ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY USING DevOps

Exp-1

By
B Manikyala Rao M.Tech(Ph.d)
Assistant Professor
Dept of Computer Science & Engineering
Aditya College of Engineering & Technology
Surampalem

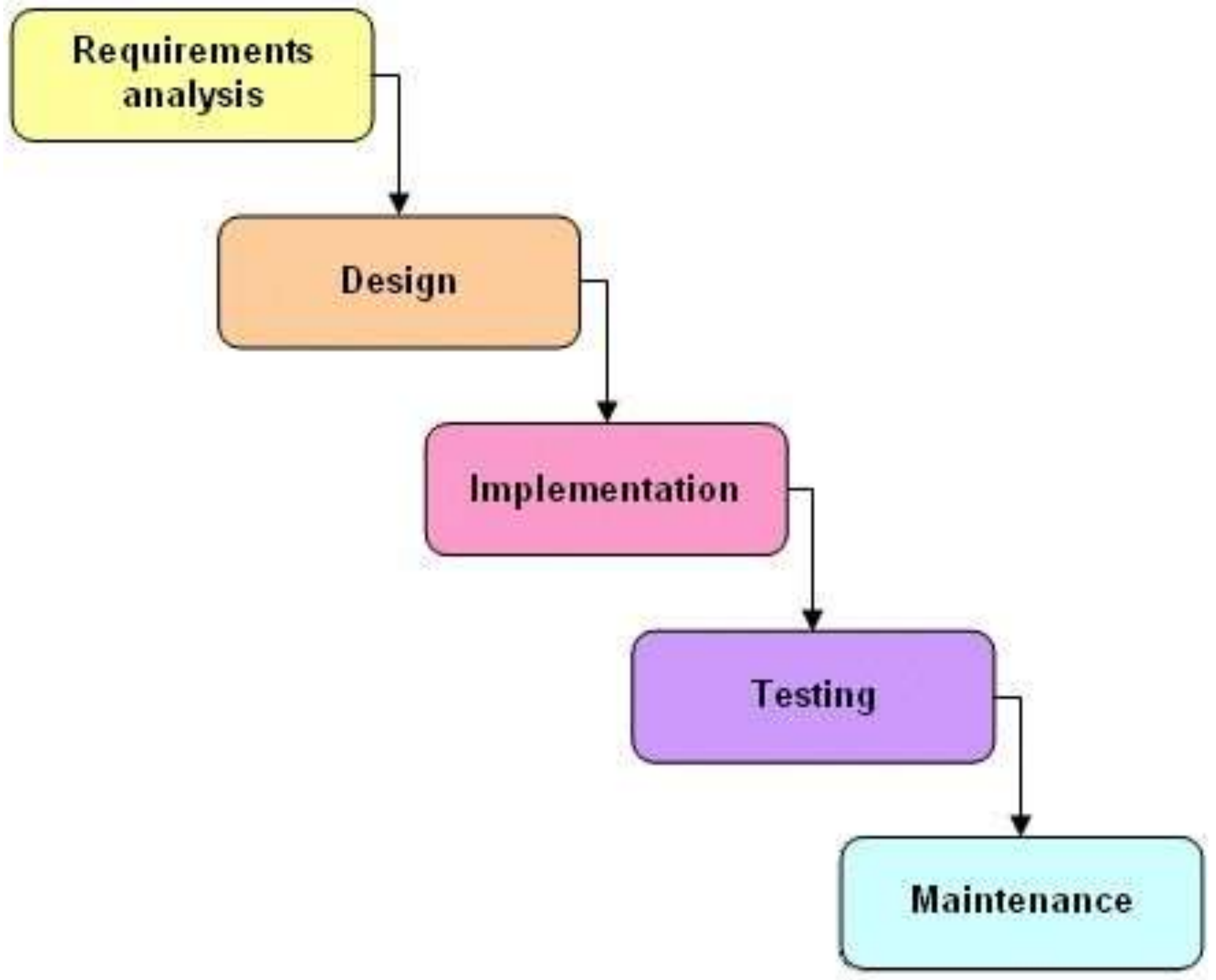
What is Software Engineering?

- The process of solving customers’ problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints.

SDLC(Software Development Life Cycle):

- SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

SDLC Phases


Requirements analysis
Design
Implementation
Testing
Maintenance

SDLC Phases

1.Requirement gathering and analysis: This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system?

2. **Design :** System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

3. **Implementation / Coding :**On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.

4. **Testing:** After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.

5**.Deployement:**Deploy the software in customer environment.

6. **Maintenance:** Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance

Software Development Process Models

- Waterfall model
- Prototyping model
- Incremental model
- Spiral model
- Agile process model

Waterfall Model

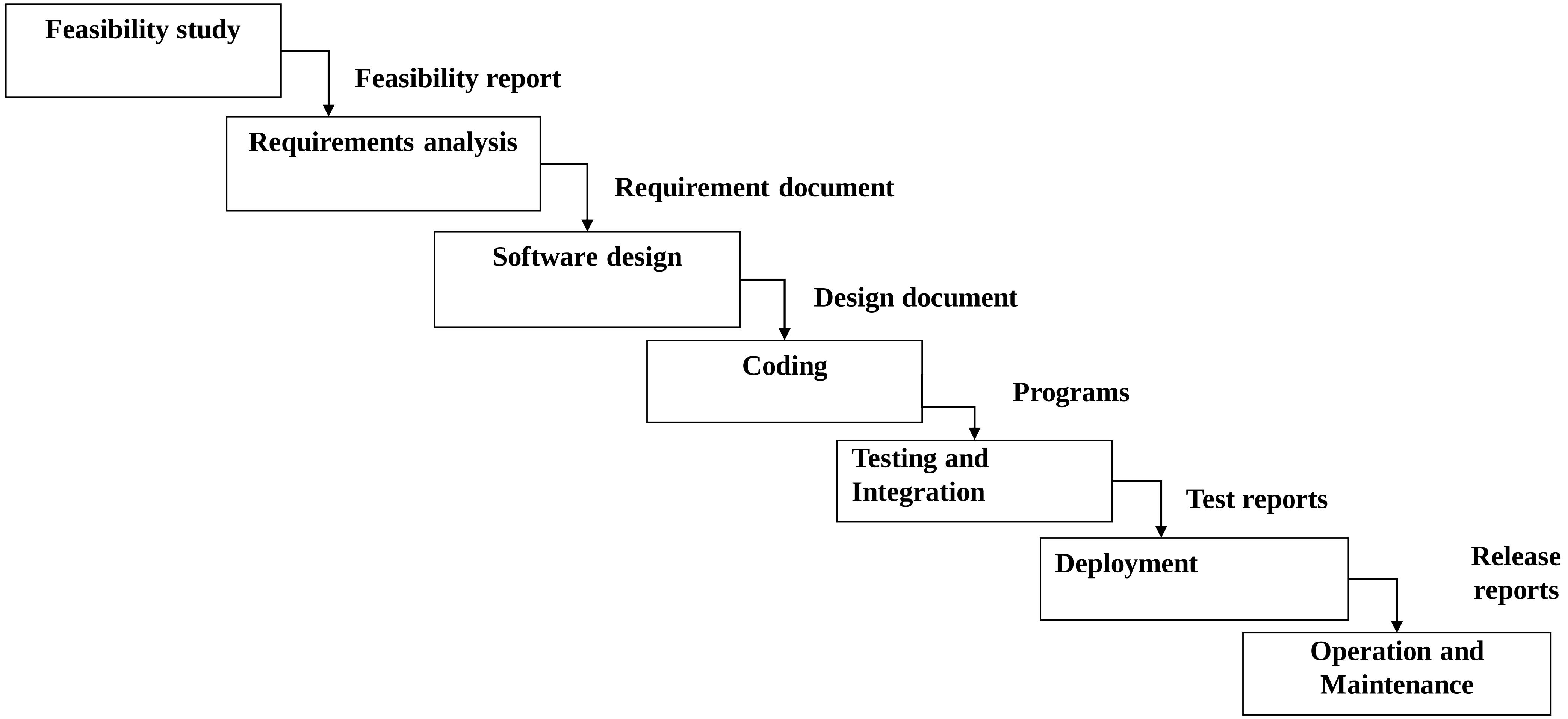


Figure 2.4: Classical waterfall model

Waterfall Model

Advantages:

- The main advantage of the waterfall model is that it is easy to understand and implement.
- Where the requirements are well understood and the developers are confident, the waterfall model works well.

Disadvantages

- The model assumes that the requirements will not change during the project. Sometimes, it is unrealistic to expect accurate requirements early in a project.
- It is very difficult to estimate the time and cost in the waterfall model.
- The people mentally ready to work in a phase will have to wait until its previous phase is completed

Prototyping Model

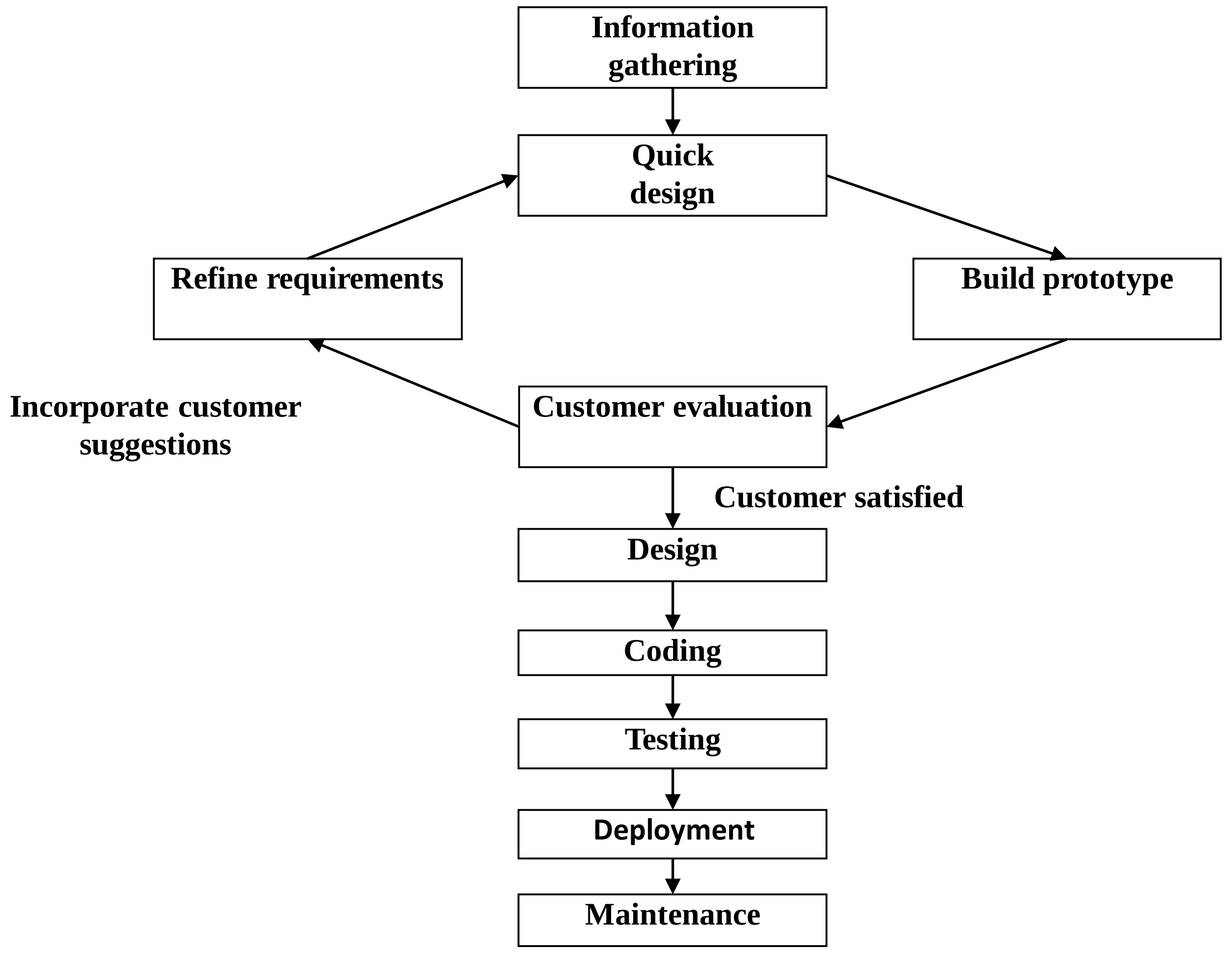


Figure 2.6: Prototyping model

Prototyping Model

- Prototyping is an alternative in which partial working software (i.e. a prototype) is initially developed instead of developing the final product.
- A prototype helps customer to understand the requirements that can further reduce the possibility of requirement changes
- The prototype model is well suited for projects where requirements are difficult to understand and the customer is not confident in illustrating and clarifying the requirements

Incremental Model

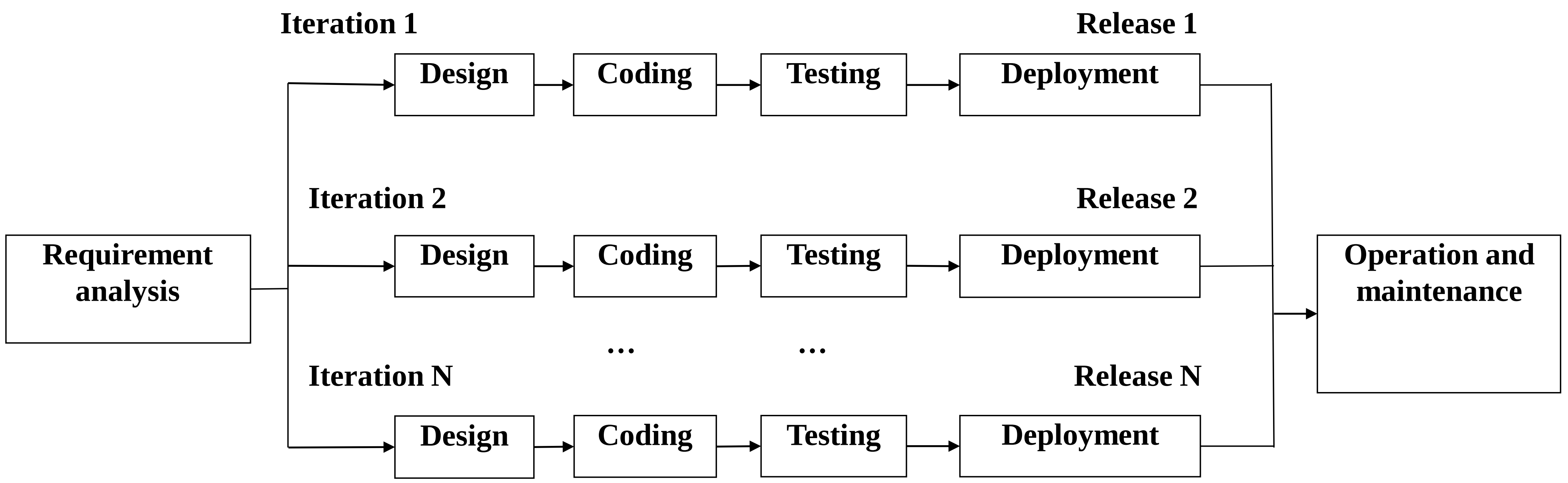


Figure 2.7: Incremental Model

Incremental Model

- The main advantage of the incremental model is the early production of working software during the software life cycle.
- Because each module is tested thoroughly, there is little possibility to change scope and requirements in the final software.
- Due to incremental development, testing and debugging of each module become easier.
- This model is also helpful in handling risks (technical, requirements, usability, etc.) because risky modules are identified and handled in a separate iteration.
- This model is suitable for larger projects where requirements are somewhat clear and which need phase-wise implementation.

Spiral Model

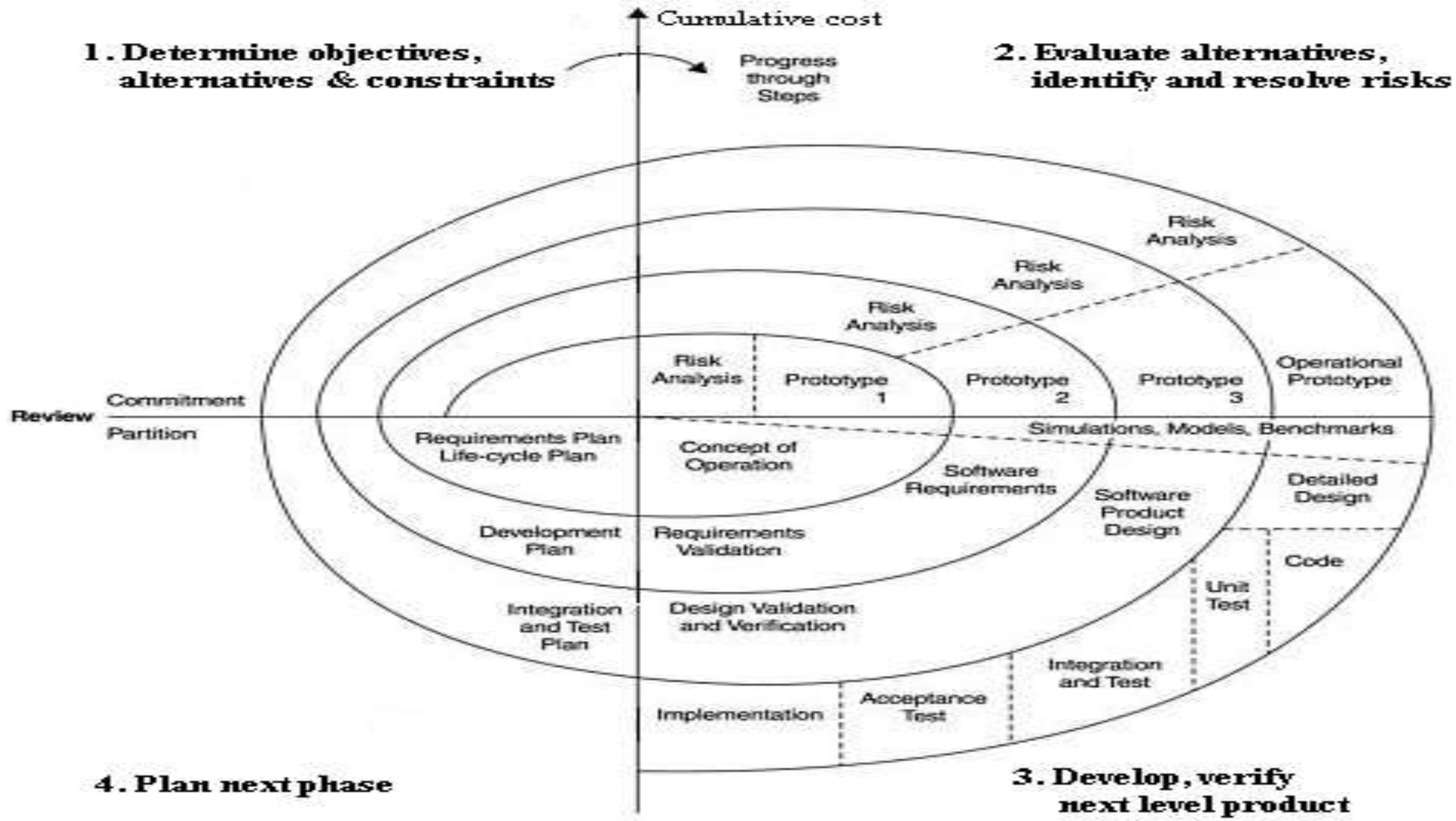


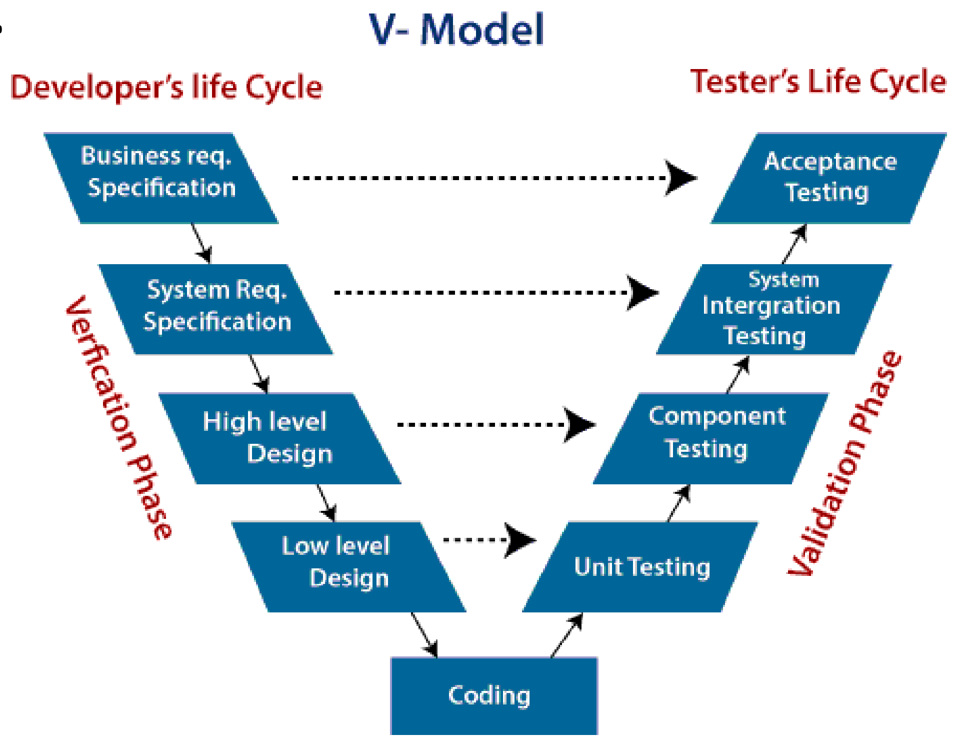
Figure 2.8: Spiral Model

Example

- This model is most suitable for projects having high risks and also for large, complex, and ambitious projects.
- The estimates (i.e. budget, schedule, etc.) get more realistic as work progresses because important issues are discovered earlier.

V-Model

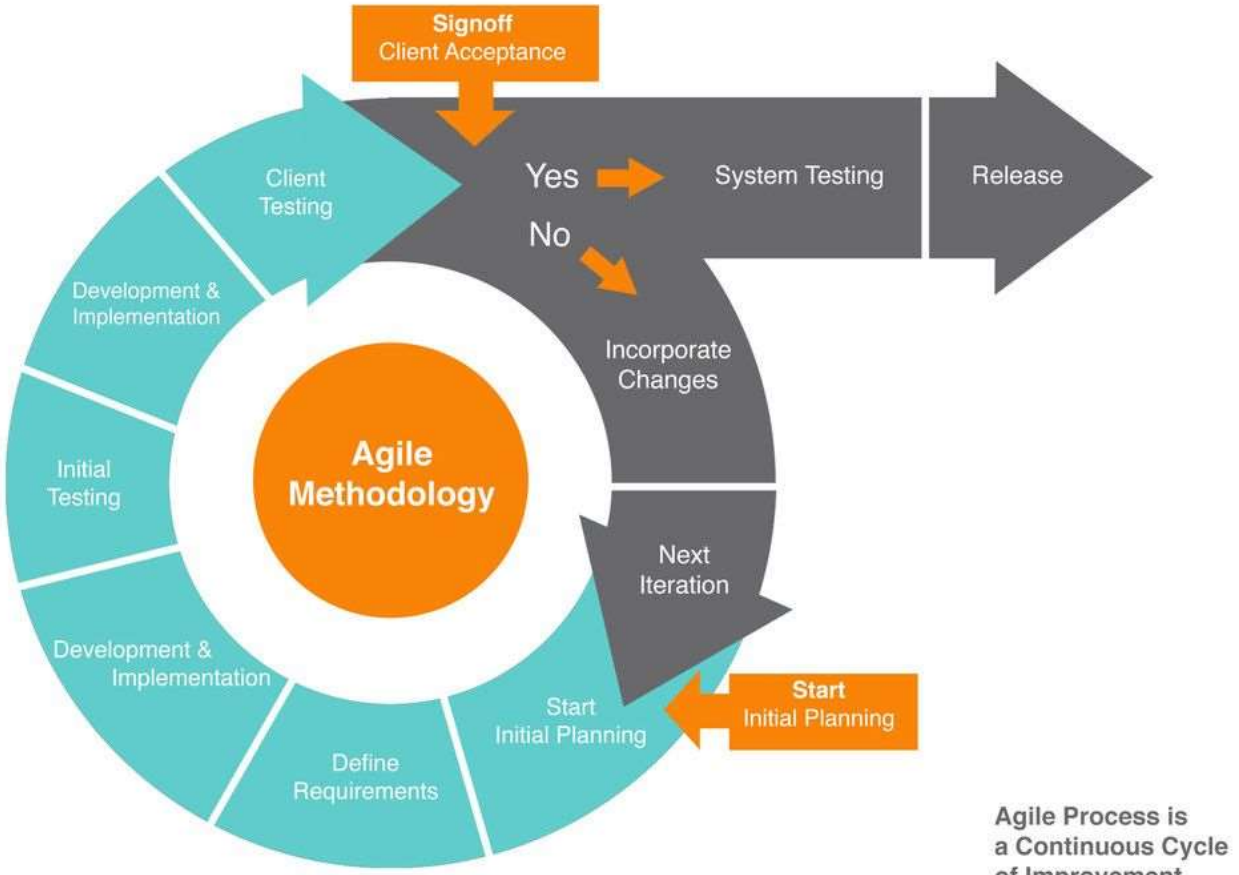
- V-Model also referred to as the Verification and Validation Model. In this, each phase of SDLC must complete before the next phase starts. It follows a sequential design process same as the waterfall model. Testing of the device is planned in parallel with a corresponding stage of development.



Agile Methodology

- Agile methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project.
- In the Agile model, both development and testing activities are concurrent, unlike the Waterfall model.
- Not a process, it's a philosophy or set of values
- The Agile software development methodology is one of the simplest and effective processes to turn a vision for a business need into software solutions.
- Agile is a term used to describe software development approaches that employ:
  - continual planning
  - Learning
  - improvement,
  - team collaboration
  - evolutionary development
  - and early delivery.
- It encourages flexible responses to change.
- The agile software development emphasizes on four core values.
  - Individual and team interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan

Agile Methodology



Agile Manifesto 12 Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development.
9. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
10. Continuous attention to technical excellence and good design enhances agility.
11. Simplicity–the art of maximizing the amount of work not done–is essential.
12. The best architectures, requirements, and designs emerge from self-organizing teams. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

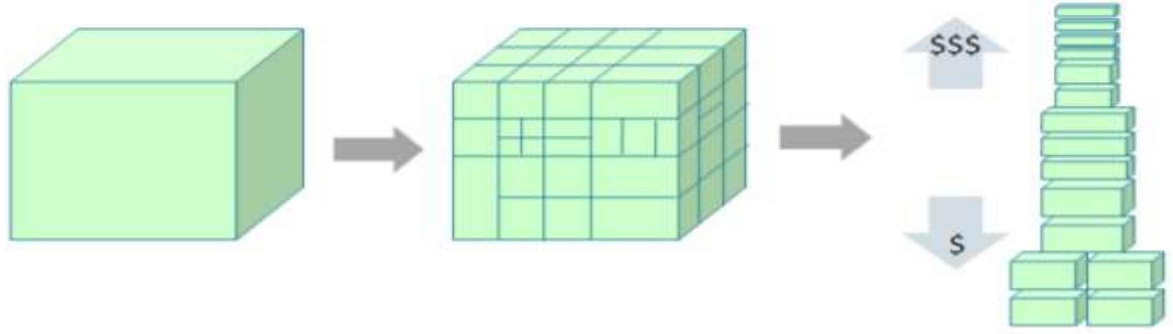
Agile Process

- Scrum
- Crystal Methodologies
- DSDM (Dynamic Software Development Method)
- Feature driven development (FDD)
- Lean software development
- Extreme Programming (XP)

Scrum

- A light-weight agile process tool
- Split your organization into small, cross-functional, self organizing teams.

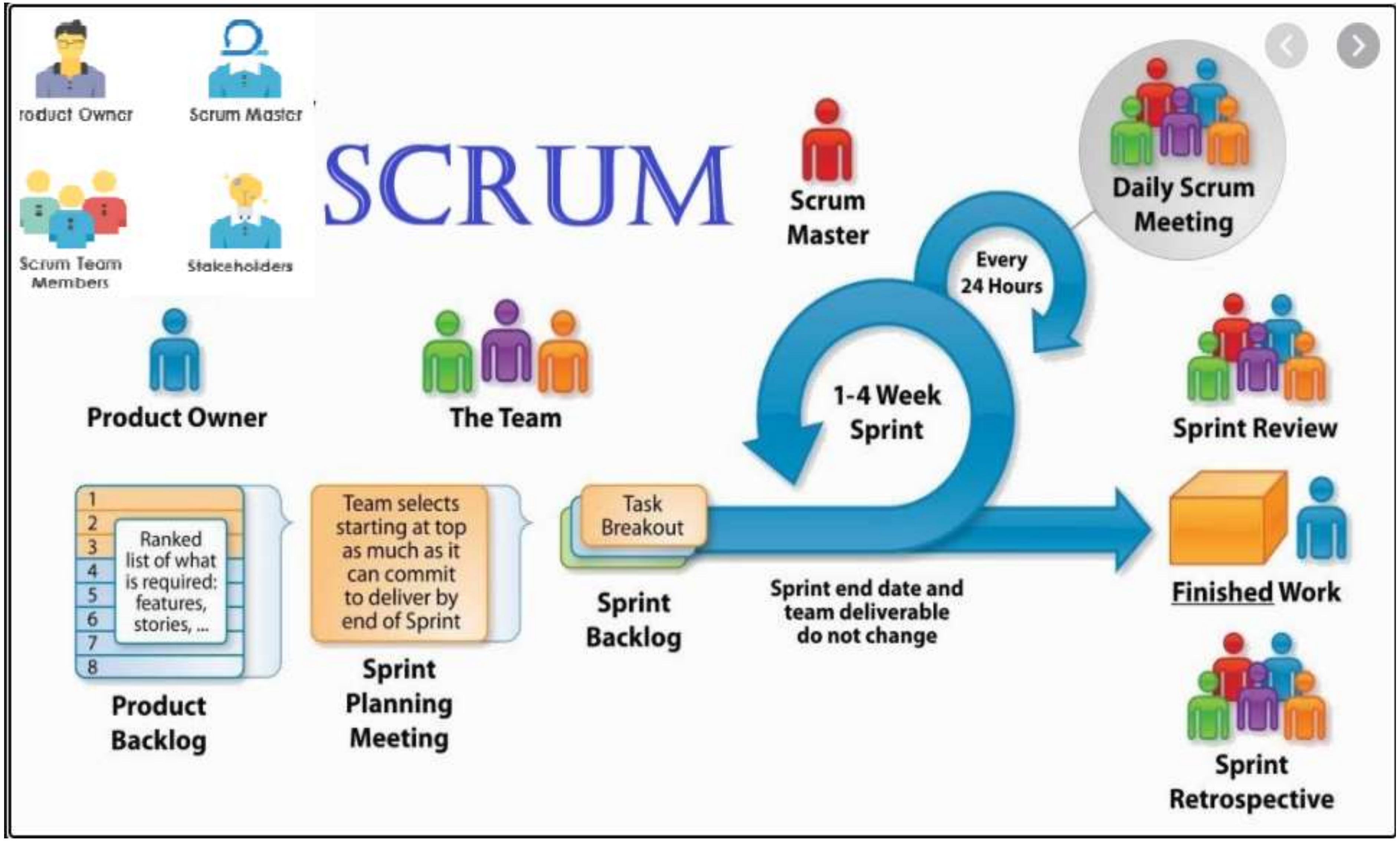
- **Split your work into a list of small, concrete deliverables. Sort the list by priority and estimate the relative effort of each item.**



Scrum

- Split time into short fixed-length iterations/ sprints (usually 2 – 4 weeks), with potentially shippable code demonstrated after each iteration.
- Optimize the release plan and update priorities in collaboration with the customer, based on insights gained by inspecting the release after each iteration.
- Optimize the process by having a

surveying after each iteration.


roduct Owner
Scrum Master
Scrum Team Members
Stakeholders
SCRUM
Scrum Master
Daily Scrum Meeting
Every 24 Hours
Product Owner
The Team
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Team selects starting at top as much as it can commit to deliver by end of Sprint
Task Breakout
Finished Work
Sprint Backlog
Sprint end date and team deliverable do not change
Product Backlog
Sprint Planning Meeting
Sprint Retrospective

ANY QUERIES