

# COSC 310: Software Engineering

Dr. Bowen Hui

University of British Columbia  
Okanagan

# Review: Choosing SDLC

- Which SDLC would you choose?
  1. An experienced team member who absolutely prefers to work alone, the project can be broken up into independent chunks

# Review: Choosing SDLC

- Which SDLC would you choose?
  1. An experienced team member who absolutely prefers to work alone, the project can be broken up into independent chunks
  2. A junior team member who needs more support in coding but there's no dedicated training time

# Review: Choosing SDLC

- Which SDLC would you choose?
  1. An experienced team member who absolutely prefers to work alone, the project can be broken up into independent chunks
  2. A junior team member who needs more support in coding but there's no dedicated training time
  3. Project that involves a lot of sensitive, personal data

# Review: Choosing SDLC

- Which SDLC would you choose?
  1. An experienced team member who absolutely prefers to work alone, the project can be broken up into independent chunks
  2. A junior team member who needs more support in coding but there's no dedicated training time
  3. Project that involves a lot of sensitive, personal data
- More in this week's lab

# Aspects of a Project

- **Scope:** If too many features to implement

# Aspects of a Project

- **Scope:** If too many features to implement
  - Team won't be able to meet expectations
- **Schedule:** If there's not enough time

# Aspects of a Project

- **Scope:** If too many features to implement
  - Team won't be able to meet expectations
- **Schedule:** If there's not enough time
  - Team won't be able to complete tasks
- **Cost:** If actual budget exceeds estimates



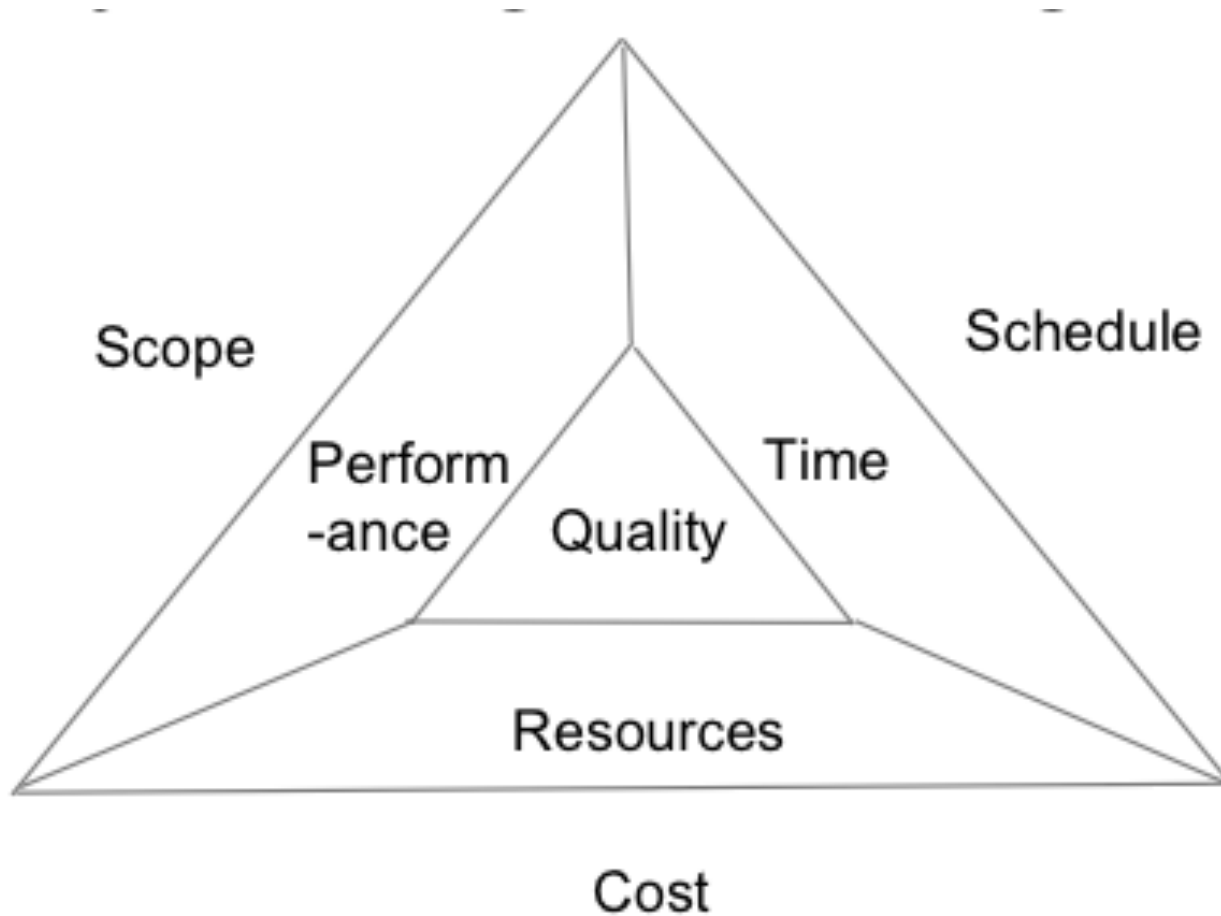
# Aspects of a Project

- **Scope:** If too many features to implement
  - Team won't be able to meet expectations
- **Schedule:** If there's not enough time
  - Team won't be able to complete tasks
- **Cost:** If actual budget exceeds estimates
  - Project cancellation
- **Quality:** If product is low quality

# Aspects of a Project

- **Scope:** If too many features to implement
  - Team won't be able to meet expectations
- **Schedule:** If there's not enough time
  - Team won't be able to complete tasks
- **Cost:** If actual budget exceeds estimates
  - Project cancellation
- **Quality:** If product is low quality
  - Users won't use the software

# Project Management Triangle



# Key to Project Success

- Good planning and organization
  - Selecting appropriate SDLC
  - Allocating suitable team members and resources
- Careful monitoring and control
  - Process
  - Product development
  - HR health
- What makes a good software project manager?

# Project Management Competencies

- **Product competencies**
  - E.g.: manage requirements, select methods
- **Project competencies**
  - E.g.: document plans, estimate costs/size
- **People competencies**
  - E.g.: leadership, communication, team building
- Factors that make software project successful are not especially technical!

# In detail...

<b>Software</b> (Product)	<b>Project</b> (Project)	<b>Management</b> (People)
1. Assess processes	12. Build WBS	23. Appraise performance
2. Awareness of proces standards	13. Document plans	24. Handle subcontractors
3. Define product	14. Estimate cost	25. Hold effective meetings
4. Evaluate alternative processes	15. Estimate effort	26. Interaction and communication
5. Manage requirements	16. Manage risks	27. Leadership
6. Handle IP	17. Monitor dev.	28. Manage change
7. Perform initial assessment	18. Scheduling	29. Negotiate successfully
8. Select methods/tools	19. Select metrics	30. Plan careers
9. Tailor processes	20. Select PM tools	31. Present effectively
10. Track product quality	21. Track process	32. Recruiting
11. Understand dev. activities	22. Track project progress	33. Select teams
		34. Team building

# Why 3 Areas of Competencies?

- Imagine, what happens if a PM has no...
  - Product competency?
  - Project competency?
  - People competency?

# PM Main Responsibilities

1. Project planning
2. Project sizing
3. Estimating effort needed
4. Scheduling and allocating resources
5. Monitoring progress
6. Quality control



# 1. Project Planning

- Formal project creation
- Work product: creation of project documents
  - Industry: **charter, statement of work**
    - **Why** project is needed
    - **What** the project is
  - Academia: **project proposals**
    - **Why** project is needed
    - **How** project will be carried out
- **Plan = WBS + staffing + cost + schedule**

## 2. Project Sizing

- Prediction of amount of code needed to fulfill requirements
- Estimate **size** of software in terms of
  - Lines of code
  - Function points or feature points
  - Number of processes in data flow diagram (DFD)
  - Number of objects, attributes, services in object-oriented diagram (usually UML notation)
- Used as input to other planning steps

# 3. Project Estimation

- Prediction of resources needed to complete project based on predicted size (#2)
- Estimate **cost = work effort**
  - Cost is always in terms of effort, not \$
- Considers
  - Calendar time
  - Staff availability, capability, productivity
  - Budget constraints
  - Creeping requirements
  - **What-if analysis**

# 4. Project Scheduling

- Reflects relationships among all the resources and activities in project
- Ways to represent schedule:
  - Simple table
  - Gantt chart
  - Pert diagram
- Main purposes:
  - Illustrates interdependencies on calendar
  - Indicates dates for major milestones
  - Compares estimates vs. actuals

# 5. Project Monitoring

- Manage scope, schedule, cost, quality
- **Metrics** = measurement of an observable unit
  - Establish baselines
  - Understand what happens during development
  - Enables control
    - Monitor progress and variance
    - Establish goals
    - Predict future performance
  - Enables current/future improvement
    - Increase productivity and/or quality
- Progress management (**earned value management**)
- May include risk management

# 6. Project Quality Control

- Defects should be highlighted and brought forth for corrective action
- Improve quality, save money, increase business value
- **Quality assurance** activity
  - Auditing, external reviews, and reporting
  - Provide management with data on product quality
  - Focus on correction and prevention
  - Reviews all work products and documented activities

# PM's Key Role

- Organize and lead team meetings
- When should meetings be held?

# PM's Key Role

- Organize and lead team meetings
- Common meeting objectives:
  - Information exchange
  - Problem solving
  - Decision making
  - Planning
  - Evaluation



# Meeting Components

- Roles:
  - Facilitator (directs meeting)
  - Scribe (takes notes and shares with team)
  - General participants
- Input:
  - **Agenda** (list of items to review)
- Output:
  - **Minutes** (documentation)

# Meeting Components

- Roles:
  - Facilitator, scribe, participants
- Input:
  - **Agenda** (list of items to review)
  - Specify:
    - Participants
    - Schedule
- Output:
  - **Minutes** (documentation)

# Meeting Components

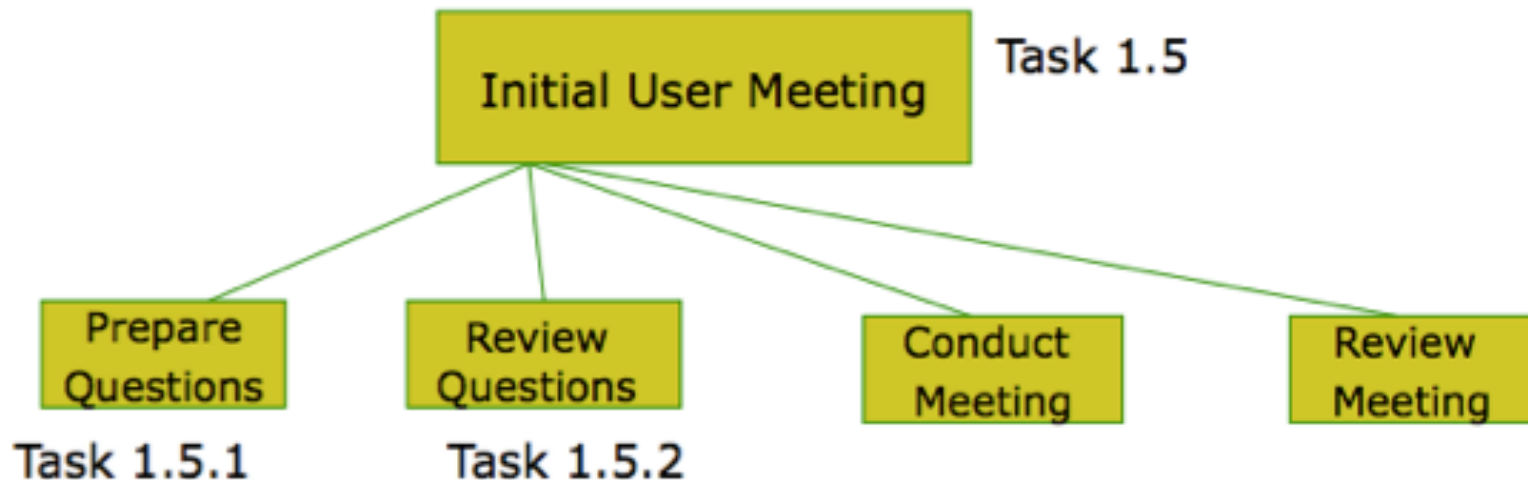
- Roles:
  - Facilitator, scribe, participants
- Input:
  - **Agenda** (list of items to review)
- Output:
  - **Minutes** (documentation)
    - Specify:
      - **Outcomes**: summary of main points per item
      - **Decisions**: what and why was decided
      - **Action items**: who will do what by when

# Creating a Project Plan

- Plan = WBS + staffing + cost + schedule
  - Decompose project into WBS
  - Assign hardware and human resources
  - Estimate schedule required
  - Budget estimation (not for this course)
- Reviewed and approved by **everyone** involved
- Serves as benchmark
  - Actuals serve as comparison

# Work Breakdown Structure (WBS)

- **WBS** = heart of a project plan
- Resembles hierarchy diagram
- Typically organized around deliverables and work products
- E.g.:



# Work Product vs. Deliverable

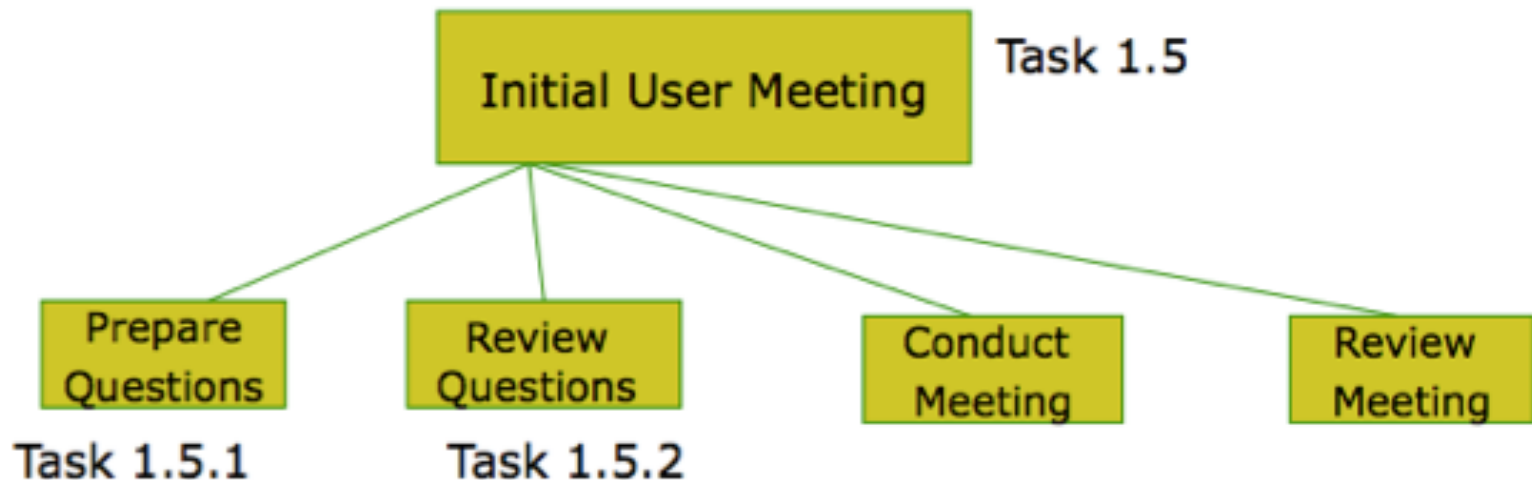
- **Work product**
  - Anything tangible produced as part of project
  - E.g., project plan, meeting minutes, test logs, etc.
  - Can be internal to team
- **Deliverable**
  - Work products that are viewed by client or another project team
  - E.g., source code, technical manual, user manual, etc.
  - External to immediate team

# Types of WBS

- **Product view**
  - Depicts hierarchical relationship between software modules
- **Process view**
  - Depicts hierarchical relationship between work activities
- **Why both?**

# Example

- Is this partial WBS product view or process view?





# Top-level WBS for A2/A3/Project

<b>Phase No</b>	<b>Phase Name</b>
1.0	Initiation
2.0	System Study
3.0	Detailed Design
4.0	Development
5.0	Implementation
6.0	Project Completion
	Total

# Ex WBS for Phase 1: Initiation

- Subtasks with subsubtasks:
  - Initial user meeting
  - Project proposal
  - Project plan
  - Repositories
- Other subtasks:
  - 1.1-1.4
  - 1.9: Contingency tasks

Task No	Task Title
1.1	Plan organize, and Control
1.2	Team meetings
1.3	Review Team Roles
1.4	Review Repositories
1.5	Initial User meeting
1.5.1	Prepare Questions
1.5.2	Prepare for User Meeting
1.5.3	Conduct User Meeting
1.5.4	Review User Meeting
1.6	Project Proposal
1.6.1	Draft Proposal
1.6.2	Review Proposal With Director
1.6.3	Revise Proposal
1.6.4	Deliver Proposal to User
1.7	Project Plan
1.7.1	Draft Project Plan
1.7.2	Review Plan With Director
1.7.3	Revise Plan
1.8	Repositories
1.8.1	Prepare Binder Filing material with IS Director
1.9	Contingency Tasks

# Ex: Plan for Phase 1

Task No	Task Title	Estimated Hours	Actual Hours	Estimated Start Date	Estimated Complete Date	Actual Start Date	Actual Complete Date	Assigned To	Performed By
1.1	Plan organize, and Control	20.00		10-Sep-99	8-Oct-99			CJ,MM	
1.2	Team meetings	20.00		10-Sep-99	8-Oct-99			TEAM	
1.3	Review Team Roles	2.00		14-Sep-99	19-Sep-99			TEAM	
1.4	Review Repositories	18.00							
1.5	Initial User meeting			21-Sep-99	25-Sep-99			TEAM	
1.5.1	Prepare Questions	10.00		21-Sep-99	22-Sep-99			TEAM	
1.5.2	Prepare for User Meeting	2.00		21-Sep-99	22-Sep-99			TEAM	
1.5.3	Conduct User Meeting	3.00		23-Sep-99	23-Sep-99			TEAM	
1.5.4	Review User Meeting	15.00		24-Sep-99	25-Sep-99			TEAM	
1.6	Project Proposal								
1.6.1	Draft Proposal	20.00		24-Sep-99	27-Sep-99			CJ,MM	
1.6.2	Review Proposal With Director	2.00		28-Sep-99	28-Sep-99			CJ	
1.6.3	Revise Proposal	6.00		29-Sep-99	29-Sep-99			CJ,MM	
1.6.4	Deliver Proposal to User	2.00		30-Sept-99	30-Sept-99			CJ,MM,FD	
1.7	Project Plan								
1.7.1	Draft Project Plan	15.00		28-Sep-99	1- Oct-99			CJ,MM	
1.7.2	Review Plan With Director	2.00		3-Oct-99	3-Oct-99			CJ	
1.7.3	Revise Plan	2.00		6-Oct-9	8-Oct-99			CJ,MM,FD	
1.8	Repositories								
1.8.1	Prepare Binder Filing material with IS Director	1.00		8-Oct-99	8-Oct-99			FD	
1.9	Contingency Tasks	10.00						TEAM	
	Total	150.00							

# Ex: Plan for Phase 1

- Headings used on previous slide
  - Task number, task title
  - Estimate hours and actual hours
  - Estimated start and end dates
  - Actual start and end dates
  - Assigned to team/sub-team
    - Could have individuals
    - Indicate lead vs. assistant/support in that task

# Ex: WBS for Phase 4.3

- Phase 4: Development
- Product view WBS

<b>4.3</b>	<b>Code Application (Ext. from working prototype)</b>
4.3.1	Code front-end screens
4.3.2	Integrate with database
4.3.3	Code queries
4.3.4	Code Statistical component
4.3.5	Code Reports
4.3.6	Develop and Code Input data conversion tool

# Creating a WBS

- Organize work according to chosen SDLC
- Arrange activities around major work products, deliverables, **milestones**
- Keep plan fairly high level (why?)
  - Decompose only to 1 or 2 levels
- Use teams or sub-teams for chunks of work
  - E.g.: analysis team, documentation team, test team, programming team

# What are Milestones

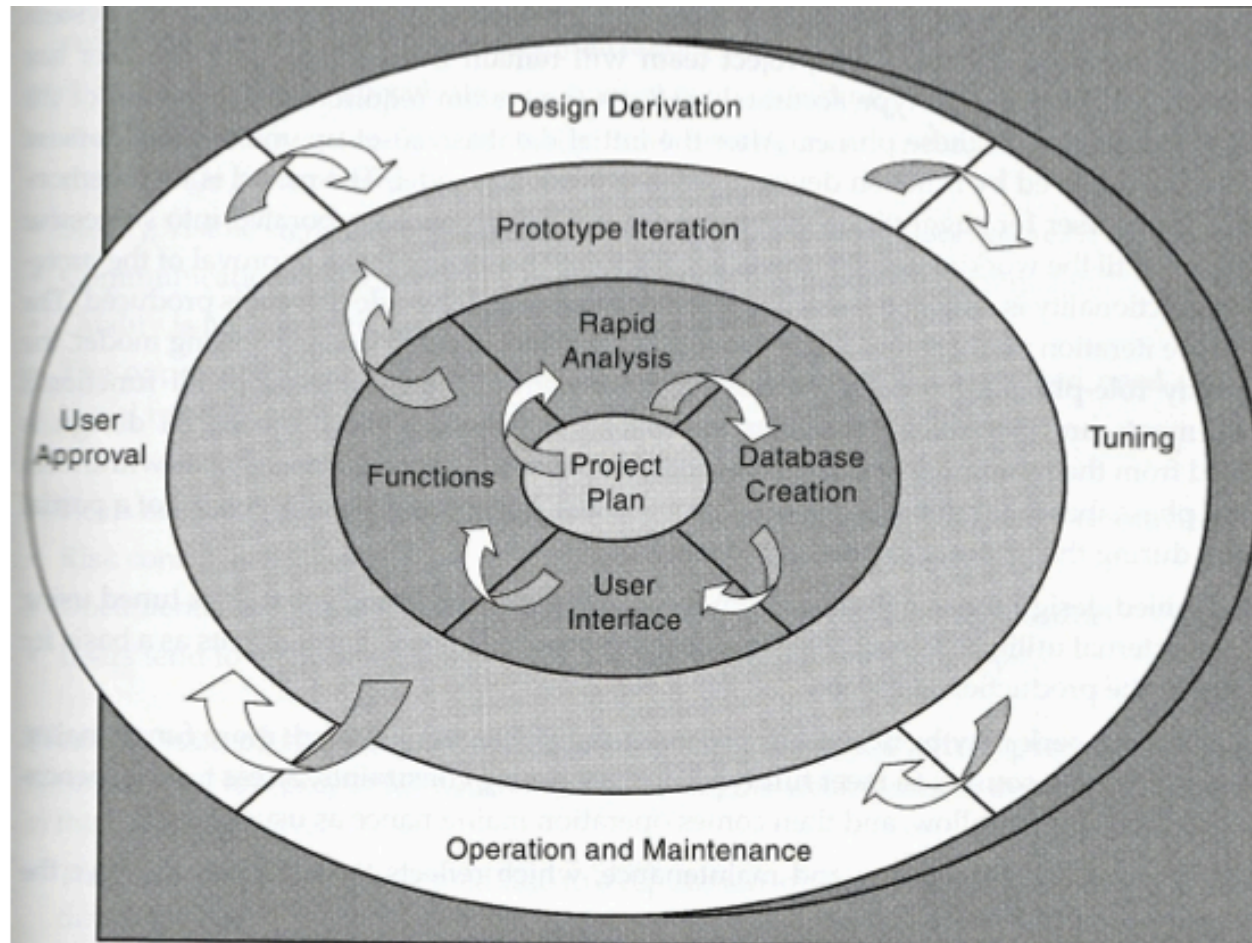
- **Milestone** = significant event in project
  - Usually associated with completion of deliverable
  - Ex. customer demo date
- Why do milestones typically have zero duration?
- Each project should have some milestones spread out throughout schedule

# Remaining Steps in Creating Plan

- After WBS, add:
  - Staffing requirements (names/positions)
  - Schedule (estimated start/end dates)
  - Cost (budget)
- Should you come up with ...
  - Schedule before staffing is assigned?
  - Budget before staffing is assigned?
- After project starts, do progress monitoring:
  - Update Actual hours
  - Update Actual Start/end dates



# Example WBS for Rapid Prototyping



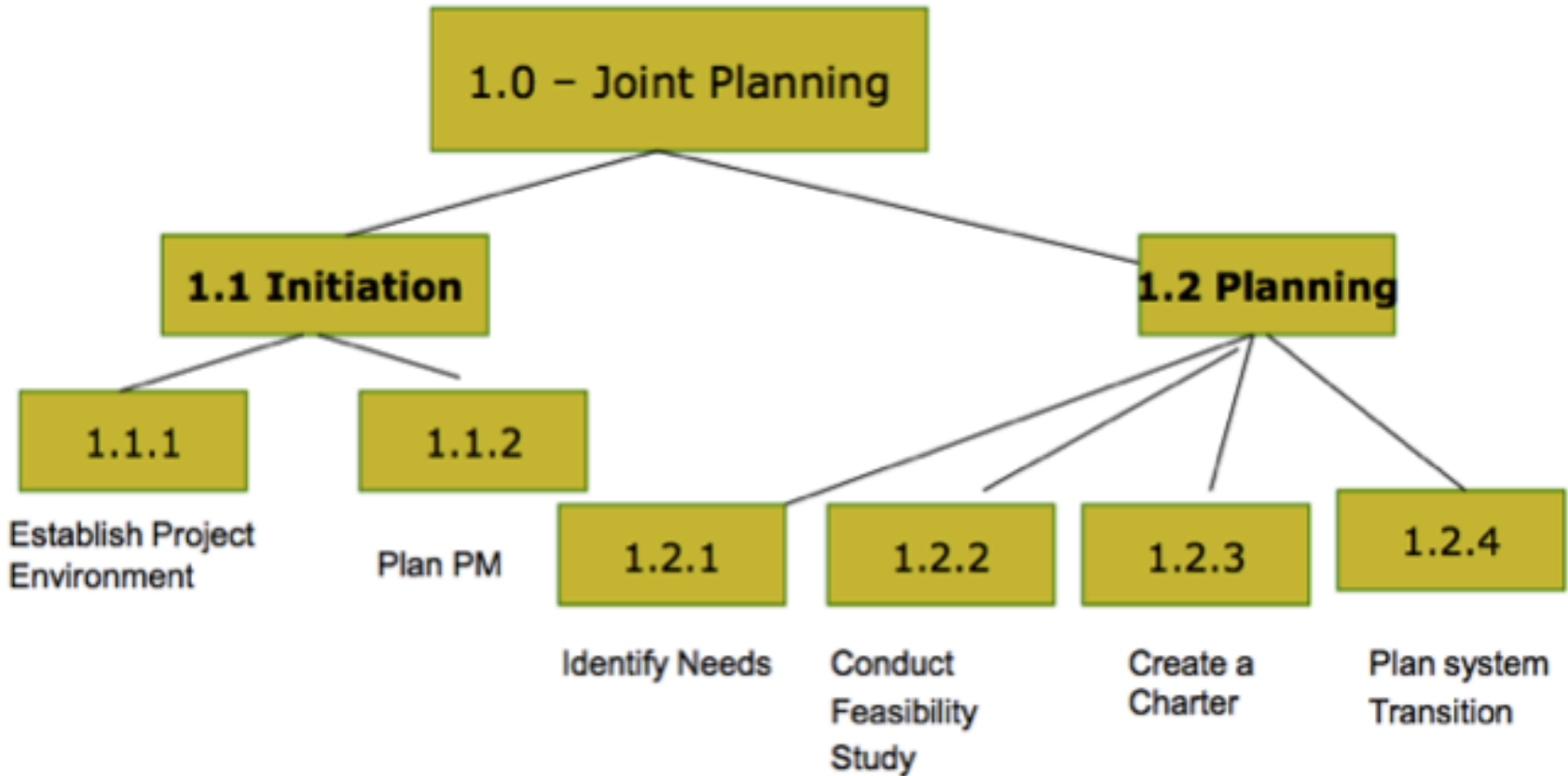
# Decide Main Phases

1. Joint Project Planning
2. Rapid Analysis
3. Database Creation
4. Design User Interface
5. Design Functions
6. Create Partial Requirements Document
7. Create Prototype
8. Evaluate System
9. Implement System

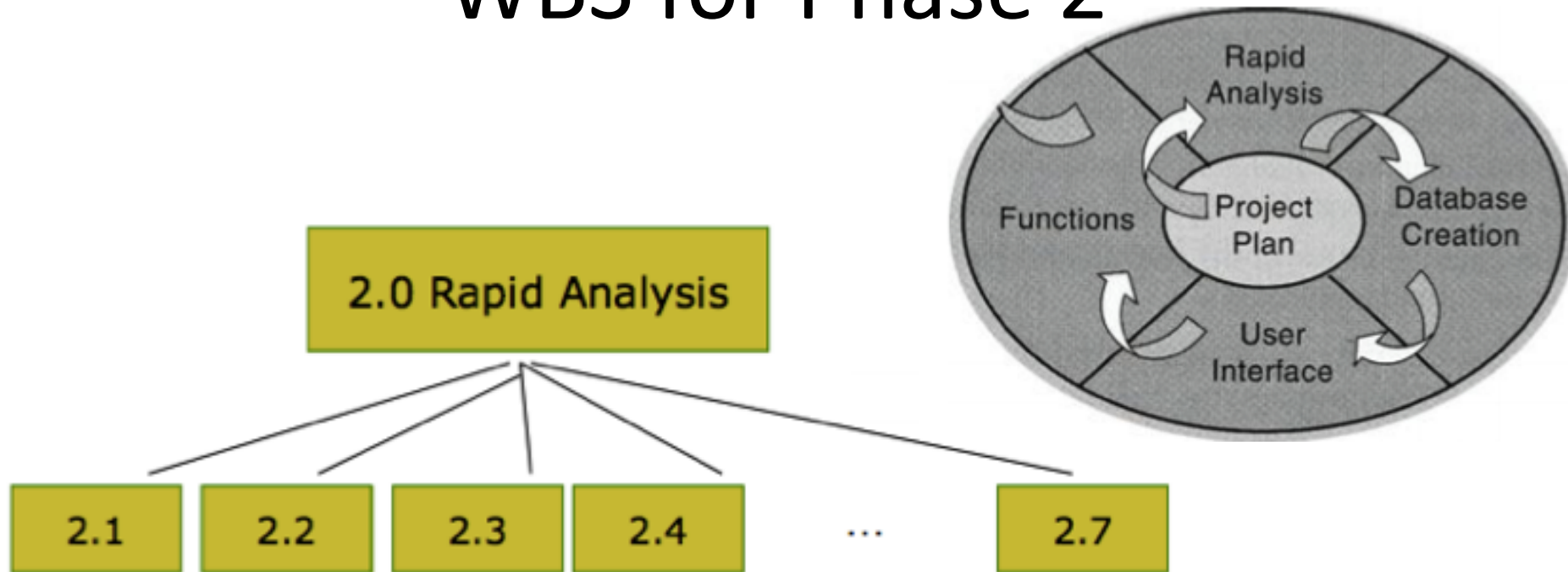
# Phase 1: Joint Project Planning

- 1.1 Initiation
  - 1.1.1 Establish project environment
  - 1.1.2 Plan project management
- 1.2 Concept Exploration
  - 1.2.1 Identify needs
  - 1.2.2 Conduct feasibility analysis
  - 1.2.3 Create project charter/proposal
  - 1.2.4 Plan system transition

# WBS for Phase 1



# WBS for Phase 2



**2.1 Analyze Business Processes**

**2.2 Analyze functions**

**2.3 Create a preliminary software architecture**

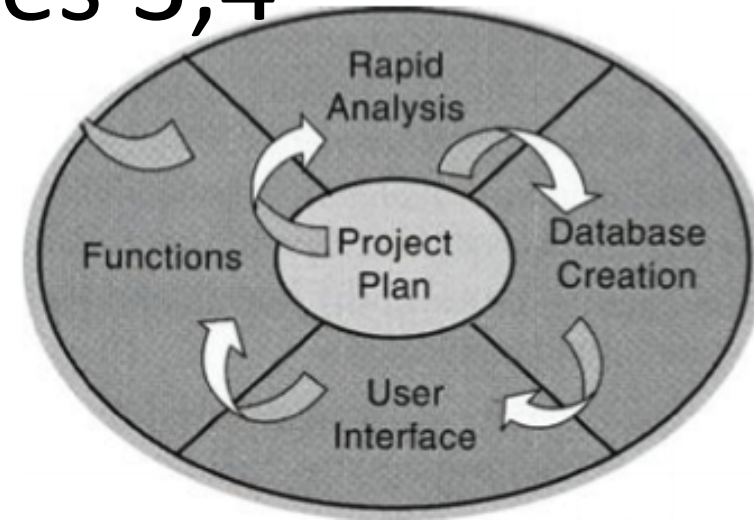
**2.4 Define and develop preliminary SW requirements**

**2.5 Define preliminary interface requirements**

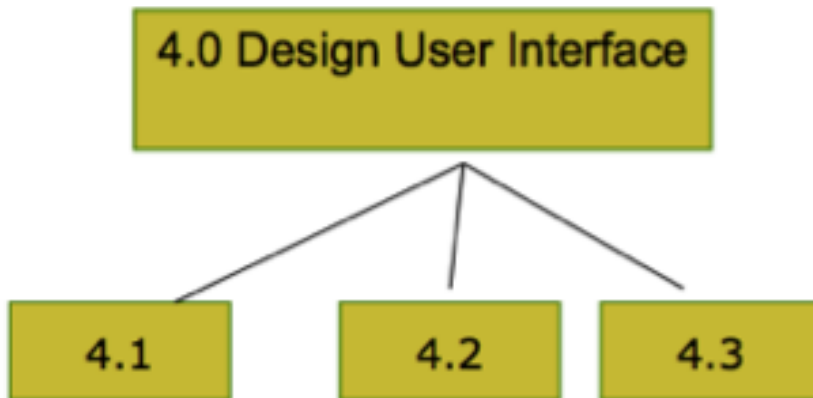
**2.6 Prioritize the requirements**

**2.7 Choose a prototyping tool**

# WBS for Phases 3,4



- 3.1 Create Preliminary ERD
- 3.2 Design Preliminary Database Model



- 4.1 Design system navigation and layout
- 4.2 Review Screen Navigation Diagram Technique
- 4.3 Design Preliminary User Interface (on paper)



# WBS for Phases 5,6,7

## 5 Design Functions

5.1 Design Algorithms using TOE

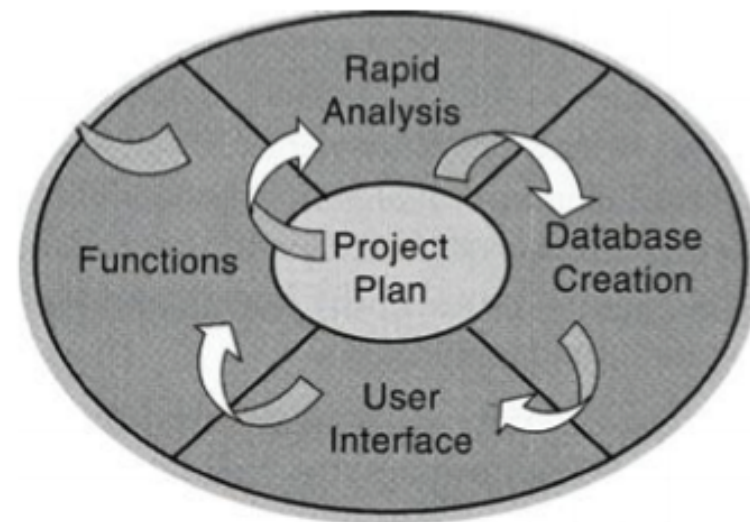
5.2 Refine Algorithms using Flowcharts

## 6 Create partial requirements document

6.1 Draft SRS

6.2 Review SRS

## 7 Build Prototype using the document created in Phase 6



# WBS for Phases 8,9

## 8 Evaluate the System

- 8.1 Prepare user feedback method
- 8.2 Conduct feedback session
- 8.3 Document evaluation results

## 9 Implement the System

- 9.1 Incorporate user feedback into the design
- 9.2 Create test data and test plan
- 9.3 Test the system
- 9.4 Plan Installation
- 9.5 Install the system

