Computer Fundamentals

Software Engineering and SDLC

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

Learning Objectives

By the end of this lesson, you will be able to:

- Elaborate Software development models
- Comprehend Software evolution
- Analyze Software Development Paradigm
- Apply Software Development life cycle

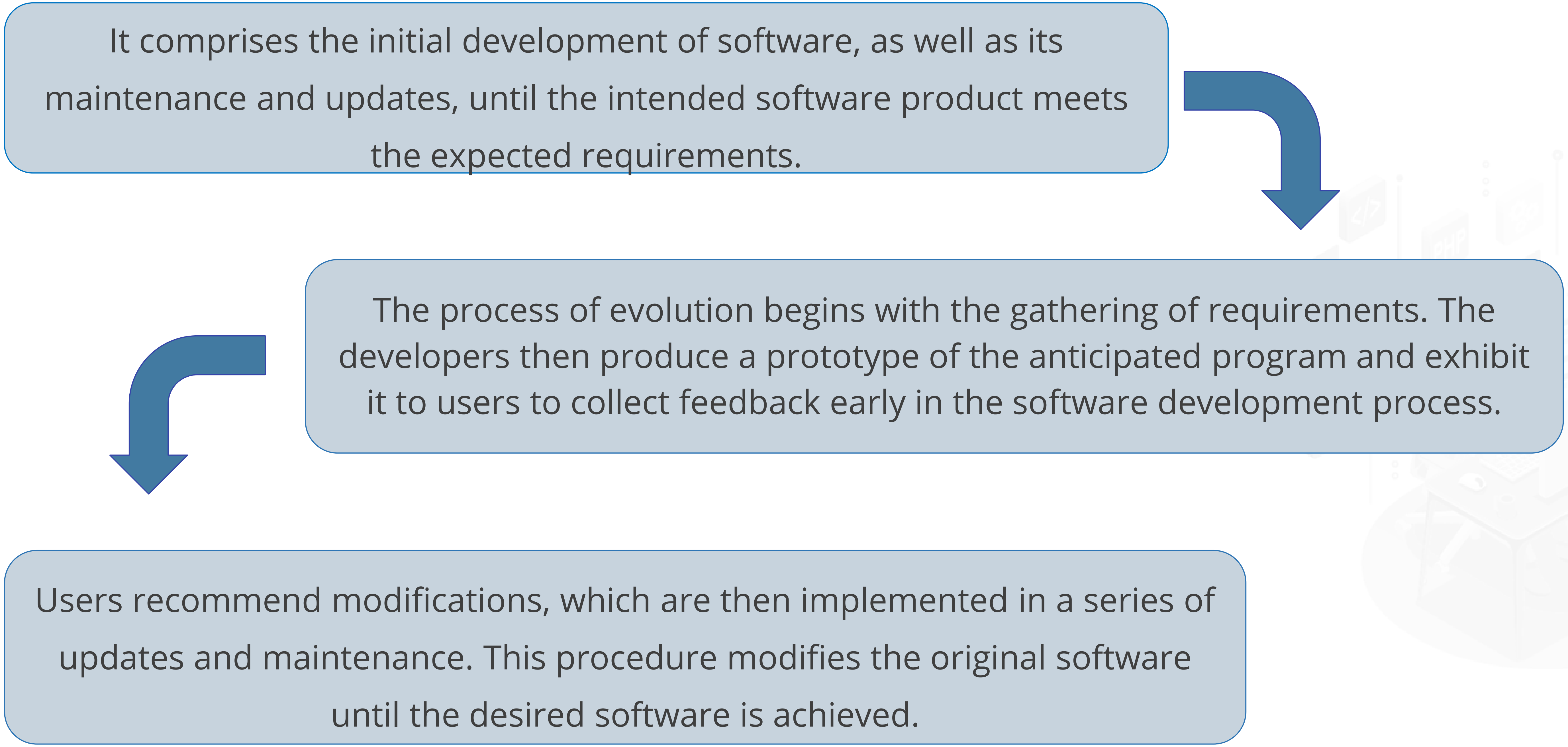
Software Evolution

Software Evolution

Software evolution, also known as the s/w engineering paradigm, is the process of creating a software product utilising software engineering ideas and methods.

The process of evolution begins with the gathering of requirements. The developers then produce a prototype of the anticipated programme and exhibit it to users to collect feedback early in the software development process.

Software Evolution



It comprises the initial development of software, as well as its maintenance and updates, until the intended software product meets the expected requirements.

The process of evolution begins with the gathering of requirements. The developers then produce a prototype of the anticipated program and exhibit it to users to collect feedback early in the software development process.

Users recommend modifications, which are then implemented in a series of updates and maintenance. This procedure modifies the original software until the desired software is achieved.

FULL STACK

Software Development Paradigm

Software Development Paradigms

The Paradigm is known as software engineering paradigms where all the engineering concepts pertaining to the development of software are applied. It includes various researches and requirement gathering which helps the software product to build.

Software Development Paradigms

It consists of :

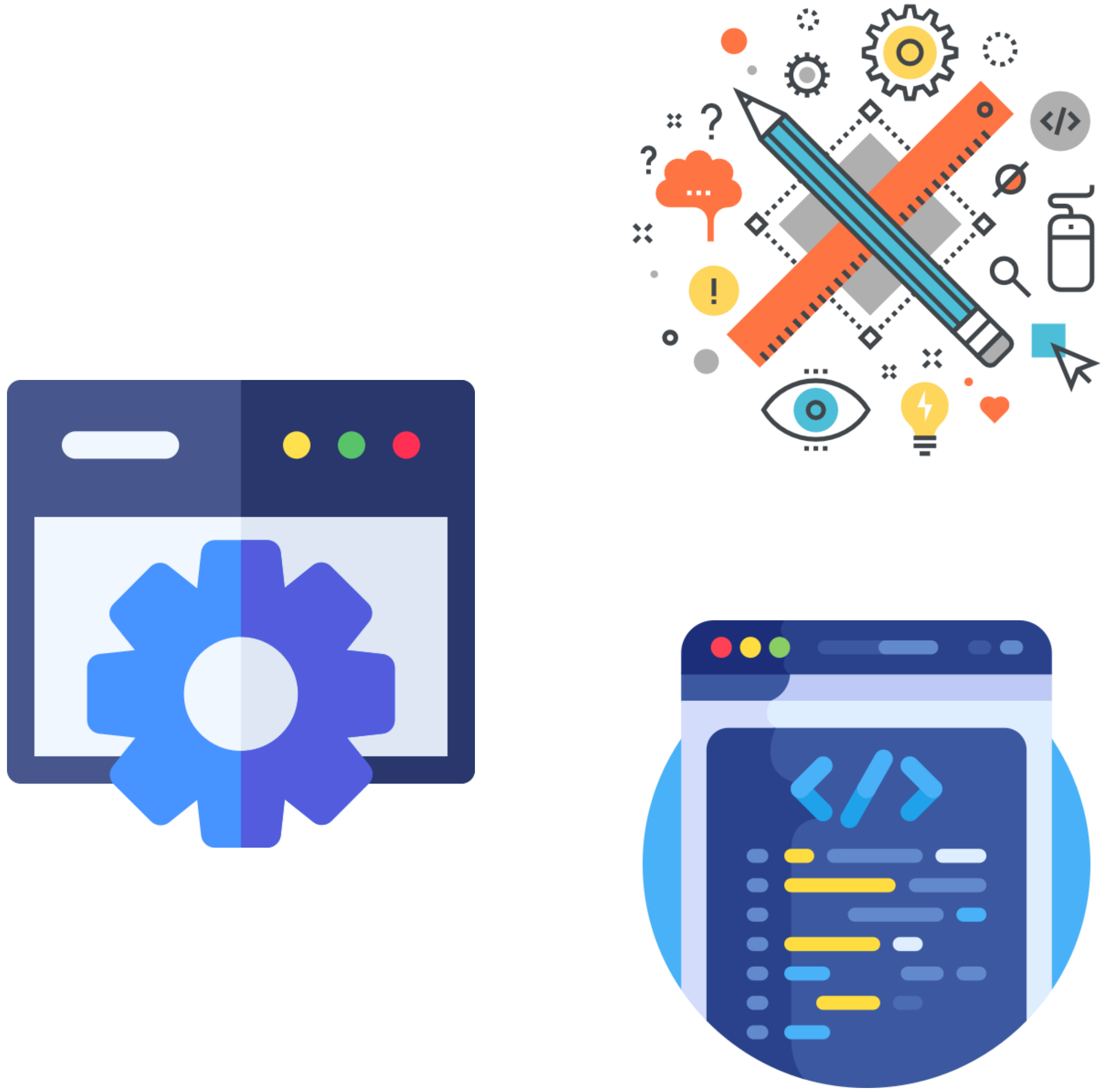
Requirement gathering

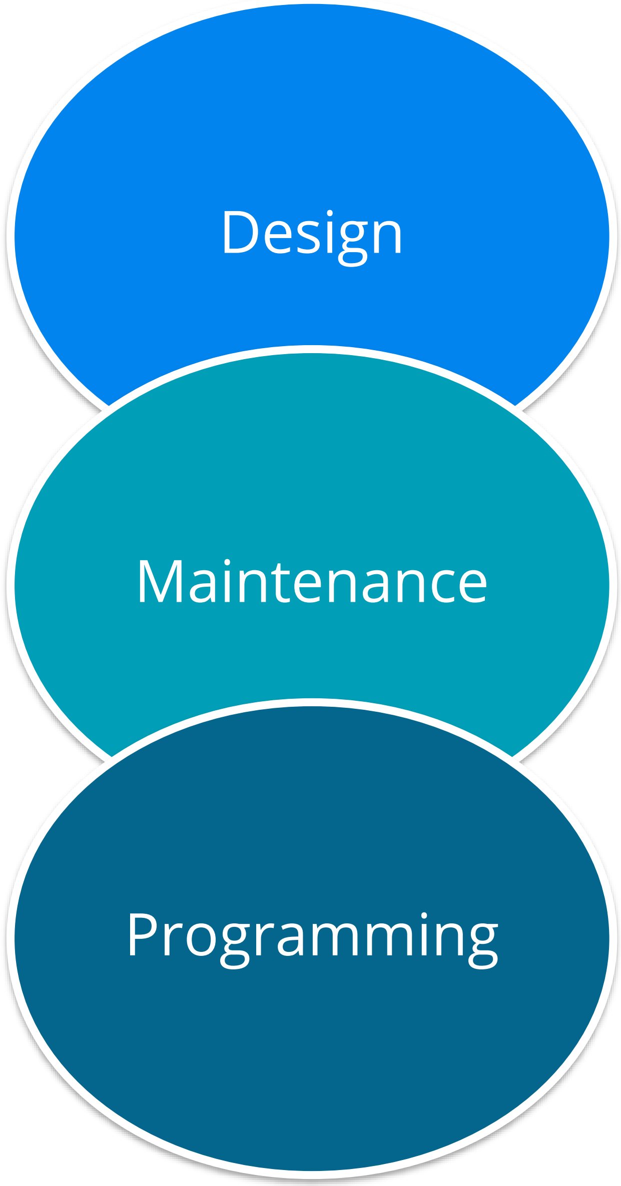
Software design

Programming

Software Design Paradigms

This paradigm is a part of Software Development and includes:





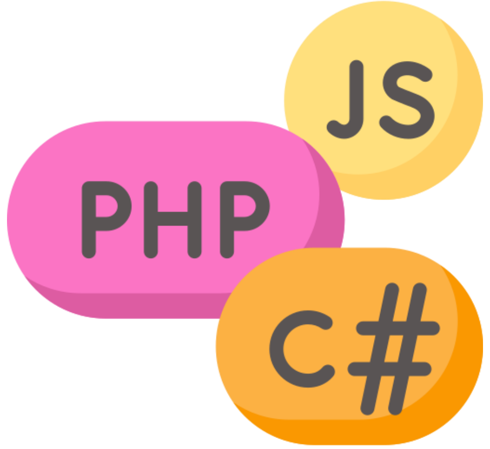
Design

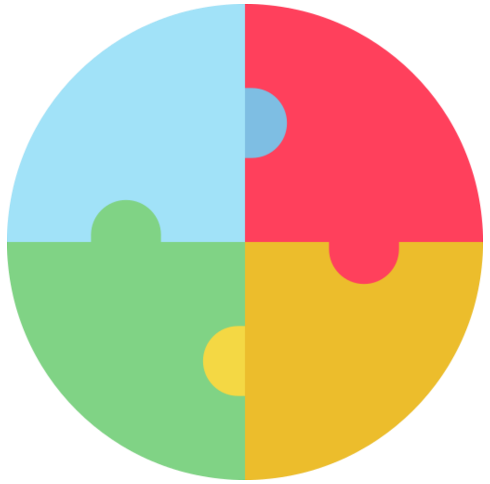
Maintenance

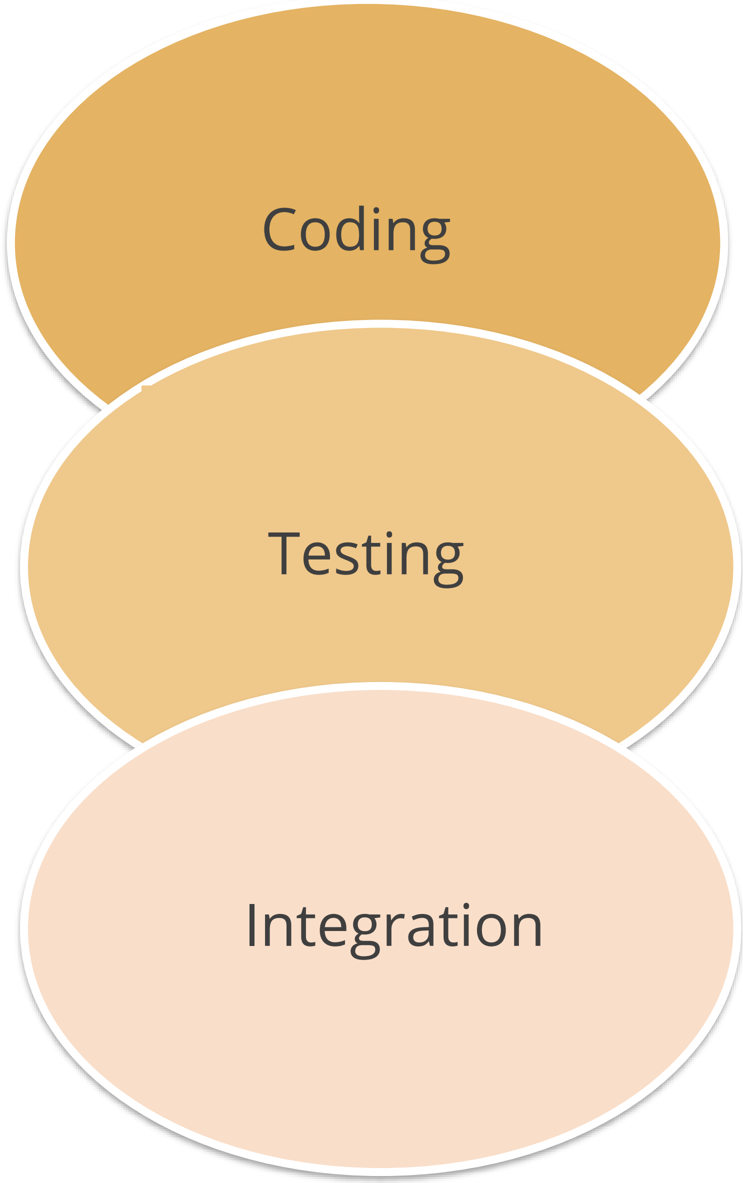
Programming

Programming Paradigms

This paradigm is related closely to programming aspect of software development. It also includes:







Coding

Testing

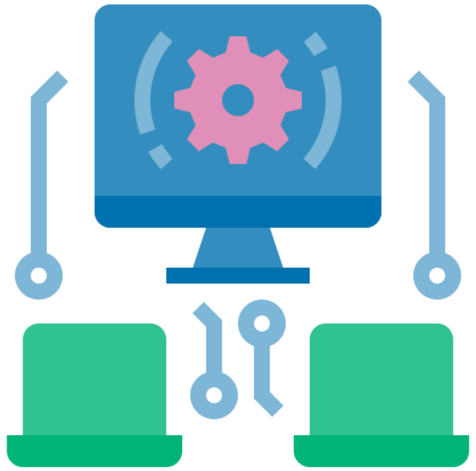
Integration

FULL STACK

Software Development Life Cycle

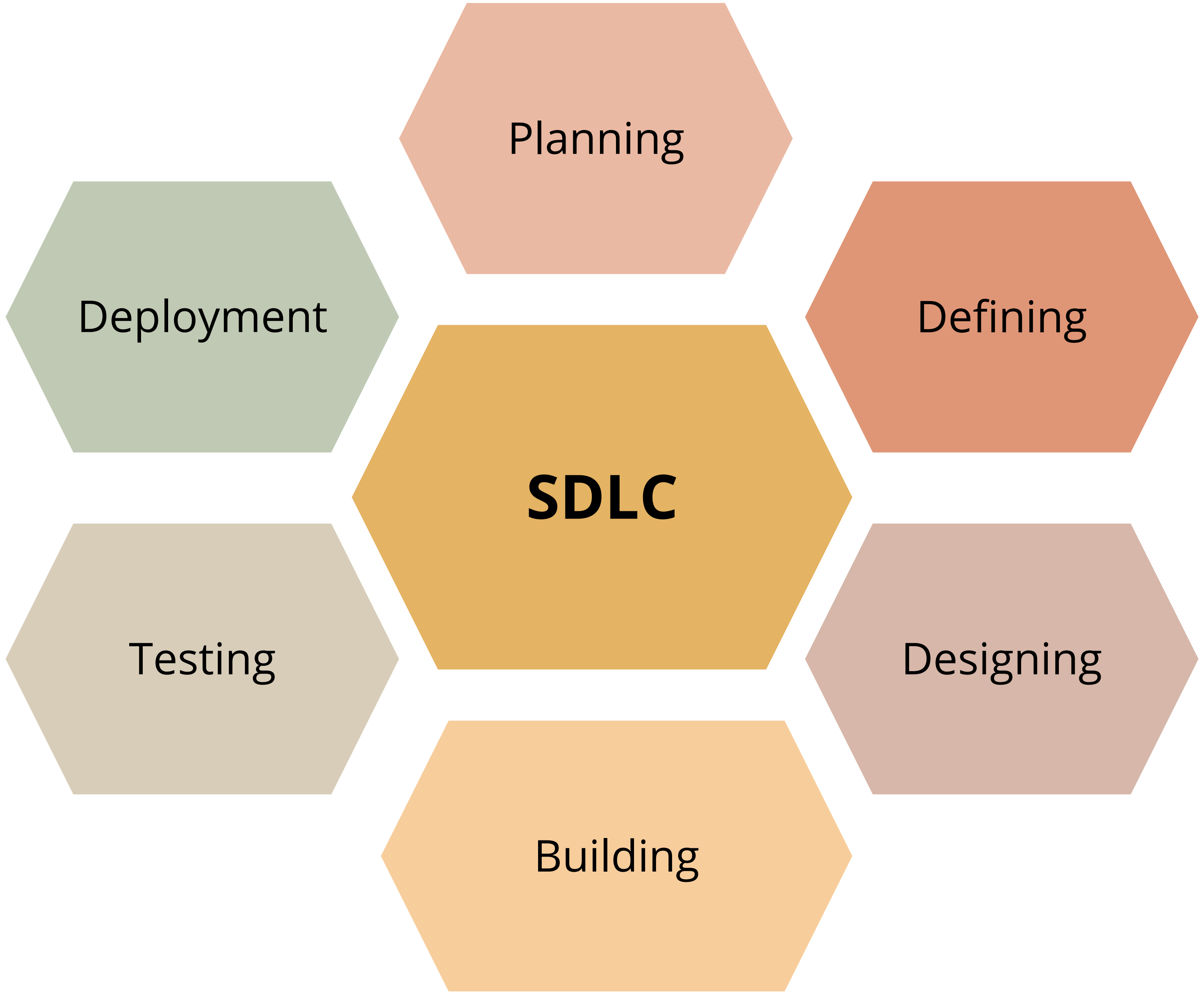
Software Development Life Cycle

The purpose of the Software Development Life Cycle (SDLC) is to generate a large system that satisfies customer expectations, performs effectively and efficiently in current and projected information technology infrastructure, and is economical to maintain and enhance.



Software Development Paradigms

The process of software development are:



Software Development Paradigms

Planning:

- The first duration of a development project is included in software development planning.
- Analysis and planning, ideation, design and development, deployment, production and launch, and maintenance are all included in the plan.

Defining:

It defines the logical framework of each module as well as the interface used to connect with it.

Software Development Paradigms

Designing:

- Design decisions depend on the requirements list you created during the definition phase.
- One or more designs are generated during the design process to accomplish the project's goal.

Implementing:

User notification, training, hardware installation, software installation on production computers, and system integration into regular work activities are all part of the implementation process.

Software Development Paradigms

Testing:

In the software development lifecycle (SDLC), the testing phase is where you focus on research and discovery. During the testing phase, programmers check to see if their code and programming meet the needs of the customer.

Deployment:

The software development life cycle (SDLC) closes with the deployment phase, which puts the product into business.

Software Development Life Cycle

Some of the reasons why SDLC is important in Software Development are:

It provides visibility of a project plan to all the involved stakeholders.

It helps us to avoid project risks.

It allows us to track and control the project.

It doesn’t conclude until all the requirements have been achieved.

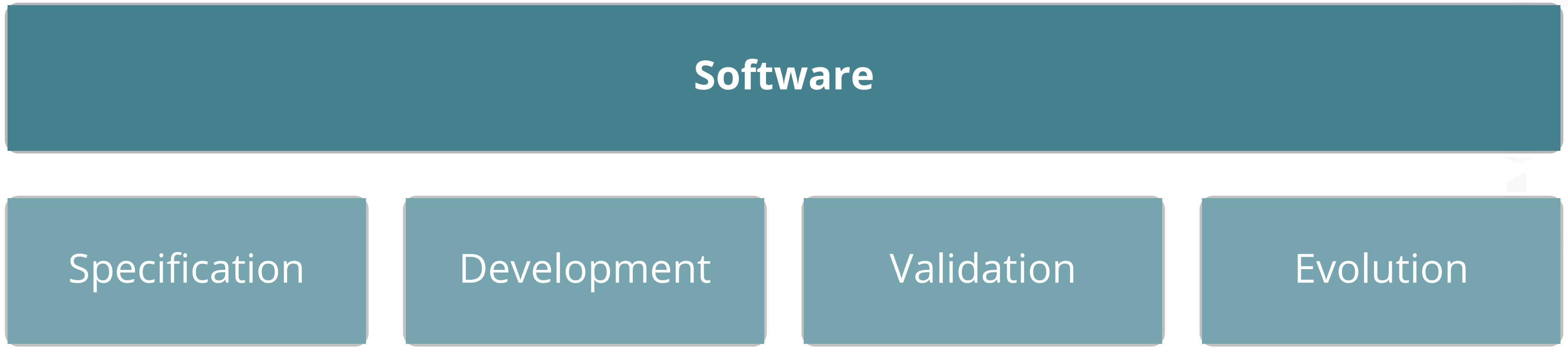
Software Development Models

Software Development

A set of activities for specifying, designing, implementing and testing software systems for associated outcome that produce a software product.

Software Development

These are four key process activities, which are common to all software processes and the activities are:



Software

Specification

Development

Validation

Evolution

Software Development Model

The software development process is based on the SDLC model.

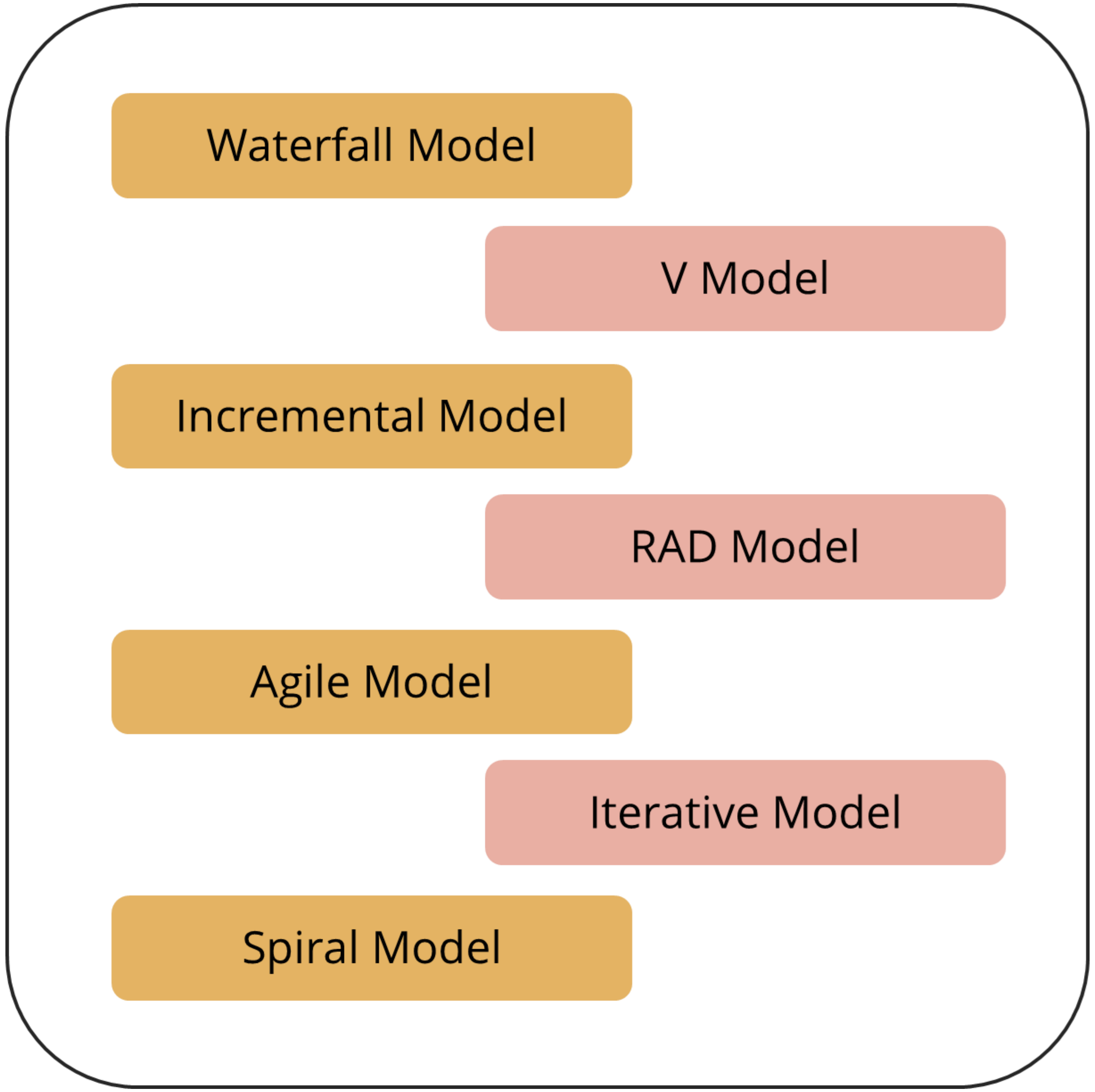
SDLC is a pictorial and diagrammatic representation of the software life cycle.

The models specify the various stages of the process in which they are carried out.

There are multiple SDLC models. They are divided into main groups, each with its features and weaknesses.

Software Development Model

The most used, popular and important SDLC models are given below:



Waterfall Model

V Model

Incremental Model

RAD Model

Agile Model

Iterative Model

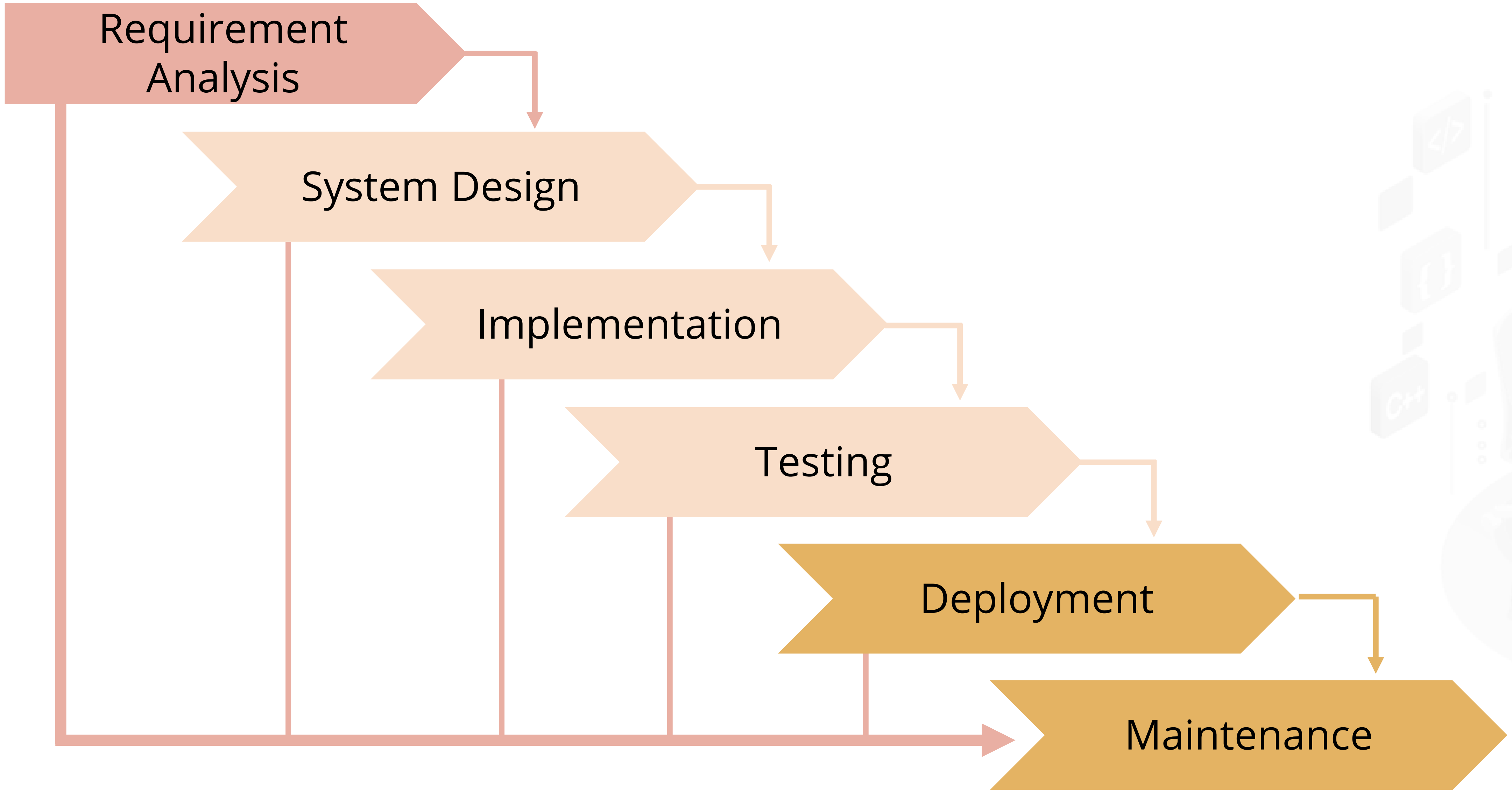
Spiral Model

Waterfall Model

- The waterfall is a cascade SDLC model that includes gradual executions of phases.
- The waterfall model is a breakdown of project activities into linear sequential phases.
- Each phase depends on the deliverables of the previous phase. The approach is typical for certain areas of engineering design.
- This process is documented and predefined with features expected to every phase.

Waterfall Model

The different steps of the Waterfall model are:



Waterfall Model

The use cases are:

Small to mid-sized projects with well-defined, recurring needs (small company website development)

Projects that require a greater level of control, and a more predictable budget and timeframe (e.g., governmental projects)

Healthcare projects where multiple rules and procedures must be maintained when working on a task

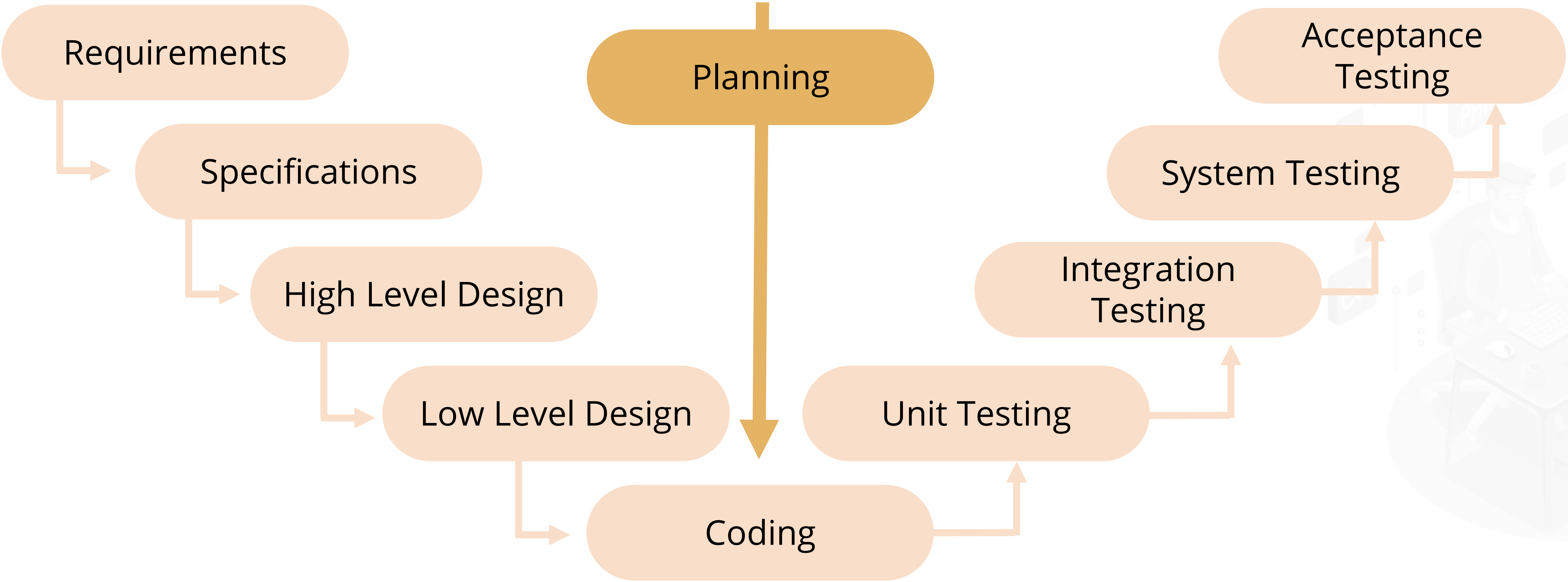
Projects that make use of some good software platform and tools

V Model

- V-shaped SDLC model is an expansion of classic waterfall model.
- This model is based on associated **test stage** for every development stage.
- This is a very strict model, and the next stage is started only after the previous phase.
- This is also called **“Validation and verification”** model.
- The **horizontal axes** represent time or project completeness (left-to-right).
- The **vertical axes** represent level of abstraction (coarsest-grain abstraction uppermost).

V Model

The different steps of the V model are:



Requirements

Specifications

High Level Design

Low Level Design

Coding

Planning

Unit Testing

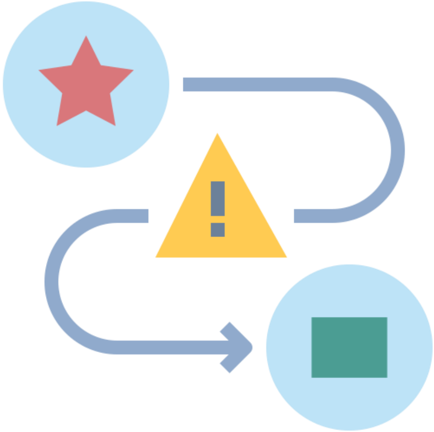
Integration Testing

System Testing

Acceptance Testing

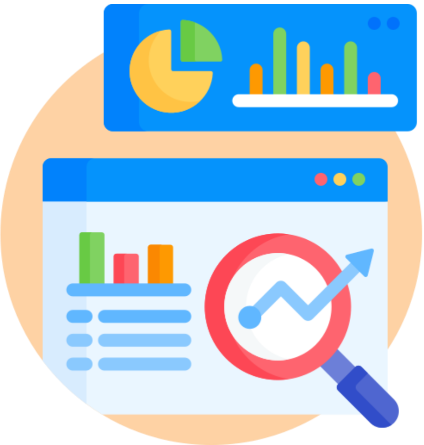
V Model

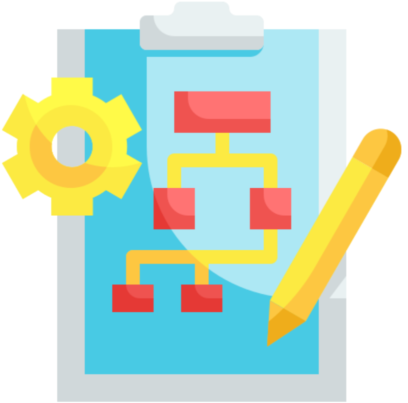
V models are used in projects where downtime and failures are unacceptably high. For example, medical software and aviation fleet management software.



Incremental Model

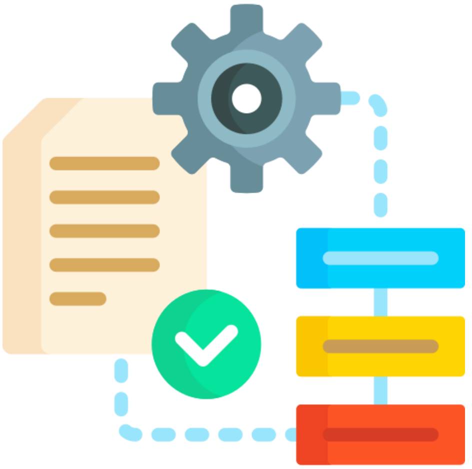
The incremental model is a software development process that distributes requirements into multiple stand-alone modules during the software development cycle.





- Analysis, implementation, testing, and maintenance are the steps in this model.
- Each iteration goes through the processes of requirements, design, coding, and testing.
- It includes both development and upkeep.

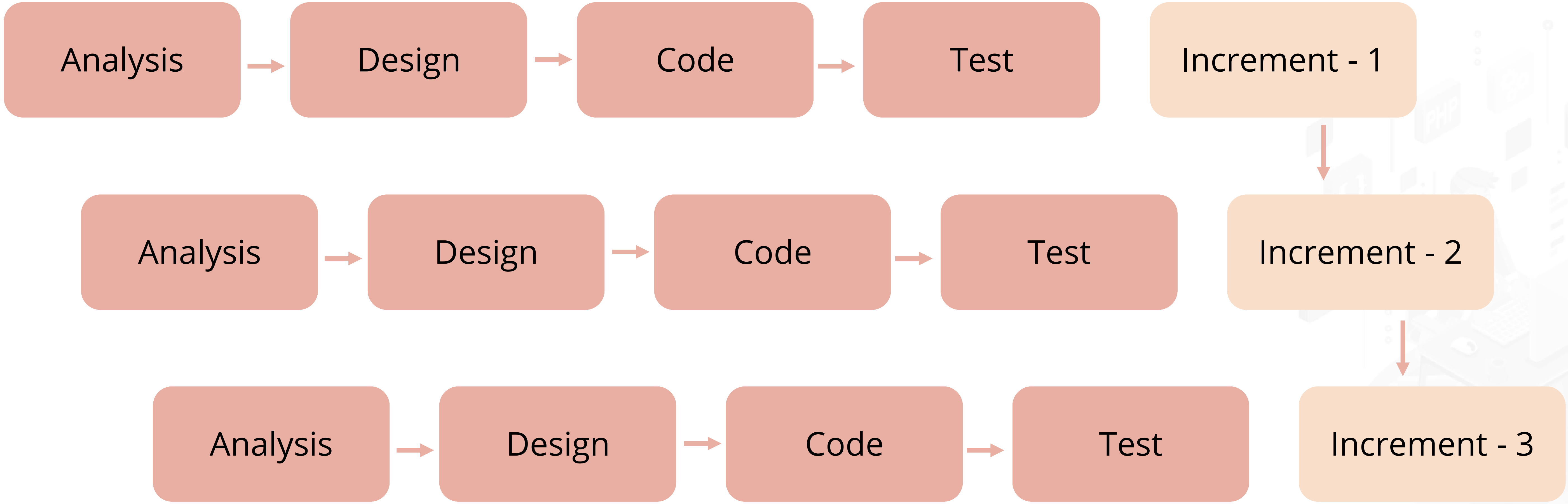
Incremental Model



- When a product meets all its specifications, it is said to be finished.
- This approach combines parts of the waterfall model with prototyping iteration.
- When the first increment is achieved, the system is put into production.
- The development of a system is split down into several smaller tasks. The most important criterion is addressed first.

Incremental Model

The different steps of the Incremental model are:



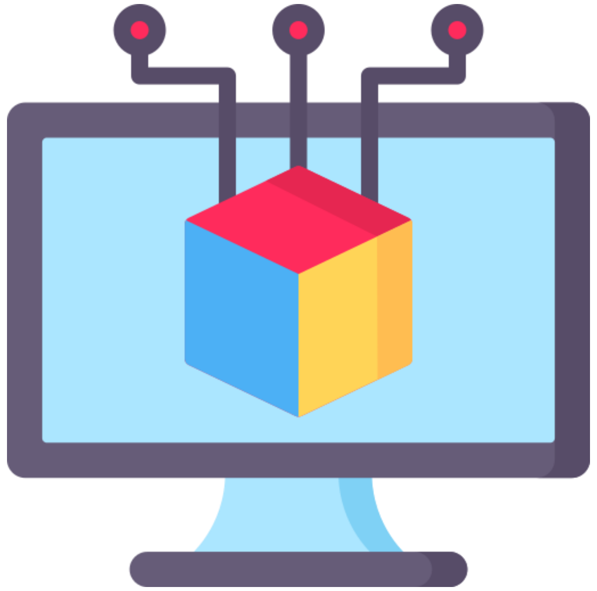
Incremental Model

The use case is:

Large, mission-critical enterprise programs made from loosely linked components like microservices or web services

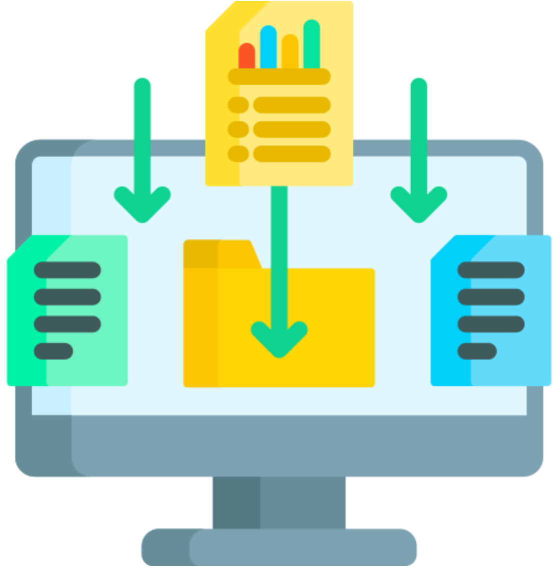
RAD Model

Adaptive software development is another term for rapid application development. The incremental prototyping technique to software development is known as RAD.



When reviewing a live system, end users can provide better input here. It places a greater emphasis on adaptability rather than planning.

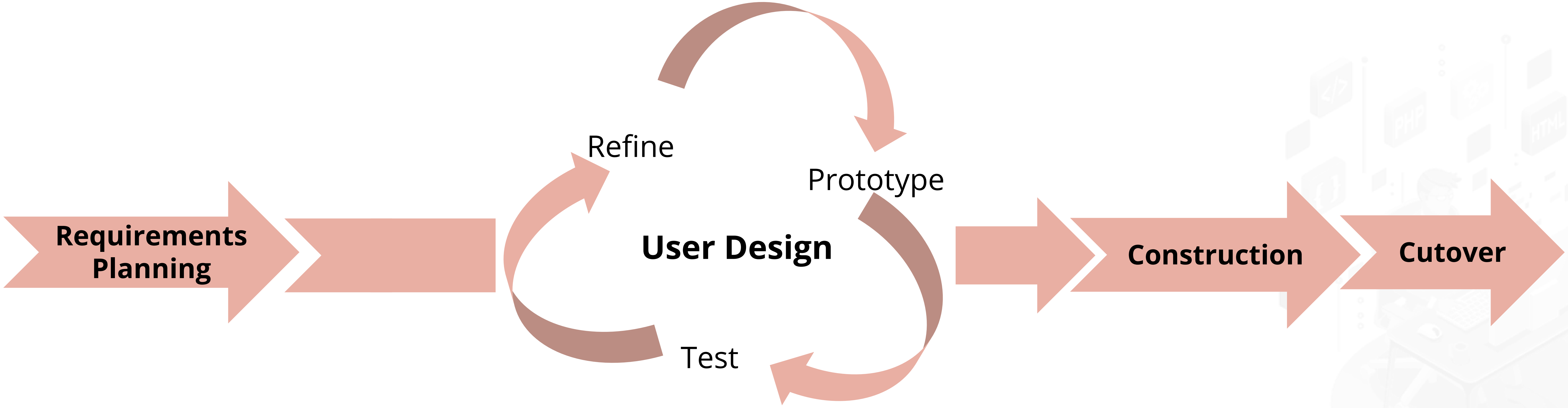
RAD Model



- RAD is a concept that states that by gathering needs through workshops or focus groups, goods may be built faster and with higher quality.
- It involves design prototyping and early iterative user testing.
- Software component reuse is a term used to describe the process of reusing software components.
- In evaluations and other team interactions, there is less formality.

RAD Model

The different steps of the RAD model are:



RAD Model

The use cases are:

RAD should only be utilized when a system can be modularized and deployed incrementally.

It is being used if there are many designers available for modeling.

It should only be utilized if the budget does not allow for the usage of automated code generation technologies.

RAD Model

The use cases are:

It should be utilized when project requirements change and workable prototypes are provided to customers in small iterations of 2-3 months.

RAD SDLC model should only be used if domain specialists with appropriate business knowledge are available.

Agile Model

A software development approach based on iterative development is referred to as "agile process model."

Agile methodologies separate projects into smaller iterations, or sections, and avoid long-term planning.

Agile Model

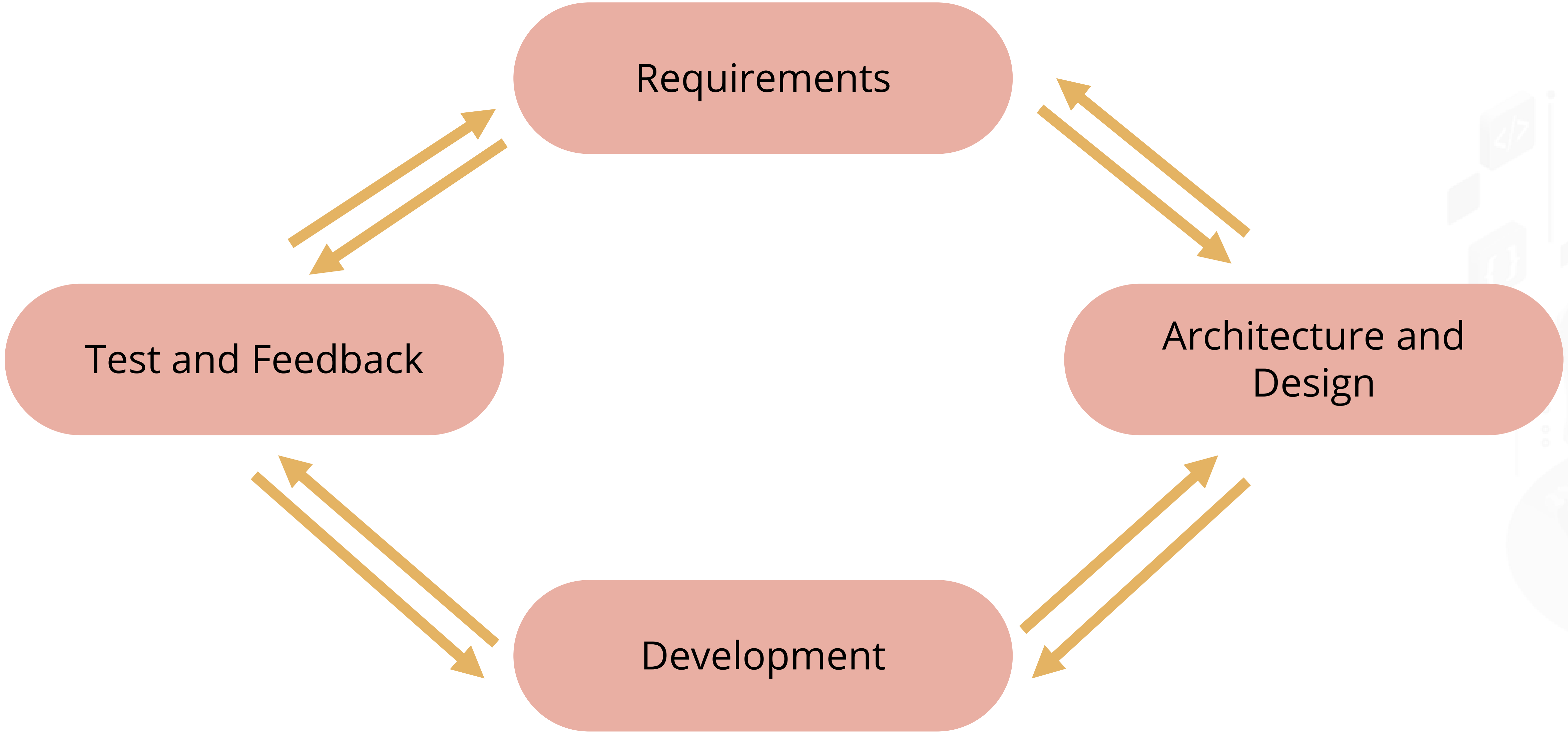
- Customers can view the result of each development iteration in the Agile methodology and decide if they are satisfied or not.
- The scrum methodology is one of the Agile model's practical applications.
- The alternatives to traditional project management are suggested by the Agile movement.

Agile Model

- Agile software development is often carried out in short, quick cycles.
- With each release, the frequency of incremental releases increases.
- To guarantee that software quality is maintained, extensive testing is performed.
- Many frameworks, such as Scrum, Kanban, Lean, and Extreme Programming (XP), can help organizations become more agile.

Agile Model

The different steps of the Agile model are:



Requirements

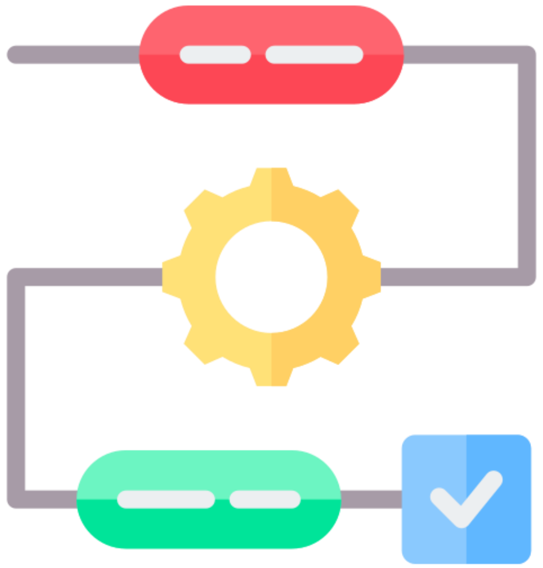
Architecture and Design

Development

Test and Feedback

Iterative Model

The Iterative Model allows the team to go back in time and visit prior phases where the changes were made. When the Software Development Life Cycle (SDLC) procedure is completed, the project's final result is renewed.



Iterative Model

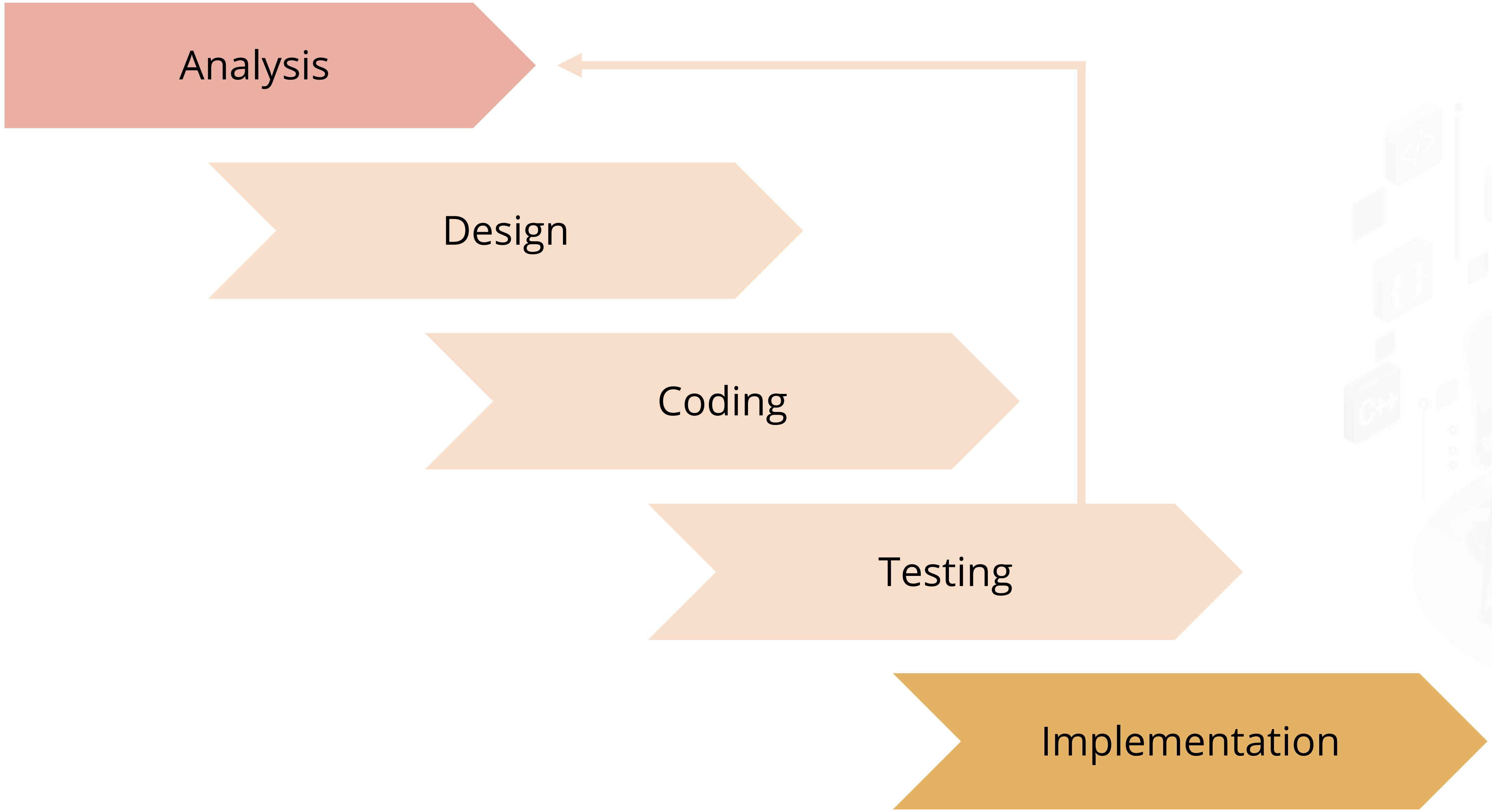
- The iterative model does not need a full list of requirements before starting a project.
- The development process may start with the requirements of the functional part.
- The process is repetitive, allowing it to make new versions of the product for every cycle.

Iterative Model

- Every iteration includes the development of a separate component of the system.
- After that, this component is added to the functional part developed earlier.
- However, in an iterative project, a formal software specification may also be required.

Iterative Model

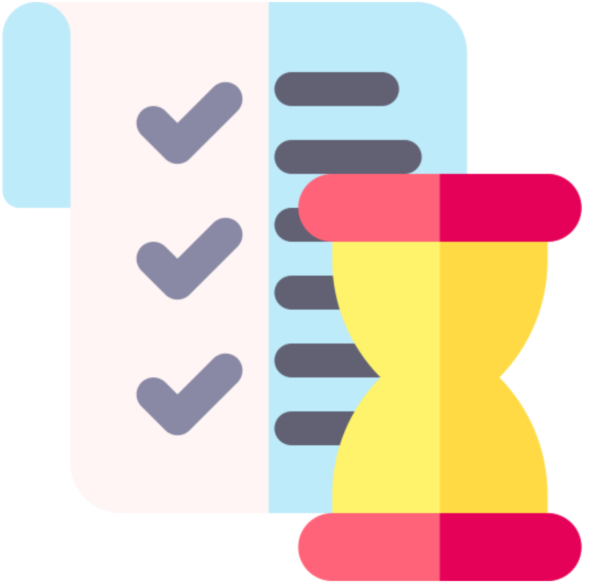
The different steps of the Iterative model are:



Spiral Model

The spiral model is a stage-based SDLC approach that integrates architecture and prototyping. It is a hybrid of the Iterative and Waterfall SDLC models, with a strong focus on risk assessment.

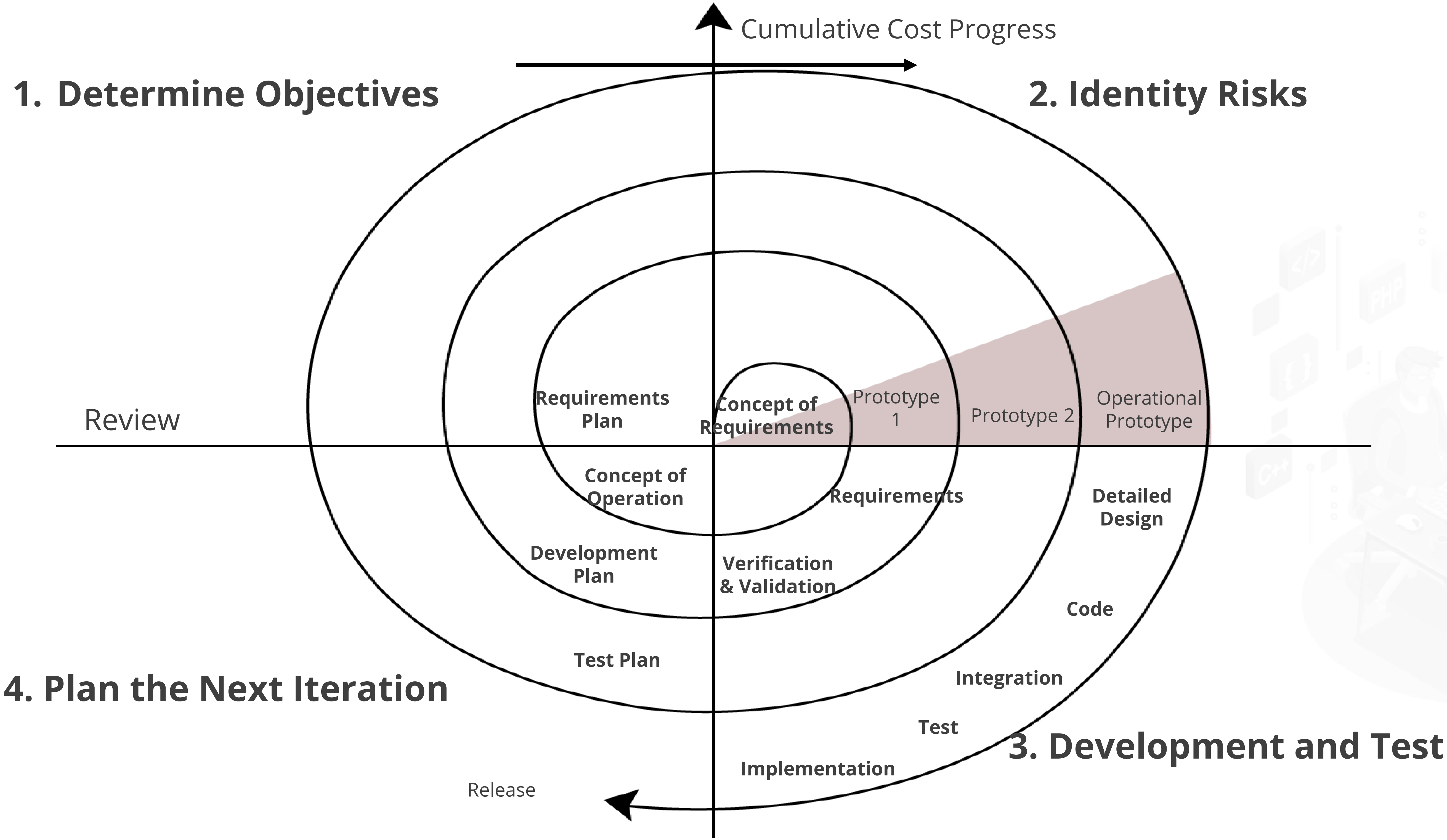
Spiral Model



Process:

- In this model, the proper time to progress to the next stage must be determined.
- The solution to this problem is to use the pre-determined time frames.
- The transition to the next level takes place as planned.
- Even if the earlier stage's work is still not completed.
- The strategy is based on statistical data gathered from past projects.

Spiral Model



Key Takeaways

- Software development is a set of activities for specifying, designing, implementing, and testing software systems.
- SDLC models are Waterfall model, V model, Incremental model, RAD model, Agile model, Iterative model, and Spiral model
- The Paradigm is known as software engineering paradigms where all the engineering concepts pertaining to the development of software are applied.
- Software Development life cycle aims to produce a high-quality system that meets or exceeds customer expectations, works effectively in the current and planned information technology infrastructure.

Thank you