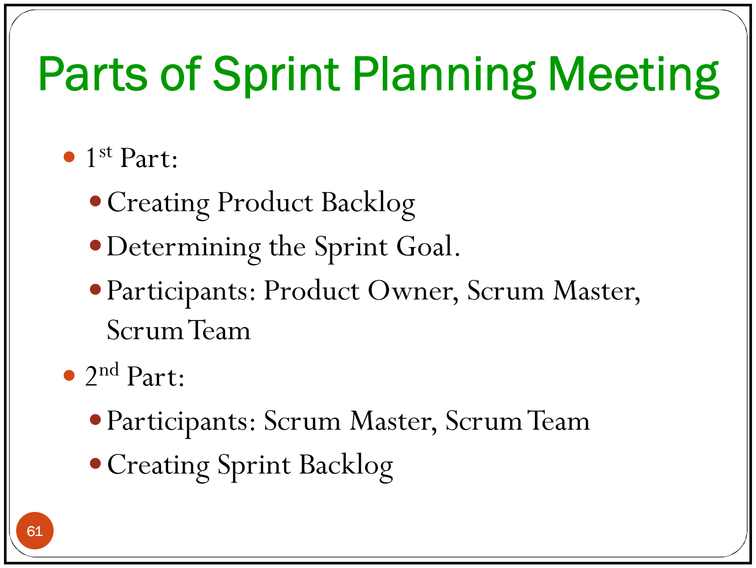
- A month-long iteration, during which is incremented a product functionality
- NO outside influence can interference with the Scrum team during the Sprint
- Each Sprint begins with the Daily Scrum Meeting

59

Sprint Planning Meeting

- A collaborative meeting in the beginning of each Sprint between the Product Owner, the Scrum Master and the Team
- Takes 8 hours and consists of 2 parts ("before lunch and after lunch")

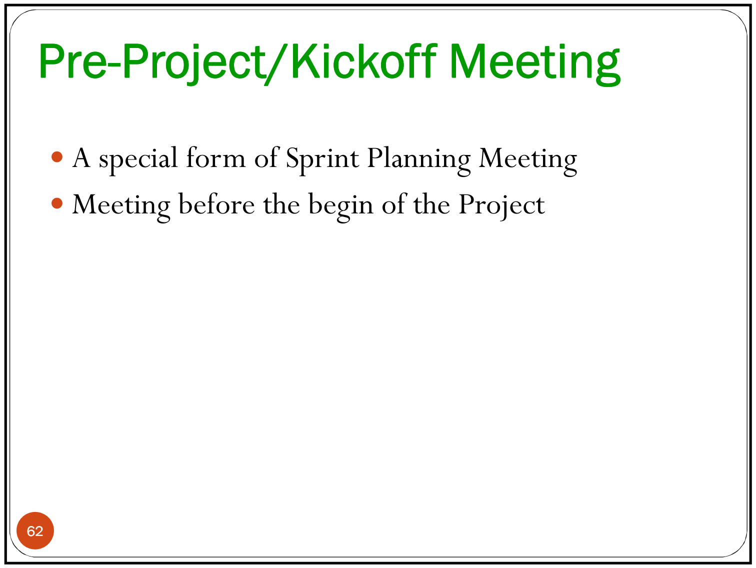
60



Parts of Sprint Planning Meeting

- 1st Part:
  - Creating Product Backlog
  - Determining the Sprint Goal.
  - Participants: Product Owner, Scrum Master, Scrum Team
- 2nd Part:
  - Participants: Scrum Master, Scrum Team
  - Creating Sprint Backlog

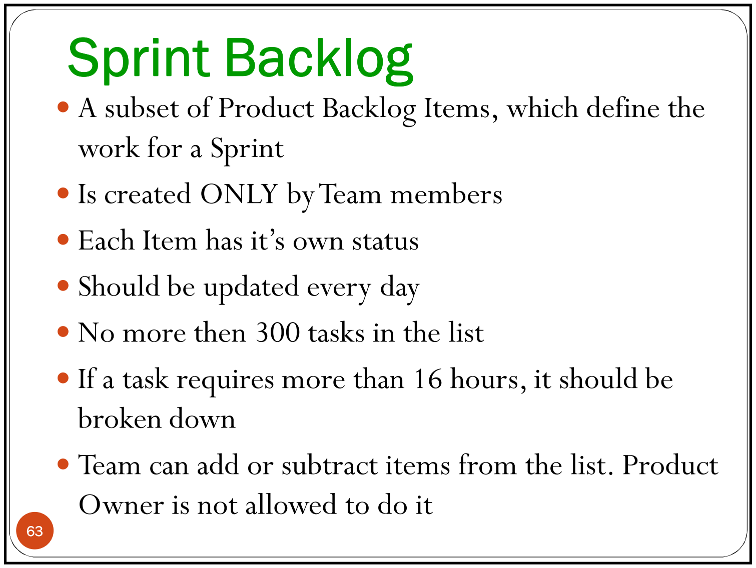
61



Pre-Project/Kickoff Meeting

- A special form of Sprint Planning Meeting
- Meeting before the begin of the Project

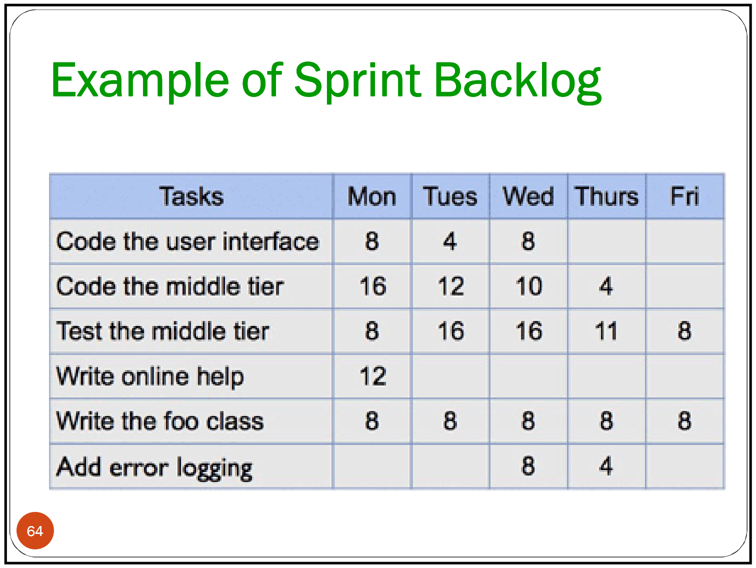
62



Sprint Backlog

- A subset of Product Backlog Items, which define the work for a Sprint
- Is created ONLY by Team members
- Each Item has it's own status
- Should be updated every day
- No more then 300 tasks in the list
- If a task requires more than 16 hours, it should be broken down
- Team can add or subtract items from the list. Product Owner is not allowed to do it

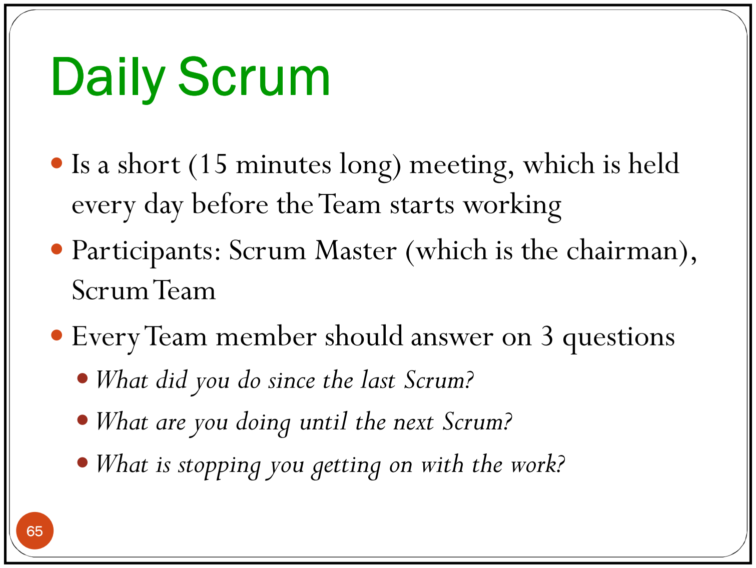
63



Example of Sprint Backlog

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

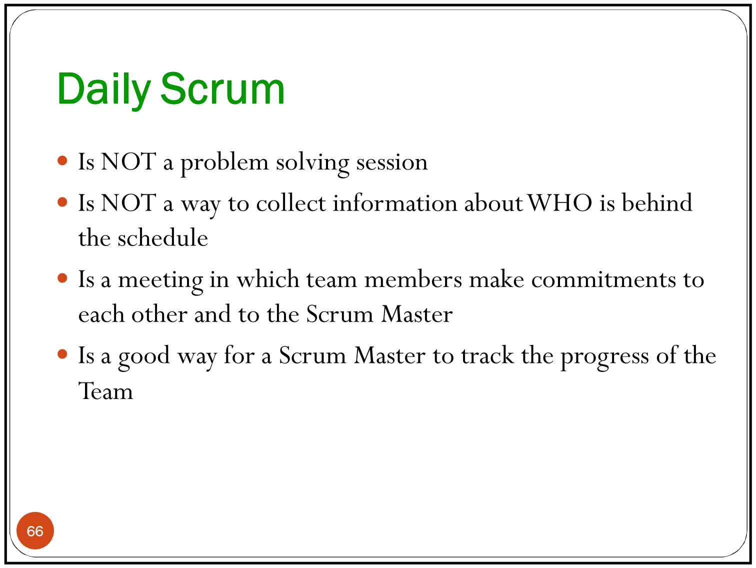
64



Daily Scrum

- Is a short (15 minutes long) meeting, which is held every day before the Team starts working
- Participants: Scrum Master (which is the chairman), Scrum Team
- Every Team member should answer on 3 questions
  - *What did you do since the last Scrum?*
  - *What are you doing until the next Scrum?*
  - *What is stopping you getting on with the work?*

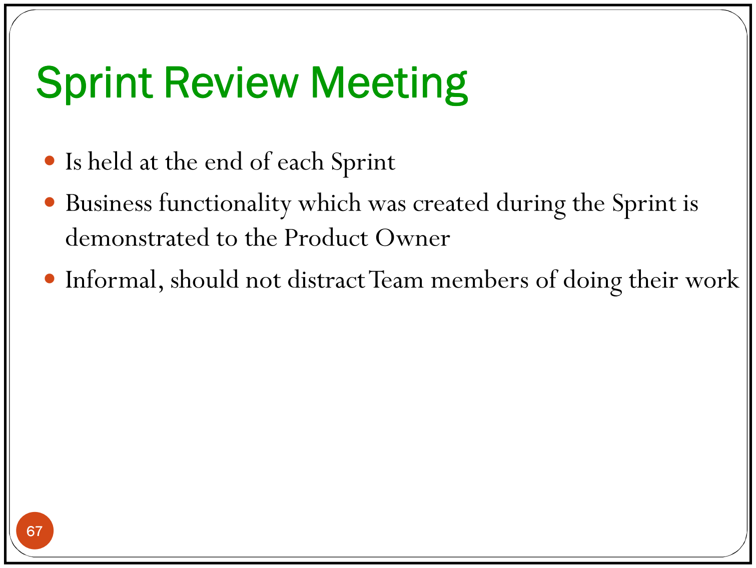
65



Daily Scrum

- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a Scrum Master to track the progress of the Team

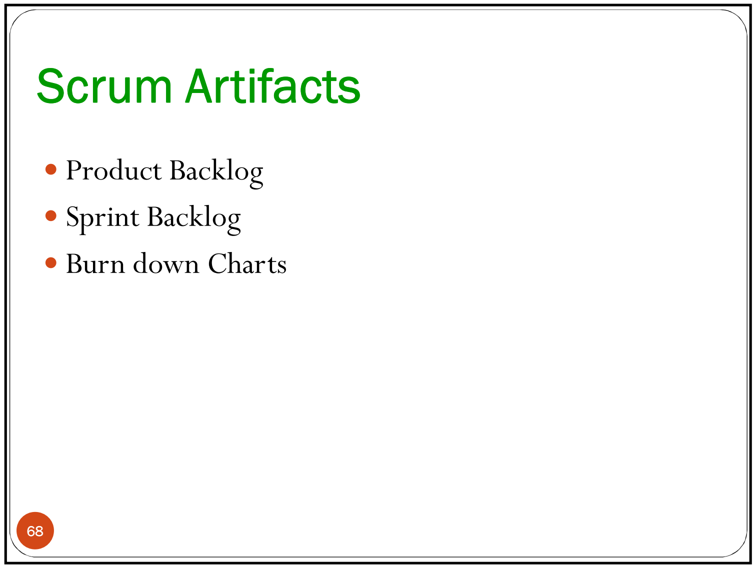
66



Sprint Review Meeting

- Is held at the end of each Sprint
- Business functionality which was created during the Sprint is demonstrated to the Product Owner
- Informal, should not distract Team members of doing their work

67



Scrum Artifacts

- Product Backlog
- Sprint Backlog
- Burn down Charts

68

Burn down Charts

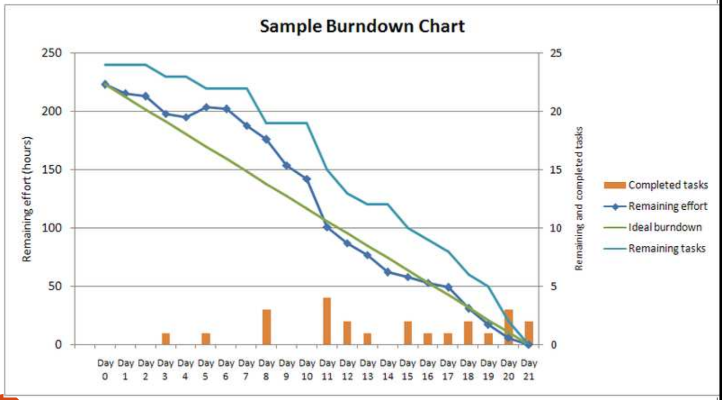
- Are used to represent “work done”.
- Are wonderful Information Radiators

  "Two characteristics are key to a good information radiator. The first is that the information changes over time. This makes it worth a person's while to look at the display... The other characteristic is that it takes very little energy to view the display.“
- 3 Types:
  - Sprint Burn down Chart (progress of the Sprint)
  - Release Burn down Chart (progress of Release)
  - Product Burn down chart (progress of the Product)
- X-Axis: time (usually in days)
- Y-Axis: remaining effort

Sprint Burn down Chart

- Depicts the total Sprint Backlog hours remaining per day
- Shows the estimated amount of time to release
- Ideally should burn down to zero to the end of the Sprint
- Actually is not a straight line
- Can bump UP

Example of Sprint burn down chart



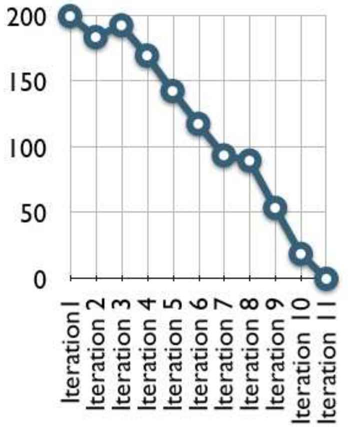
Release Burn down Chart

- Will the release be done on right time?
- X-axis: sprints
- Y-axis: amount of hours remaining
- The estimated work remaining can also burn up

Product Burn down Chart

- Is a “big picture” view of project’s progress (all the releases)

Release Burn down Chart



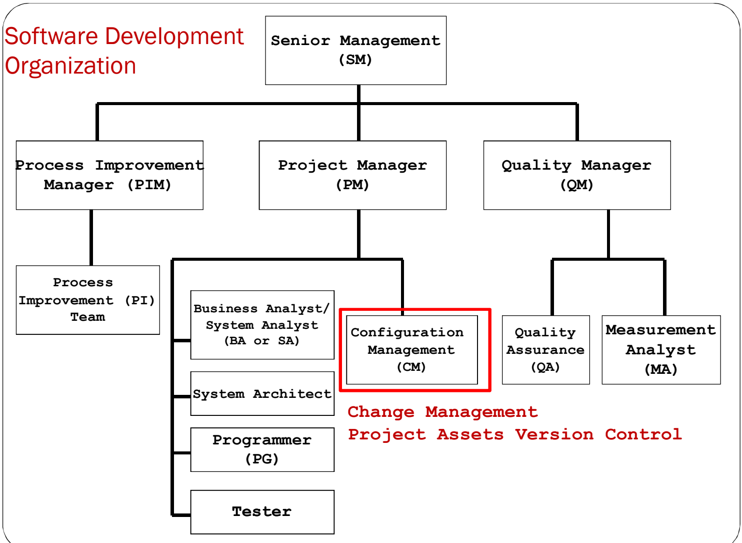
XP@Scrum

Scrum is an effective project management wrapper for XP development practices, which enables agile projects to become scalable and developed by distributed teams of developers.

Pro/Con of Agile Methods

- Advantages
  - Completely developed and tested features in short iterations
  - Simplicity of the process
  - Clearly defined rules
  - Increasing productivity
  - Self-organizing
  - each team member carries a lot of responsibility
  - Improved communication
  - Combination with Extreme Programming
- Drawbacks
  - "Undisciplined hacking" (no written documentation)
  - Violation of responsibility
  - Current mainly carried by the inventors

Software Development Organization



Senior Management (SM)

Process Improvement Manager (PIM)

Project Manager (PM)

Quality Manager (QM)

Process Improvement (PI) Team

Business Analyst/ System Analyst (BA or SA)

Configuration Management (CM)

Quality Assurance (QA)

Measurement Analyst (MA)

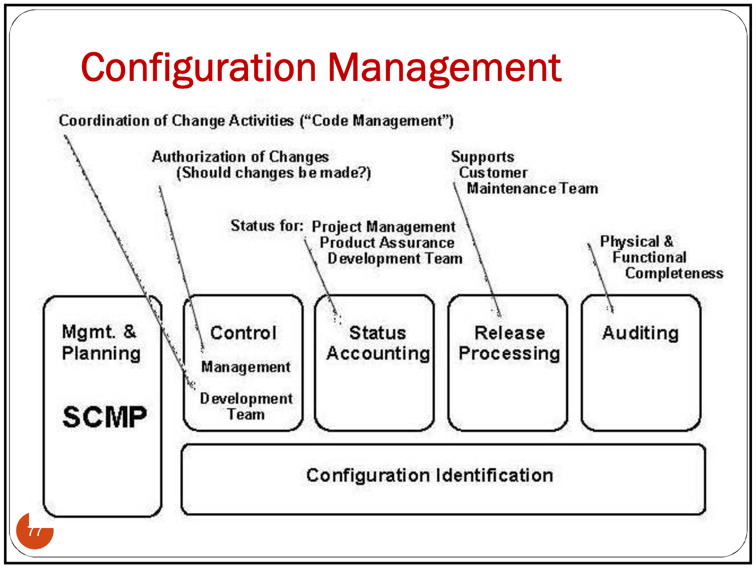
System Architect

Programmer (PG)

Tester

Change Management

Project Assets Version Control



Configuration Management

Coordination of Change Activities ("Code Management")

Authorization of Changes (Should changes be made?)

Supports Customer Maintenance Team

Status for: Project Management Product Assurance Development Team

Physical & Functional Completeness

Mgmt. & Planning

SCMP

Control

Management

Development Team

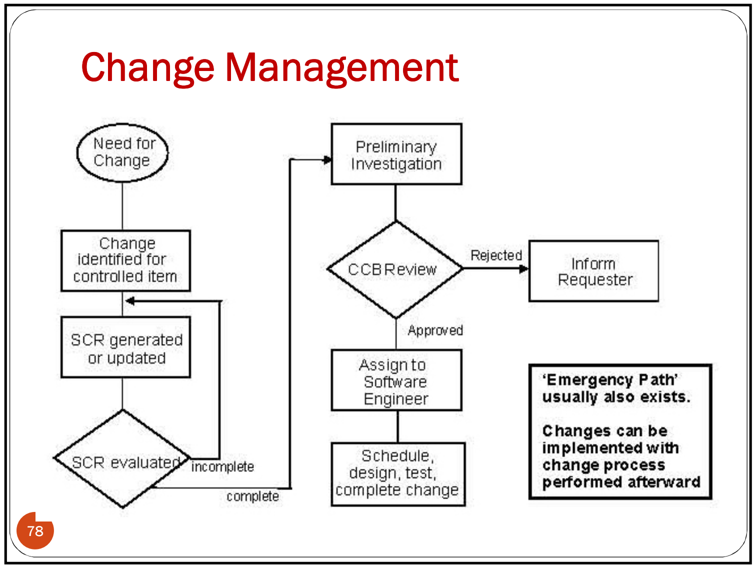
Status Accounting

Release Processing

Auditing

Configuration Identification

77



Change Management

Need for Change

Change identified for controlled item

SCR generated or updated

SCR evaluated

incomplete

complete

Preliminary Investigation

CCB Review

Rejected

Inform Requester

Approved

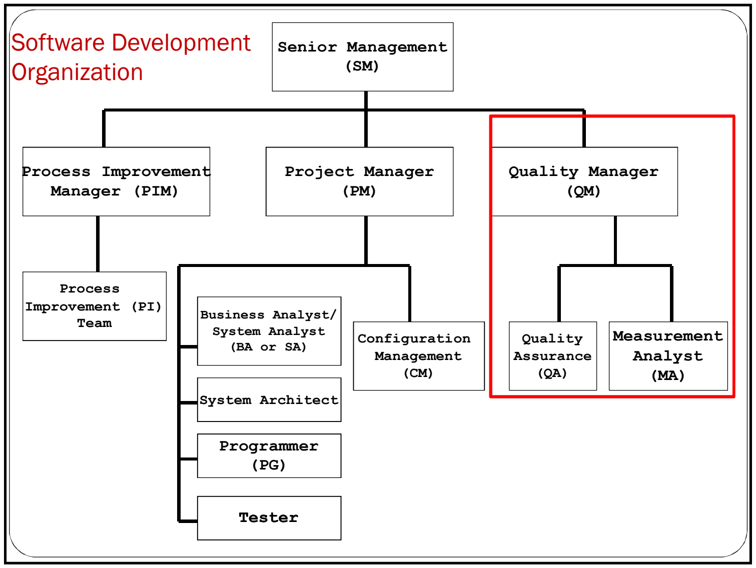
Assign to Software Engineer

Schedule, design, test, complete change

'Emergency Path' usually also exists.

Changes can be implemented with change process performed afterward

78



Software Development Organization

Senior Management (SM)

Process Improvement Manager (PIM)

Process Improvement (PI) Team

Project Manager (PM)

Business Analyst/ System Analyst (BA or SA)

System Architect

Programmer (PG)

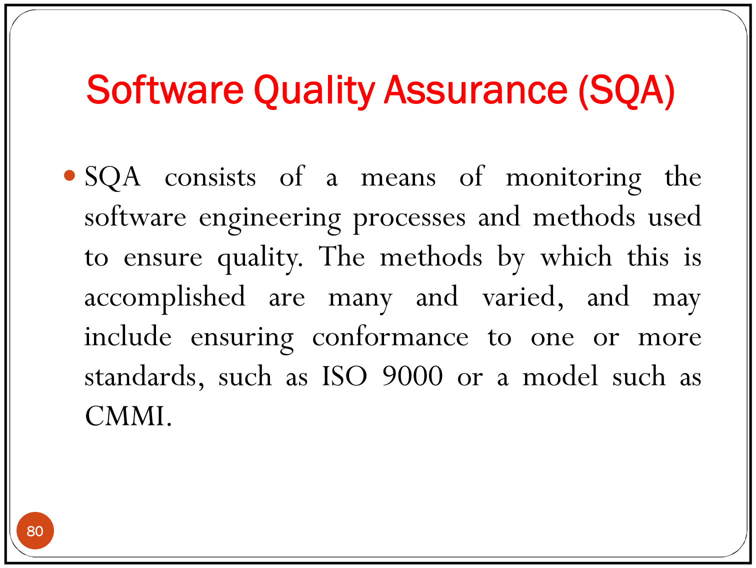
Tester

Configuration Management (CM)

Quality Manager (QM)

Quality Assurance (QA)

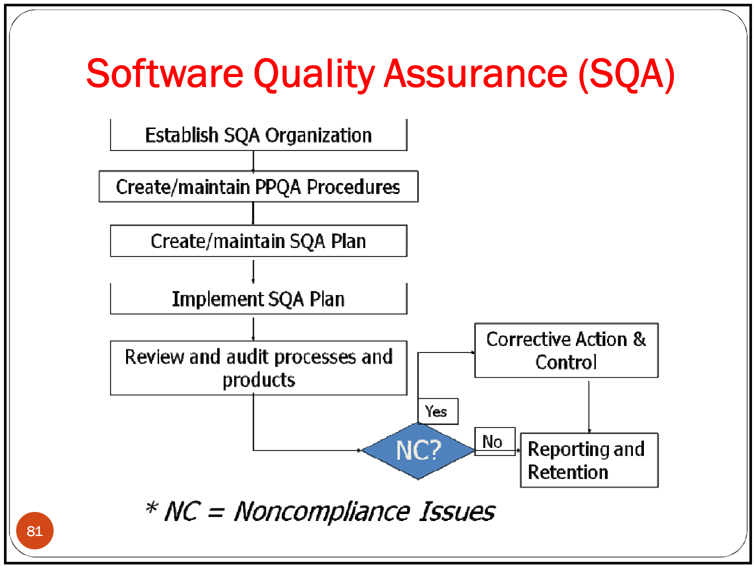
Measurement Analyst (MA)



Software Quality Assurance (SQA)

- SQA consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or a model such as CMMI.

80



Software Quality Assurance (SQA)

Establish SQA Organization

Create/maintain PPQA Procedures

Create/maintain SQA Plan

Implement SQA Plan

Review and audit processes and products

NC?

Yes

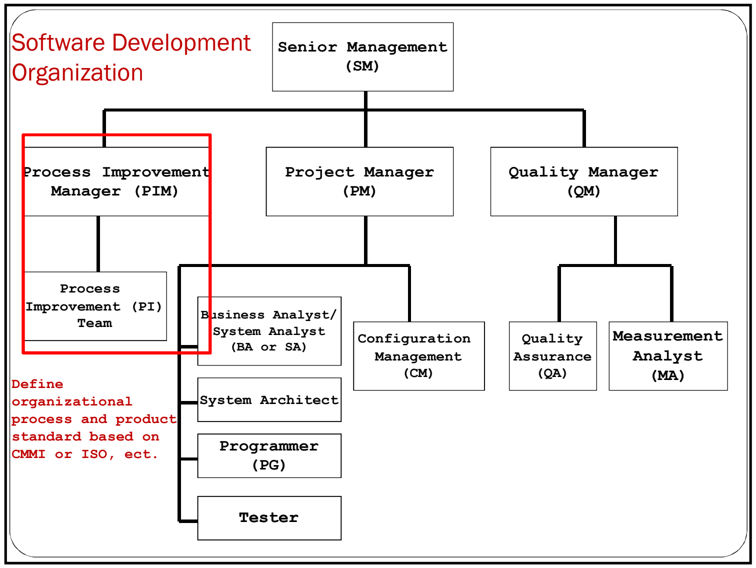
Corrective Action & Control

No

Reporting and Retention

** NC = Noncompliance Issues*

81



Software Development Organization

Senior Management (SM)

Process Improvement Manager (PIM)

Project Manager (PM)

Quality Manager (QM)

Process Improvement (PI) Team

Business Analyst/ System Analyst (BA or SA)

Configuration Management (CM)

Quality Assurance (QA)

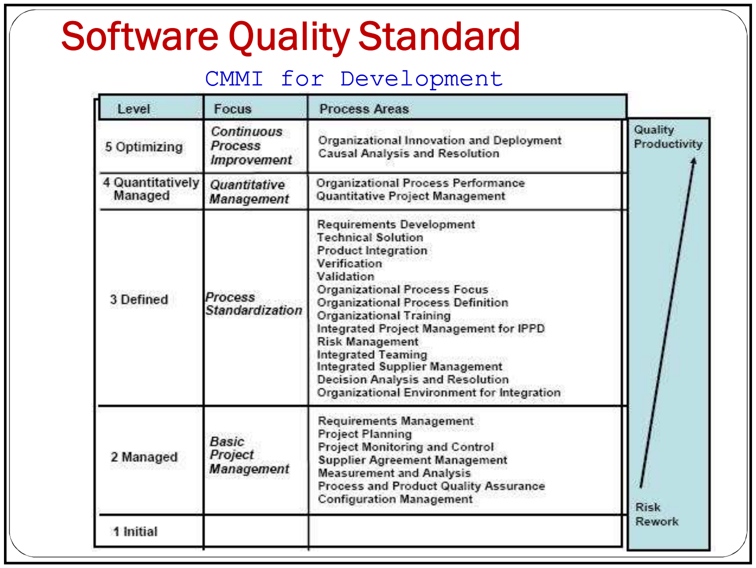
Measurement Analyst (MA)

System Architect

Programmer (PG)

Tester

Define organizational process and product standard based on CMMI or ISO, ect.

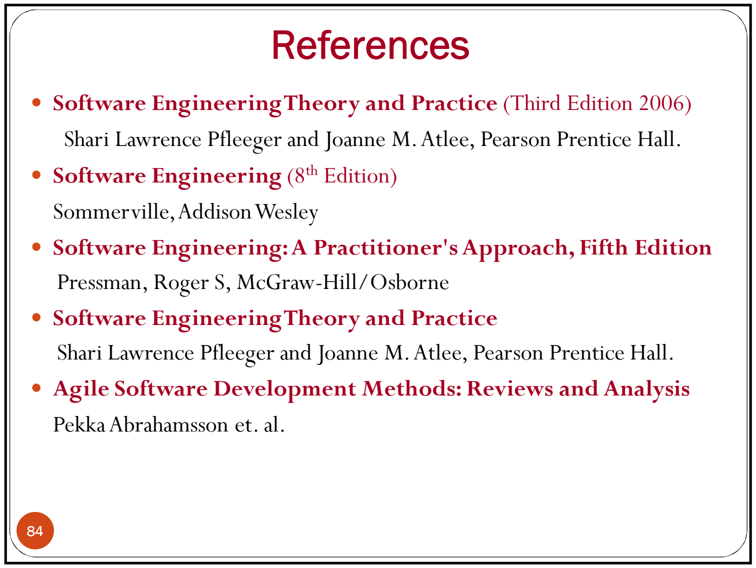


Software Quality Standard

CMMI for Development

Level	Focus	Process Areas
5 Optimizing	*Continuous Process Improvement*	Organizational Innovation and Deployment Causal Analysis and Resolution
4 Quantitatively Managed	*Quantitative Management*	Organizational Process Performance Quantitative Project Management
3 Defined	*Process Standardization*	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management for IPPD Risk Management Integrated Teaming Integrated Supplier Management Decision Analysis and Resolution Organizational Environment for Integration
2 Managed	*Basic Project Management*	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
1 Initial		

Quality
Productivity
Risk
Rework

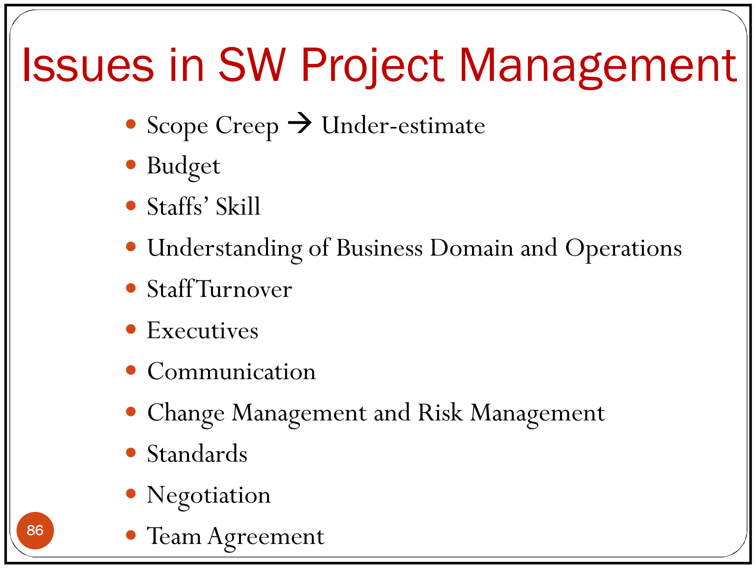


References

- **Software Engineering Theory and Practice** (Third Edition 2006)
  Shari Lawrence Pfleeger and Joanne M. Atlee, Pearson Prentice Hall.
- **Software Engineering** (8th Edition)
  Sommerville, Addison Wesley
- **Software Engineering: A Practitioner's Approach, Fifth Edition**
  Pressman, Roger S, McGraw-Hill/Osborne
- **Software Engineering Theory and Practice**
  Shari Lawrence Pfleeger and Joanne M. Atlee, Pearson Prentice Hall.
- **Agile Software Development Methods: Reviews and Analysis**
  Pekka Abrahamsson et. al.

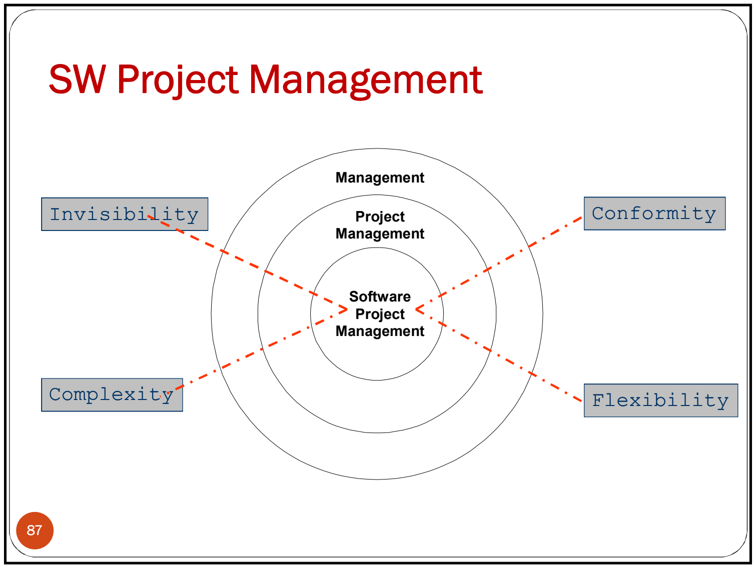
84

Software Project Management



Issues in SW Project Management

- Scope Creep → Under-estimate
- Budget
- Staffs' Skill
- Understanding of Business Domain and Operations
- Staff Turnover
- Executives
- Communication
- Change Management and Risk Management
- Standards
- Negotiation
- Team Agreement



SW Project Management

What is Project Management?

What is a project?

A temporary endeavor undertaken to create a unique product or service

Most projects meet the following criteria:

- Involve sequenced of planned activities or tasks that combine to achieve a specified goal
- Have a specific objective to be completed within certain quality or operational specifications
- Have a defined start and end dates
- Use people, money and/or equipment (collectively called resources in project management terminology)
- Have a budget or other resource limitations
- Involve a group of people who temporarily work together to achieve a desired result

What are the benefits and costs of implementing project management?

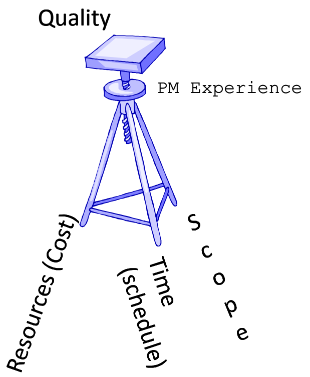
Benefits of Project Management

1. Ability to define your project's outcome and avoid "scope creep"
2. Ability to accurately estimate the time and resources necessary to complete your project successfully
3. Ability to schedule tasks and resources to avoid conflicts
4. Ability to anticipate problems and plan accordingly
5. Ability to bring your projects in on time, on target, and within budget

Costs of Project Management

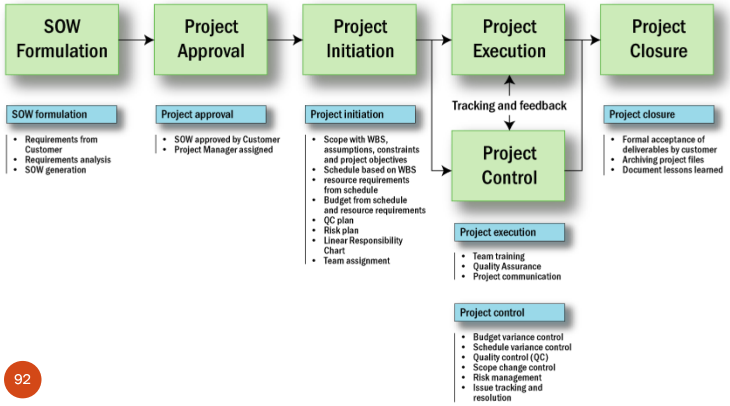
1. Time to learn project management and practice the process
2. Discipline to pay attention to the tools of project management
3. Getting agreement among the team and stakeholders

Another view of TRIPLE CONSTRAINTS in PM



Project Management Life Cycle

Localization Project Management Phases and Activities



SOW Formulation → Project Approval → Project Initiation → Project Execution → Project Closure

Tracking and feedback

Project Control

SOW formulation
- Requirements from Customer
- Requirements analysis
- SOW generation

Project approval
- SOW approved by Customer
- Project Manager assigned

Project initiation
- Scope with WBS, assumptions, constraints and project objectives
- Schedule based on WBS
- resource requirements from schedule
- Budget from schedule and resource requirements
- QC plan
- Risk plan
- Linear Responsibility Chart
- Team assignment

Project execution
- Team training
- Quality Assurance
- Project communication

Project control
- Budget variance control
- Schedule variance control
- Quality control (QC)
- Scope change control
- Risk management
- Issue tracking and resolution

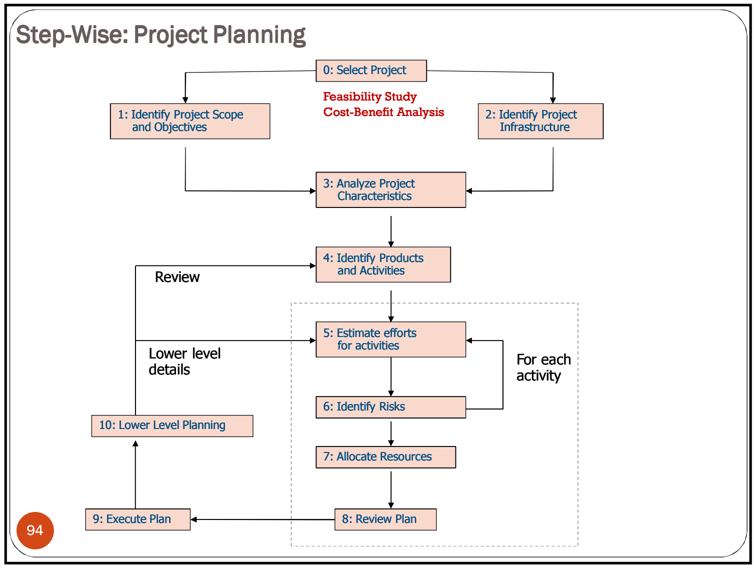
Project closure
- Formal acceptance of deliverables by customer
- Archiving project files
- Document lessons learned

Session 2

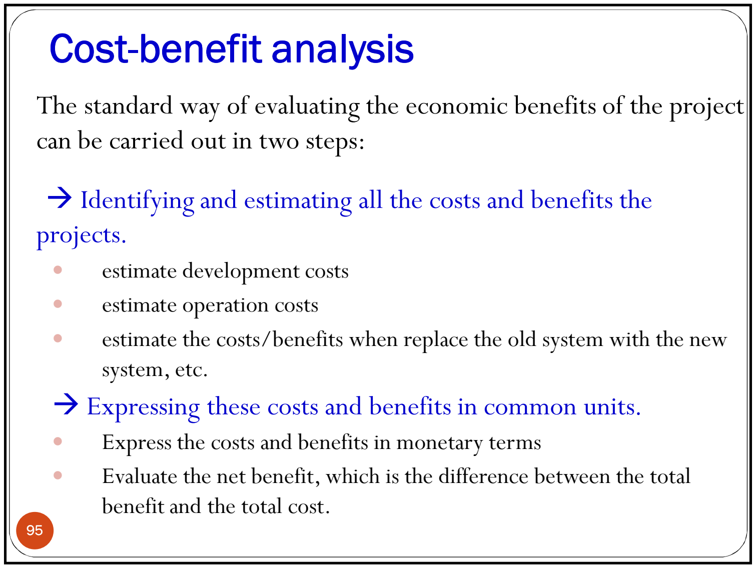
Step-wise

Project Planning

Referenced Textbook:
Software Project Management by Bob Hughes & Mike Cotterell.



Step-Wise: Project Planning



Cost-benefit analysis

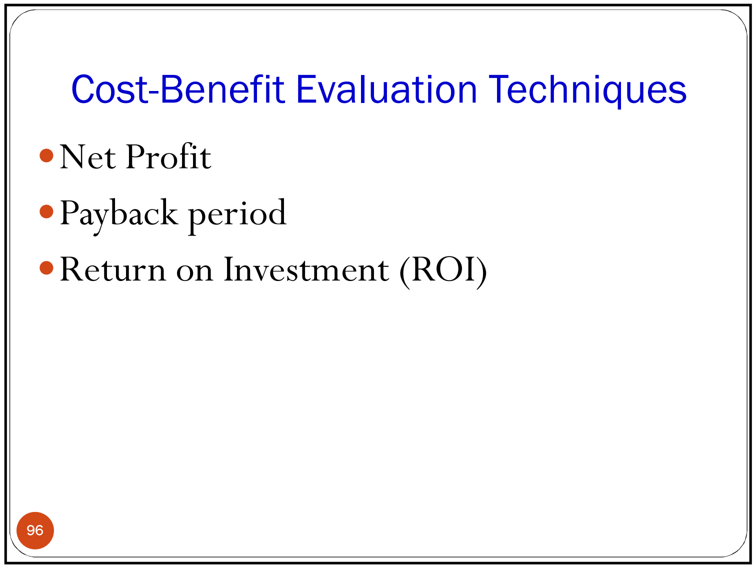
The standard way of evaluating the economic benefits of the project can be carried out in two steps:

→ Identifying and estimating all the costs and benefits the projects.

- estimate development costs
- estimate operation costs
- estimate the costs/benefits when replace the old system with the new system, etc.

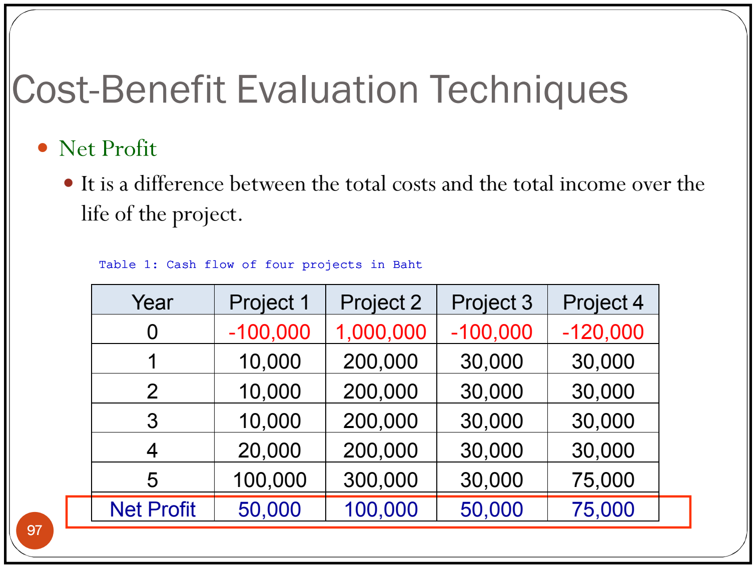
→ Expressing these costs and benefits in common units.

- Express the costs and benefits in monetary terms
- Evaluate the net benefit, which is the difference between the total benefit and the total cost.



Cost-Benefit Evaluation Techniques

- Net Profit
- Payback period
- Return on Investment (ROI)

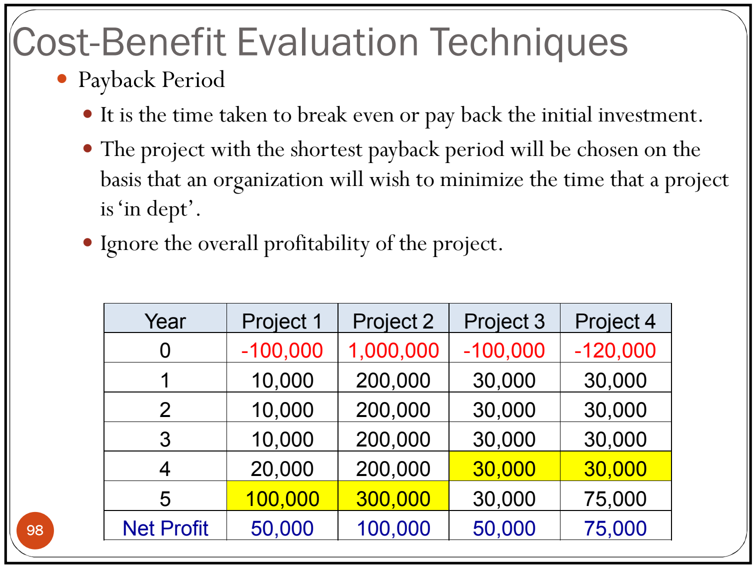


Cost-Benefit Evaluation Techniques

- Net Profit
  - It is a difference between the total costs and the total income over the life of the project.

Table 1: Cash flow of four projects in Baht

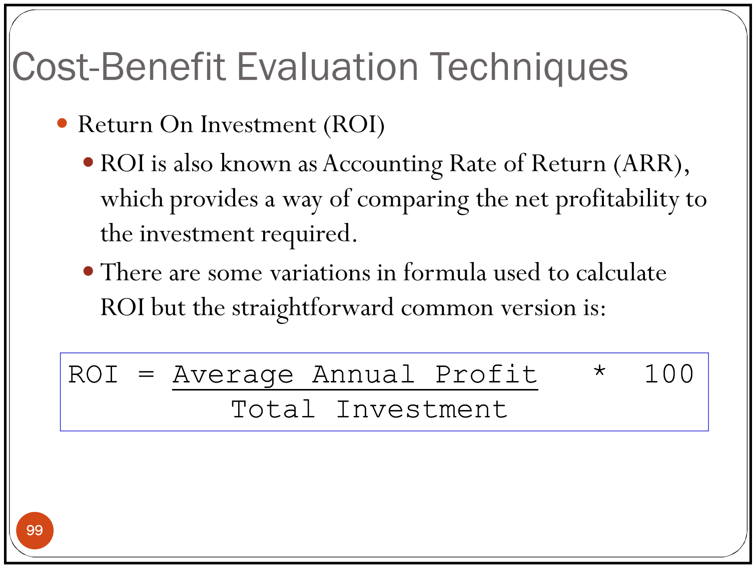
Year	Project 1	Project 2	Project 3	Project 4
0	-100,000	1,000,000	-100,000	-120,000
1	10,000	200,000	30,000	30,000
2	10,000	200,000	30,000	30,000
3	10,000	200,000	30,000	30,000
4	20,000	200,000	30,000	30,000
5	100,000	300,000	30,000	75,000
Net Profit	50,000	100,000	50,000	75,000



Cost-Benefit Evaluation Techniques

- Payback Period
  - It is the time taken to break even or pay back the initial investment.
  - The project with the shortest payback period will be chosen on the basis that an organization will wish to minimize the time that a project is 'in dept'.
  - Ignore the overall profitability of the project.

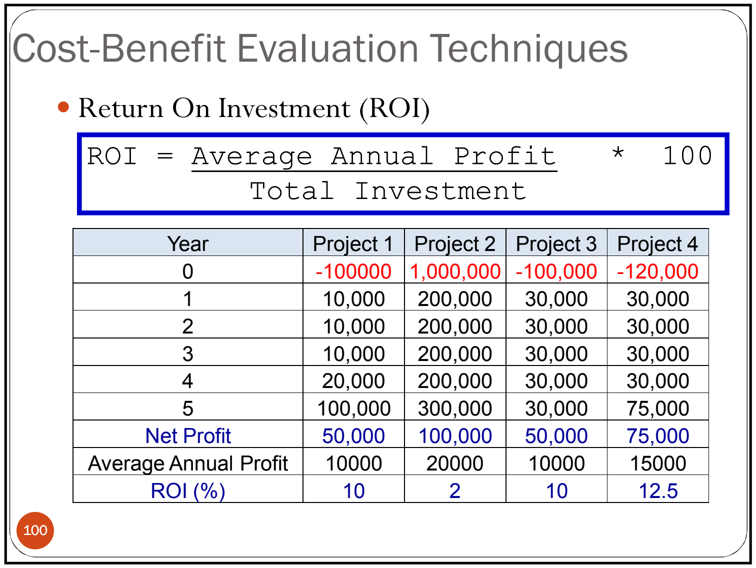
Year	Project 1	Project 2	Project 3	Project 4
0	-100,000	1,000,000	-100,000	-120,000
1	10,000	200,000	30,000	30,000
2	10,000	200,000	30,000	30,000
3	10,000	200,000	30,000	30,000
4	20,000	200,000	30,000	30,000
5	100,000	300,000	30,000	75,000
Net Profit	50,000	100,000	50,000	75,000



Cost-Benefit Evaluation Techniques

- Return On Investment (ROI)
  - ROI is also known as Accounting Rate of Return (ARR), which provides a way of comparing the net profitability to the investment required.
  - There are some variations in formula used to calculate ROI but the straightforward common version is:

$$ROI = \frac{\text{Average Annual Profit}}{\text{Total Investment}} * 100$$



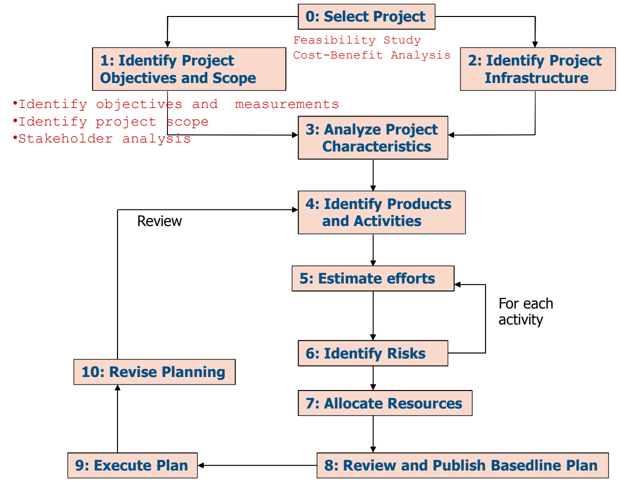
Cost-Benefit Evaluation Techniques

- Return On Investment (ROI)

$$ROI = \frac{\text{Average Annual Profit}}{\text{Total Investment}} * 100$$

Year	Project 1	Project 2	Project 3	Project 4
0	-100000	1,000,000	-100,000	-120,000
1	10,000	200,000	30,000	30,000
2	10,000	200,000	30,000	30,000
3	10,000	200,000	30,000	30,000
4	20,000	200,000	30,000	30,000
5	100,000	300,000	30,000	75,000
Net Profit	50,000	100,000	50,000	75,000
Average Annual Profit	10000	20000	10000	15000
ROI (%)	10	2	10	12.5

Step-Wise: Project Planning



0: Select Project
Feasibility Study
Cost-Benefit Analysis
1: Identify Project Objectives and Scope
2: Identify Project Infrastructure
•Identify objectives and measurements
•Identify project scope
•Stakeholder analysis
3: Analyze Project Characteristics
4: Identify Products and Activities
Review
5: Estimate efforts
For each activity
6: Identify Risks
10: Revise Planning
7: Allocate Resources
9: Execute Plan
8: Review and Publish Basedline Plan

Step 1: Identify project objectives and Scope

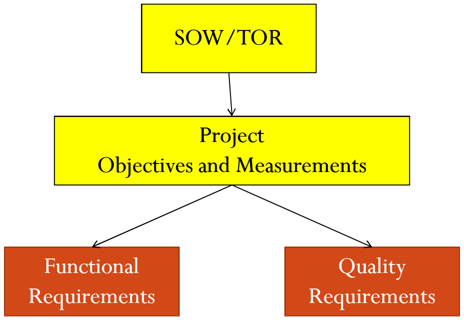
1. Identify project objectives and measurements of effectiveness in meeting those objectives
2. Identify project scope
3. Stakeholder analysis

   Identify all stakeholders in the project and their expectations
4. Modify objectives in the light of stakeholder analysis

Step 1: Identify project scope and objectives

Identify Objectives and Project Scope

- Analyze Requirements



103

Functional Requirements

- Functionality
  - What will the system do?
  - When will the system do it?
  - Are there several modes of operations?
  - What kinds of computations or data transformations must be performed?
  - What are the appropriate reactions to possible actions?
- Data
  - What should be the format of input and output?
  - Must any data be retained for any period of time?

24-Feb-11

Tools for Capturing Functional Requirements

- Form-based Document

```
Function:________________________
   ___
Description:_____________________
   ____
Inputs:__________________________
   ___
Outputs:_________________________
   ___
Pre-
   condition:____________________
   ___
Post-
   condition:____________________
   __
         :
     :
```

24-Feb-11

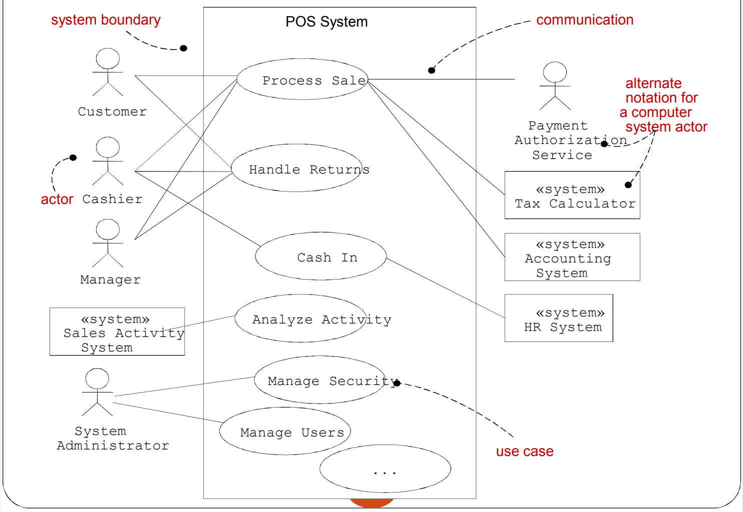
Tools for Capturing Functional Requirements

- Tabular Specification

Example:

Condition	Action
Member = yes	discount = 5% then Net Price = total price – (total price*discount)
Member = no	discount = 0% then Net Price = total price

An Example: A Point-Of-Sale System



Quality Requirements/ Non-Functional Requirements

- Performance
- Usability and Human Factors
- Security
- Reliability and Availability
- Maintainability
- Precision and Accuracy

24-Feb-11

Quality Requirements/ Non-Functional Requirements

Quality Requirements	Measurement
Speed	- Execution time (sec.) - Response time (sec.)
Size	- Kbyte - Size of required RAM
Usability	-Time required for user training - Help Topics
Reliability	- Average number of bugs (errors) - Possibility of System Failure - Rate of System Failure
Portability	- numbers of platforms

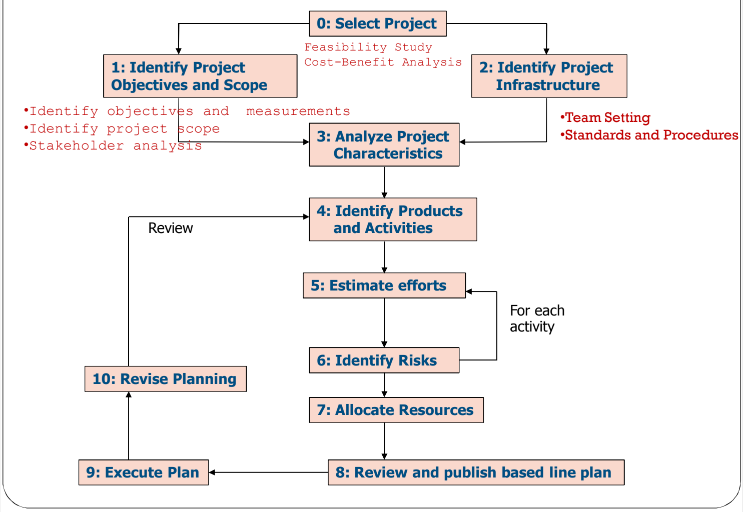
24-Feb-11

Project Stakeholders

- **Stakeholders** are the people involved in or affected by project activities
- Stakeholders include
  - project sponsors
  - project manager
  - project team
  - support staffs
  - customers
  - users
  - suppliers

110

Step-Wise: Project Planning

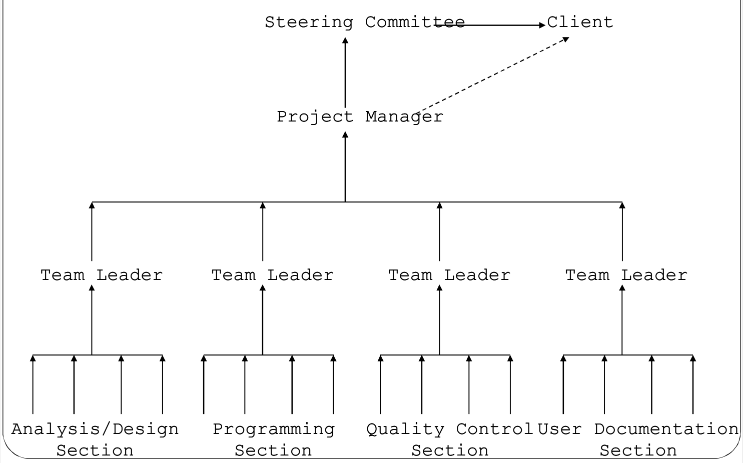

0: Select Project
Feasibility Study
Cost-Benefit Analysis
1: Identify Project Objectives and Scope
2: Identify Project Infrastructure
•Identify objectives and measurements
•Identify project scope
•Stakeholder analysis
•Team Setting
•Standards and Procedures
3: Analyze Project Characteristics
4: Identify Products and Activities
Review
5: Estimate efforts
For each activity
6: Identify Risks
10: Revise Planning
7: Allocate Resources
9: Execute Plan
8: Review and publish based line plan

Step 2: Identify project infrastructure

- Identify project team organization
- Identify process standards and procedures
  - ex. CMMI, ITIL, ISO, etc.

112

Team Organization and Project Reporting Structures



Categories of Reporting

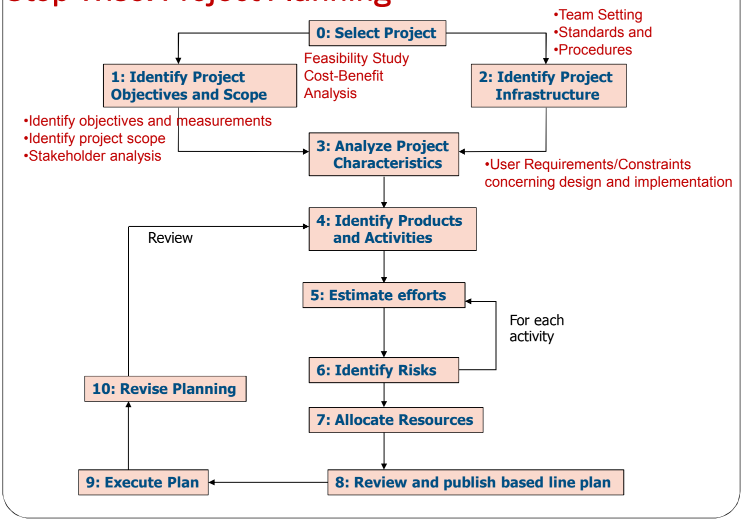
Report Type	Examples	Comment
Oral Formal (Regular)	Weekly or Monthly Progress Meetings	While reports may be oral, formal written MOM should be kept.
Oral Formal (Ad hoc)	End-of-Stage Review Meetings	While largely oral, likely to receive and generate written reports.
Written Formal (Regular)	Job Sheets, Progress Reports	Normally weekly using forms.
Written Formal (Ad hoc)	Exception Reports, Change Reports	
Oral Informal (Ad hoc)	Canteen Discussion, Social Interaction	Often provides early warning; must be backed up by formal reporting.

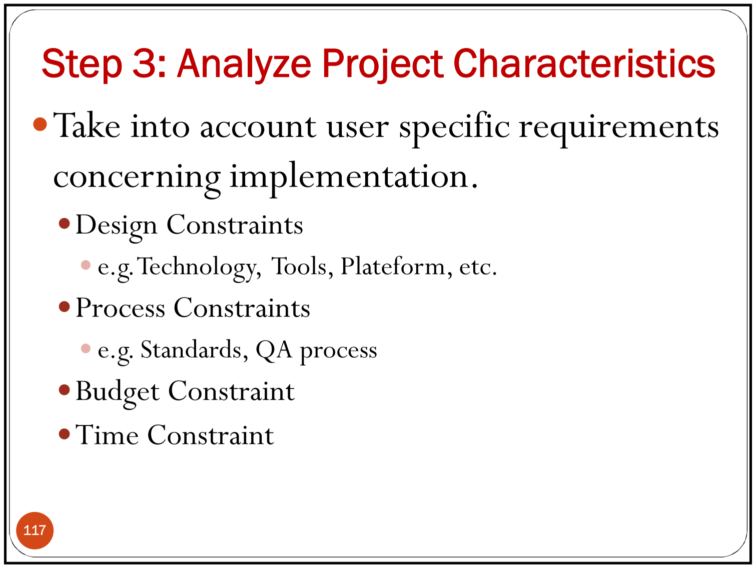
Step 2: Identify project infrastructure

- Identify project team organization
- Identify process standards and procedures
  - ex. CMMI, ITIL, ISO, etc.
  - Templates

115

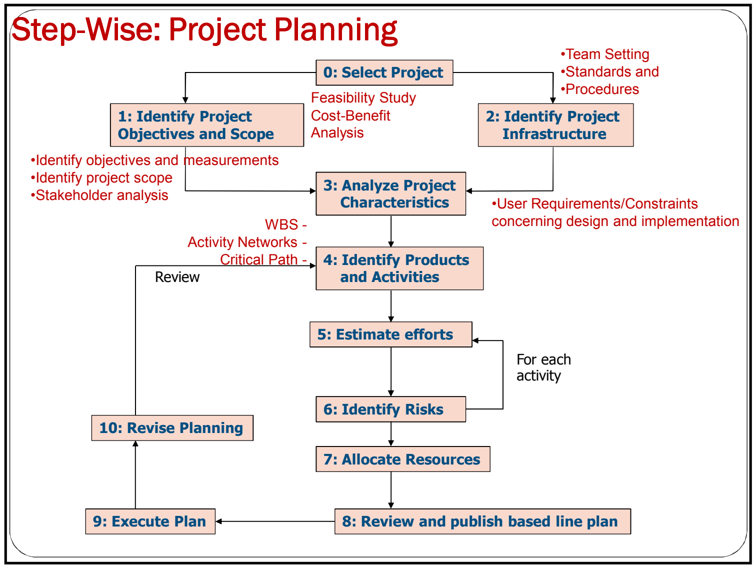
Step-Wise: Project Planning


0: Select Project
Feasibility Study
Cost-Benefit Analysis
•Team Setting
•Standards and
•Procedures
1: Identify Project Objectives and Scope
2: Identify Project Infrastructure
•Identify objectives and measurements
•Identify project scope
•Stakeholder analysis
3: Analyze Project Characteristics
•User Requirements/Constraints concerning design and implementation
4: Identify Products and Activities
Review
5: Estimate efforts
For each activity
6: Identify Risks
10: Revise Planning
7: Allocate Resources
9: Execute Plan
8: Review and publish based line plan

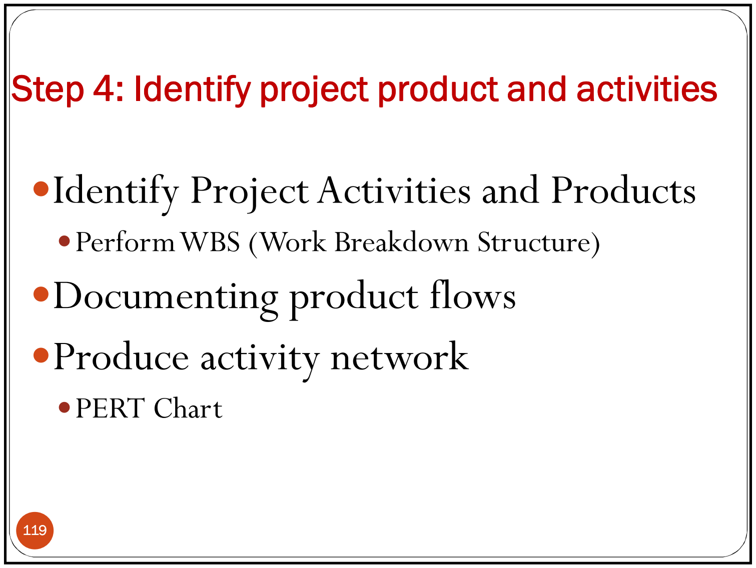


Step 3: Analyze Project Characteristics

- Take into account user specific requirements concerning implementation.
  - Design Constraints
    - e.g. Technology, Tools, Plateform, etc.
  - Process Constraints
    - e.g. Standards, QA process
  - Budget Constraint
  - Time Constraint

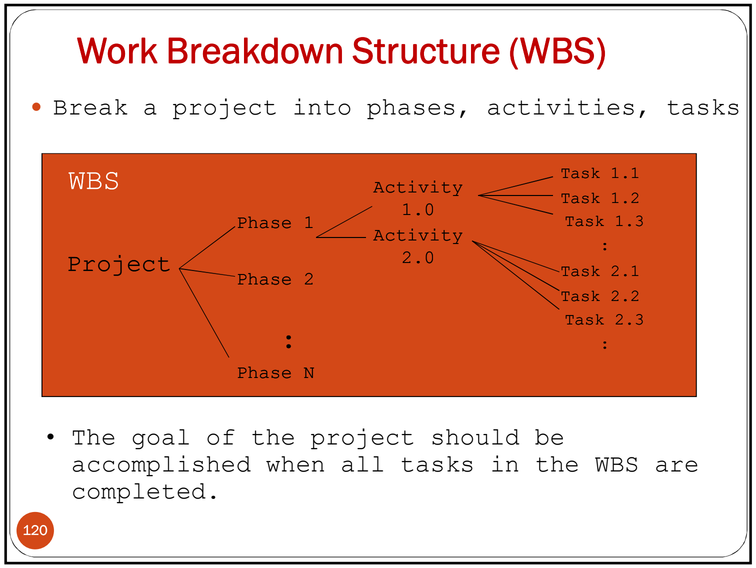


Step-Wise: Project Planning



Step 4: Identify project product and activities

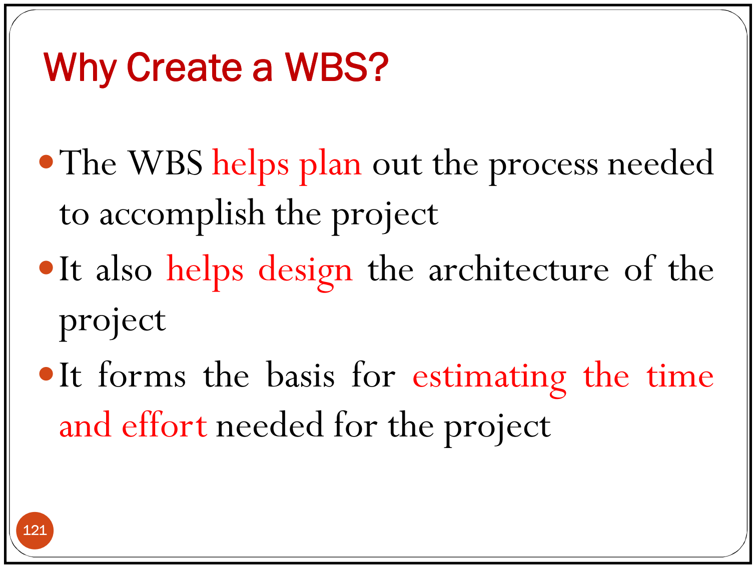
- Identify Project Activities and Products
  - Perform WBS (Work Breakdown Structure)
- Documenting product flows
- Produce activity network
  - PERT Chart



Work Breakdown Structure (WBS)

- Break a project into phases, activities, tasks

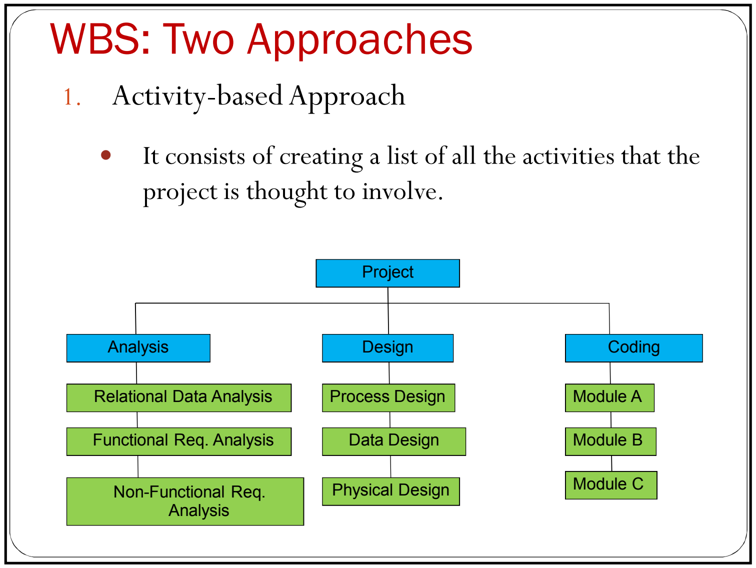
- The goal of the project should be accomplished when all tasks in the WBS are completed.



Why Create a WBS?

- The WBS helps plan out the process needed to accomplish the project
- It also helps design the architecture of the project
- It forms the basis for estimating the time and effort needed for the project

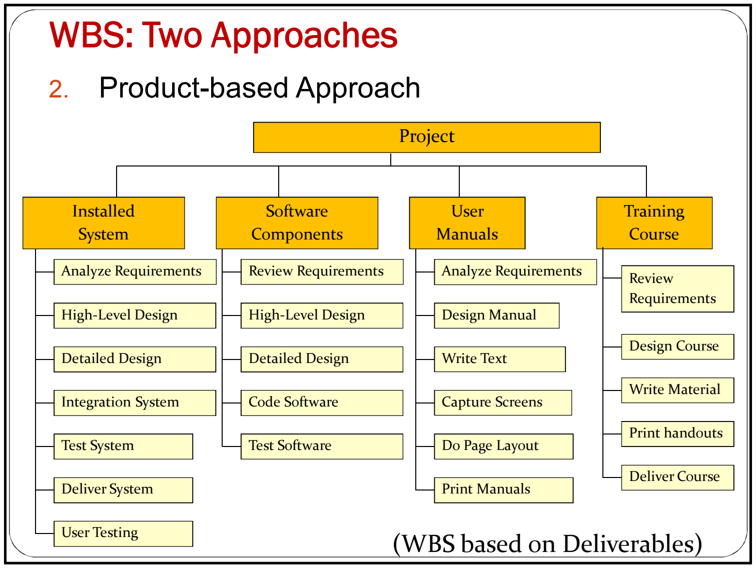
121



WBS: Two Approaches

1. Activity-based Approach
   - It consists of creating a list of all the activities that the project is thought to involve.

- Project
  - Analysis
    - Relational Data Analysis
    - Functional Req. Analysis
    - Non-Functional Req. Analysis
  - Design
    - Process Design
    - Data Design
    - Physical Design
  - Coding
    - Module A
    - Module B
    - Module C

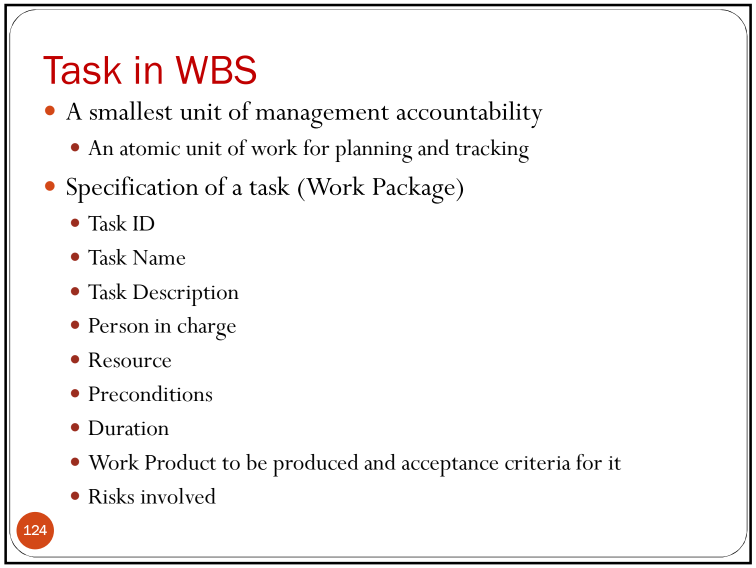


WBS: Two Approaches

2. Product-based Approach

- Project
  - Installed System
    - Analyze Requirements
    - High-Level Design
    - Detailed Design
    - Integration System
    - Test System
    - Deliver System
    - User Testing
  - Software Components
    - Review Requirements
    - High-Level Design
    - Detailed Design
    - Code Software
    - Test Software
  - User Manuals
    - Analyze Requirements
    - Design Manual
    - Write Text
    - Capture Screens
    - Do Page Layout
    - Print Manuals
  - Training Course
    - Review Requirements
    - Design Course
    - Write Material
    - Print handouts
    - Deliver Course

(WBS based on Deliverables)

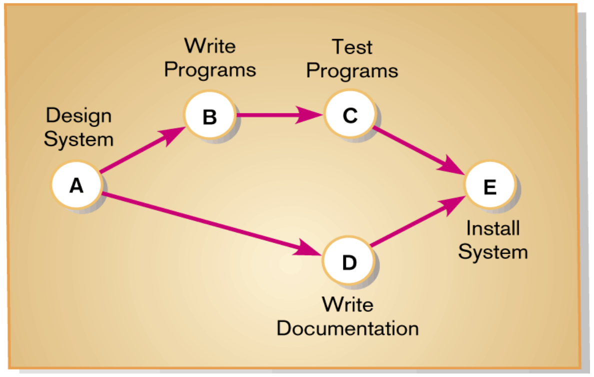


Task in WBS

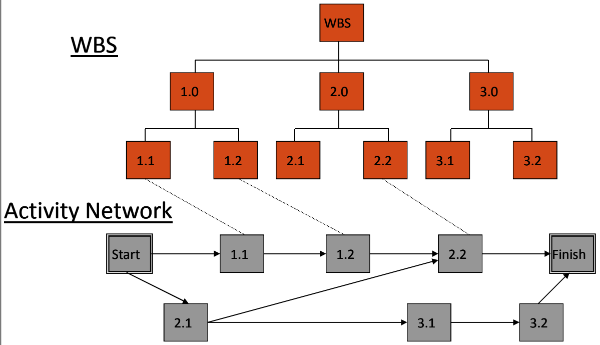
- A smallest unit of management accountability
  - An atomic unit of work for planning and tracking
- Specification of a task (Work Package)
  - Task ID
  - Task Name
  - Task Description
  - Person in charge
  - Resource
  - Preconditions
  - Duration
  - Work Product to be produced and acceptance criteria for it
  - Risks involved

124

Activity Network (PERT Chart)

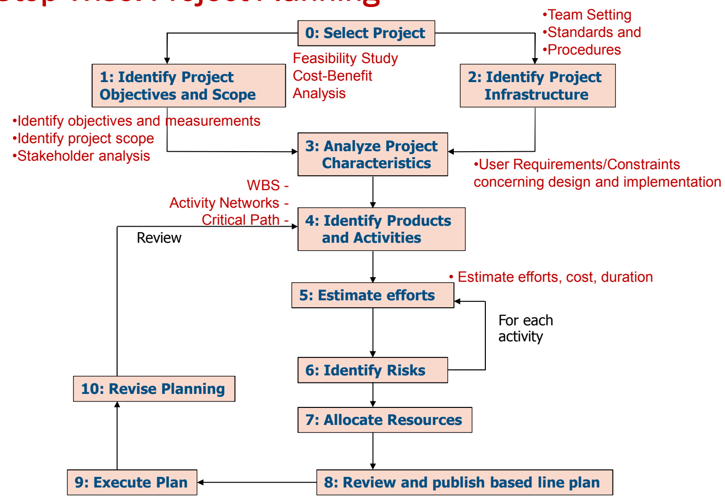


Network Relationship to the WBS


WBS

Activity Network

Step-Wise: Project Planning


0: Select Project

•Team Setting
•Standards and
•Procedures

Feasibility Study
Cost-Benefit
Analysis

1: Identify Project Objectives and Scope

2: Identify Project Infrastructure

•Identify objectives and measurements
•Identify project scope
•Stakeholder analysis

3: Analyze Project Characteristics

•User Requirements/Constraints concerning design and implementation

WBS -
Activity Networks -
Critical Path -

4: Identify Products and Activities

Review

• Estimate efforts, cost, duration

5: Estimate efforts

For each activity

6: Identify Risks

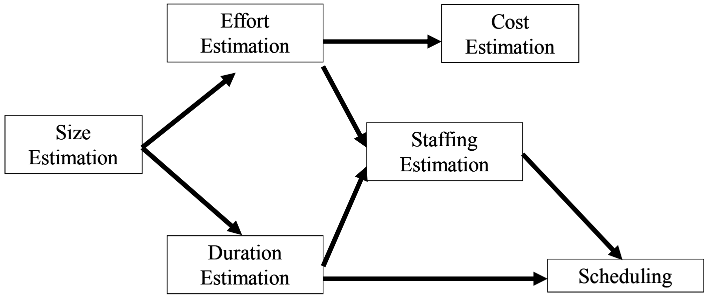
10: Revise Planning

7: Allocate Resources

9: Execute Plan

8: Review and publish based line plan

Project Estimation


Effort Estimation

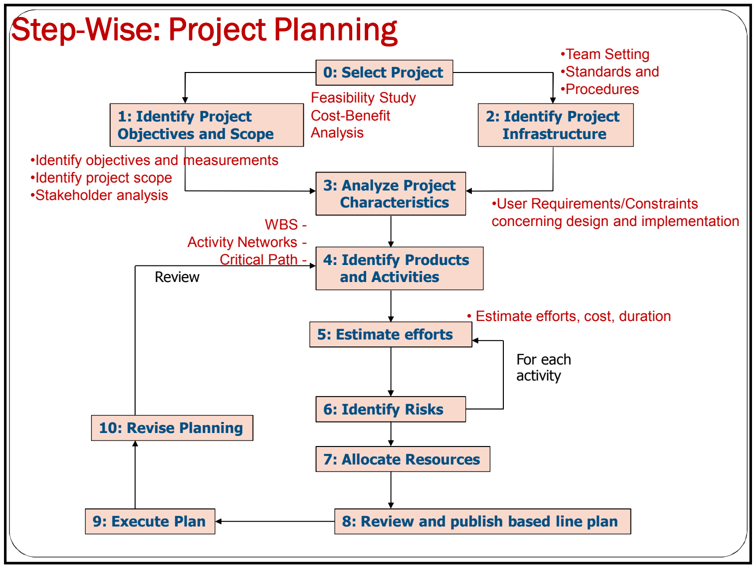
Cost Estimation

Size Estimation

Staffing Estimation

Duration Estimation

Scheduling



Step-Wise: Project Planning

0: Select Project

Feasibility Study
Cost-Benefit
Analysis

•Team Setting
•Standards and
•Procedures

1: Identify Project Objectives and Scope

•Identify objectives and measurements
•Identify project scope
•Stakeholder analysis

2: Identify Project Infrastructure

3: Analyze Project Characteristics

•User Requirements/Constraints
concerning design and implementation

WBS -
Activity Networks -
Critical Path -

4: Identify Products and Activities

Review

5: Estimate efforts

• Estimate efforts, cost, duration

For each
activity

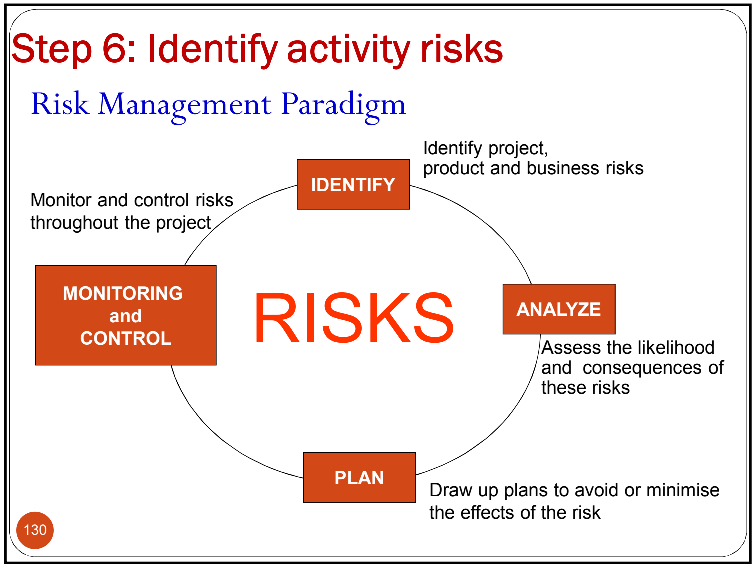
6: Identify Risks

7: Allocate Resources

8: Review and publish based line plan

9: Execute Plan

10: Revise Planning



Step 6: Identify activity risks

Risk Management Paradigm

RISKS

IDENTIFY

Identify project,
product and business risks

ANALYZE

Assess the likelihood
and consequences of
these risks

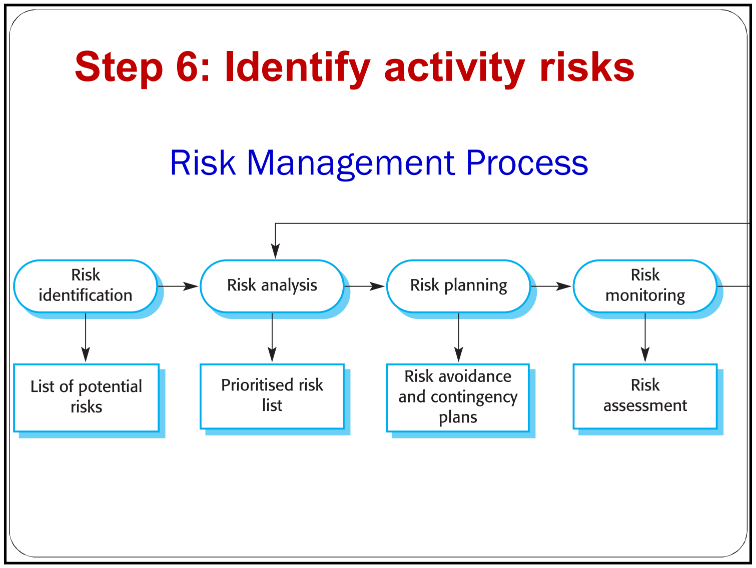
PLAN

Draw up plans to avoid or minimise
the effects of the risk

MONITORING
and
CONTROL

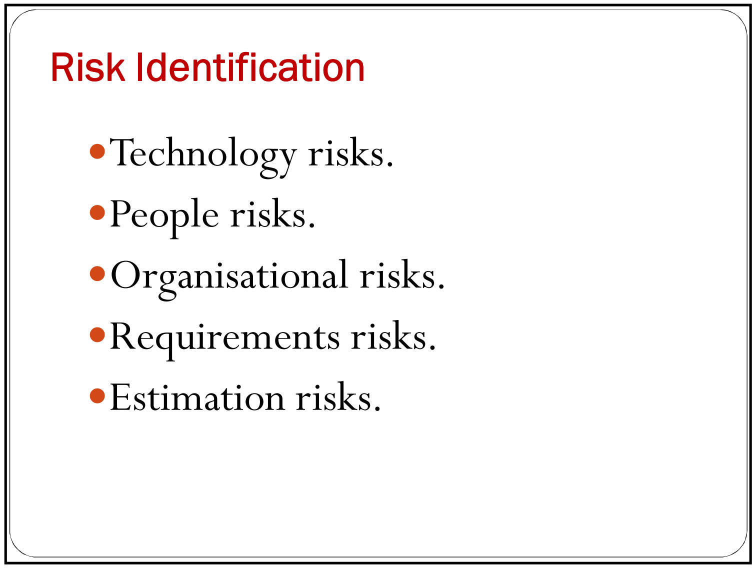
Monitor and control risks
throughout the project

130



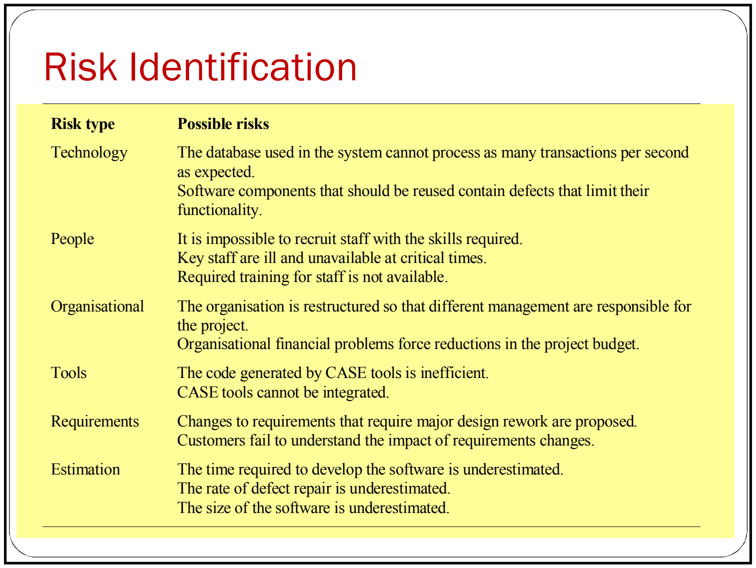
Step 6: Identify activity risks

Risk Management Process



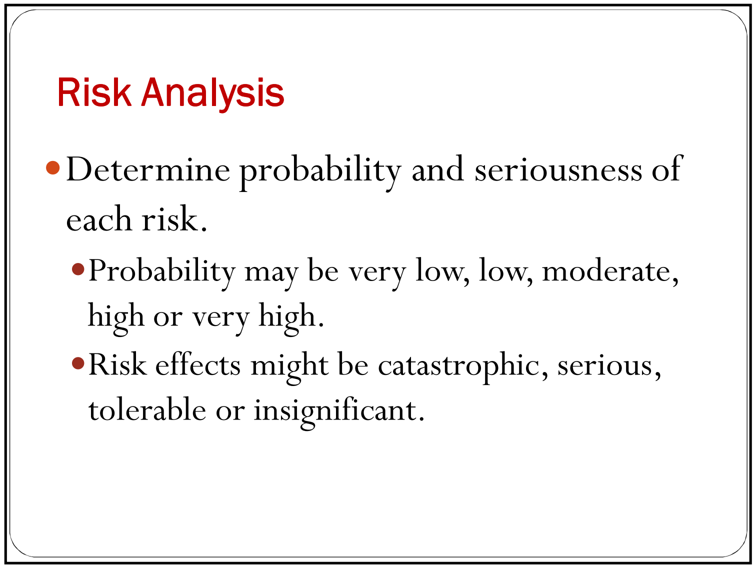
Risk Identification

- Technology risks.
- People risks.
- Organisational risks.
- Requirements risks.
- Estimation risks.



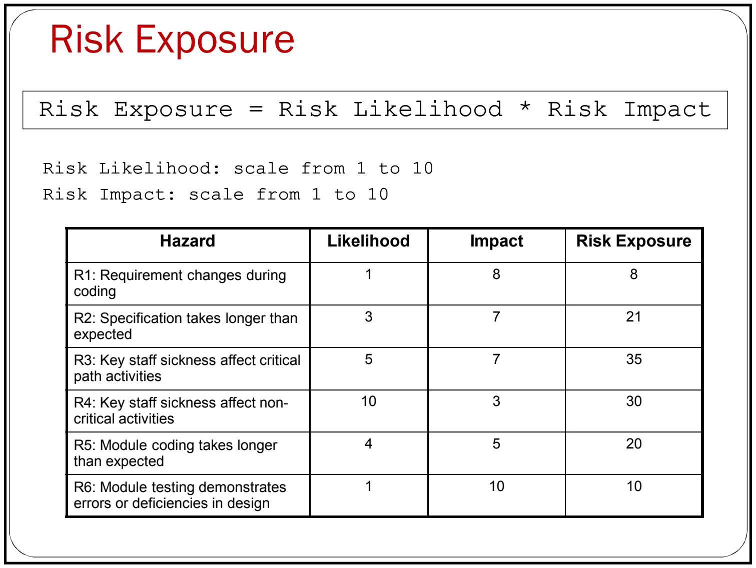
Risk Identification

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. Software components that should be reused contain defects that limit their functionality.
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.
Organisational	The organisation is restructured so that different management are responsible for the project. Organisational financial problems force reductions in the project budget.
Tools	The code generated by CASE tools is inefficient. CASE tools cannot be integrated.
Requirements	Changes to requirements that require major design rework are proposed. Customers fail to understand the impact of requirements changes.
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.



Risk Analysis

- Determine probability and seriousness of each risk.
  - Probability may be very low, low, moderate, high or very high.
  - Risk effects might be catastrophic, serious, tolerable or insignificant.

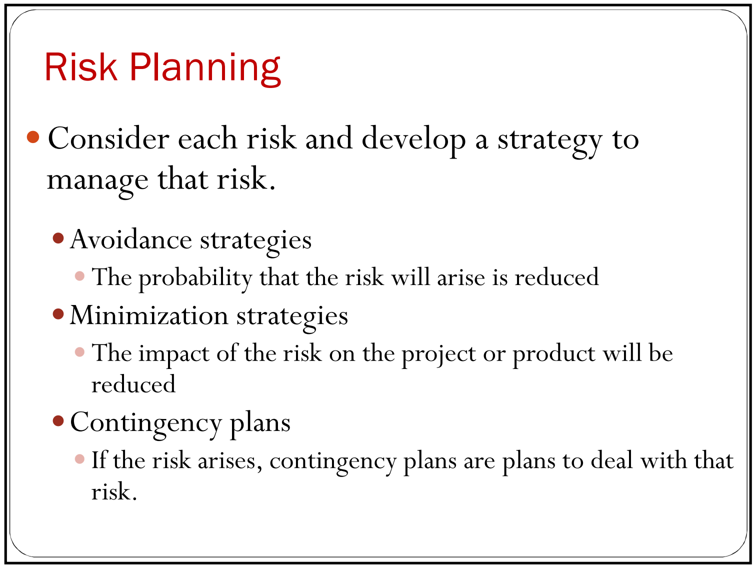


Risk Exposure

```
Risk Exposure = Risk Likelihood * Risk Impact
```

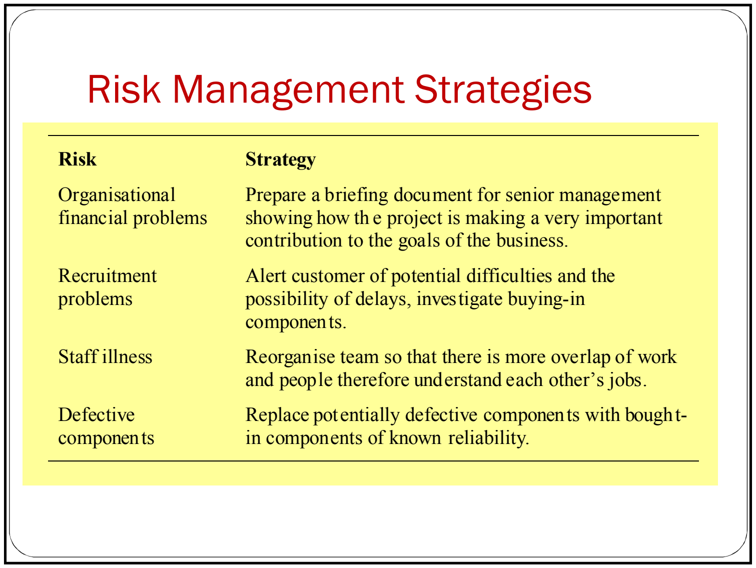
```
Risk Likelihood: scale from 1 to 10
Risk Impact: scale from 1 to 10
```

Hazard	Likelihood	Impact	Risk Exposure
R1: Requirement changes during coding	1	8	8
R2: Specification takes longer than expected	3	7	21
R3: Key staff sickness affect critical path activities	5	7	35
R4: Key staff sickness affect non-critical activities	10	3	30
R5: Module coding takes longer than expected	4	5	20
R6: Module testing demonstrates errors or deficiencies in design	1	10	10



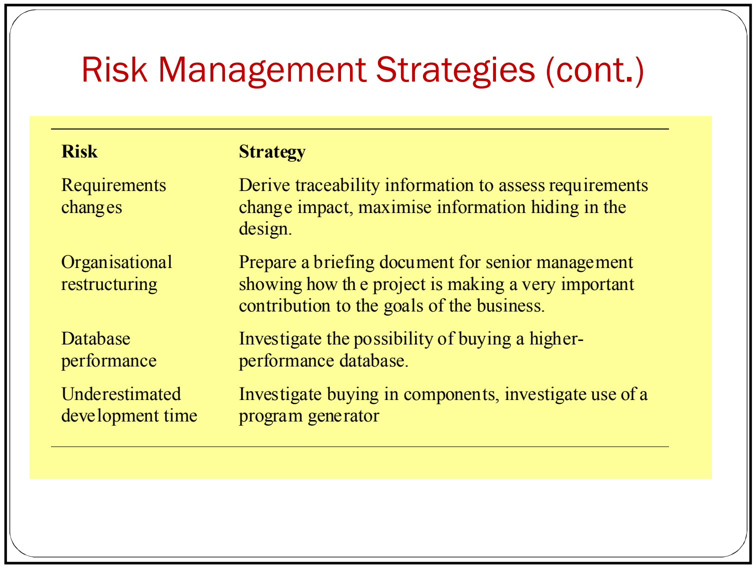
Risk Planning

- 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.



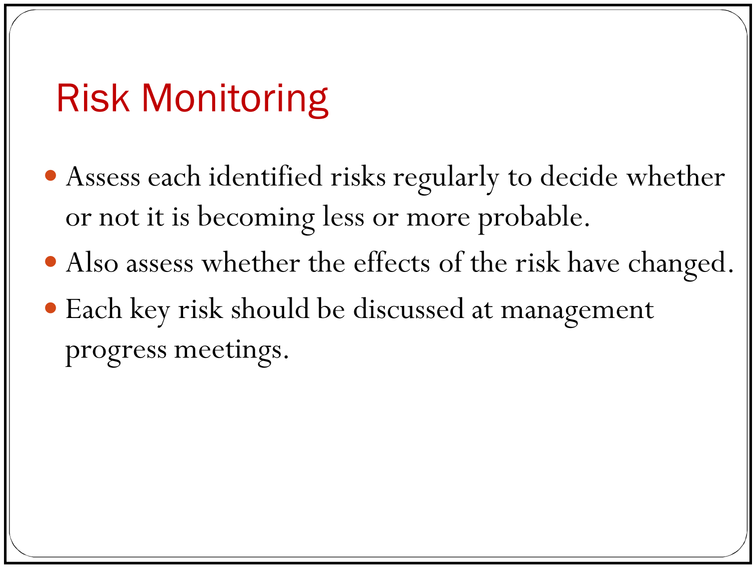
Risk Management Strategies

Risk	Strategy
Organisational 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.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components.
Staff illness	Reorganise 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.



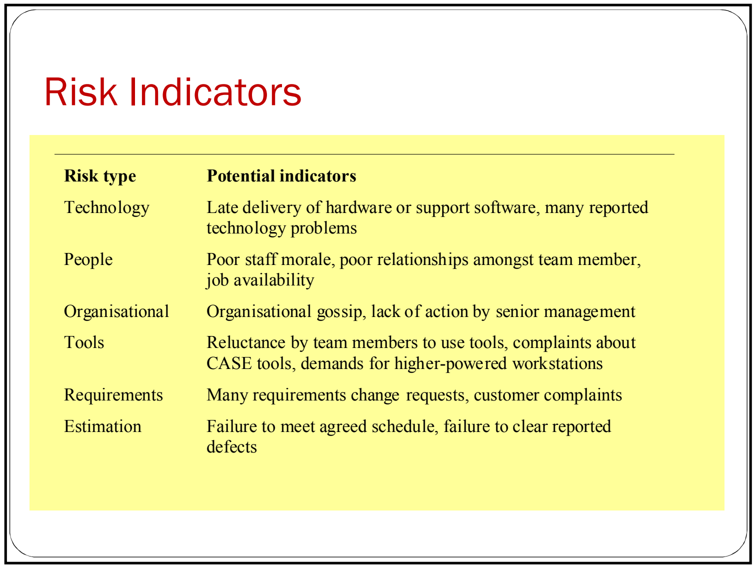
Risk Management Strategies (cont.)

Risk	Strategy
Requirements changes	Derive traceability information to assess requirements change impact, maximise information hiding in the design.
Organisational 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

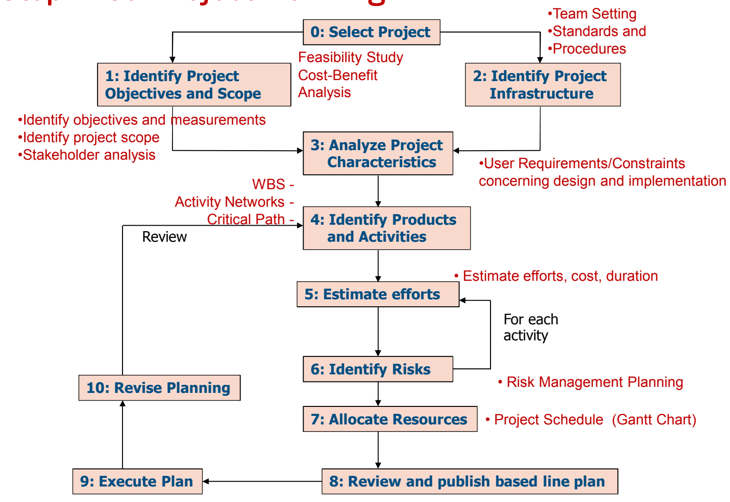
- Assess each identified risks regularly to decide whether or not it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Each key risk should be discussed at management progress meetings.



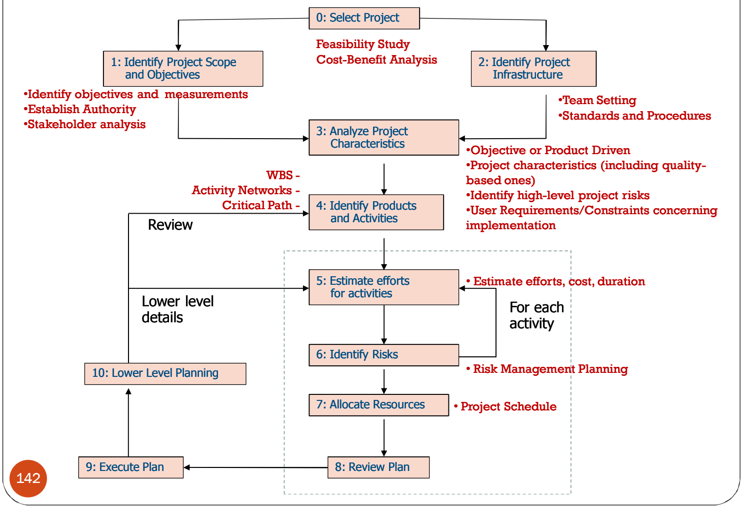
Risk Indicators

Risk type	Potential indicators
Technology	Late delivery of hardware or support software, many reported technology problems
People	Poor staff morale, poor relationships amongst team member, job availability
Organisational	Organisational 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

Step-Wise: Project Planning



Step-Wise: Project Planning



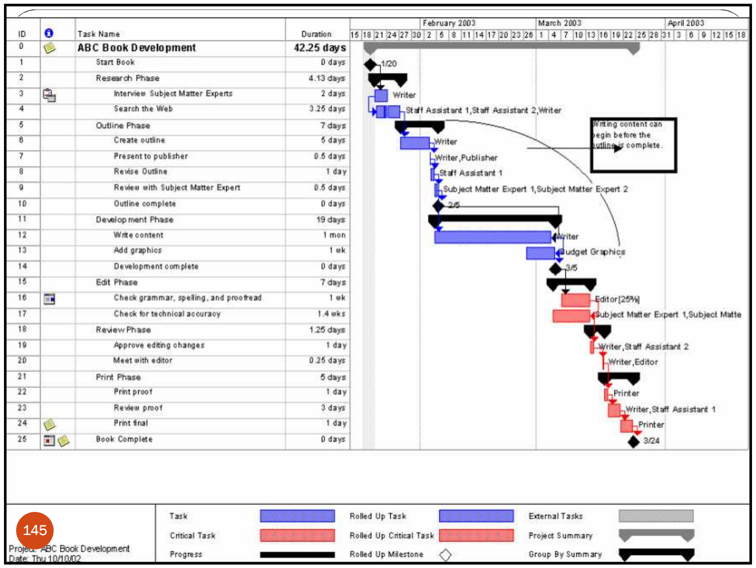
Step 7: Allocate Resources

- Identify and allocate resources (Scheduling)
- Revise plans and estimates to take into account resource constraints

Project Schedule

Gantt Chart (Bar Chart)

- Graphical notations used to illustrate the project schedule.
- Show project breakdown into tasks. Tasks should not be too small. They should take about a week or two.
- Bar charts show schedule against calendar time.

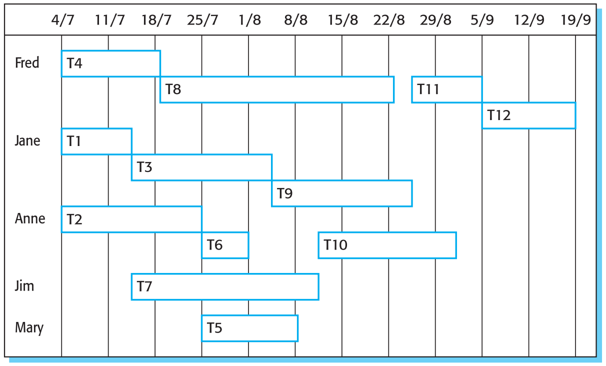


ID		Task Name	Duration
0		**ABC Book Development**	**42.25 days**
1		Start Book	0 days
2		Research Phase	4.13 days
3		Interview Subject Matter Experts	2 days
4		Search the Web	3.25 days
5		Outline Phase	7 days
6		Create outline	5 days
7		Present to publisher	0.5 days
8		Revise Outline	1 day
9		Review with Subject Matter Expert	0.5 days
10		Outline complete	0 days
11		Development Phase	19 days
12		Write content	1 mon
13		Add graphics	1 wk
14		Development complete	0 days
15		Edit Phase	7 days
16		Check grammar, spelling, and proofread	1 wk
17		Check for technical accuracy	1.4 wks
18		Review Phase	1.25 days
19		Approve editing changes	1 day
20		Meet with editor	0.25 days
21		Print Phase	5 days
22		Print proof	1 day
23		Review proof	3 days
24		Print final	1 day
25		Book Complete	0 days

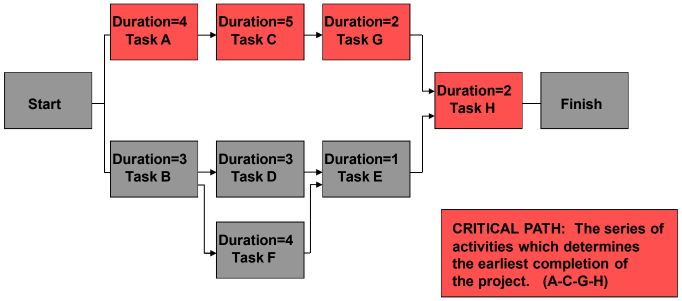
Project: ABC Book Development
Date: Thu 10/10/02

Task	🟦	Rolled Up Task	🟦	External Tasks	▨
Critical Task	🟥	Rolled Up Critical Task	🟥	Project Summary	▬
Progress	▬	Rolled Up Milestone	◇	Group By Summary	▬

Staff Allocation Chart

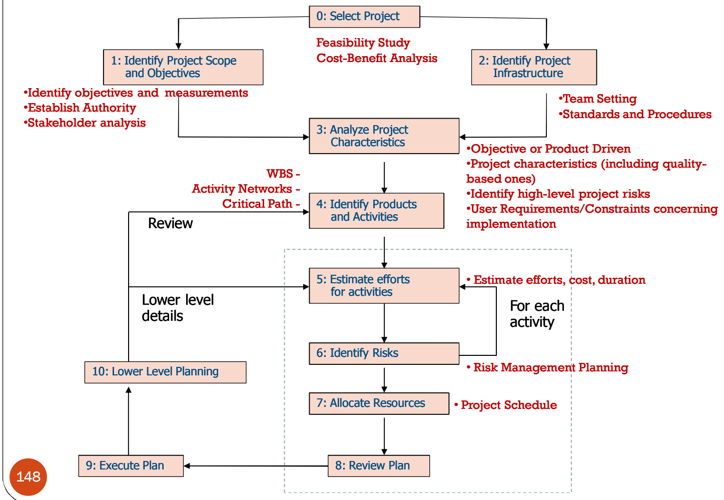


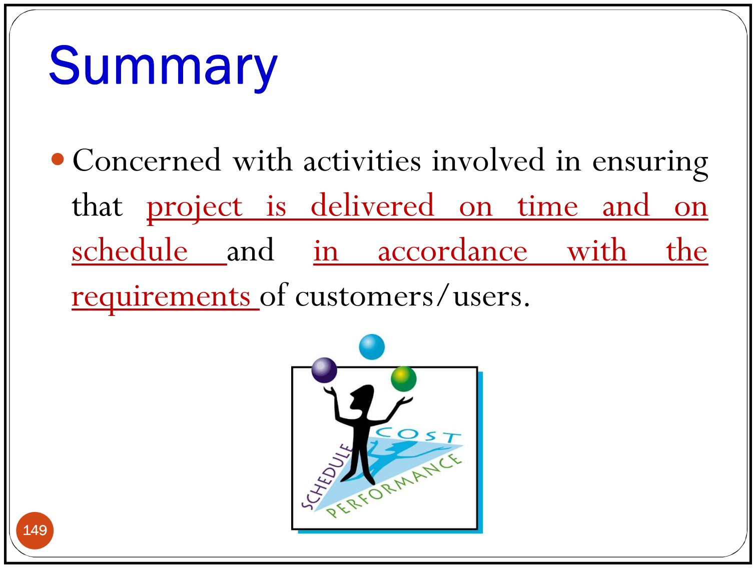
Critical Path



CRITICAL PATH: The series of activities which determines the earliest completion of the project. (A-C-G-H)

Step-Wise: Project Planning


0: Select Project
Feasibility Study
Cost-Benefit Analysis
1: Identify Project Scope and Objectives
•Identify objectives and measurements
•Establish Authority
•Stakeholder analysis
2: Identify Project Infrastructure
•Team Setting
•Standards and Procedures
3: Analyze Project Characteristics
•Objective or Product Driven
•Project characteristics (including quality-based ones)
•Identify high-level project risks
•User Requirements/Constraints concerning implementation
WBS -
Activity Networks -
Critical Path -
4: Identify Products and Activities
Review
5: Estimate efforts for activities
• Estimate efforts, cost, duration
Lower level details
For each activity
6: Identify Risks
• Risk Management Planning
10: Lower Level Planning
7: Allocate Resources
• Project Schedule
9: Execute Plan
8: Review Plan



Summary

- Concerned with activities involved in ensuring that project is delivered on time and on schedule and in accordance with the requirements of customers/users.

COST
SCHEDULE
PERFORMANCE

Questions