



Software Development Methodologies

Lecturer : Adel Vahdati



SDLC and Methodologies

- The SDLC provides the foundation for the processes used to develop an information system.
- A methodology is a formalized approach to implementing the SDLC
 - tasks, steps, deliverables
- Various methodologies differ in progression through SDLC phases
- Project characteristics affecting methodology selection



Project Characteristics

- Clarity of User Requirements
 - Understanding of needed functions and capabilities
- Familiarity with Technology
 - Experience with the technology to be used
- System Complexity
 - Anticipated complexity, feature array, integration needs, organizational span
- System Reliability
 - Need for high reliability vs. tolerable downtime
- Short Time Schedules
 - Tight project time frames
- Schedule Visibility
 - Anxiety of sponsors, users, managers to see progress

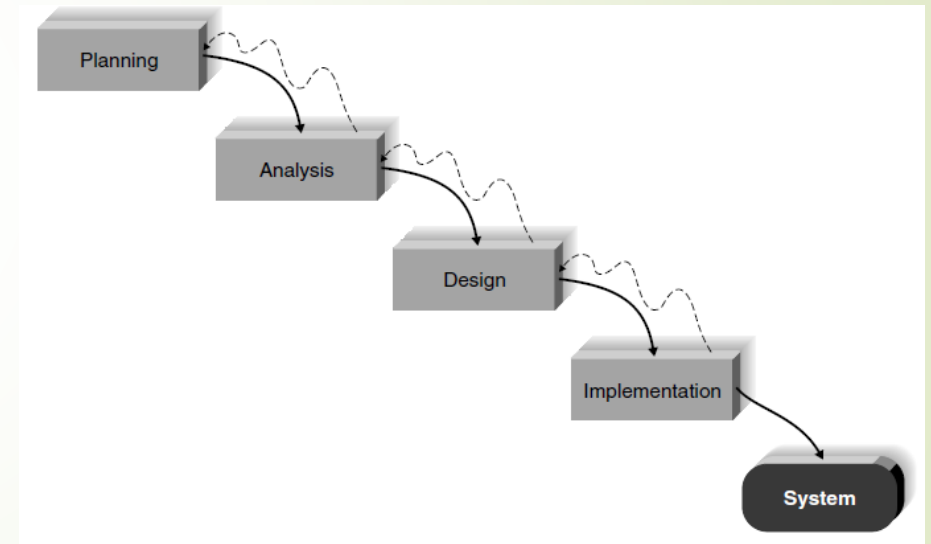
Waterfall Development

► Process

- Sequential progression from one phase to the next
- Key deliverables for each phase are voluminous
- Approval required to move from phase to phase
- Difficult to go backward through phases

► Advantages

- Early identification of requirements
- Limited requirement changes as project progresses





Waterfall Development (2)

➤ **Disadvantages**

- Complete design specification needed before programming
- Significant time between analysis completion and system delivery
- Testing often treated as an afterthought
- Poor communication through voluminous deliverables
- Potential for expensive post-implementation programming

➤ **Challenges in Dynamic Environments**

- Systems may need rework to match changing conditions
- Rework requires revisiting initial phases

Parallel Development

► Process

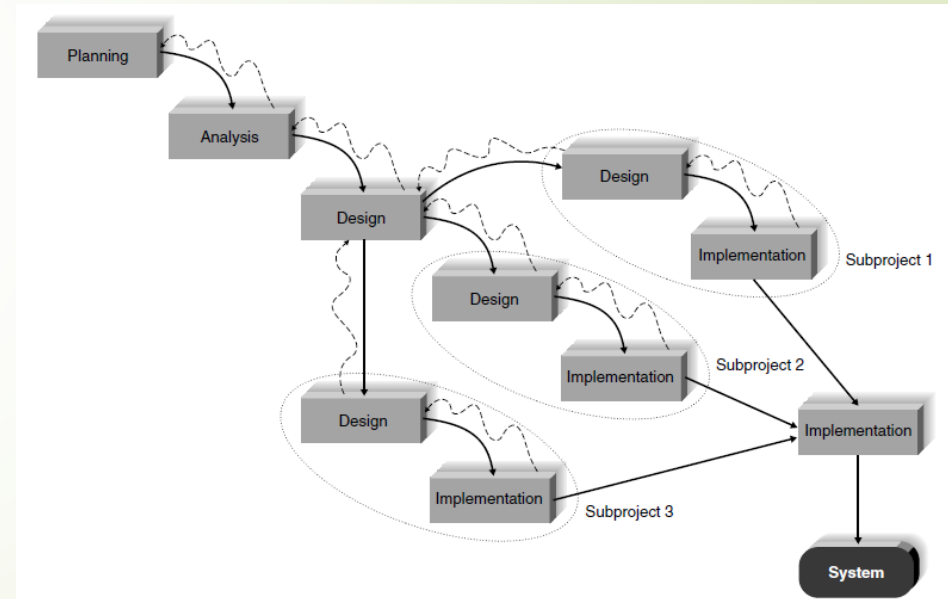
- Variant of Waterfall
- General design created after analysis phase
- Project divided into subprojects for parallel design and implementation
- Final integration of subprojects before system delivery

► Advantages

- Reduced time to deliver system
- Less rework due to business environment changes

► Disadvantages

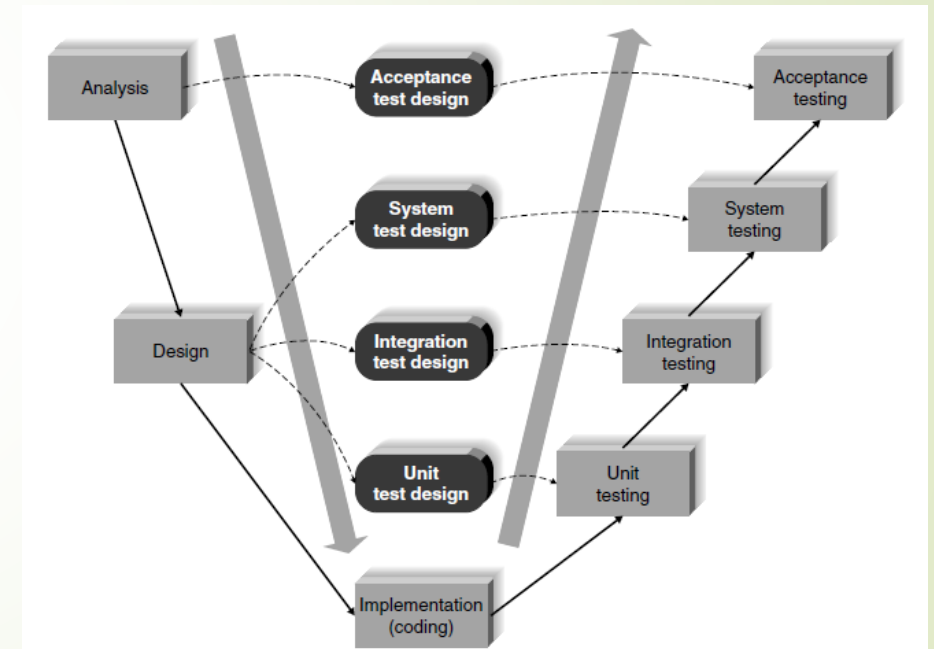
- Voluminous deliverables
- Integration challenges if subprojects are not independent



V-Model Development

➤ Process

- Left-hand slope: Define requirements and design system components
- Base of V: Code is written
- Right-hand slope: Component testing, integration testing, acceptance testing
- Testing defined alongside requirements and design
- Ensures high-quality, relevant testing





Rapid Application Development (RAD)

► Overview

- Emerged to address weaknesses of waterfall development
- Uses special techniques and tools to speed up development phases
- Involves CASE tools, JAD sessions, 4th generation/visual programming languages, code generators
- **CASE (Computer-Aided Software Engineering) tools** are software applications designed to support various stages of the software development life cycle (SDLC)
- A **JAD (Joint Application Development)** session is a structured workshop that brings together stakeholders from different disciplines to collaboratively define business requirements and brainstorm technical solutions for a project.

► Advantages

- Speeds up system development
- Provides early user feedback and evaluation

► Challenges

- Managing user expectations
- Potential for expanding system requirements

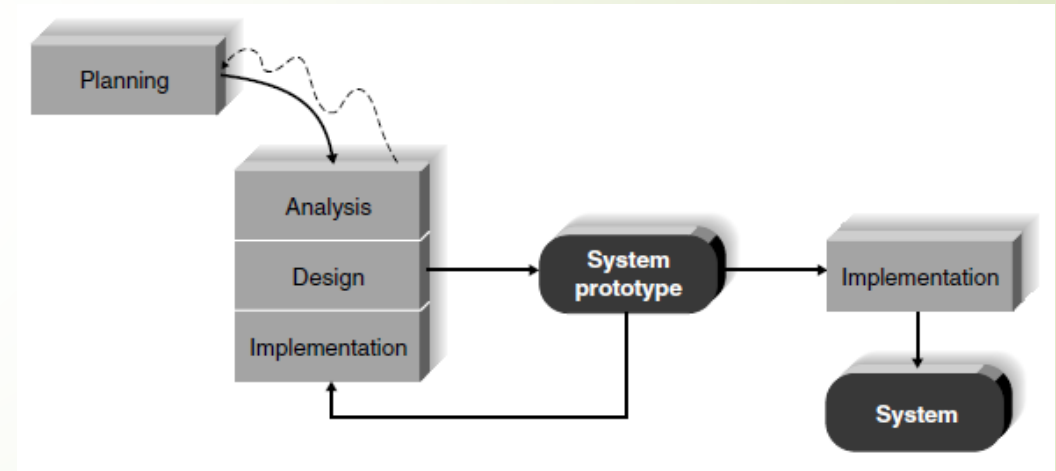


RAD Approaches - Iterative Development

- Breaks project into sequential versions
- First version includes fundamental requirements
- Users provide feedback for subsequent versions
- Provides early business value
- **Disadvantage:**
 - Users work with incomplete system initially

RAD Approaches - System Prototyping

- Concurrent analysis, design, and implementation
- Develops simplified version for user evaluation
- Iterative cycle of feedback and improvement
- Useful for users with difficulty expressing requirements
- **Disadvantage:**
 - Potential design limitations due to inadequate early analysis



RAD Approaches - Throwaway Prototyping

► Purpose

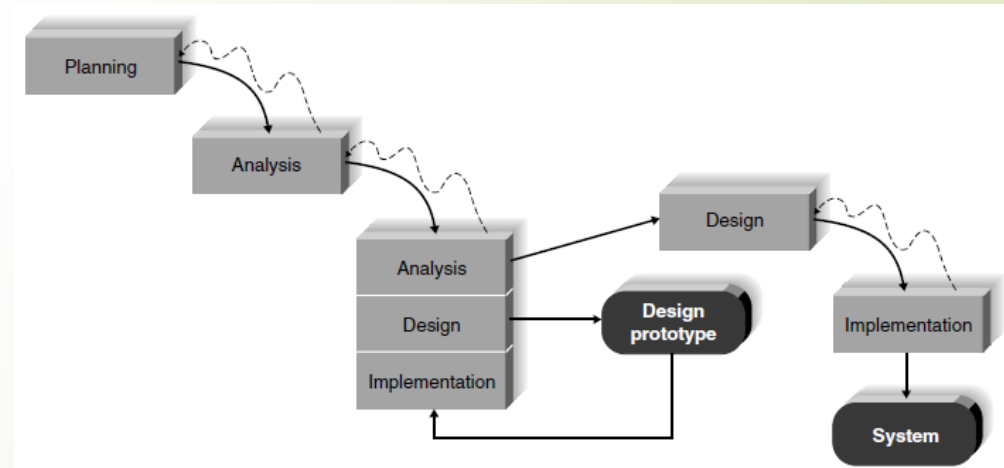
- Develops prototypes to explore design alternatives
- Prototypes are not the actual new system

► Process

- Thorough analysis phase to gather requirements and develop system concepts
- Design prototypes created to understand user-suggested features and solve technical issues
- Design prototypes are not working systems, just detailed enough for user understanding

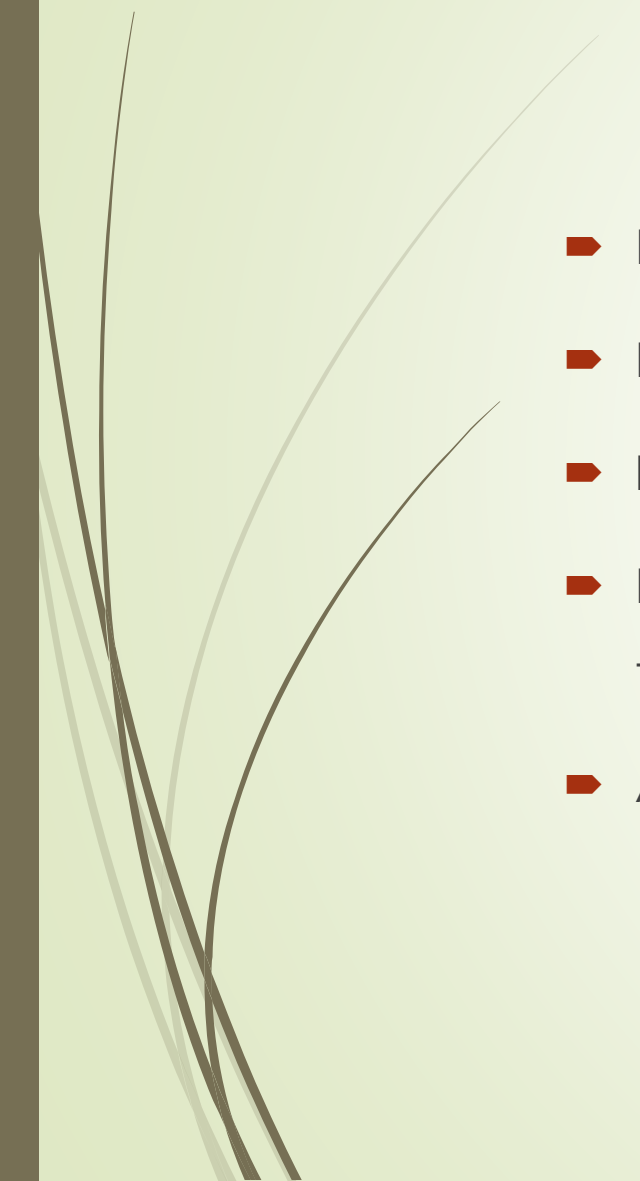
► Usage

- Multiple design prototypes used during analysis and design phases
- Minimizes risk by confirming important issues before building the real system
- Once issues are resolved, move to design and implementation
- Design prototypes are discarded after use





Agile Development: Overview

- Reduces modeling and documentation overhead
 - Prefers face-to-face communication
 - Iterative application development (1–4 week cycles)
 - Each iteration includes planning, requirements analysis, design, coding, testing, and documentation
 - Adapts to current business environment
- 



Agile Methodologies

- Extreme Programming (XP)
- Scrum
- Disciplined Agile Delivery (DAD)
- Dynamic Systems Development Method (DSDM)



Extreme Programming (XP)

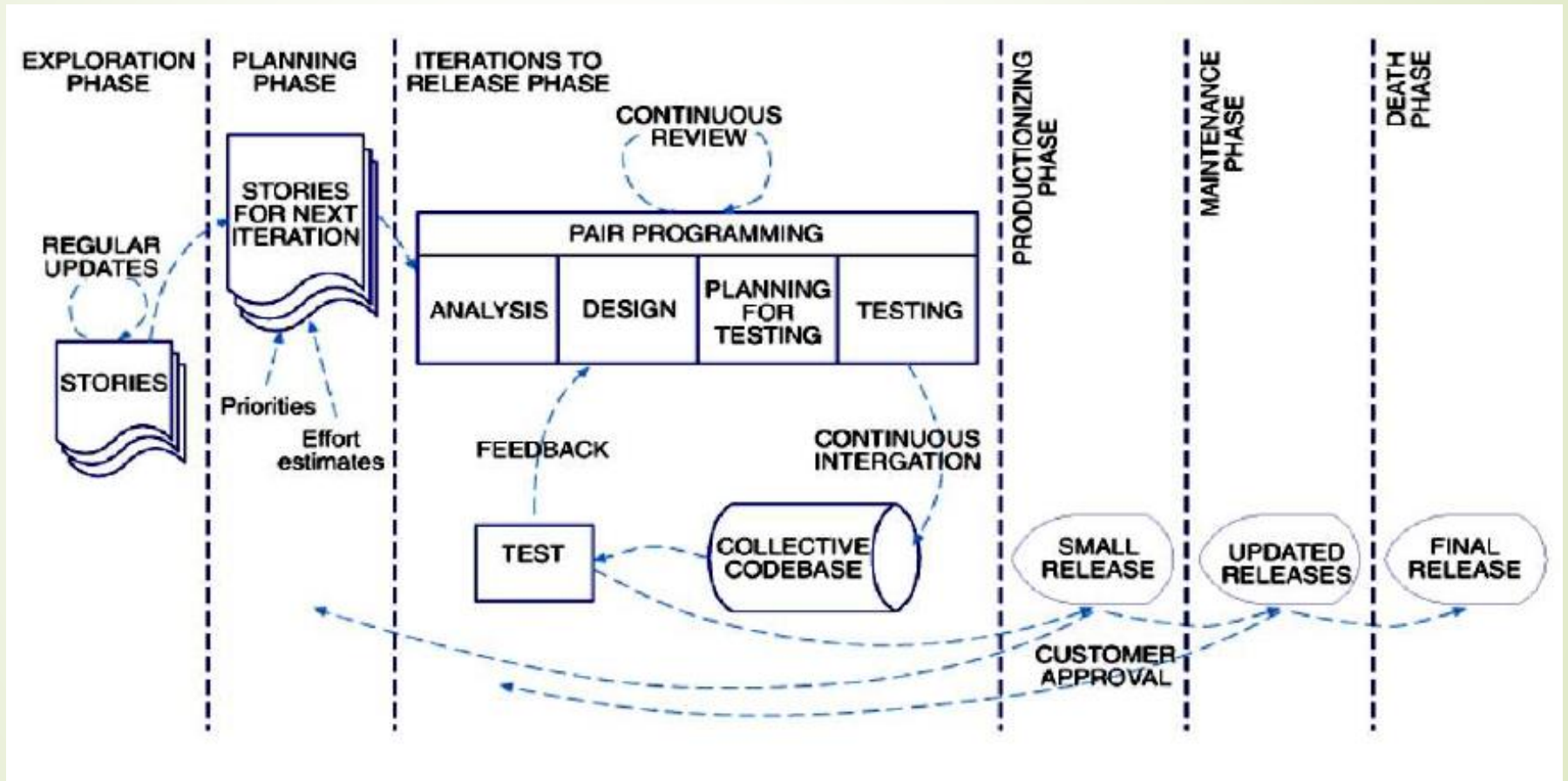
► Core Values

- Customer satisfaction
- Teamwork
- Communication
- Simplicity
- Feedback
- Courage

► Process

- Begins with user stories
- Small, simple modules coded and tested
- Users available for clarifications
- Common standards for names, descriptions, coding practices

XP Process





Extreme Programming (XP)

➤ **Advantages**

- Early and frequent testing
- Delivers results sooner than RAD
- Suitable for motivated, cohesive, stable, experienced teams

➤ **Challenges**

- Requires discipline to avoid chaos
- Recommended for small teams (≤ 10 developers)
- Not advised for mission-critical applications
- Limited analysis and design documentation
- Maintenance of large systems may be difficult
- Doubtful utility for long-term business information systems



References

- Systems Analysis and Design, 8th Edition by Alan Dennis, Barbara Haley Wixom, and Roberta M. Roth. Published by John Wiley & Sons, 2021
- Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J., Agile Software Development Methods: Review and Analysis. VTT Publications, 2002.