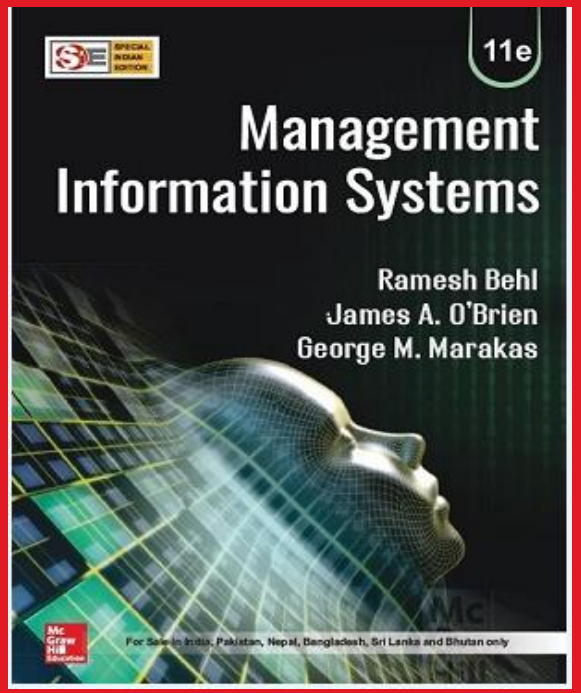
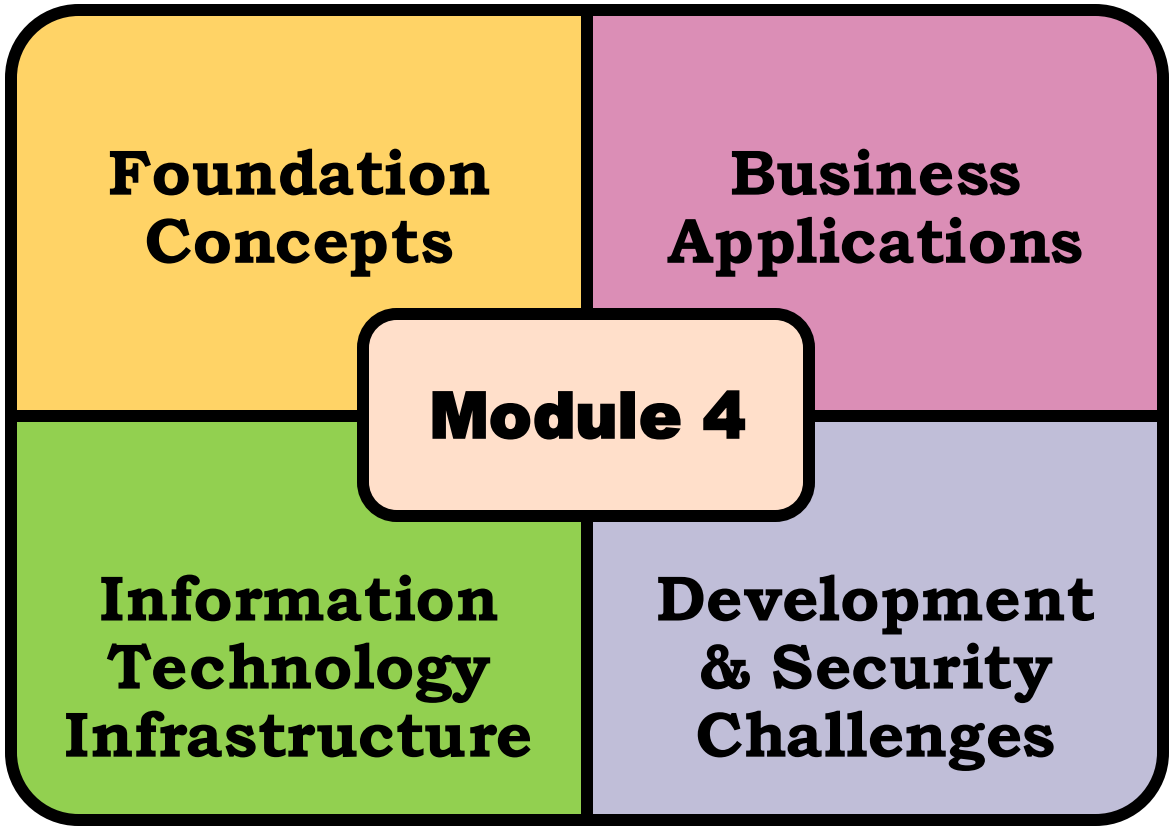
Because learning changes everything.®

Management Information Systems

Eleventh Edition



Chapter 10 : Planning and Developing Business/IT Solutions


Foundation Concepts
Business Applications
Module 4
Information Technology Infrastructure
Development & Security Challenges

Learning Objectives

Understand the role of planning in the business use of information technology, using the scenario approach and planning for competitive advantage as examples.

Define the role of planning and business models in the development of business/IT strategies, architectures and applications.

Outline the steps involved in information systems development life cycle to develop and implement a business information system.

Explain different models of systems development including prototyping.

Learning Objectives

Discuss the concept of design thinking and its importance to a successful system development effort.

Understand the project management concepts and its role in system development.

Compare and contrast the four basic system conversion strategies.

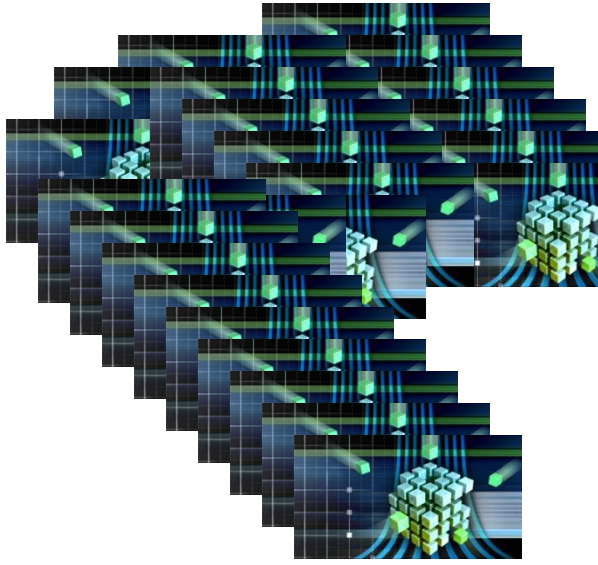
Describe several evaluation factors that should be considered in evaluating the acquisition of hardware, software, and IS services.

Identify several change management solutions for end-user.

IS Development

Information Systems Development

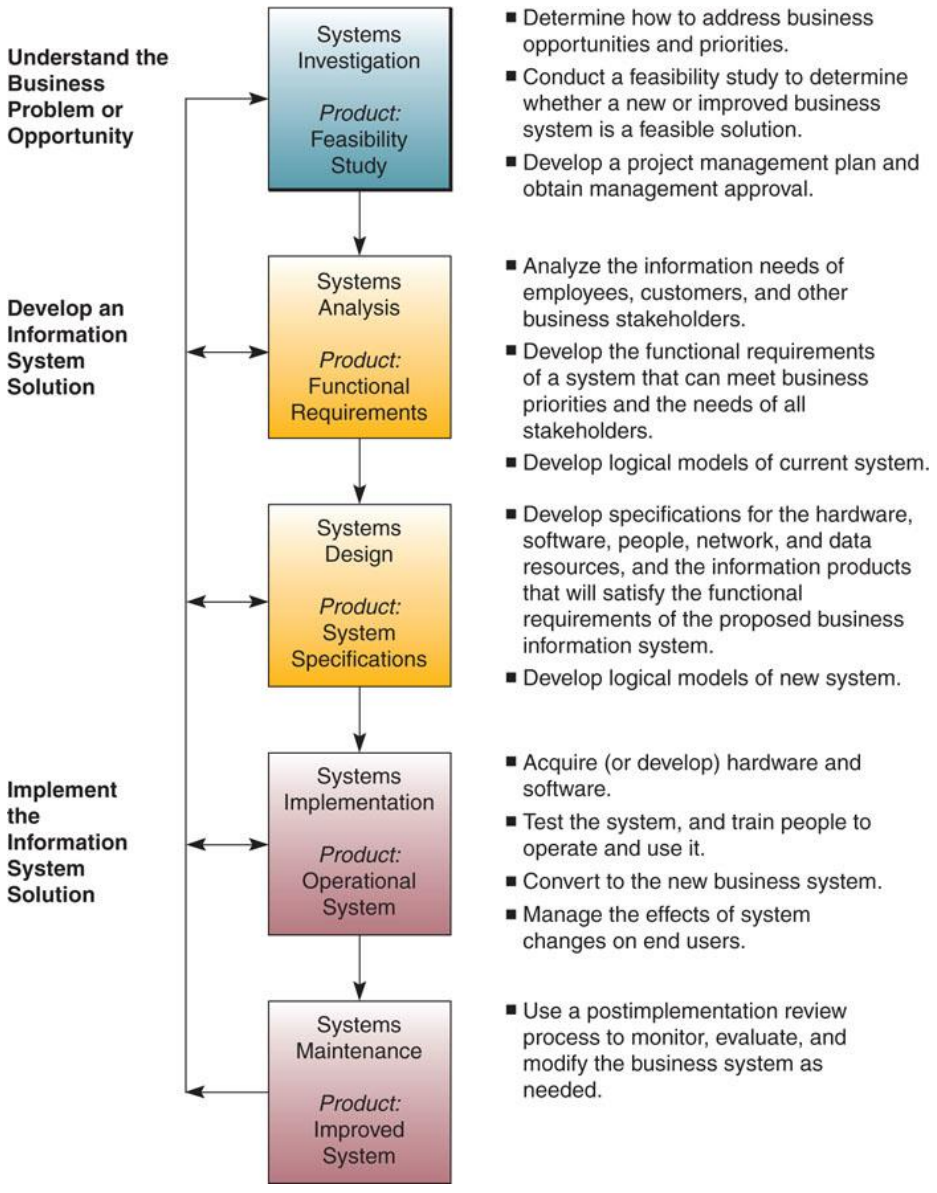
- Applying the Systems Approach to IS Development
- Also called Application Development



The Systems Approach

- Recognize and Define:
  - Identify the problem or opportunity.
  - Apply systems thinking for a holistic understanding.
  - Clearly define the problem's scope and impact.
- Develop and Evaluate Alternatives:
  - Brainstorm various solutions.
  - Evaluate each alternative's pros and cons.
  - Use decision matrices for objective assessment.
- Select the Best Solution:
  - Establish criteria for selection.
  - Consider stakeholder preferences.
  - Follow a systematic decision-making process.
- Design the Solution:
  - Develop a detailed design for the chosen solution.
  - Outline components, architecture, and integrations.
  - Ensure alignment with requirements.
- Implement and Evaluate Success:
  - Execute the implementation plan step by step.
  - Allocate resources and adhere to timelines.
  - Continuously monitor and evaluate success against predefined criteria.

Systems Development Lifecycle (SDLC)



Systems Development Process

Systems Investigation

- First step
- Consider multiple proposals
- Preliminary feasibility study
  - Information needs of prospective users
  - Resource requirements
  - Costs
  - Benefits
  - Legal environment
- Feasibility study may be unnecessary
  - Government mandate

Feasibility Study

goal:

- To evaluate the alternative system solutions and

- To propose the most feasible and desireable business application for development.

Types of Feasibility Studies

Operational

- Focuses on the degree to which
  - the proposed development project fits in the existing business environment and objectives regards to development schedule, corporate culture, and existing business process
  - the project meet the specific business objectives set during the proposal phase

Types of feasibility Study

Economical

- determines the extent to which the proposed system will provide positive economic benifits to the organization.
- involves
  - identification of all expected benifits from the system

    tangible benefit(decrese in payroll cost, decrese in inventory cost)

    intangible benefit(better customer service, faster and accurate information for management)
  - identification of all expected costs of the project

    tangible cost(cost of hardware and software, employee salaries, ETC)

    intangible cost(lost of custumer goodwill or employee morale)

Types of feasibility Study

Technical Feasibility

- helps to determine the present technical resources of the organization and their applicability to the expected needs of the proposed system
- technical resources includes:
  - hardware
  - software
  - operating environments
- After feasibility study, one must determine these resources is sufficent ored to be upgraded or added.

Types of feasibility Study

Human Factors Feasibility

- focuses on the manpower components of the successful system implementation

Legal/Political Feasibility

- includes a through analysis of any potential lal rampifications resulting from construction and implementation of new system
- legal issues(copyright, patent infrigements, voilation of existing law, foreigh trade restrictions,)
- political issues
  - Stakeholder Analysis: Identify stakeholders and their interests in the software project.
  - Political Climate: Consider the impact of the prevailing political environment on the project.
  - Governance and Decision-Making: Analyze decision-making processes and power dynamics.
  - Funding and Resources: Assess availability of resources and political support for funding.
  - Stakeholder Engagement and Communication: Develop strategies to engage and communicate with stakeholders.
  - Risk Assessment: Identify and address potential risks associated with political factors.
  - Collaborative Approach: Foster collaboration and build support among stakeholders.

Systems Analysis

- An in-depth study of existing system
  - It produces the functional requirements used as the basis for the design of an information system

It typically involves a detailed study of the

- Information needs of a company and end users
- Activities, resources, and products of one or more of the information systems currently being used
- Information system capabilities required to meet the information needs of business stakeholders

Types of Analysis

Organizational

Study of the organization, including…

- Management structure
- People
- Business activities
- Environmental systems
- Current information systems
  - Input, processing, output, storage, and control

Present System

- Hardware, software, network, input, output, processing

Logical

A logical model is a blueprint of the current system

- It displays what the current system does, without regard to how it does it
- It allows an analyst to understand the processes, functions, and data associated with a system without getting bogged down with hardware and software

Functional Requirements

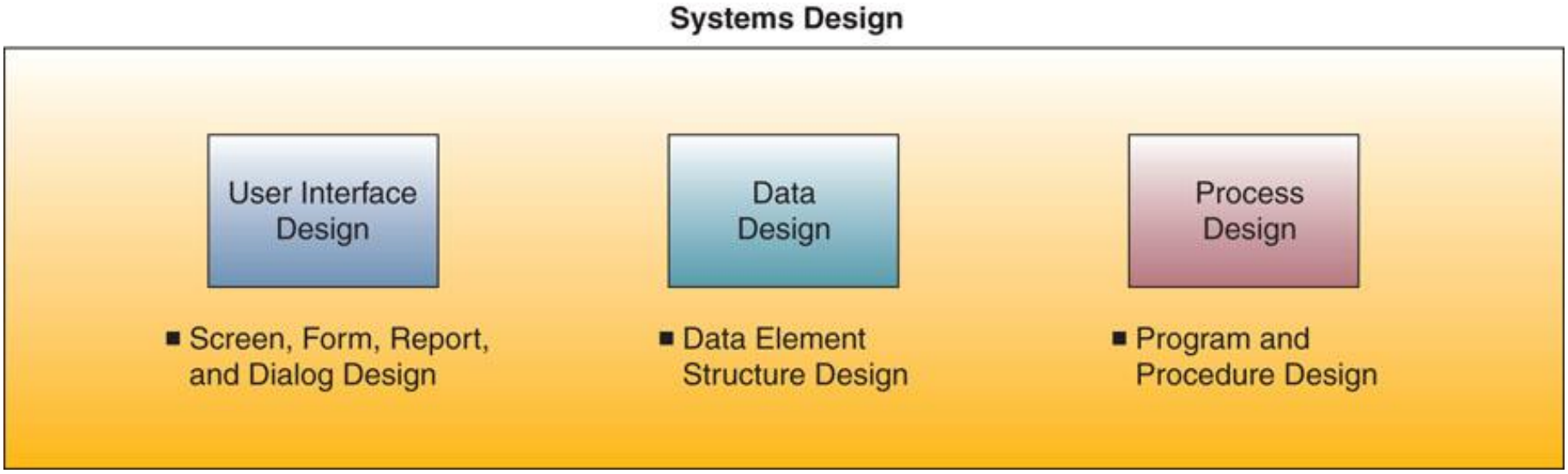
- Functional requirements specify the specific functionalities and behaviors a system should have to meet end user needs.
- Functional requirements are independent of the specific hardware, software, network, data, or people resources used in the system.
- They focus on the desired capabilities and operations of the system, rather than the technical details or resources.

Functional Requirements

Examples of Functional Requirements
• **User Interface Requirements** Automatic entry of product data and easy-to-use data entry screens for Web customers.
• **Processing Requirements** Fast, automatic calculation of sales totals and shipping costs.
• **Storage Requirements** Fast retrieval and update of data from product, pricing, and customer databases.
• **Control Requirements** Signals for data entry errors and quick e-mail confirmation for customers.

Systems Design

Focuses on three areas



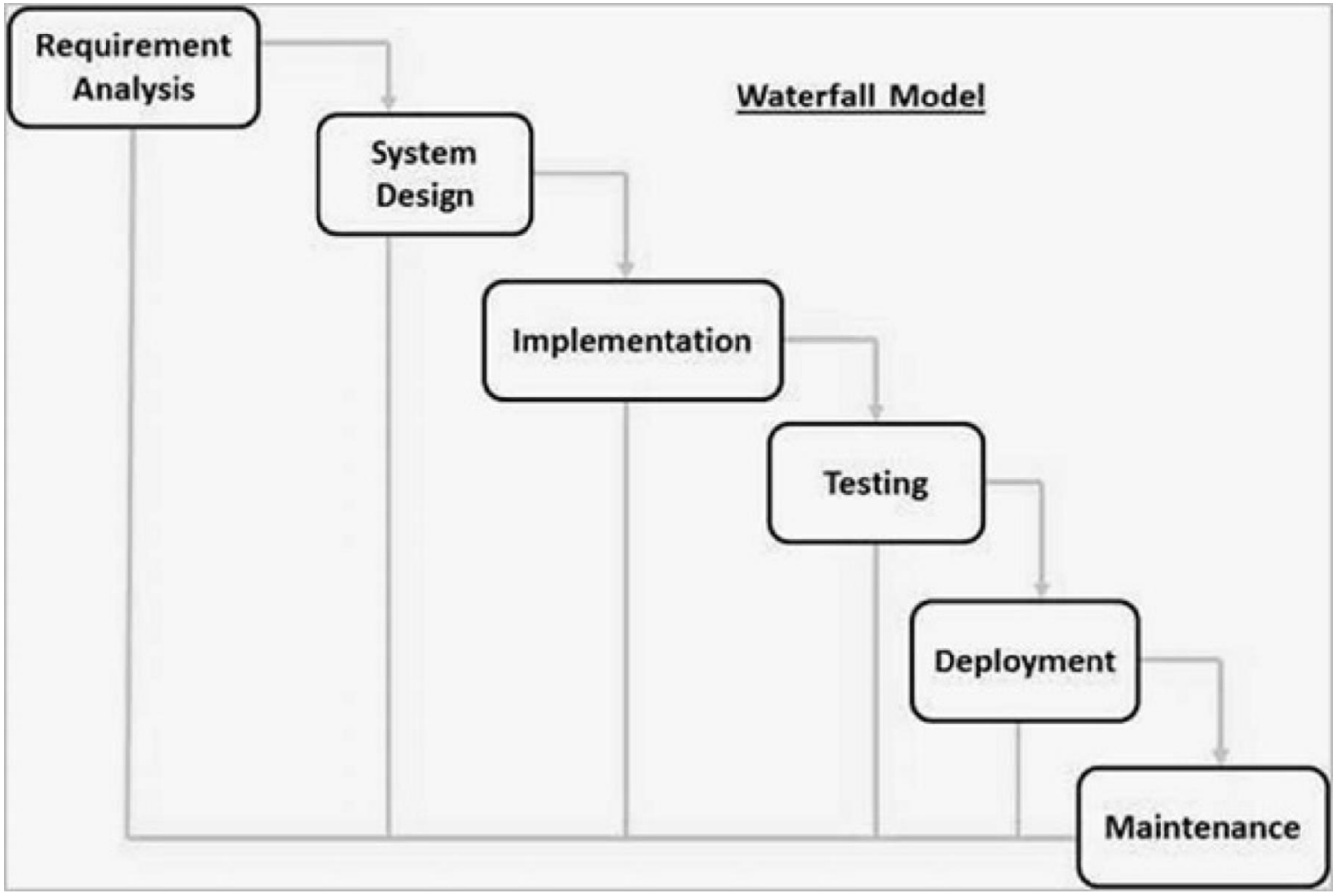
System Development Methodologies

System Development Methodology refers to the framework that is used to define necessary steps that are used to formulate, plan, and control the process of developing an information system.

- Waterfall Model
- Prototyping
- Incremental
- Spiral
- Rapid Application Development
- Agile Methods
- Extreme Programming
- Joint Application Development
- Rational Unified Process

Waterfall Model

Waterfall model is a linear sequential life cycle model of system development.



Advantages & Disadvantages of Waterfall Model

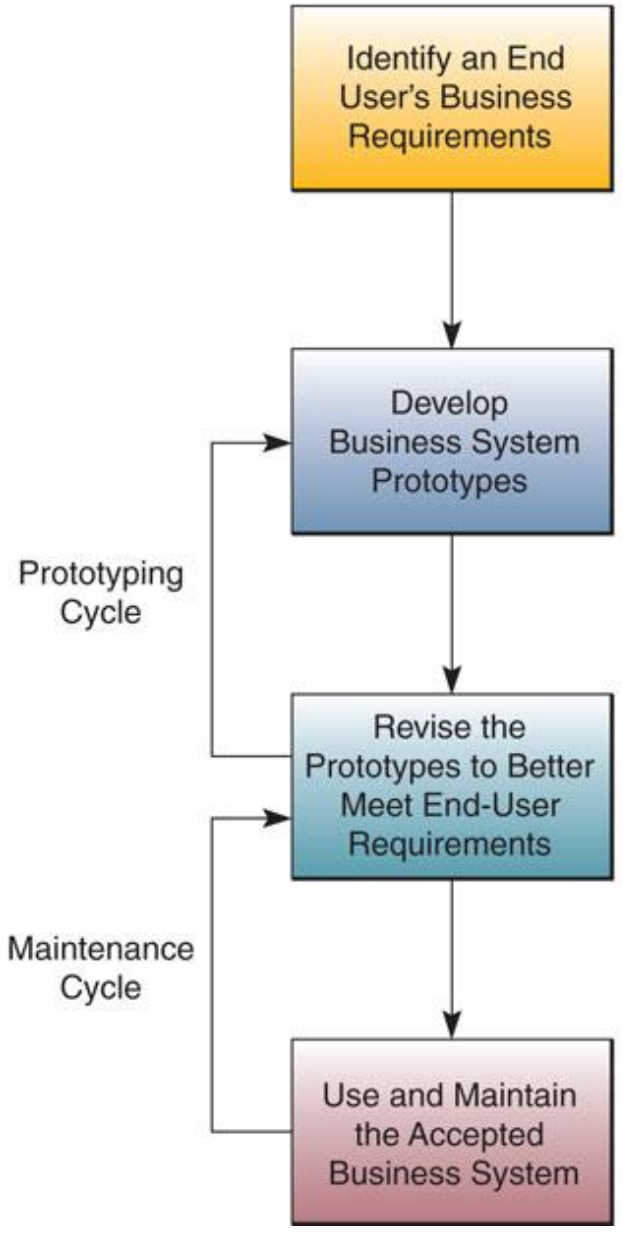
Advantages of Waterfall Model

- Simple and easy to understand
- Each phase is clearly defined with specific output
- Each pahse is processed and completed on time
- Good for smaller projects
- Processes and Outputs are very well documented
- Progress of system development is measurable and helps in optimizing resources
- Best methodology for initial smaller projects

Disadvantages of Waterfall Model

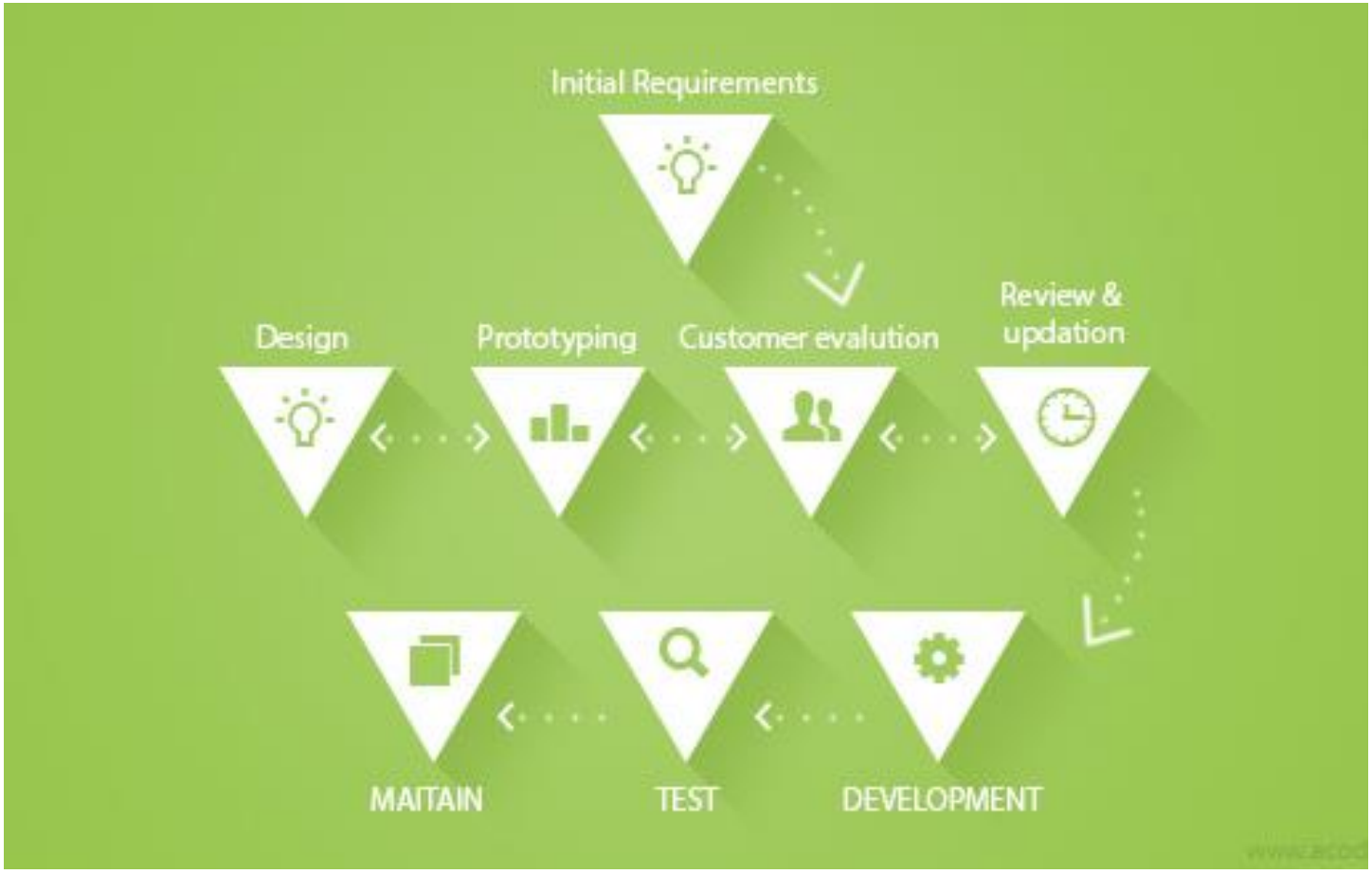
- Complete life cycle needs to be complted to get a working solution
- Very risky and uncertain model
- Not recommended for projects where the requirements are changing
- Not good for complex projects
- Cannot accommodate changing requirements
- No option of going back to complted phases for corrections
- Not good for Real-time and event-driven systems.

Prototyping Life Cycle



- **Investigation/Analysis.** End users identify their business needs and assess the feasibility of several alternative information system solutions.
- **Analysis/Design.** End users and/or IS specialists use application development tools to interactively design and test prototypes of information system components that meet end-user business needs.
- **Design/Implementation.** The business system prototypes are tested, evaluated, and modified repeatedly until end users find them acceptable.
- **Implementation/Maintenance.** The accepted business system can be modified easily since most system documentation is stored on disk.

Stages in Prototyping



Advantages of Prototyping

Advantages of Prototyping

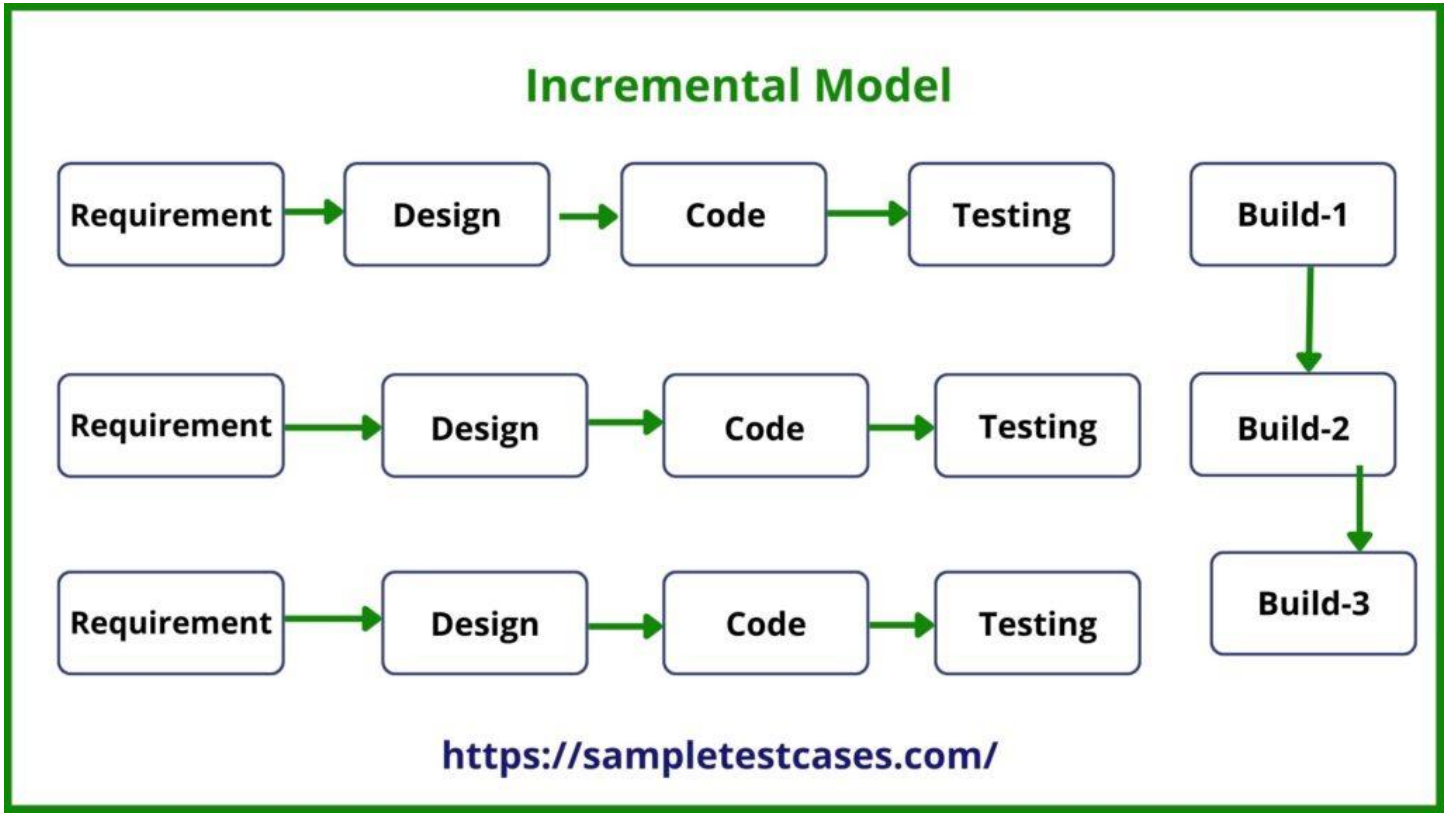
- Increased user involvement throughout the project
- User is able to get better understanding of the system while it is being developed
- Helps in building acceptable system
- Reduces time & cost as the defects can be detected at early stage
- Attempts to reduce project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
- Addresses the inability of many users to specify their information needs
- Useful for resolving unclear objectives
- Developing and validating user requirements
- Experimenting with or comparing various design solutions;
- Encourages innovation and flexible designs.
- Provides quick implementation of an incomplete, but functional, application.

Disadvantages of Prototyping

- Risk of incomplete requirement analysis
- User may get confused between prototype and actual system
- Not a standalone, complete development methodology
- It may increase the complexity of the system
- A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problem.
- Approval process and control is not strict.
- Requirements may frequently change significantly.
- Identification of non-functional elements is difficult to document.
- Designers may neglect documentation, resulting in insufficient justification for the final product and inadequate records for the future.
- Can lead to poorly designed systems.
- Iterations add to project budgets and schedules
- Prototype may not have sufficient checks and balances incorporated.

Incremental Models

Incremental models support combination of linear and iterative methods of system development.



Incremental Models

Incremental models support combination of linear and iterative methods of system development.

Advantages of Incremental Models

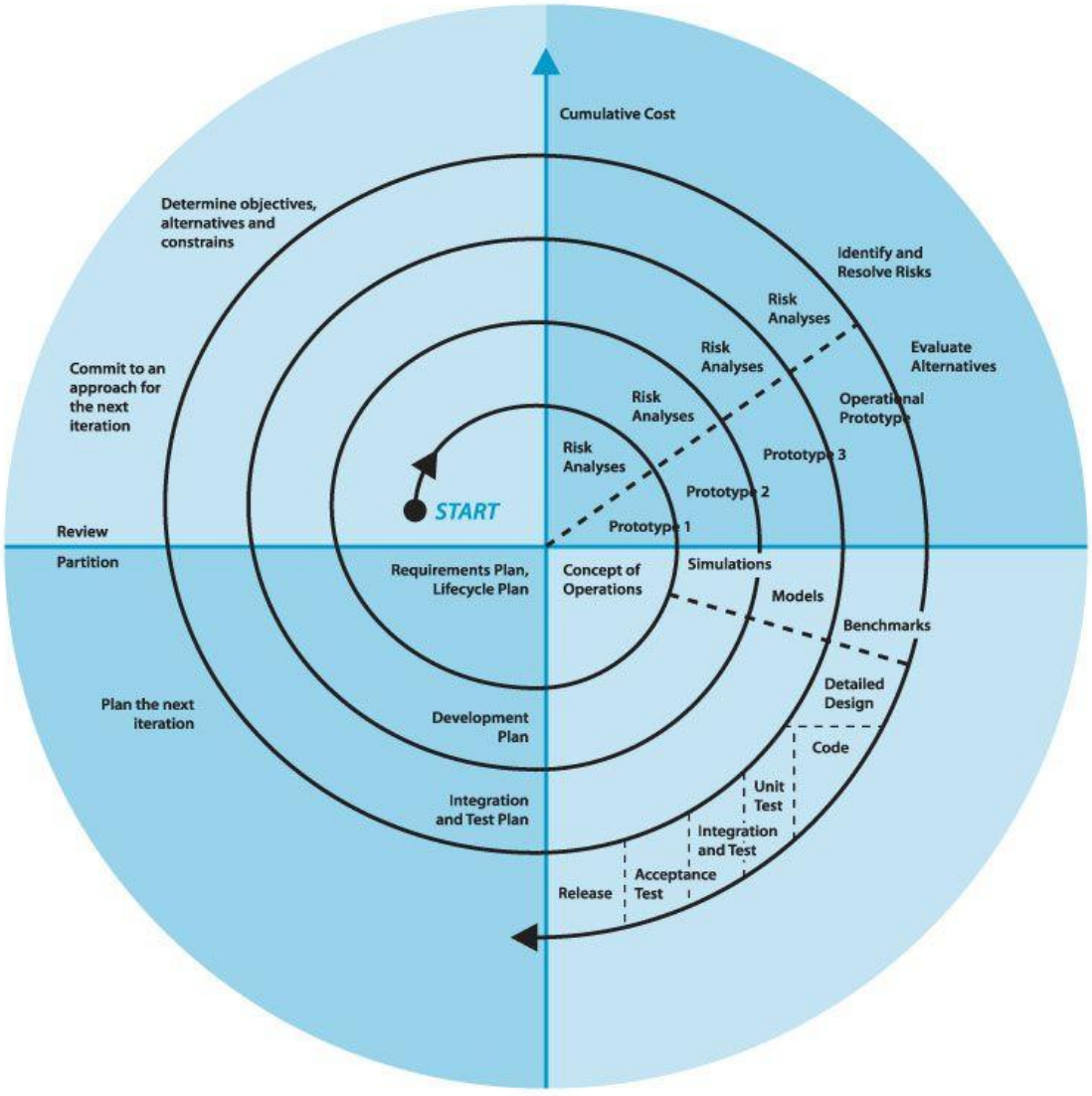
- Knowledge gained in an early increment can be used while developing later increments.
- Helps to mitigate integration and architectural risks earlier in the project.
- Limited control is maintained over the life of the project
- User is able to see the concrete evidence of project status throughout the life cycle.
- Gradual implementation provides the ability to monitor the effect of incremental changes, isolate issues and make adjustments before the organization is negatively impacted.
- Allows delivery of a series of implementations to speed up the project.
- Good for large projects where requirements are not very well defined/understood.

Disadvantages of Incremental Models

- Lacks overall consideration of the business problem and technical requirements for the overall system.
- Well-defined interfaces are required for smaller modules
- Difficult problems tend to be pushed to the future to demonstrate early success to management.
- Not good for small projects

Spiral Models

Spiral models are combination of linear and interative methods of system development which focuses on risk assessment and on minimizing project risk.



Spiral Models

Advantages of Spiral

- Enhances risk avoidance.
- Helps in selecting the best methodology for development
- Can incorporate Waterfall, Prototyping, and Incremental methodologies as special cases in the framework, and provide guidance as to which combination of these models best fits a given software iteration, based upon the type of project risk.
- Good for Real-time or safety-critical systems.
- Resource optimization
- Requirement exists for strong documentation control.
- Project benefit from a mix of other development methodologies.

Disadvantages of Spiral

- Project manager needs to be highly skilled and experienced.
- Composition of development methodologies to be used for each iteration around the Spiral is always a challenge
- Quite complex model.
- No established controls for moving from one cycle to another cycle.
- There is an inherent risk of not meeting budget or schedule.
- Possibility exists that project ends up following only the Waterfall framework.
- Risk avoidance is a low priority.

Rapid Application Development (RAD)

Rapid Application Development (RAD) model is an iterative method for fast development and delivery of a high quality system at a relatively low cost.

Advantages of RAD

- It is faster method of system development as compared to any of the earlier methods.
- This is one of the cheapest method of system development
- Users and other stakeholders are more committed to the new system because of their involvement and quick response
- Concentrates on essential system elements.
- Ability to rapidly change system design as per the demand.
- Produces a tighter fit between user requirements and system specifications.
- Helps in savings time, money, and human effort.
- Users possess detailed knowledge of the application area.
- Senior management commitment exists to ensure end-user involvement.
- Requirements of the system are unknown or uncertain.

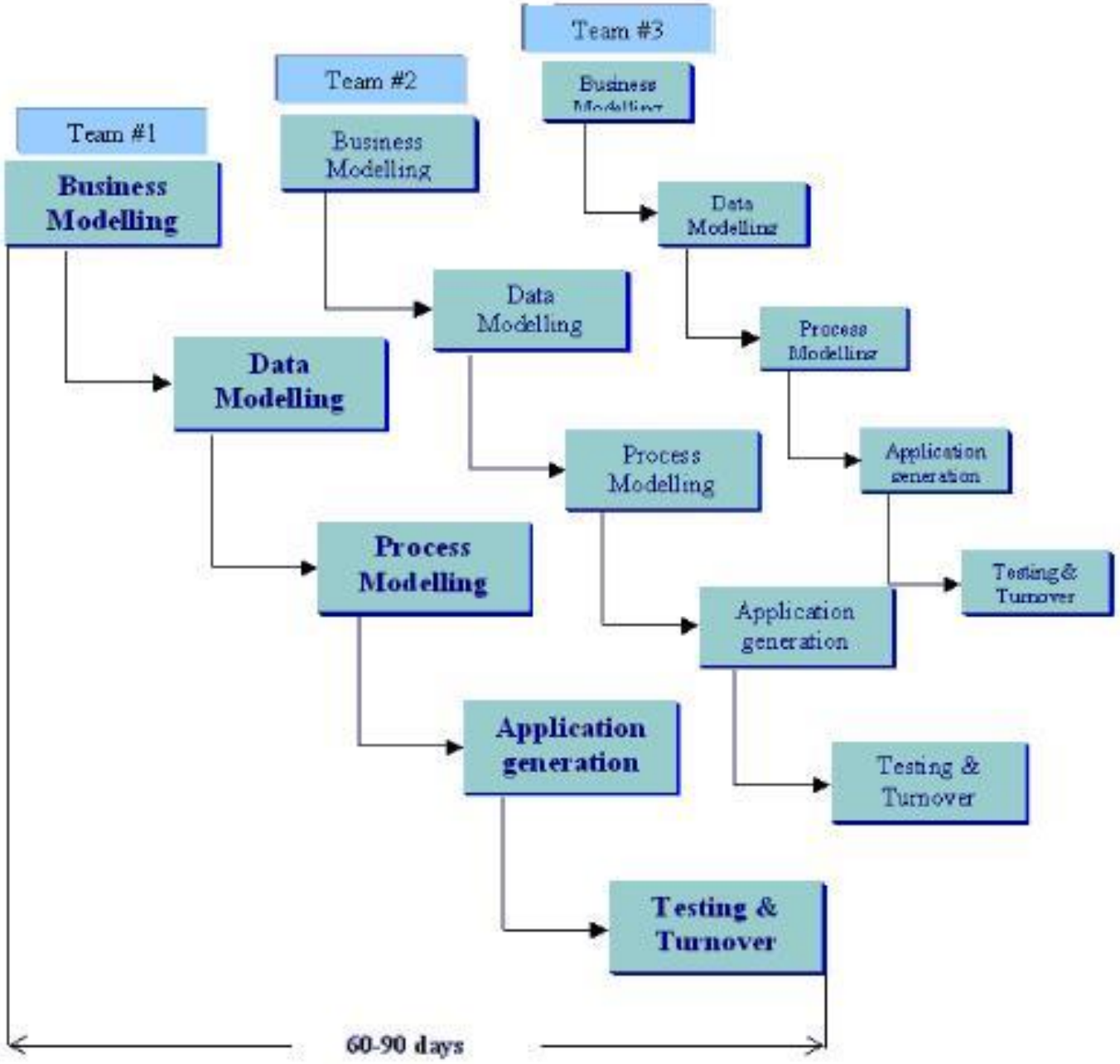


Figure 1.5 – RAD Model

Disadvantages of RAD

- High speed and lower cost may lead to impact quality
- Project may end up with more requirements than needed
- Potential for feature creep where more and more features are added to the system over the course of development.
- Potential for inconsistent designs within and across systems.
- Difficulty with module reuse for future systems.
- Formal reviews and audits are more difficult to implement than for a complete system.
- Requires skilled manpower to handle complexity
- Not applicable for small projects
- Not good for Real-time or safety-critical systems.

Agile Methods

- provide a flexible and iterative approach to project management.
- In contrast to traditional waterfall models, Agile methodologies embrace change, collaboration, and rapid delivery to meet the dynamic needs of today's software development projects.

Agile Manifesto

The Agile Manifesto, created in 2001 by a group of software development thought leaders, outlines the values and principles that underpin Agile methodologies. It consists of four core values:

1. Individuals and interactions over processes and tools:
   - Focus on people, collaboration, and teamwork.
   - Emphasize effective communication and a positive team culture.
2. Working software over comprehensive documentation:
   - Deliver functional software that adds value to the customer.
   - Prioritize tangible results and progress over extensive documentation.
3. Customer collaboration over contract negotiation:
   - Actively involve customers and stakeholders throughout the development process.
   - Seek continuous feedback, collaboration, and understanding of customer needs.
4. Responding to change over following a plan:
   - Embrace change and adaptability in response to evolving requirements.
   - Value the ability to quickly adjust plans and priorities to deliver the best outcome

Agile Principles

1. Satisfy the customer through early and continuous delivery of valuable software
2. Welcome changing requirements, even late in development
3. Deliver working software frequently, with a preference for shorter timescales
4. Business people and developers must work together daily
5. Build projects around motivated individuals
6. The most efficient and effective method of conveying information is face-to-face conversation
7. Working software is the primary measure of progress
8. Maintain a sustainable pace for the team
9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams produce the best architectures, requirements, and designs
12. Regularly reflect on how to become more effective and adjust behavior accordingly

Popular Agile methodologies

- Scrum: Framework for managing complex projects
- Kanban: Visual system for managing work
- Extreme Programming (XP): Focused on quality and collaboration
- Lean Software Development: Eliminating waste and delivering value
- Feature-Driven Development (FDD): Iterative and incremental process

Agile Methods

Advatnges of Agile Methods

- Adaptive approach that responds to changes
- Supports effective communication to maintain transparency
- Better quality solution as it helps in identifying defect early
- Helps in identifying gaps in expectations

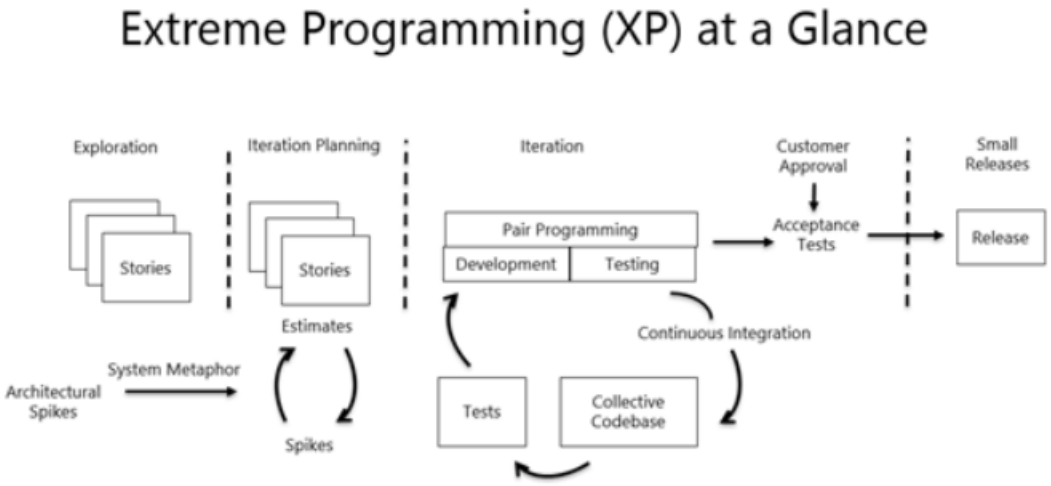
Disadvatnges of Agile Methods

- Focus more on software development and less on documentation
- Possibility of loosing track as objective are not clearly defined

Extreme Programming

Agile software development methodology.

It emphasizes collaboration, feedback, and adaptability.



Extreme Programming

Key Principles of Extreme Programming:

- Communication:
  - Emphasizes effective and open communication within the team and with stakeholders.
  - Encourages collaboration, feedback exchange, and transparency.
- Simplicity:
  - Focuses on keeping things simple and avoiding unnecessary complexity.
  - Prioritizes straightforward solutions and minimalistic design.
- Feedback:
  - Values continuous feedback throughout the development process.
  - Enables timely adjustments, validation, and iterative improvements.
- Courage:
  - Encourages ownership, decision-making, and embracing challenges.
  - Fosters a culture of trust, empowerment, and innovation.
- Respect:
  - Promotes mutual respect, diversity, and collaboration among team members.
  - Values empathy, support, and a positive working environment.

Extreme Programming

Advantages of XP

- Focus on customer involvement
- Establishes rational plans and schedules
- Developers are exceptionally committed to the project
- Equipped with modernistic methods for quality software

Disadvantages of XP

- Effectiveness depends on the people involved
- Requires frequent meeting for development raising total costs
- Necessitates for excessive development changes
- Exact possibilities and future outcomes are really unknown

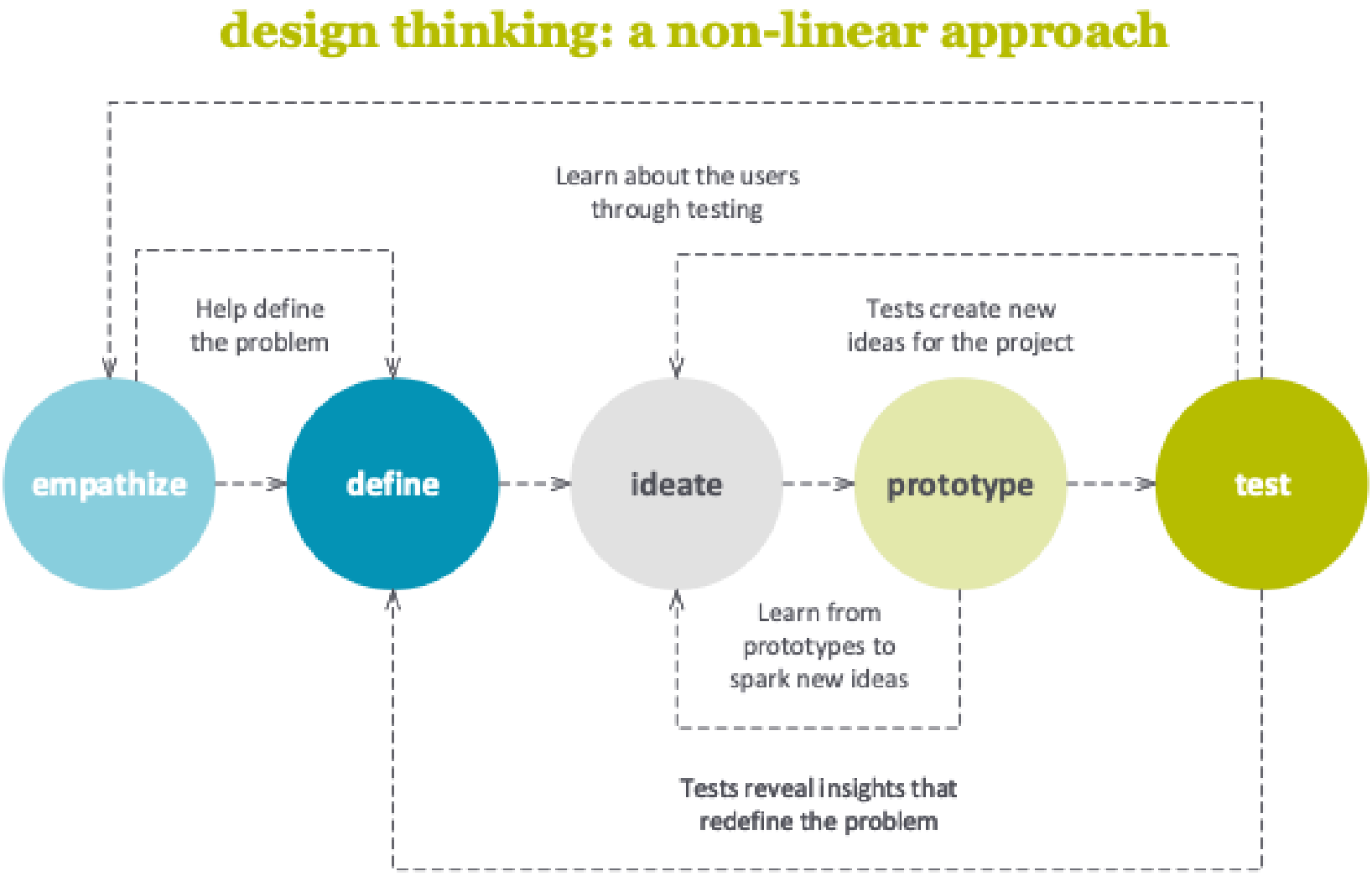
Design Thinking

Design Thinking is a methodology that provides a solution-based approach to problem solving.

Design thinking tackles complex problem by:

- Empathising: Understanding the human needs involved.
- Defining: Re-framing & defining the problem in human-centric ways.
- Ideating: Creating many ideas in ideation stage.
- Prototyping: Adopting a hands-on approach in prototyping.
- Testing: Developing a prototype/solution to the problem.

5-Stage process of Design Thinking



User Interface Design

Supports interactions between end users and computer-based applications

- Get help from end-users
- Designers create attractive and efficient forms
- Frequently a prototyping process
- Produces detailed design specifications

End User Development

Definition: End-user development refers to the process of creating or modifying software applications by non-professional developers, allowing users to customize or extend software systems to meet their specific needs.

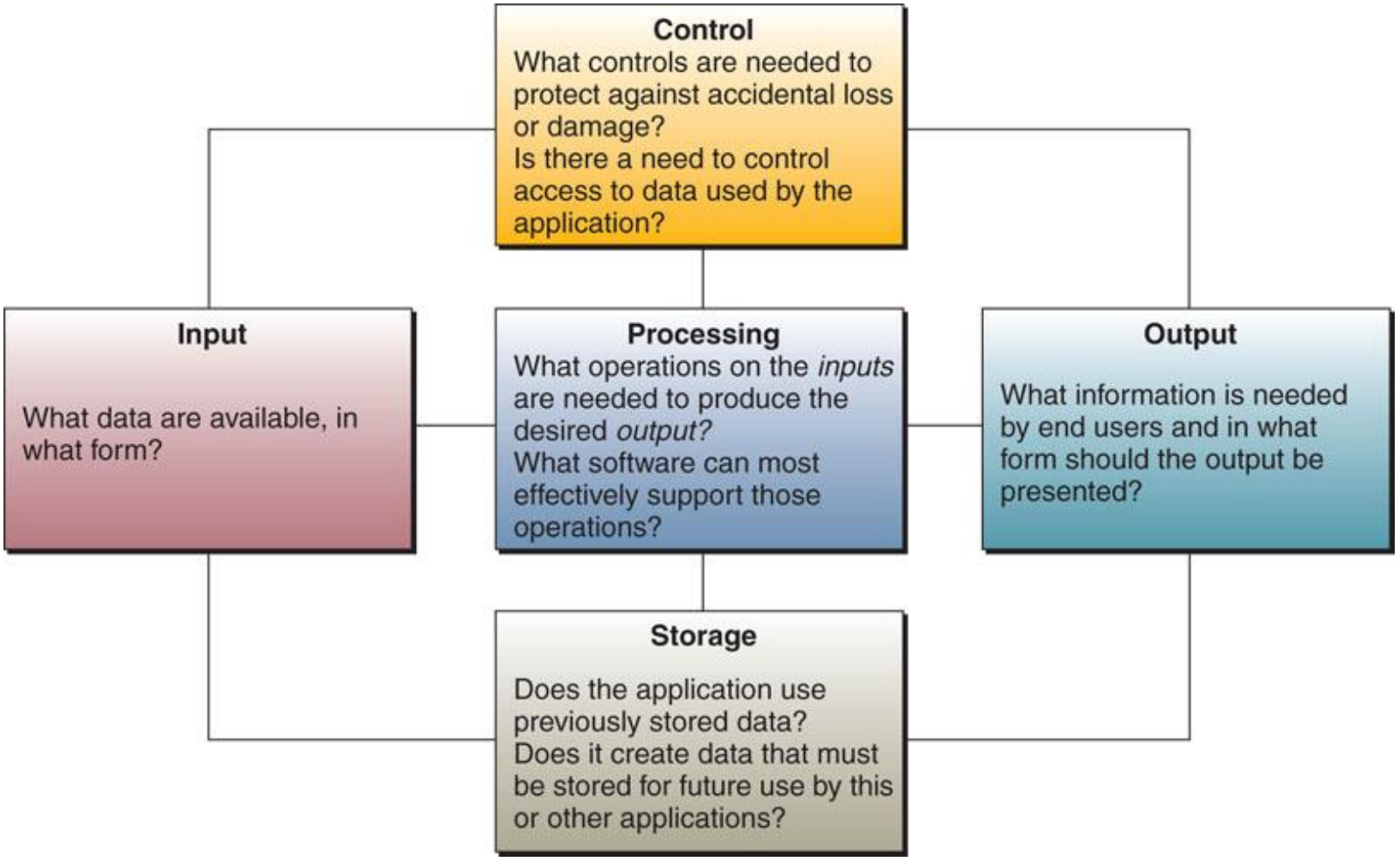
Empowerment: End-user development empowers non-technical users to take control of software customization without relying on professional developers or programmers.

Tools and Platforms: End-user development often involves the use of user-friendly tools and platforms that offer high-level abstractions, drag-and-drop interfaces, and predefined templates to simplify the development process

Role of IS professional in End User Development

- **Requirements:** They gather and analyze user requirements, translating them into technical specifications.
- **Tool Selection and Training**: IS professionals select suitable end-user development tools and provide training to users for effective utilization.
- **Governance and Quality Assurance**: They establish governance processes, ensure compliance with standards, and conduct quality assurance checks.
- **Technical Support:** They offer troubleshooting assistance during the development and deployment stages.
- **Integration:** IS professionals assist in integrating end-user-developed applications with existing systems.
- **Scalability and Maintenance:** They consider scalability and long-term maintenance of end-user-developed applications.
- **Security and Data Protection:** They implement security measures and ensure compliance with data protection protocols.
- **Collaboration and Communication**: IS professionals facilitate effective communication and collaboration between end users and developers.

End User Development



Encouraging End User Web Development

Look for tools that make sense

Spur creativity

Set some limits

Give managers responsibility

Make users comfortable

Outsourcing

- Outsourcing is the practice of delegating specific tasks, projects, or business processes to external third-party entities.
- Instead of handling these activities internally, organizations contract with external vendors or service providers who specialize in the respective areas.
- Outsourcing can encompass various functions such as information technology (IT), customer support, manufacturing, human resources, accounting, and more.

Advantages of outsourcing

- **Access to specialized skills and expertise**: Outsourcing provides access to external professionals with specialized knowledge and experience in specific areas.
- **Increased efficiency and focus on core competencies**: By outsourcing non-core functions, organizations can streamline operations and concentrate on their core business activities.
- **Scalability and flexibility:** Outsourcing allows businesses to easily scale resources up or down based on changing needs and demands.
- **Improved service levels:** Outsourcing to specialized service providers can enhance service quality, speed, and customer satisfaction.
- **Access to a global talent pool:** Outsourcing opens up opportunities to leverage the skills and expertise of professionals from around the world.
- **Risk mitigation**: Outsourcing allows organizations to transfer certain risks to external partners who have expertise in managing those risks.
- **Time savings**: By outsourcing tasks, companies can save time and allocate resources more effectively, focusing on strategic initiatives and core business objectives.

Disadvantages of Outsourcing

1. **Loss of Control:** Outsourcing involves relinquishing control over processes and outcomes to external service providers.
2. **Communication and Coordination Issues:** Challenges can arise due to communication gaps, time zone differences, and coordination difficulties with the outsourcing partner.
3. **Quality and Performance Concerns:** There is a risk of variations in the quality and performance of work provided by the outsourcing partner.
4. **Security and Data Privacy Risks**: Outsourcing involves sharing sensitive data, which can increase the risk of data breaches and unauthorized access.
5. **Dependency on External Providers:** Over-reliance on external providers can make organizations vulnerable to disruptions if the provider encounters difficulties.
6. **Cultural and Language Differences:** Offshore outsourcing may lead to challenges related to cultural and language differences.
7. **Hidden Costs and Contractual Challenges**: There may be hidden costs associated with contract negotiation, management, and monitoring, and organizations need to carefully review contracts.
8. **Negative Impact on Internal Workforce:** Outsourcing can impact employee morale, job security, and organizational culture if not managed effectively.

Object-Oriented Analysis and Design

Object – anything a programmer needs manipulated

Object-Oriented Programming (OOP) techniques:

- Inheritance
- Modularity
- Polymorphism
- Encapsulation

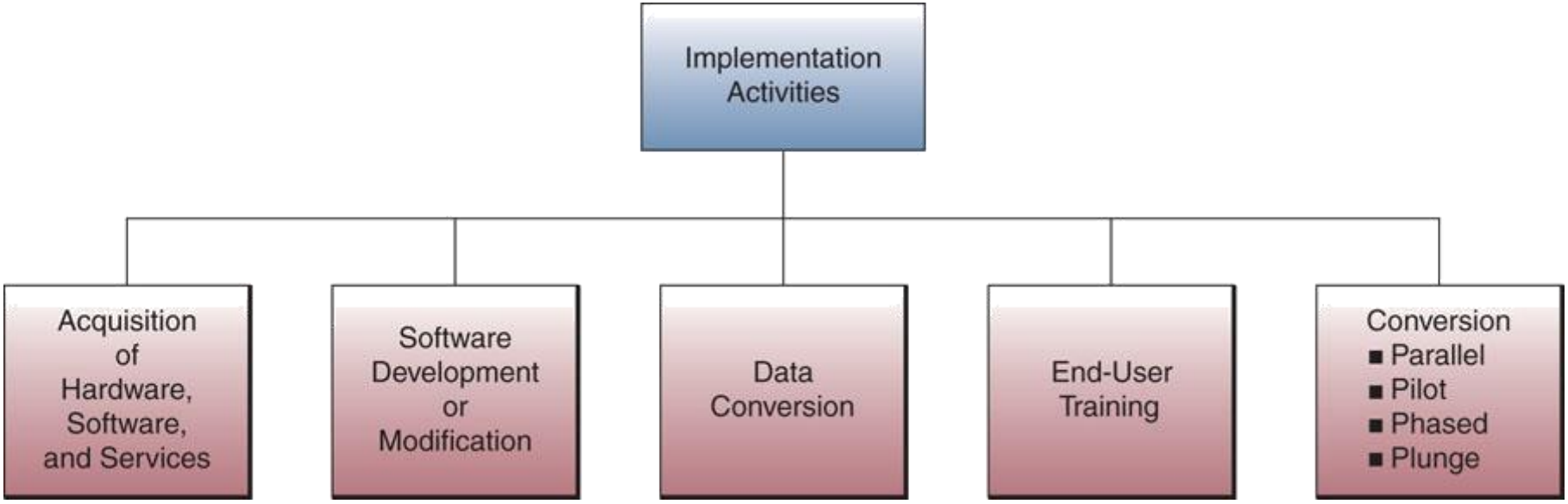
Object-Oriented Analysis (OOA)

- Model of object interaction, not solution

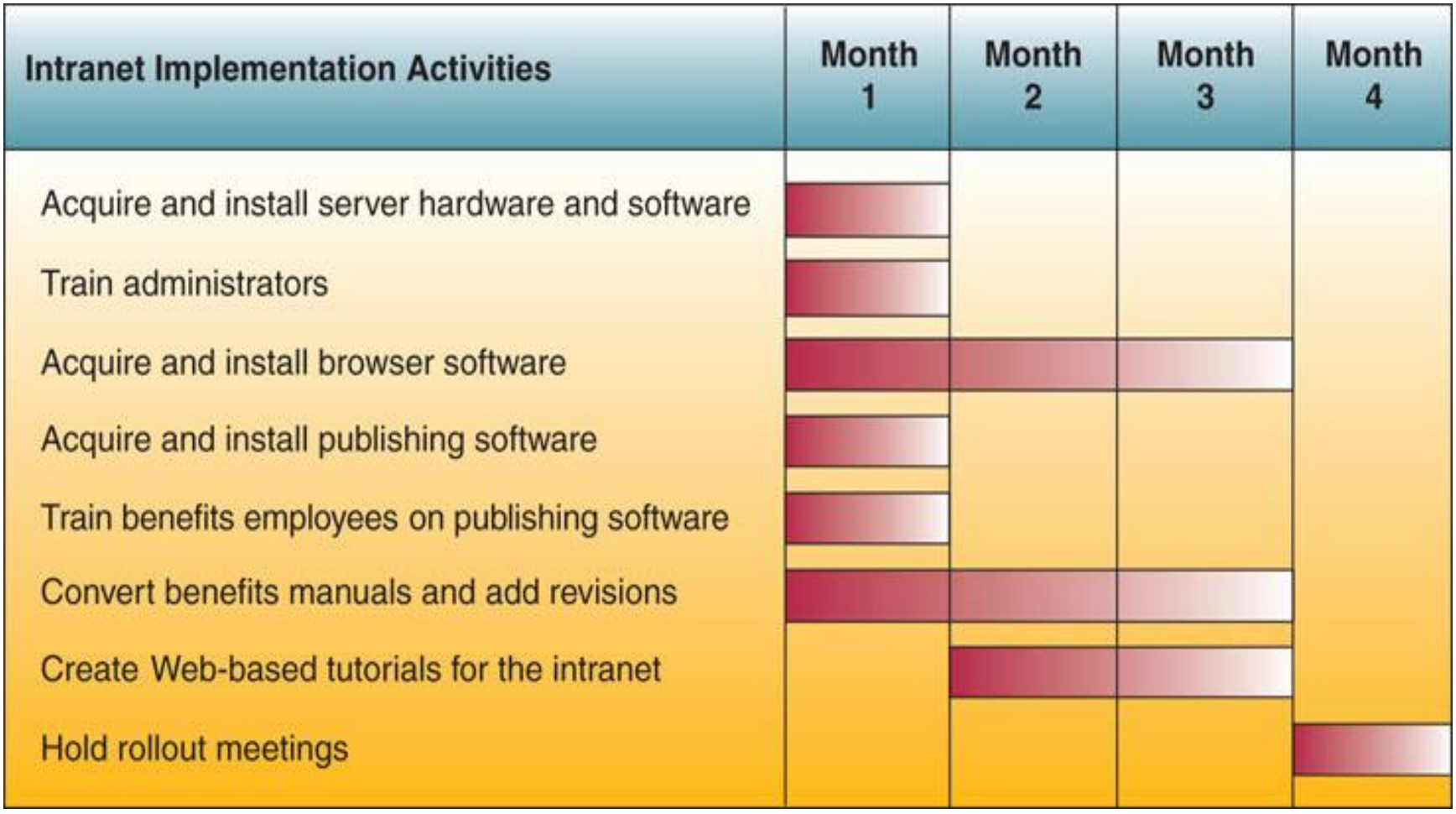
Object-Oriented Design (OOD)

- Solution based on constraints

Implementation Process



Sample Implementation Process



Other Implementation Activities

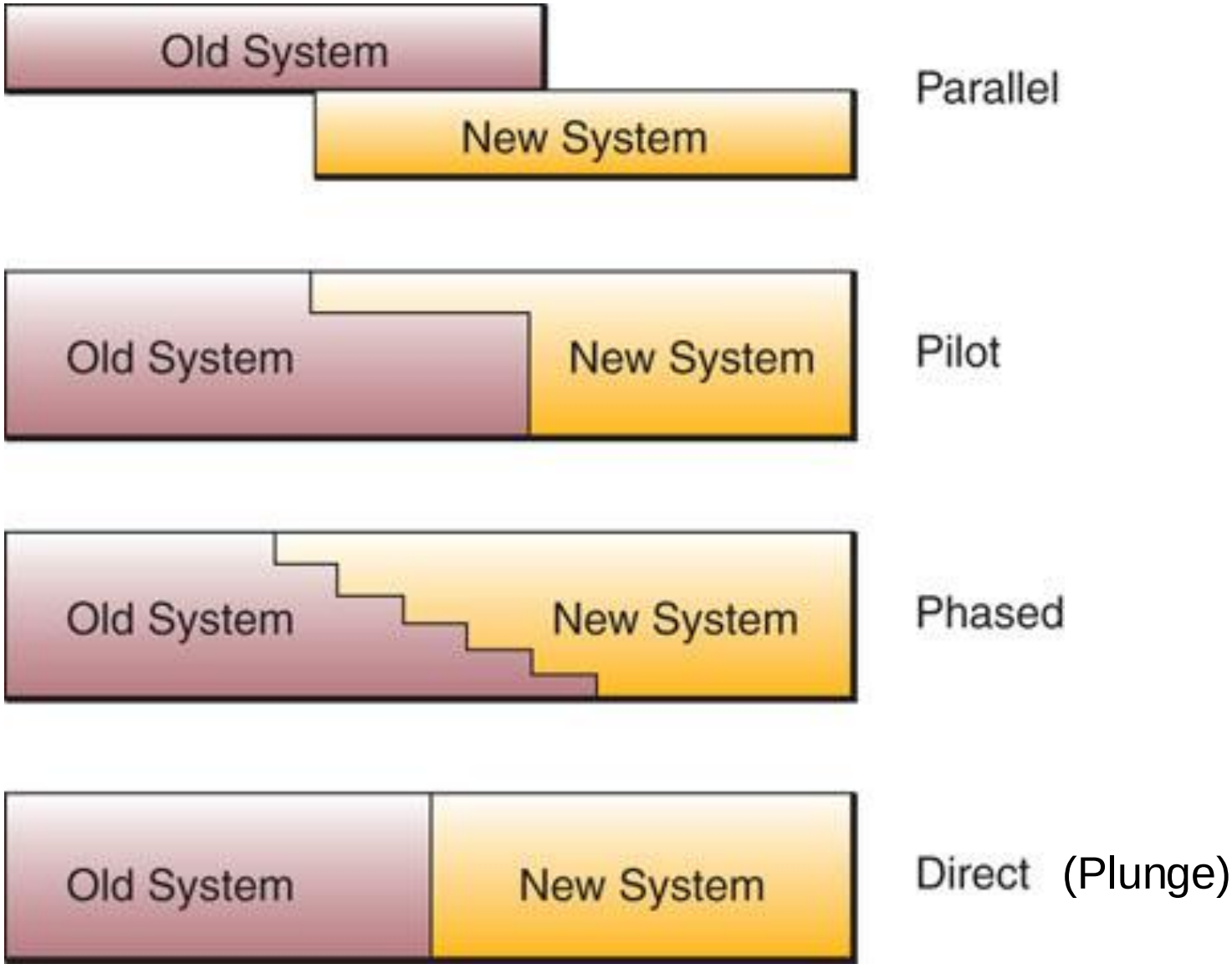
Testing

Data conversion

Documentation

Training

Major System Conversion Strategies

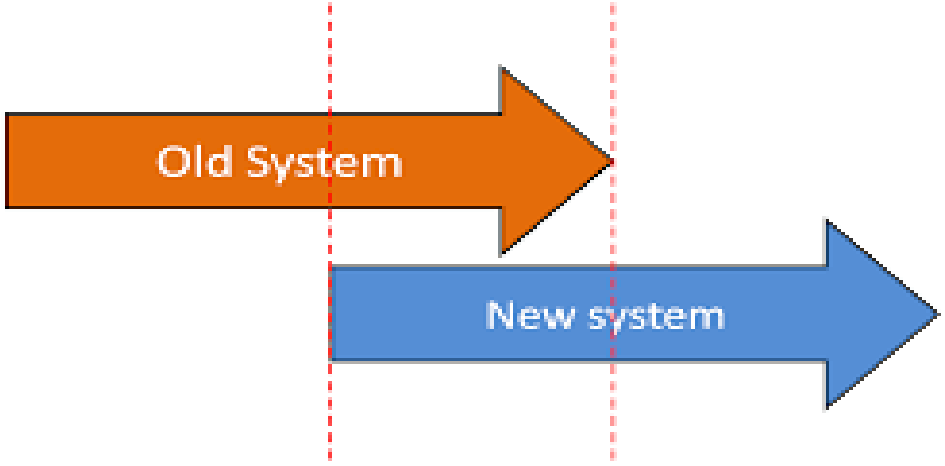


Parallel Conversion:

Definition: In parallel conversion, both the old and new systems run simultaneously for a certain period.

Pros: Allows for a side-by-side comparison of the old and new systems, ensuring that the new system functions correctly. Reduces the risk of data loss or system failure.

Cons: Requires additional resources to maintain and operate both systems simultaneously. Can be time-consuming and complex to manage.

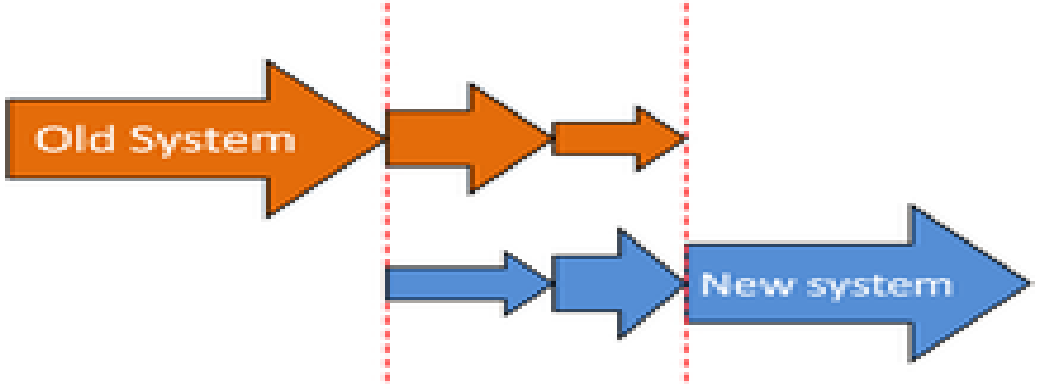


Phased Conversion:

Definition: Phased conversion involves a gradual transition from the old system to the new system in stages or modules.

Pros: Reduces the impact of change by implementing the new system in manageable increments. Provides opportunities for testing, training, and addressing issues in smaller batches.

Cons: Requires careful planning and coordination to ensure smooth transitions between phases. May result in a longer overall conversion timeline.



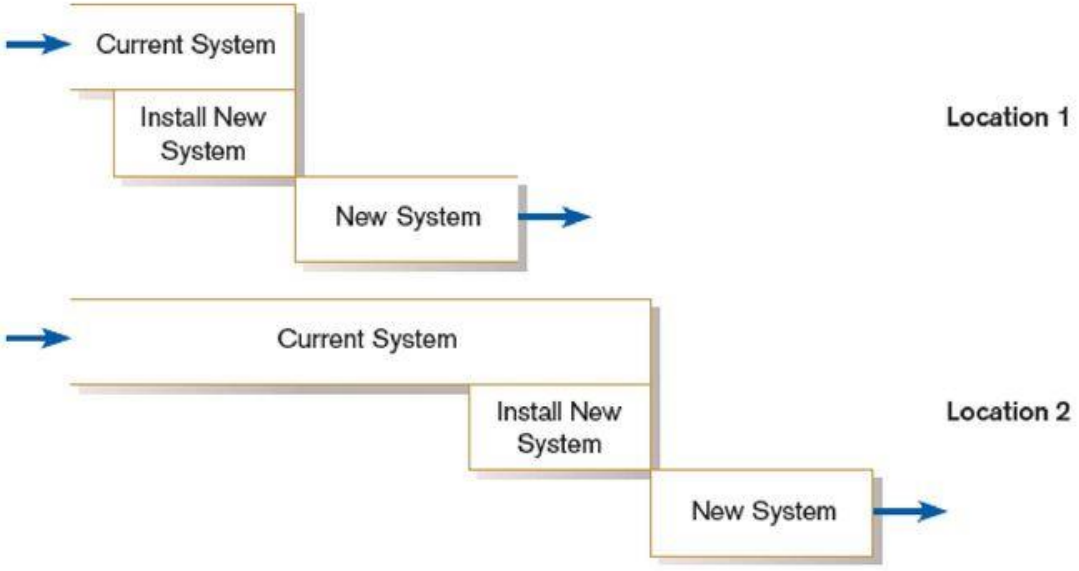
Pilot Conversion

Definition: In pilot conversion, the new system is implemented in a small, controlled environment or department before rolling it out to the entire organization.

Pros: Allows for comprehensive testing, user feedback, and refinement before full-scale implementation. Identifies and resolves issues on a smaller scale, reducing the impact on the entire organization.

Cons: May not fully represent the complexities and challenges of the entire organization. Requires effective communication and coordination between the pilot group and the rest of the organization.

Pilot Conversion

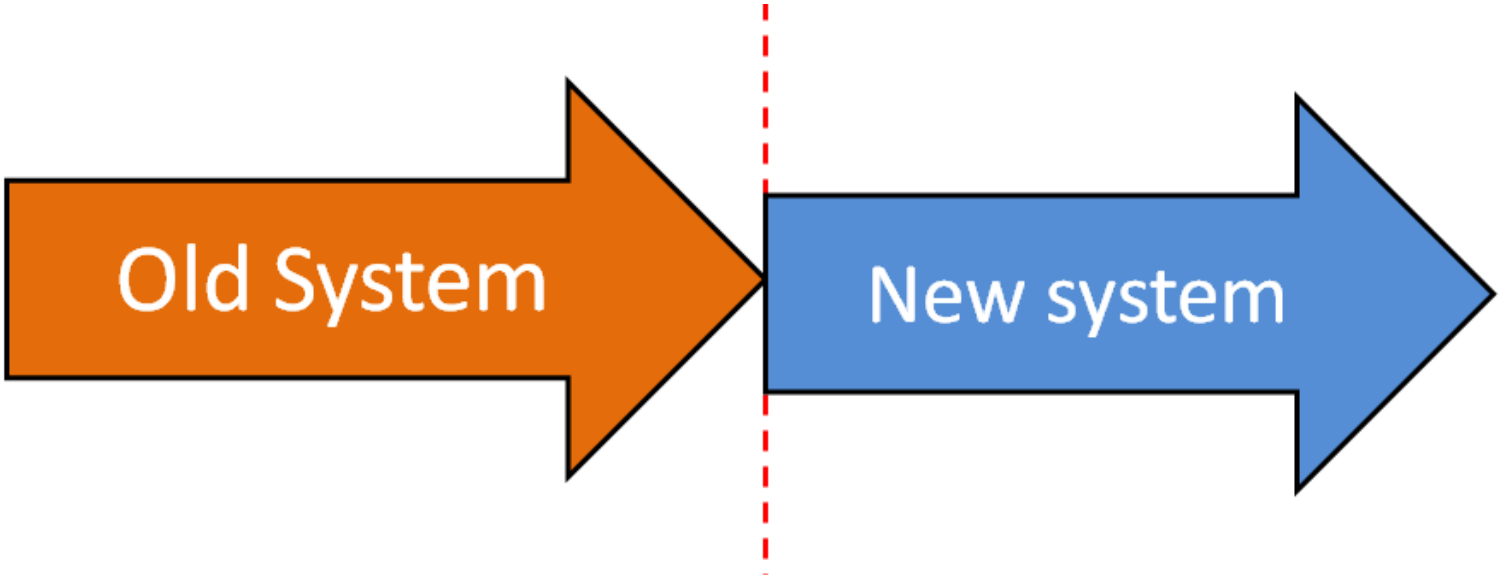


Direct Conversion

Definition: In direct conversion, the old system is completely replaced by the new system on a specific go-live date.

Pros: Swift transition to the new system, minimizing the need for ongoing dual operations. Can provide immediate benefits and cost savings once the new system is implemented.

Cons: High risk as the organization must quickly adapt to the new system. Limited opportunity for testing or identifying potential issues before going live.



Post-Implementation Activities

System Maintenance

- Corrective: fix bugs and logical errors
- Adaptive: add new functionality
- Perfective: improve performance
- Preventive: reduce chances of failure

Post Implementation Review

- Correct Errors
- Periodic review/audit

Single most costly activity