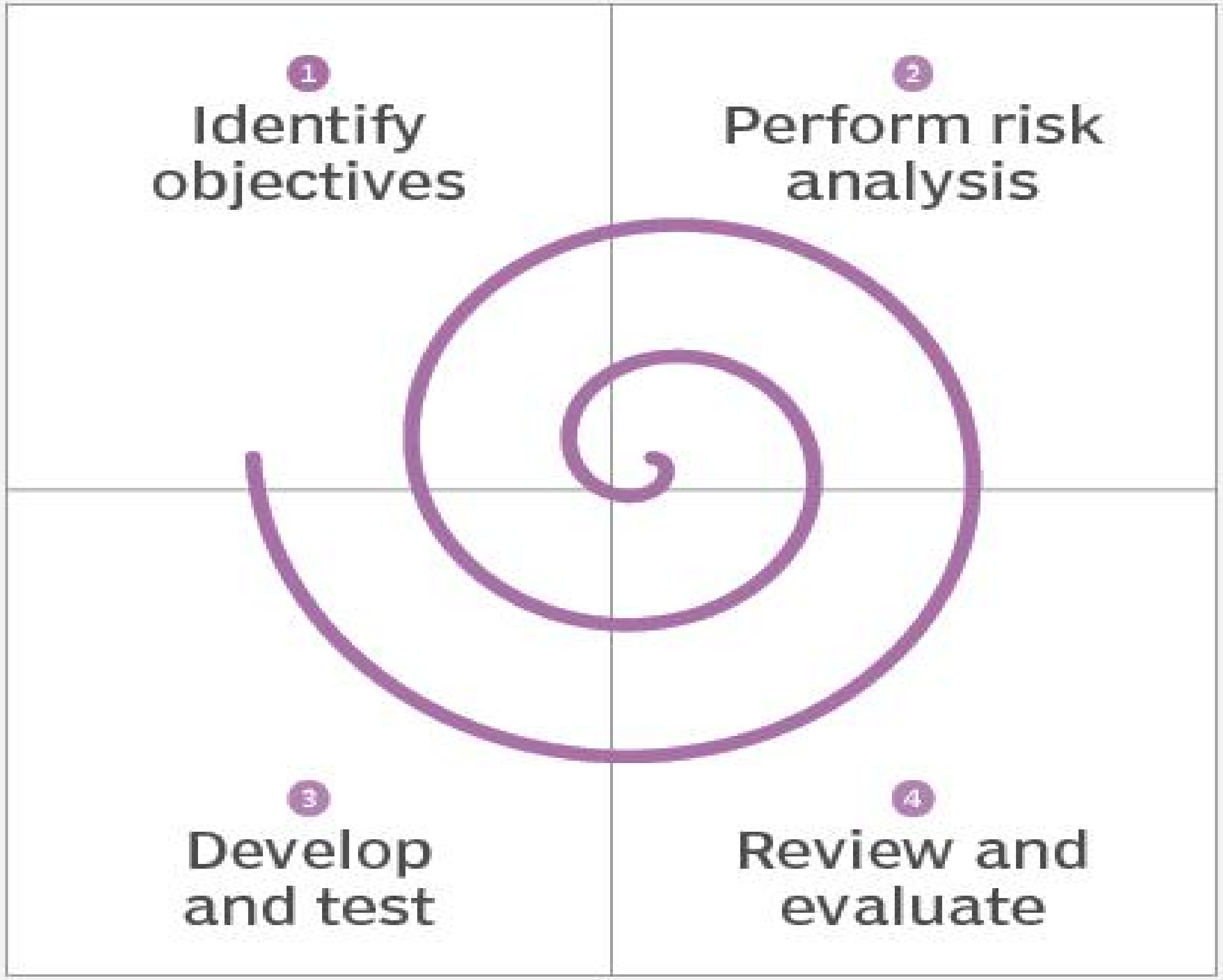
SDLC Models-Types

Spiral model



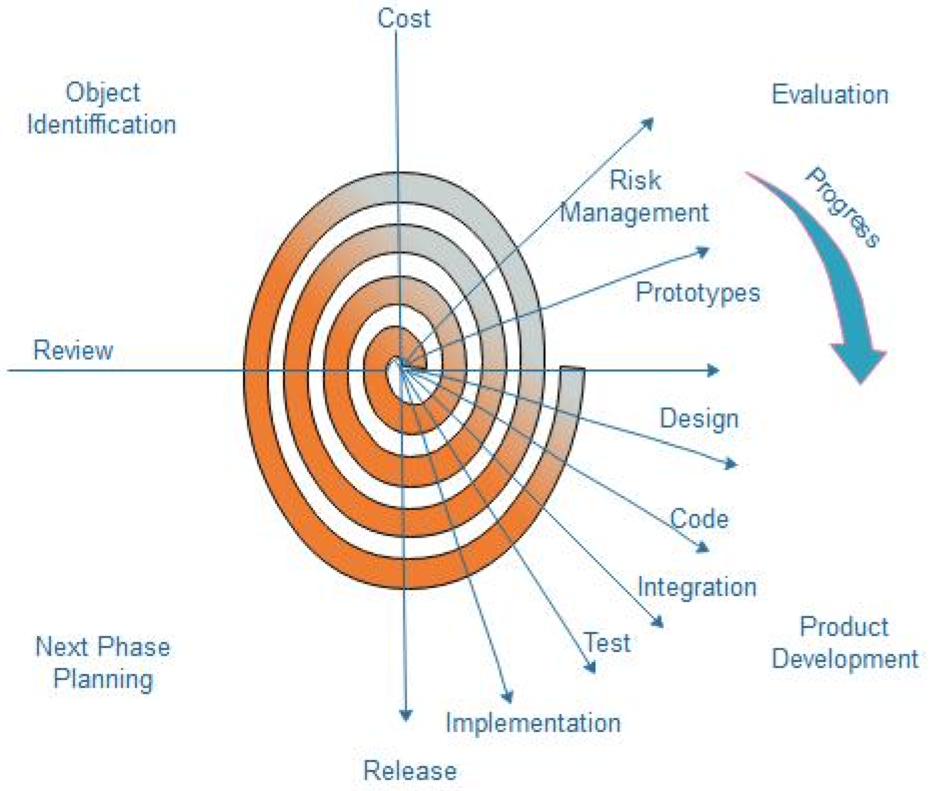


Fig. Spiral Model

Spiral Model

- Waterfall takes long duration to complete product. Spiral model application is created module by module and handed to customer early. Additional requirements may be obtained from client between process.

- Each phase has four quadrants:
- overall goal of the phase should be determined and all objectives should be elaborated and analyzed. It is important to also identify alternative solutions in case the attempted version fails to perform.

- risk analysis should be performed on all possible solutions in order to find any faults or vulnerabilities, such as running over the budget or areas within the software that could be open to cyber attacks. Each risk should then be resolved using the most efficient strategy.

Spiral Model

- prototype is built and tested. This step includes: architectural design, design of modules, physical product design and the final design.

fourth quadrant, the test results of the newest version are evaluated.

- planning for the next phase begins and the cycle repeats.

- Applications:
- deliverance is required to be frequent.
- requirements are unclear and complex
- changes may require at any time.
- Risk management

Spiral Model

- **Advantages:**
- Flexibility - Changes made to the requirements after development has started can be easily adopted and incorporated.
- Risk handling - The spiral model involves risk analysis and handling in every phase
- Customer satisfaction. evaluate their product in every phase.
- Good for large and mission-critical projects.
- More clarity for developers and testers.

- **Limitations:**
- High cost- expensive, not suitable for small projects
- Dependence of risk analysis needs expertise
- Complexity. Protocols to be followed to operate efficiently.
- Hard to manage time.

Agile Model

- Break tasks into smaller iterations, or parts do not directly involve long term planning, helps to minimize the project risk and to reduce the overall project delivery time requirements.

- project scope and requirements are laid down at the beginning of the development process.

- entire project into smaller parts **Requirements gathering**: define the requirements. explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

- **Design the requirements**: Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

Agile Model

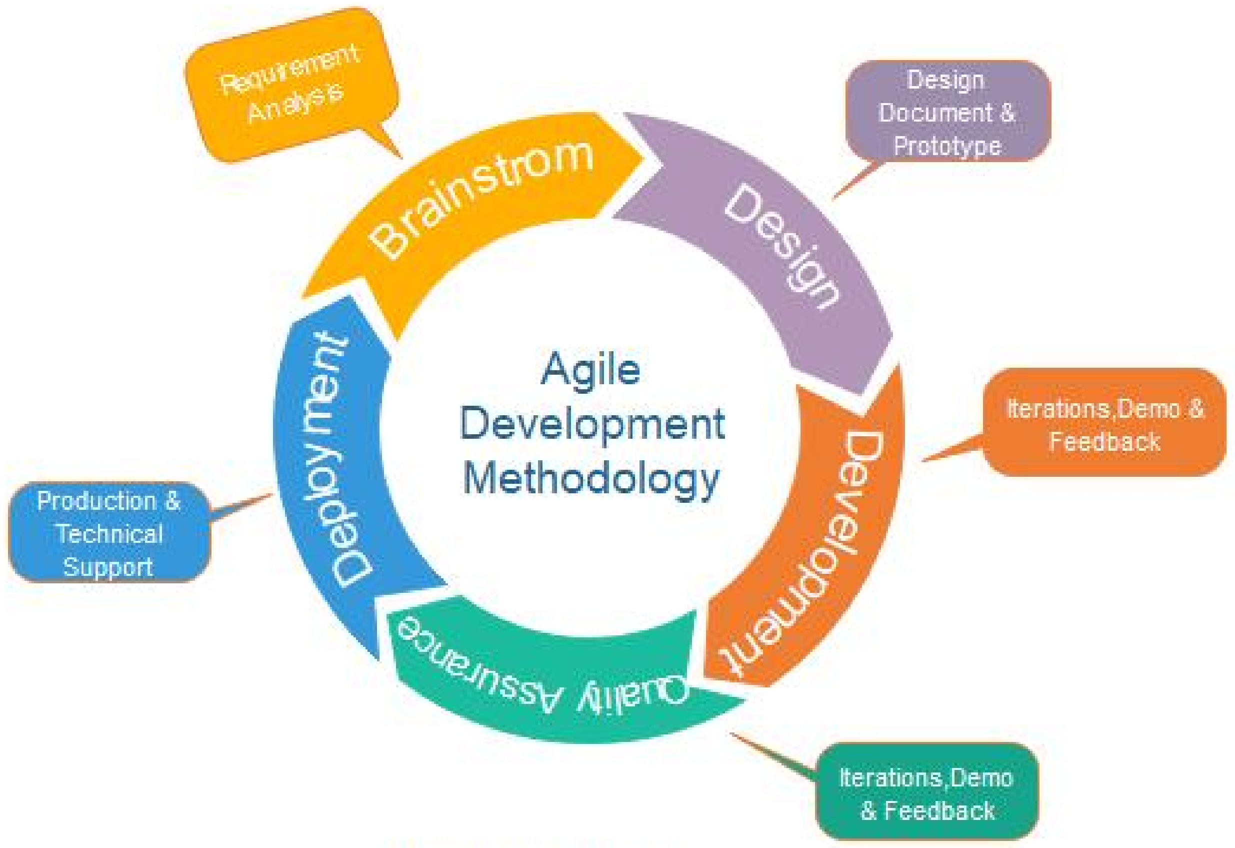

Requirement Analysis
Brainstrom
Design
Design Document & Prototype
Agile Development Methodology
Development
Iterations,Demo & Feedback
Quality Assurance
Iterations,Demo & Feedback
Deployment
Production & Technical Support

Fig. Agile Model

Agile Model

- **Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.
- **Deployment:** In this phase, the team issues a product for the user's work environment.
- **Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.
- **Applications:**
- When a highly qualified and experienced team is available.
- Continuous interaction of client with the software team all the time.
- project size is small.

Agile Model

- Advantages:
- Face-to-Face Communication with clients.
- Efficient design and fulfils the business requirement.
- Anytime changes are acceptable.
- It reduces total development time.

- Disadvantages:
- creates confusion and crucial decisions taken throughout various phases
- maintenance of the finished project can become a difficulty

V-model

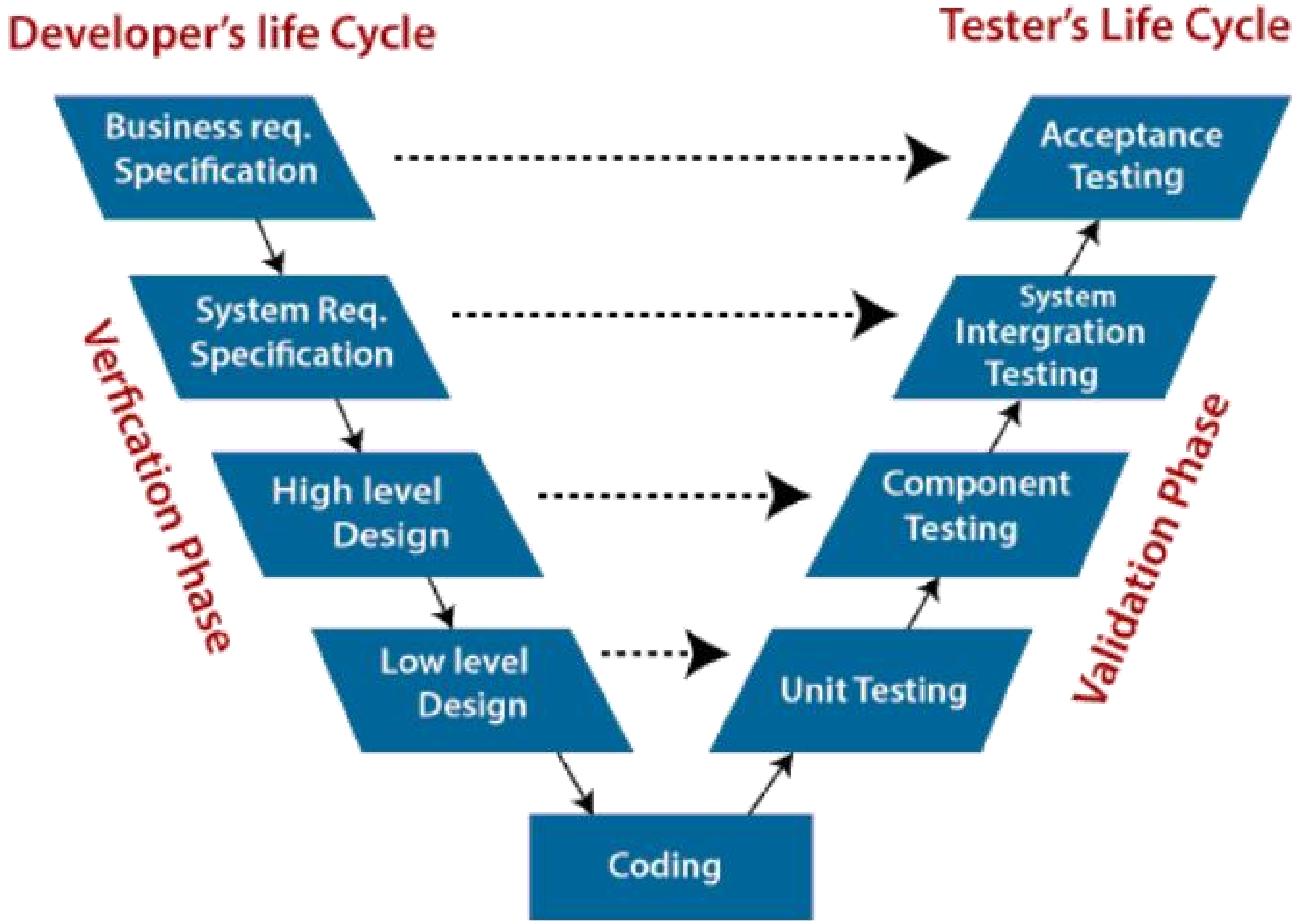
- **Verification and Validation model**.execution of processes happens in a sequential manner in a V-shape.

- Similar to water fall model

- Testing phase and development phase to be parallel.
- **Business requirement analysis:** This phase contains detailed communication to understand customer's expectations and exact requirements.

- **Applications:**
- Requirements well defined, and documented
- Technology not dynamic and understood by project team.

V- Model


Developer's life Cycle
Tester's Life Cycle
Business req. Specification
Acceptance Testing
System Req. Specification
System Intergration Testing
High level Design
Component Testing
Low level Design
Unit Testing
Coding
Verfication Phase
Validation Phase

V-model

- Unit Testing: testing at code level to eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

- Integration Testing: Integration tests are performed to test the coexistence and communication of the internal modules within the system.

- System Testing: Check the entire system functionality and the communication of the system with external systems. Most of the software and hardware compatibility issues can be covered during this system test execution.

- Acceptance Testing: associated with the business requirement analysis phase and involves testing the product in user environment. covers load and performance defects in the actual user environment.