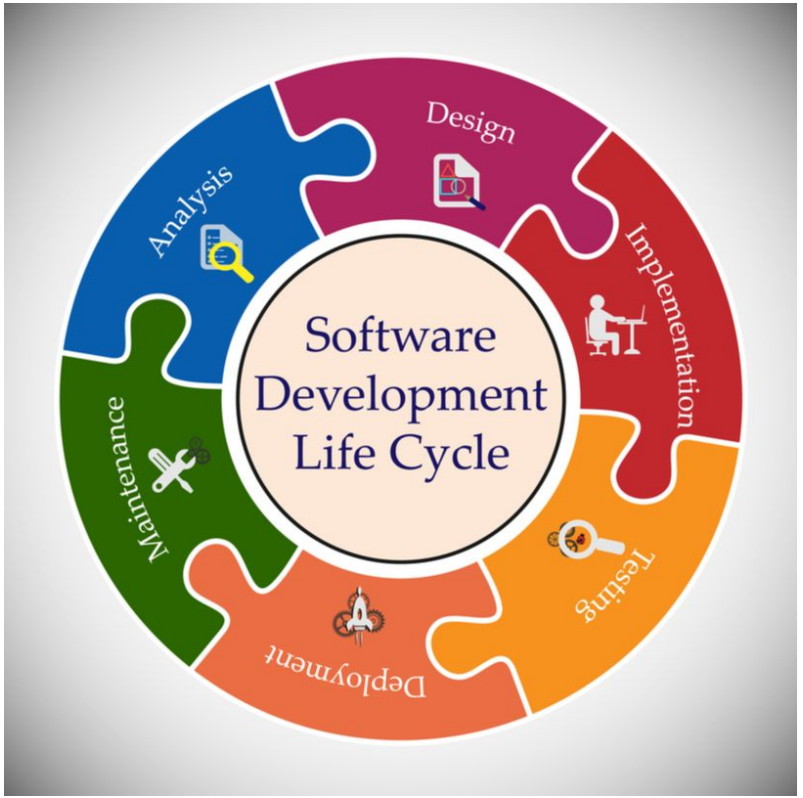
SDLC
Session - 1

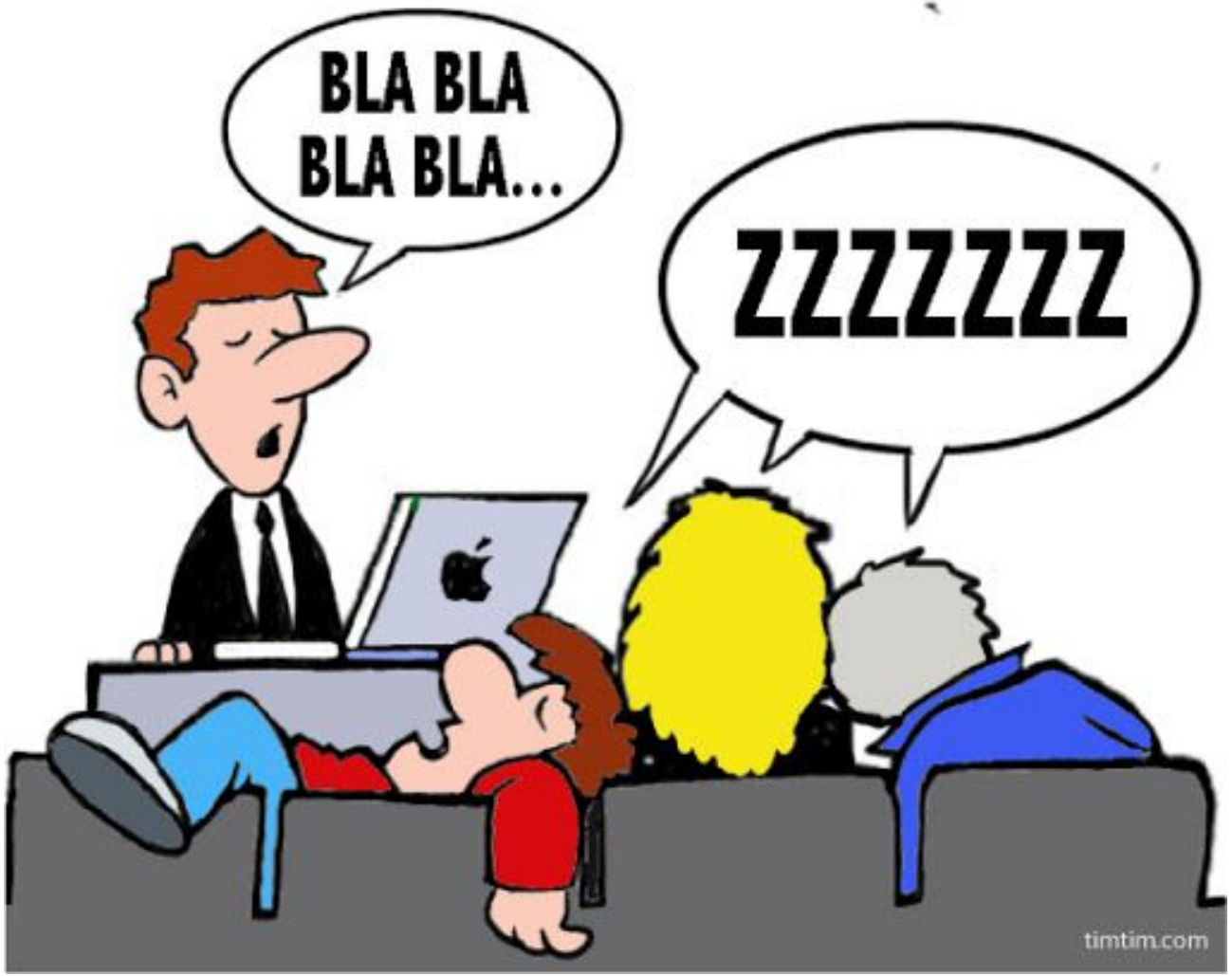


SDLC

Software Development Life Cycle

Circle how you are feeling:

Students, draw anywhere on this slide!


BLA BLA
BLA BLA...
ZZZZZZZ
timtim.com

BEFORE (in-class session)

What do you know about SDLC.

(Please write shortly on PEAR DECK slide)

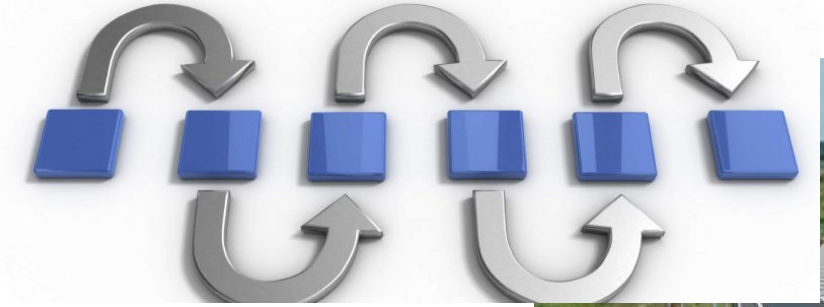
Students, write your response!

Table of Contents

* New approaches (agile and devops) and their implementations (Jira) will be explained later as separate lessons.

1 What is SDLC ?

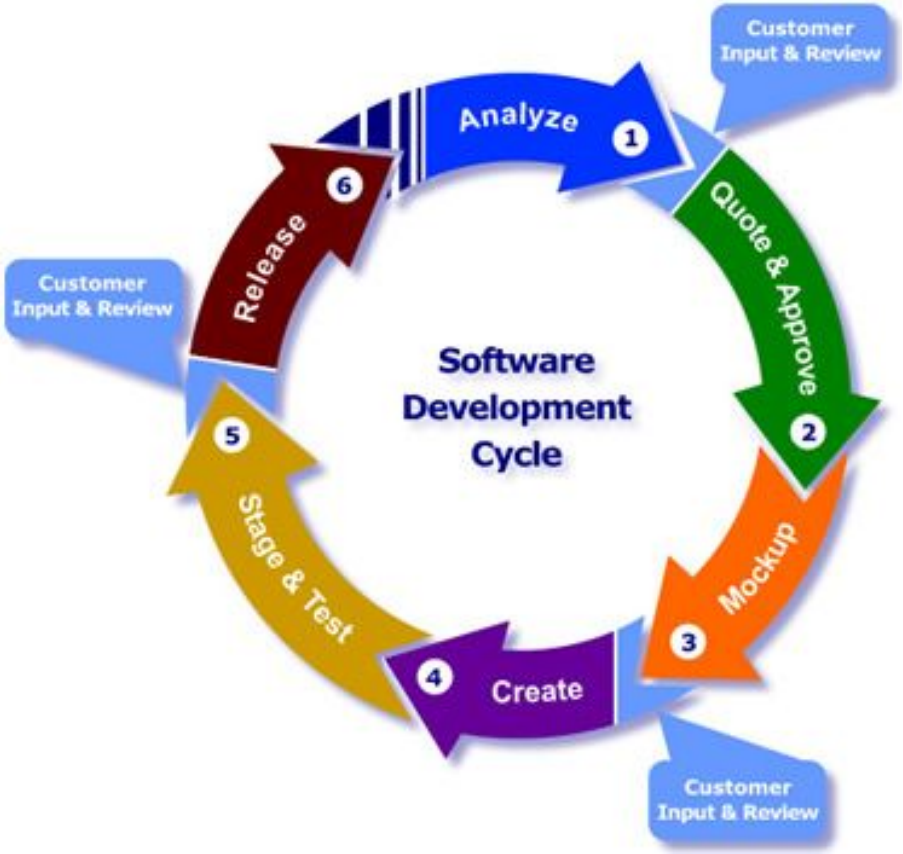
What is SDLC



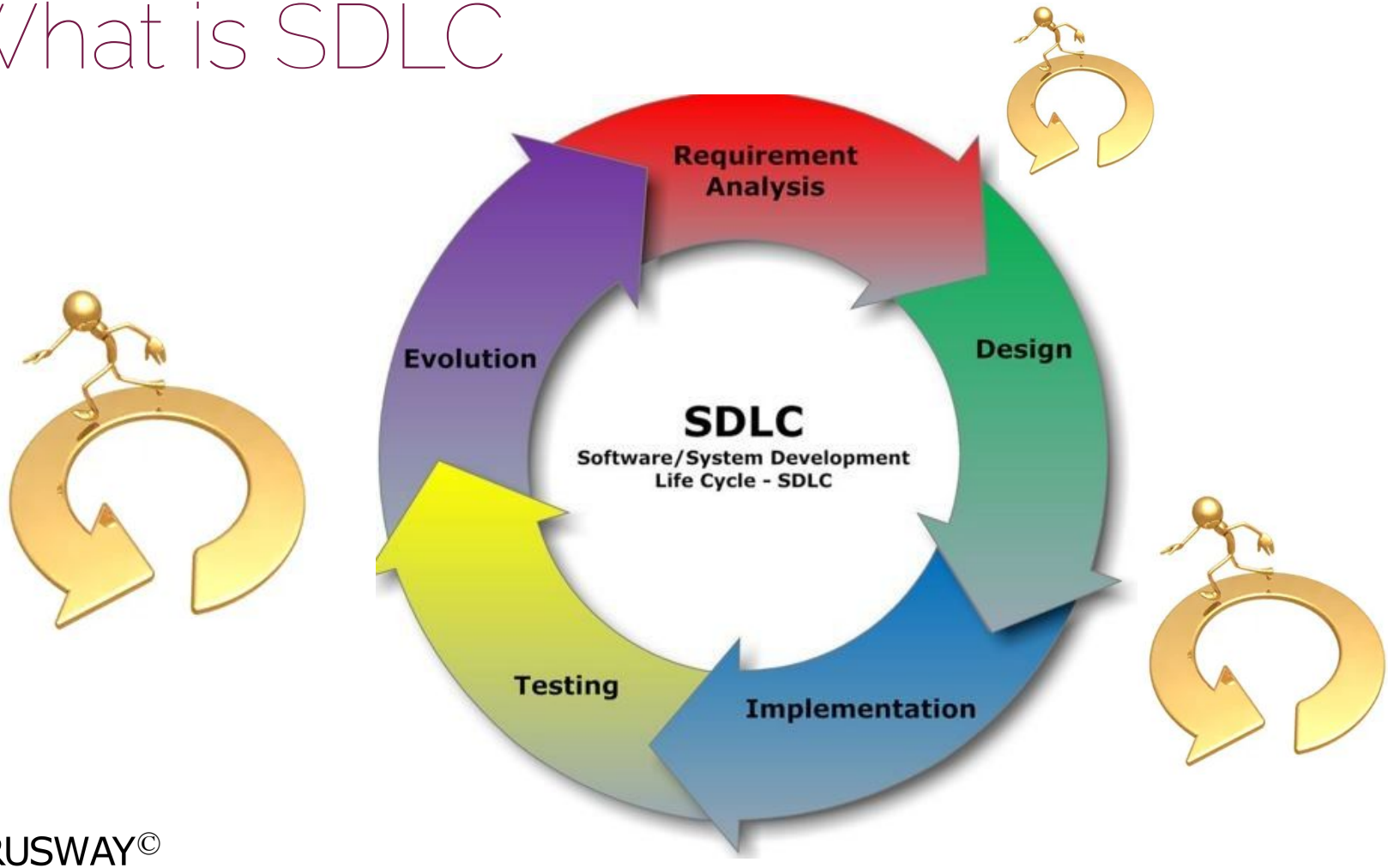
What is SDLC

- Systematic process to be followed for a software project.
- Structured way to create and develop software.

What is SDLC

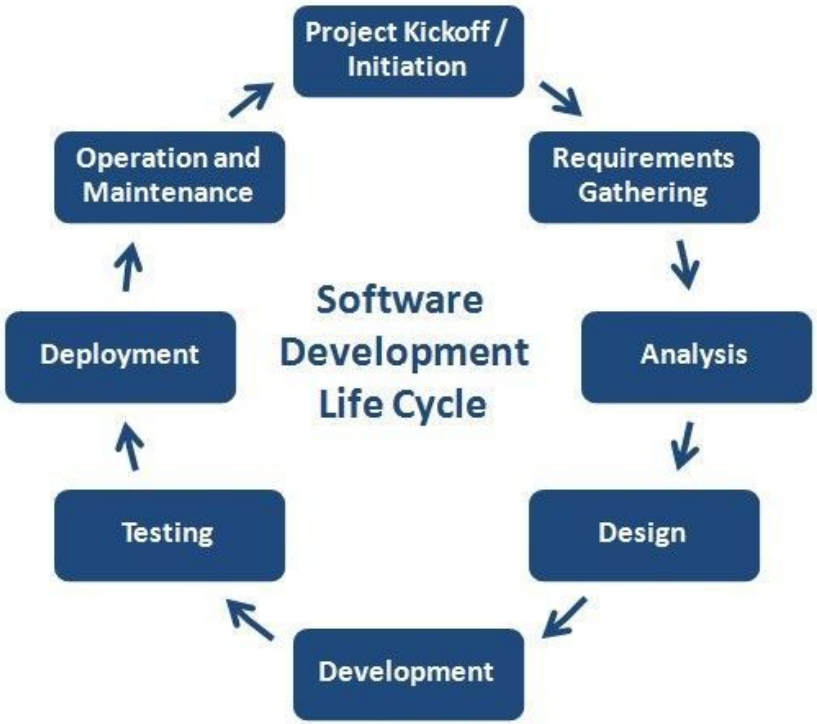

Customer Input & Review
Analyze
1
Quote & Approve
2
Mockup
3
Create
4
Stage & Test
5
Release
6
Customer Input & Review
Customer Input & Review
Software Development Cycle

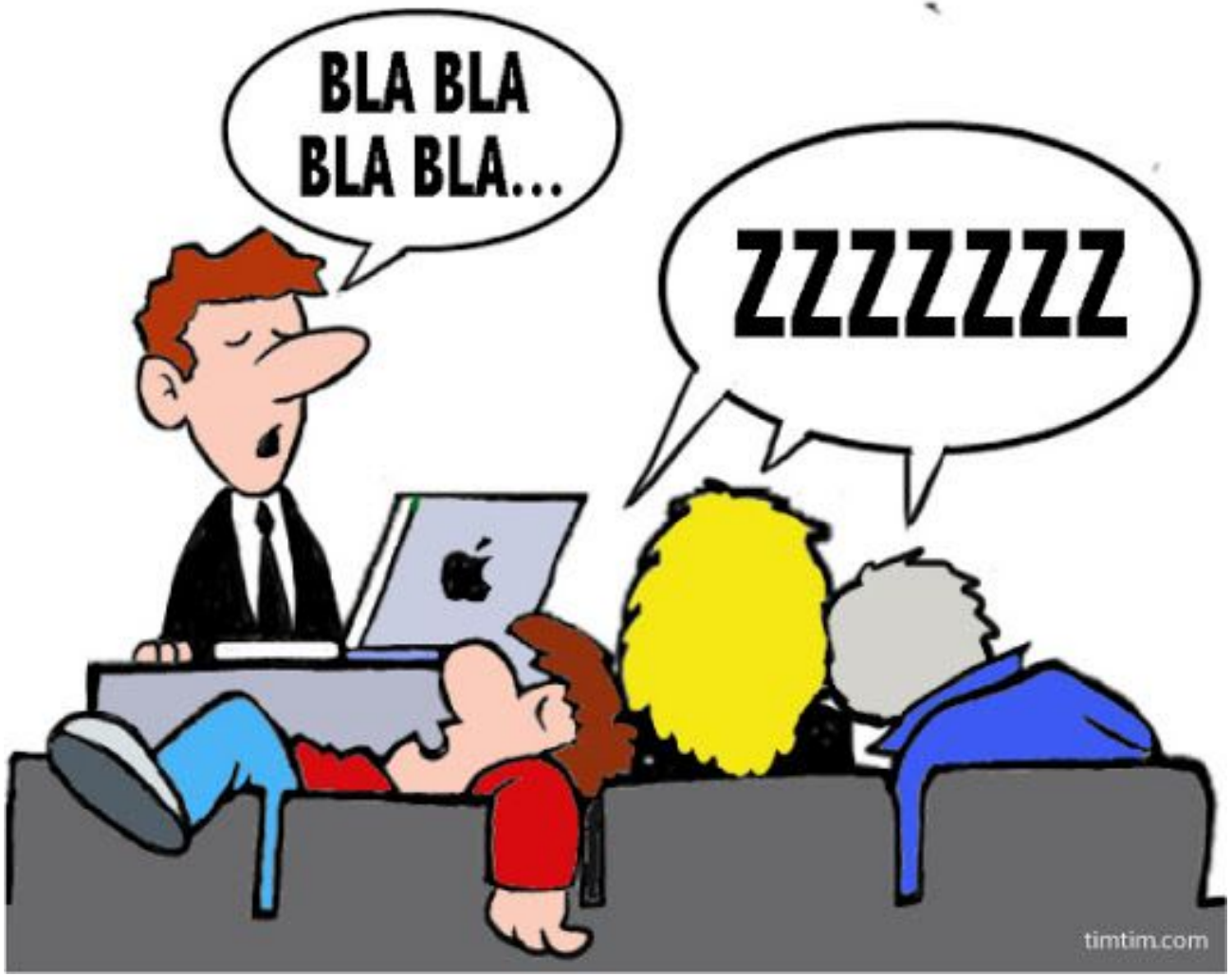
What is SDLC


Requirement Analysis
Design
Implementation
Testing
Evolution
SDLC
Software/System Development Life Cycle - SDLC

What is SDLC

V1.02.03



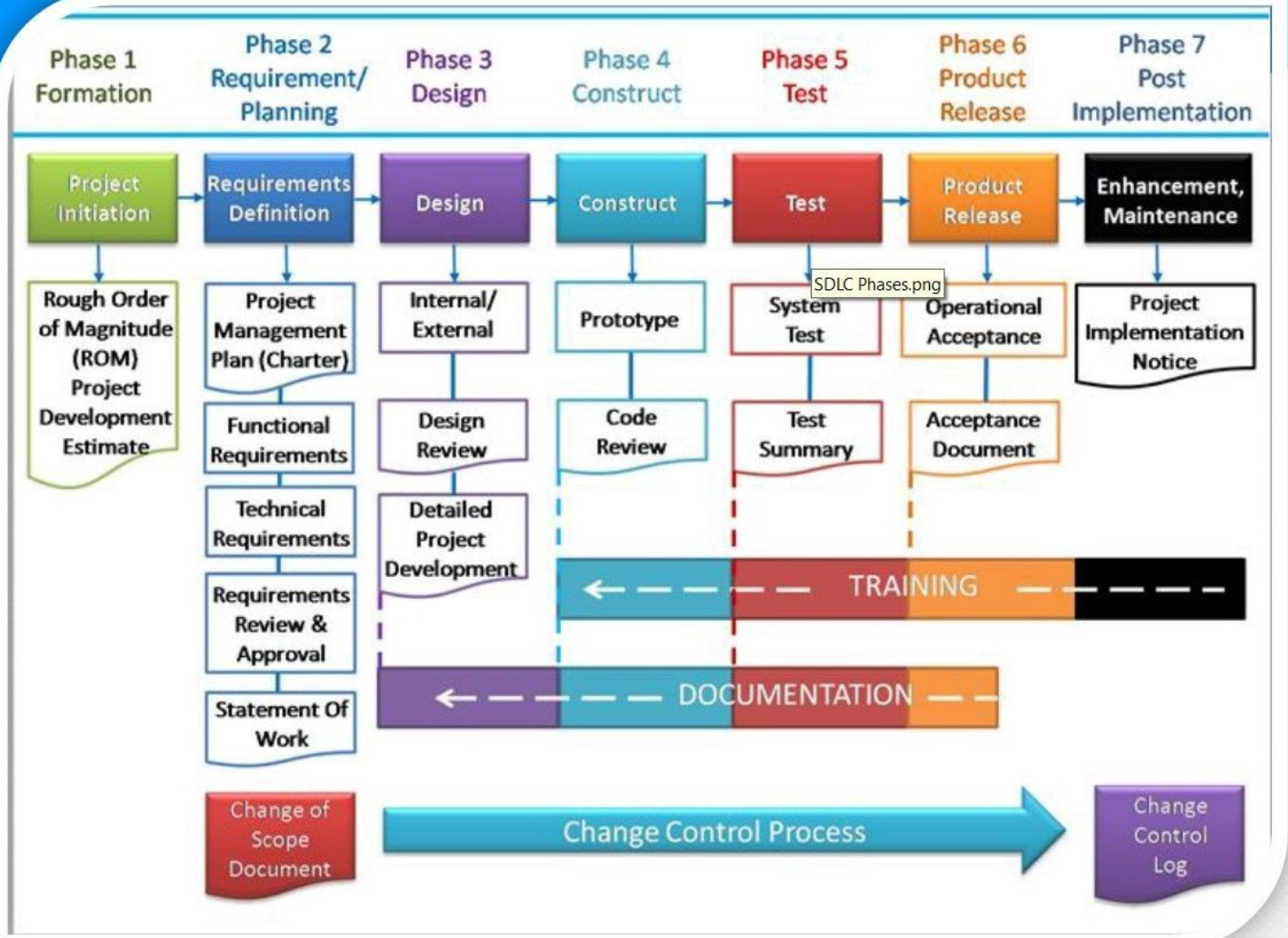

BLA BLA
BLA BLA...
ZZZZZZZ
timtim.com

2 Phases of SDLC

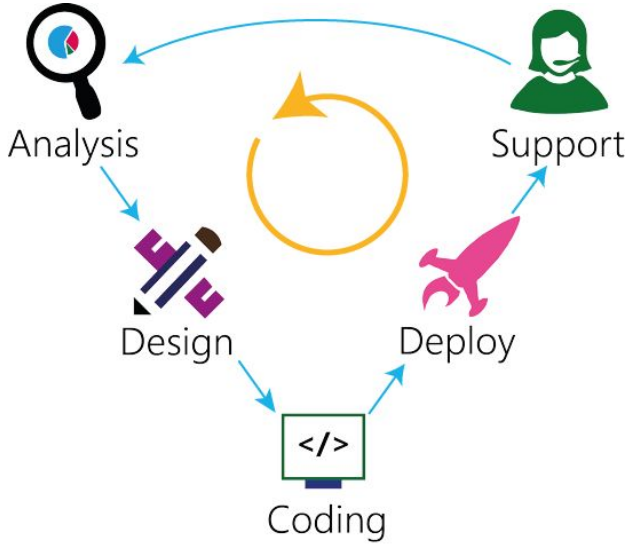
Phases of SDLC

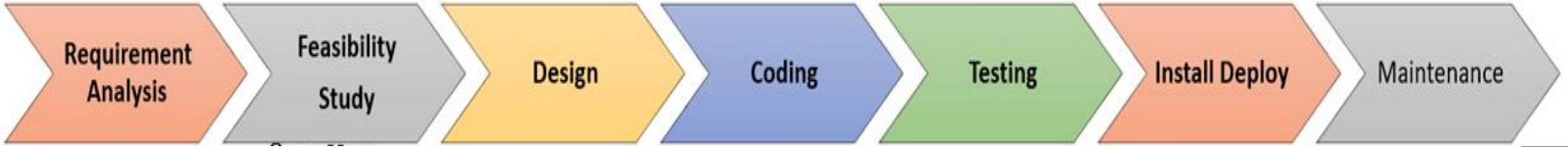
How many phases does SDLC have?

Students choose an option

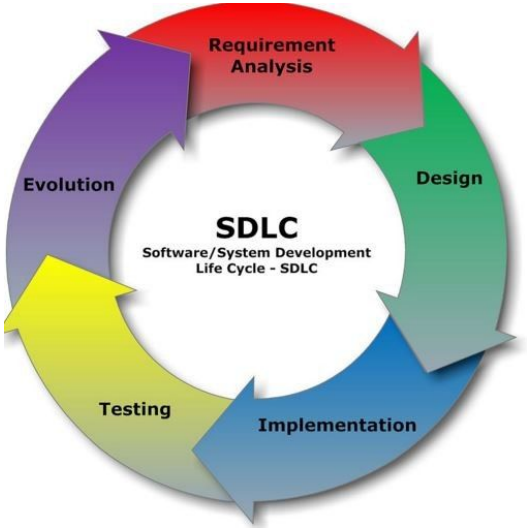

Phase 1 Formation
Phase 2 Requirement/ Planning
Phase 3 Design
Phase 4 Construct
Phase 5 Test
Phase 6 Product Release
Phase 7 Post Implementation
Project Initiation
Requirements Definition
Design
Construct
Test
Product Release
Enhancement, Maintenance
Rough Order of Magnitude (ROM) Project Development Estimate
Project Management Plan (Charter)
Functional Requirements
Technical Requirements
Requirements Review & Approval
Statement Of Work
Internal/ External
Design Review
Detailed Project Development
Prototype
Code Review
System Test
Test Summary
Operational Acceptance
Acceptance Document
Project Implementation Notice
SDLC Phases.png
TRAINING
DOCUMENTATION
Change of Scope Document
Change Control Process
Change Control Log

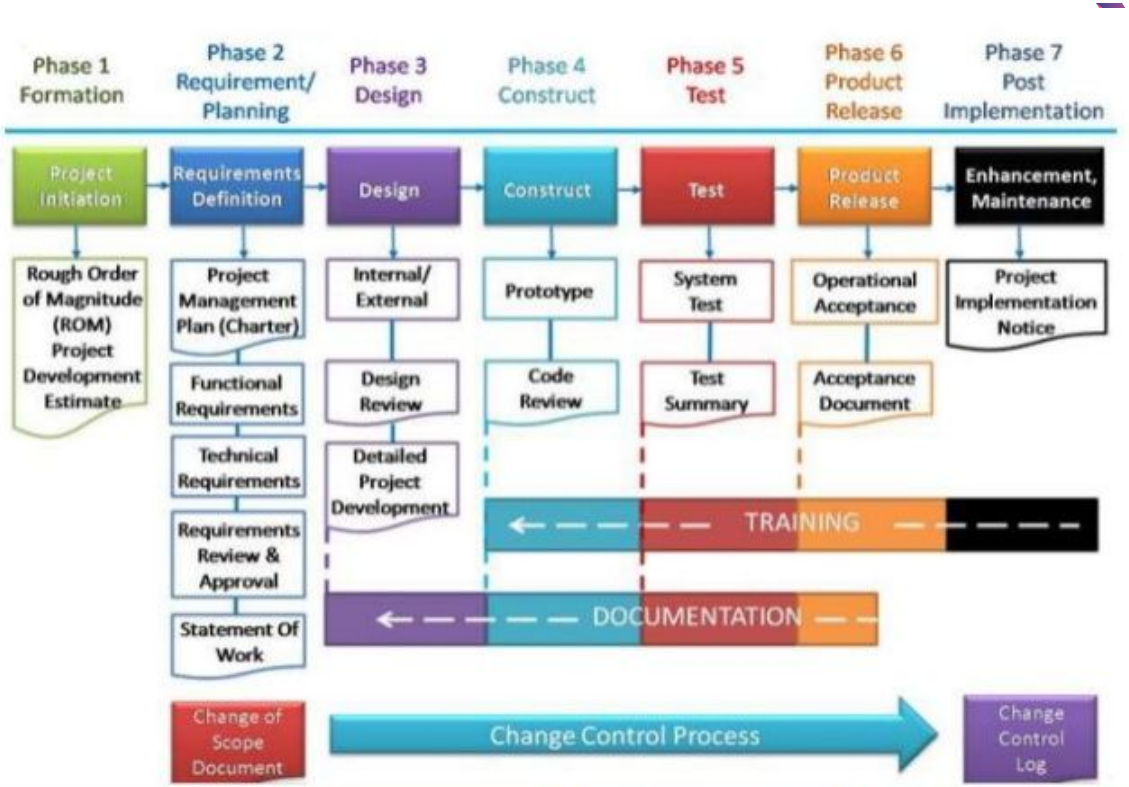
Phases of SDLC

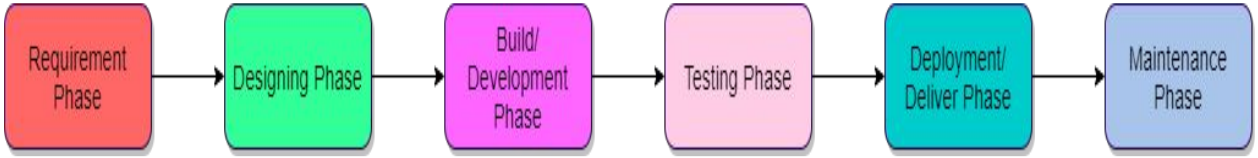




Phases of SDLC



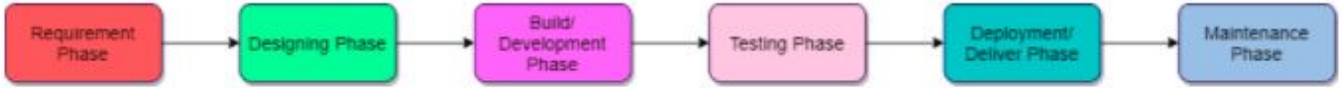




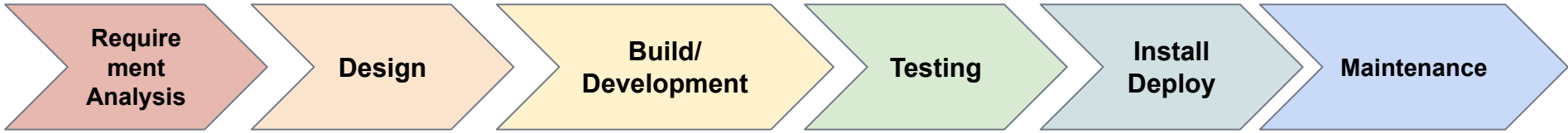
Phases of SDLC

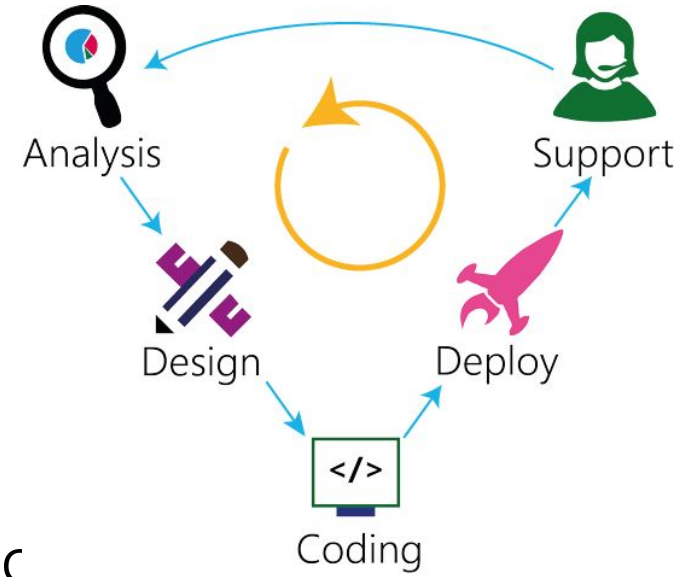
The SDLC process consists essentially of the following phases:

- Requirement Phase
- Design Phase
- Build/Development Phase
- Testing Phase
- Deployment/Deliver Phase
- Maintenance



Phases of SDLC





Phases of SDLC

What is the most critical phase?

Students choose an option

Phases of SDLC

What is the name of the document that consists of all necessary requirements to be designed?

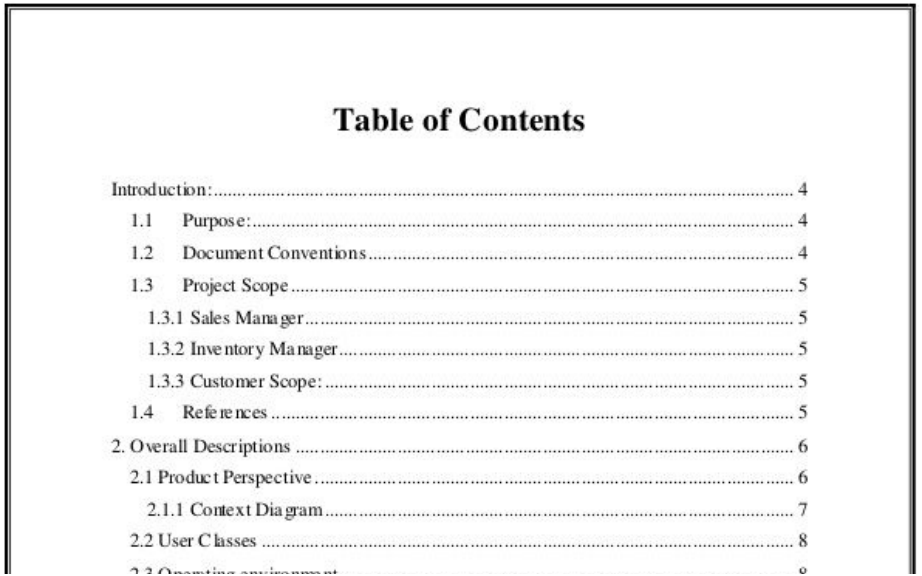
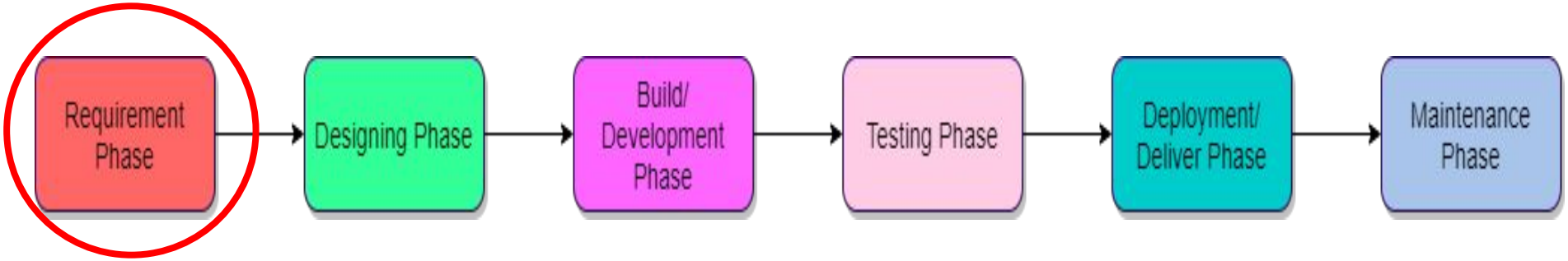
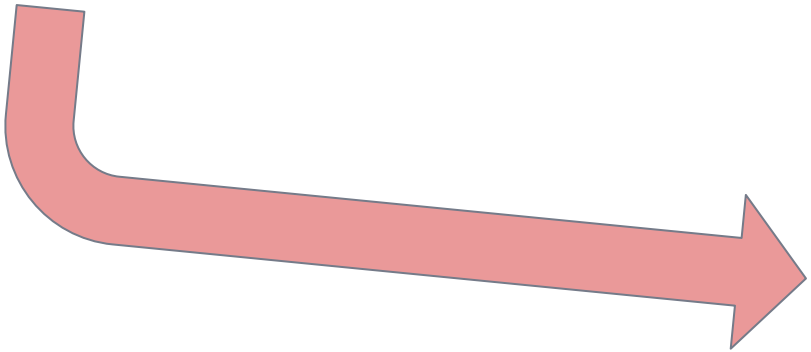


Table of Contents

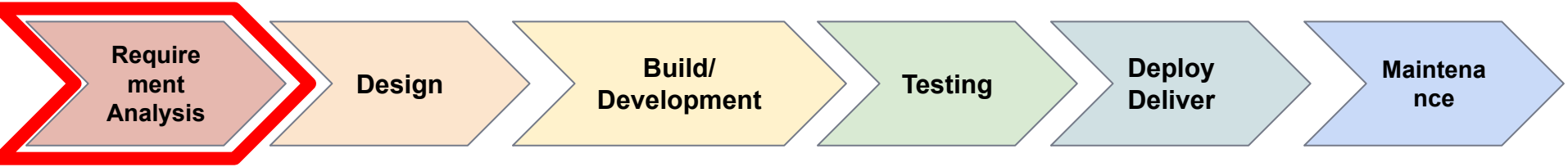
Requirements Phase

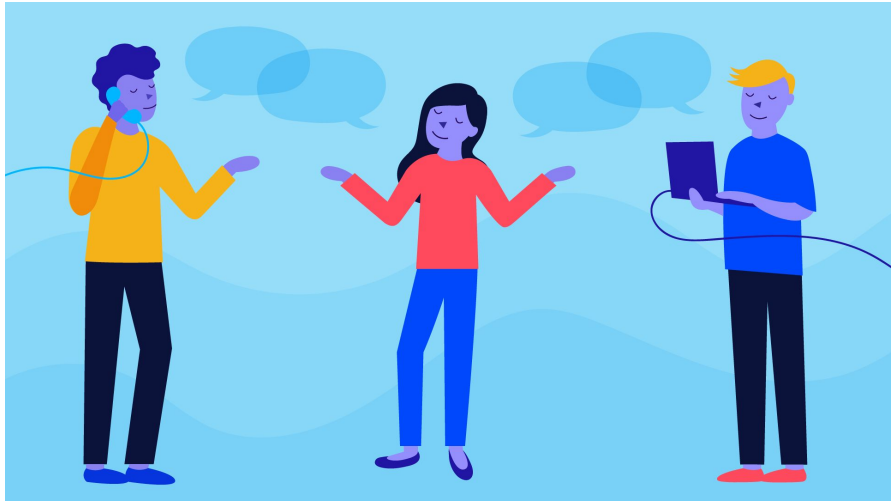




SRS

Requirements Phase





Requirement Phase

STRUCTURE OF SRS

Chapter no. 1	**Preface**	**It briefly explains about project.**
Chapter no. 2	Introduction	Highlights the projects with its title and briefly describe the projects.
Chapter no. 3	Scope	What is the capability of the product?
Chapter no. 4	Glossary	Definition, acronyms and abbreviation.
Chapter no. 5	User requirement definition	Describes non-functional requirements
Chapter no. 6	Architecture	Specifies system architecture
Chapter no. 7	System requirements	System description with function and non-function requirement.
Chapter no. 8	System model	System model used to represent relationship.
Chapter no. 9	System evaluation	How system is evolved?
Chapter no. 10	Appendices	Annexure, application, data requirements.
Chapter no. 11	indexes	Indices of diagram, tables, functions.

Requirement Phase

SRS Document Structure

Introduction

- Purpose, Definitions, System overview
- Scope of Work, References

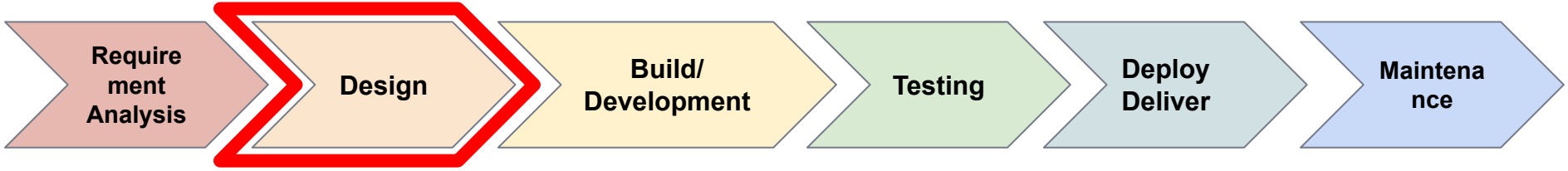
Overall description

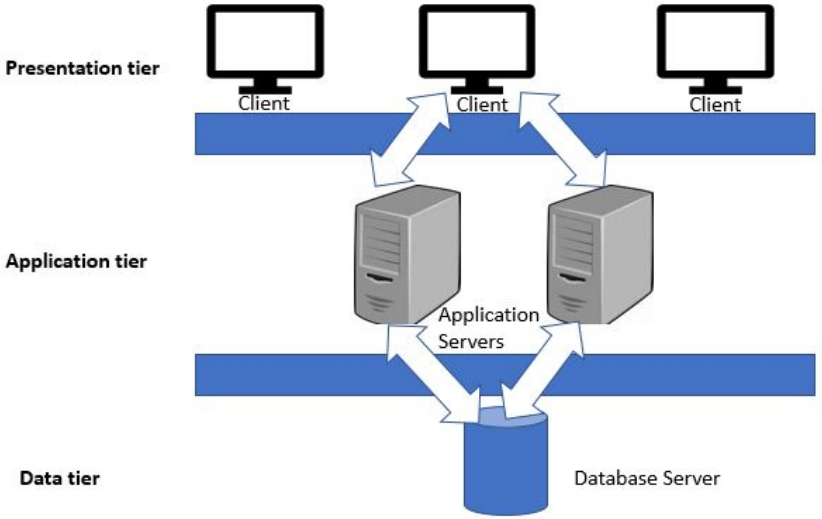
- Product perspective: System Interfaces, User Interfaces, Hardware interfaces, Software interfaces, Communication Interfaces, Memory Constraints, Operations, Site Adaptation Requirements
- Product functions and User characteristics
- Constraints, assumptions and dependencies

Specific requirements

- External interface requirements
- Functional requirements
- Performance requirements
- Design constraints: Standards Compliance
- Logical database requirement
- Software System attributes: Reliability, Availability, Security, Maintainability, Portability
- Other requirements

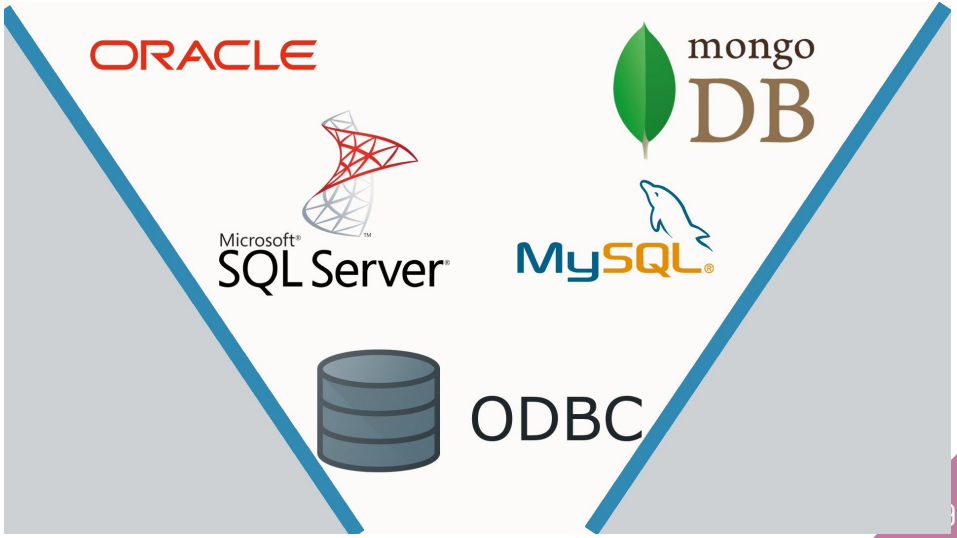
Design Phase



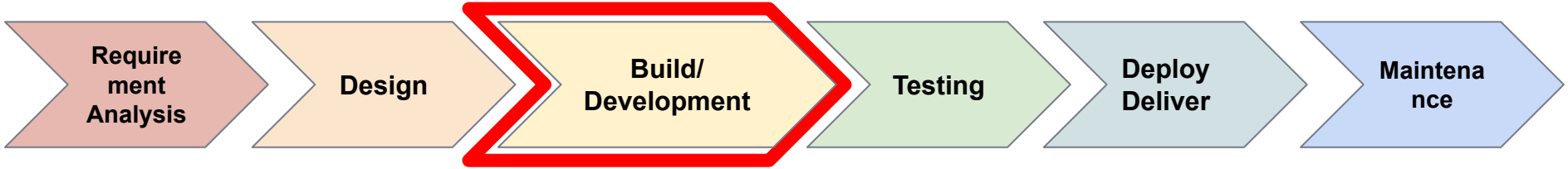


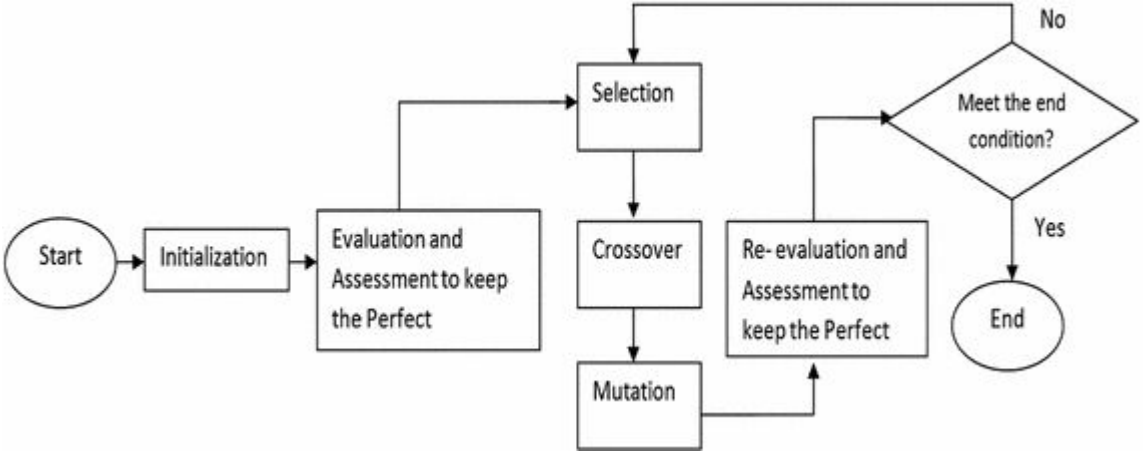
Design Phase

Requirement Analysis → **Design** → Build/ Development → Testing → Deploy Deliver → Maintenance



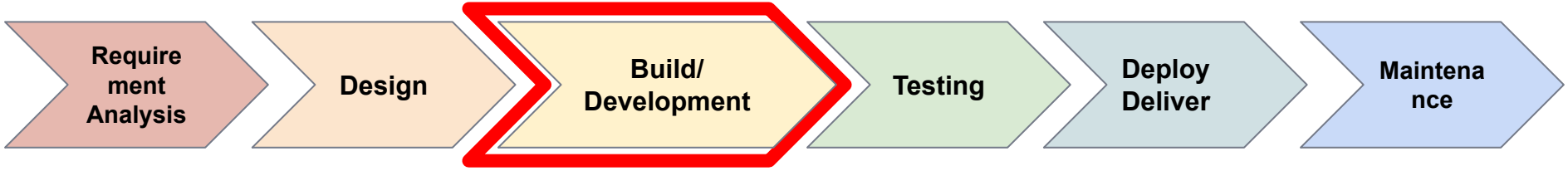
Build/Development Phase

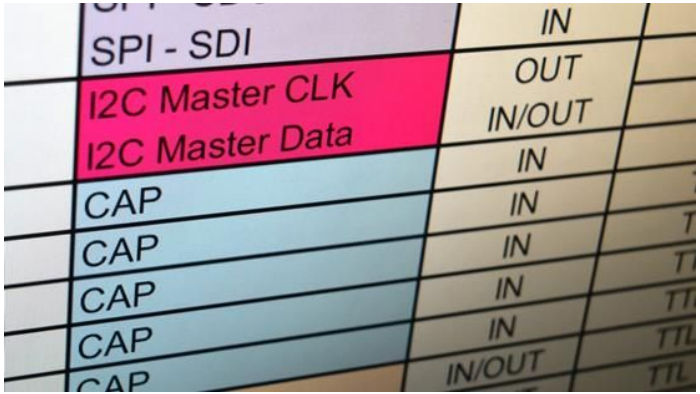




Build/ Development Phase

Design Phase





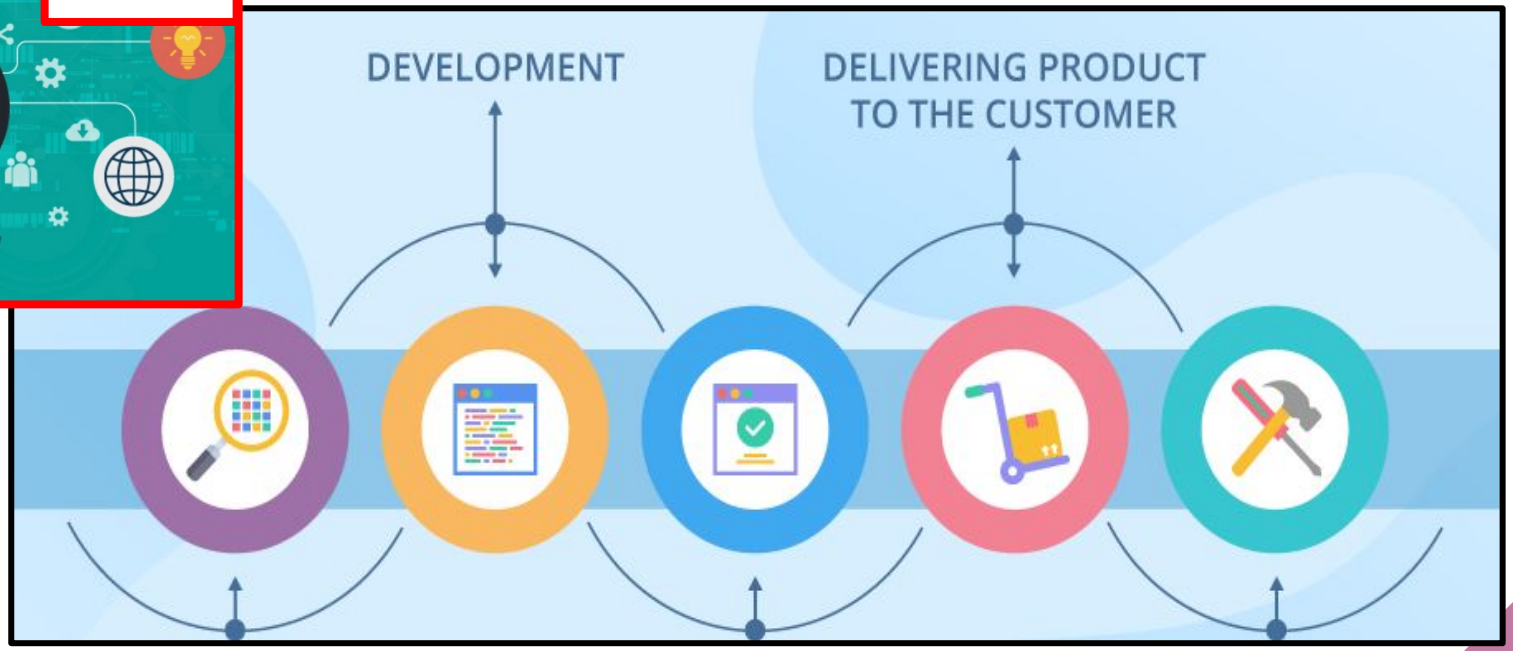
Phases of SDLC

At what phase we focus on the investigation?

Testing Phase

Requirement Analysis → Design → Build/ Development → **Testing** → Deploy Deliver → Maintenance

Require ment Analysis → Design → Build/ Development → Testing → **Deploy Deliver** → Maintena nce



Require ment Analysis | Design | Build/ Development | Testing | Deploy Deliver | Maintena nce

Require ment Analysis | Design | Build/ Development | Testing | Deploy Deliver | Maintena nce

Verilerimizi gerçekten şifrelemek zorunda mıyız? Zaten başlangıçta iletişimimizin büyük bölümünü anlamak mümkün değil ki...

"Do we really need to encrypt our data? Most of our communications are impossible to understand in the first place."

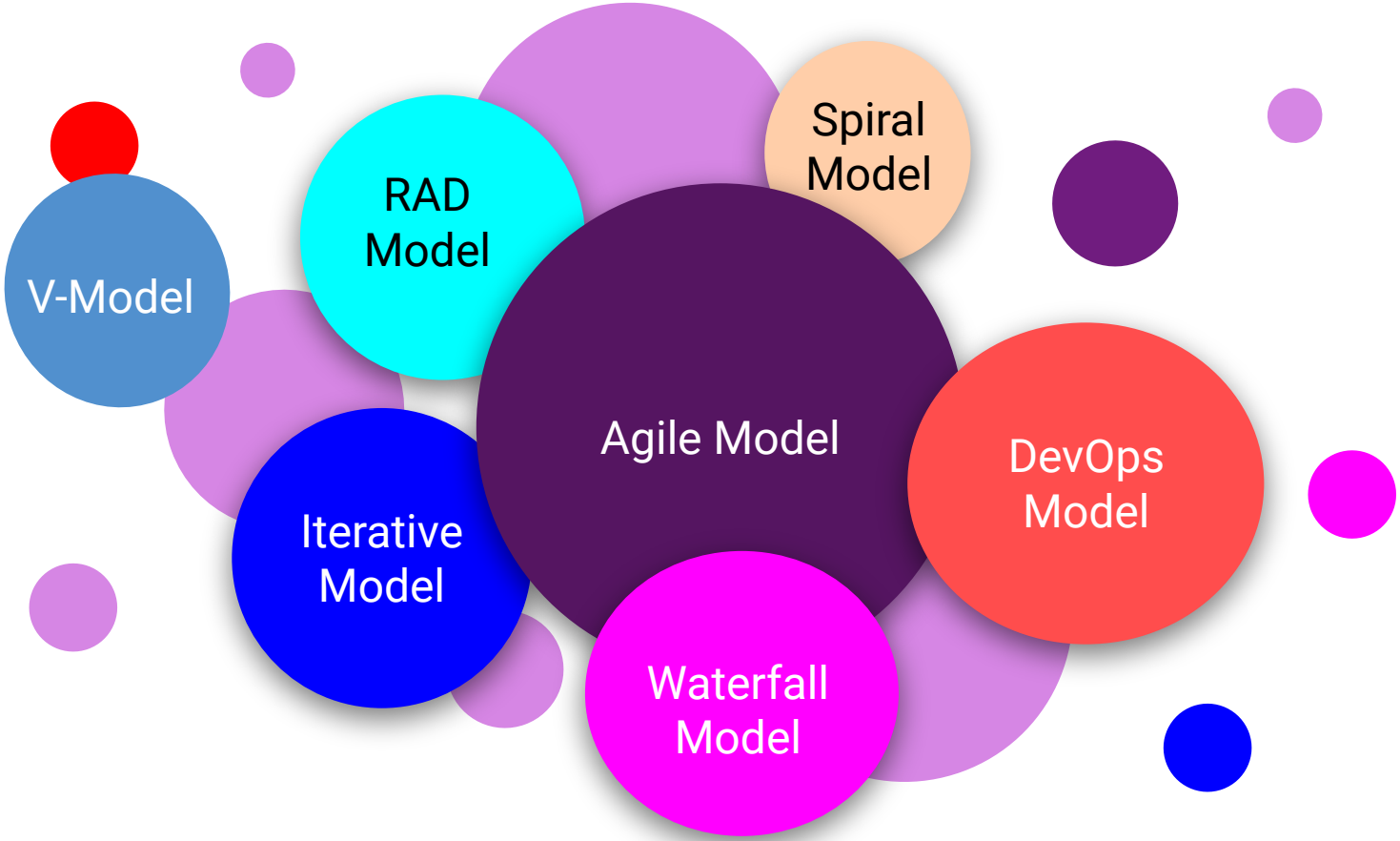
SDLC Models

SDLC Models

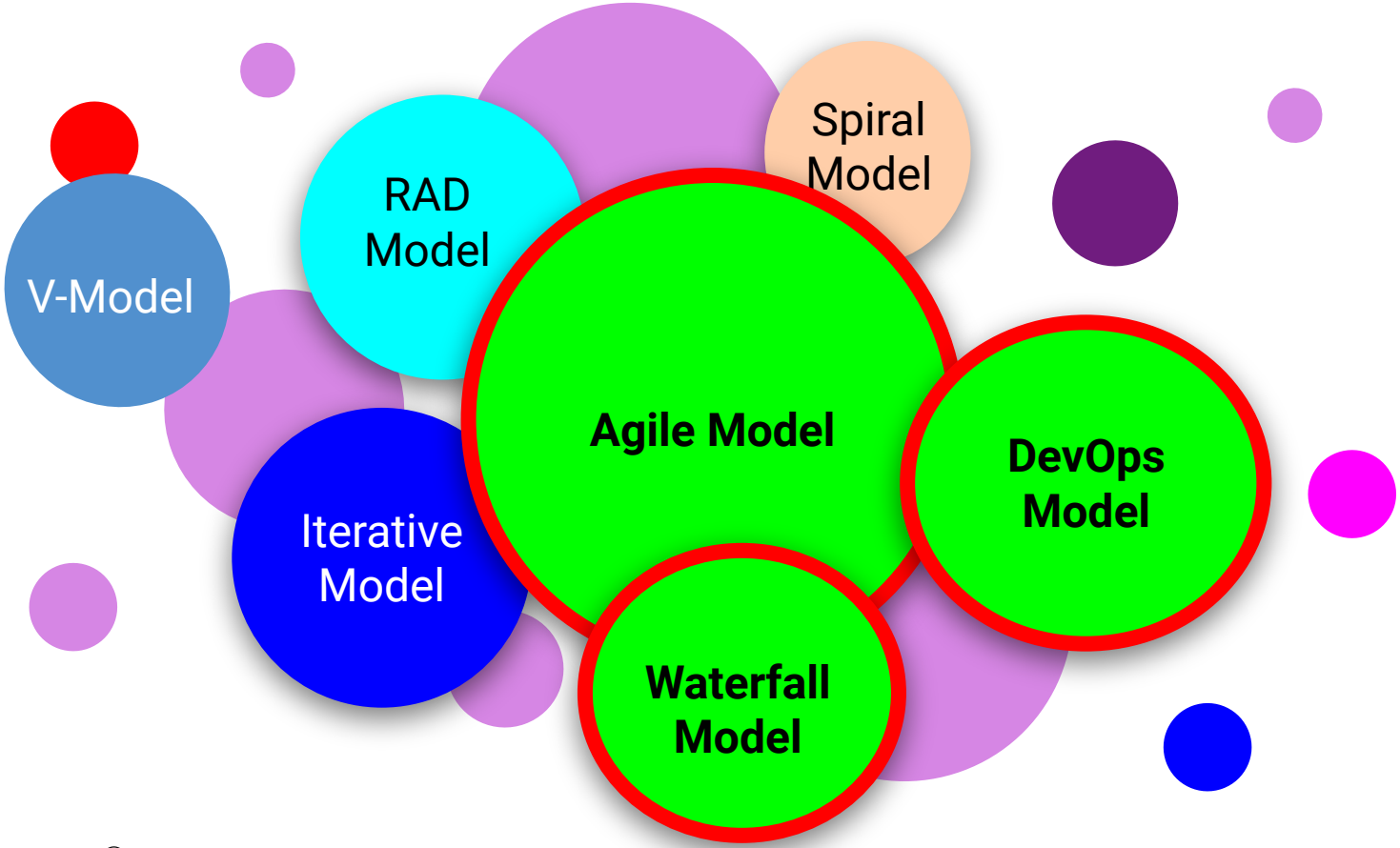
List the common SDLC models.

Students, write your response!

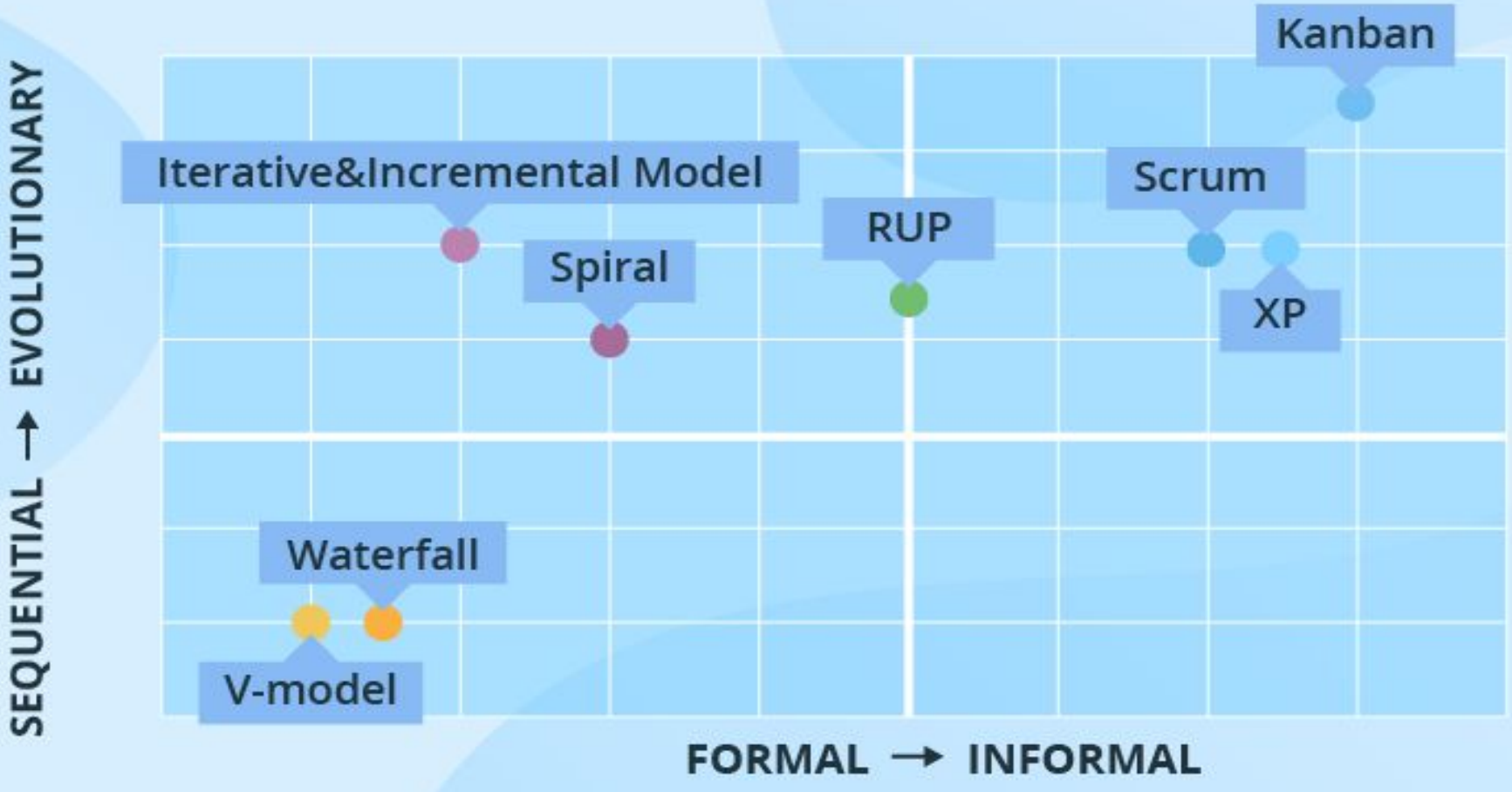
SDLC Models


V-Model
RAD Model
Spiral Model
Agile Model
DevOps Model
Iterative Model
Waterfall Model

SDLC Models


V-Model
RAD Model
Spiral Model
Agile Model
DevOps Model
Iterative Model
Waterfall Model

TYPES OF POPULAR SDLC MODELS



TYPES OF POPULAR SDLC MODELS

SEQUENTIAL → EVOLUTIONARY

Kanban

Iterative&Incremental Model

Scrum

RUP

Spiral

XP

Waterfall

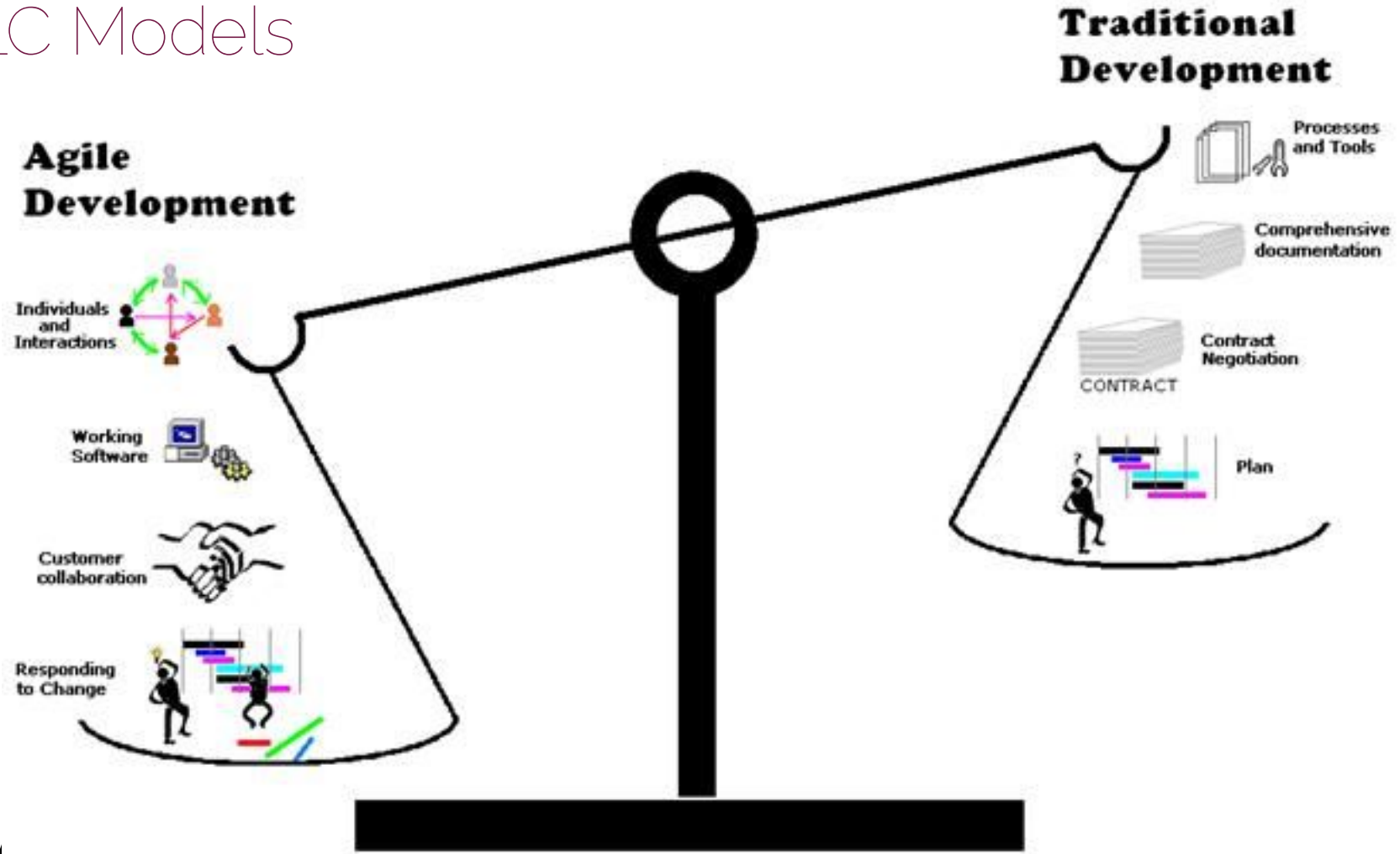
V-model

SDLC Models

Which one is the traditional SDLC model?

Students choose an option

SDLC Models


Agile Development
Individuals and Interactions
Working Software
Customer collaboration
Responding to Change
Traditional Development
Processes and Tools
Comprehensive documentation
Contract Negotiation
CONTRACT
Plan

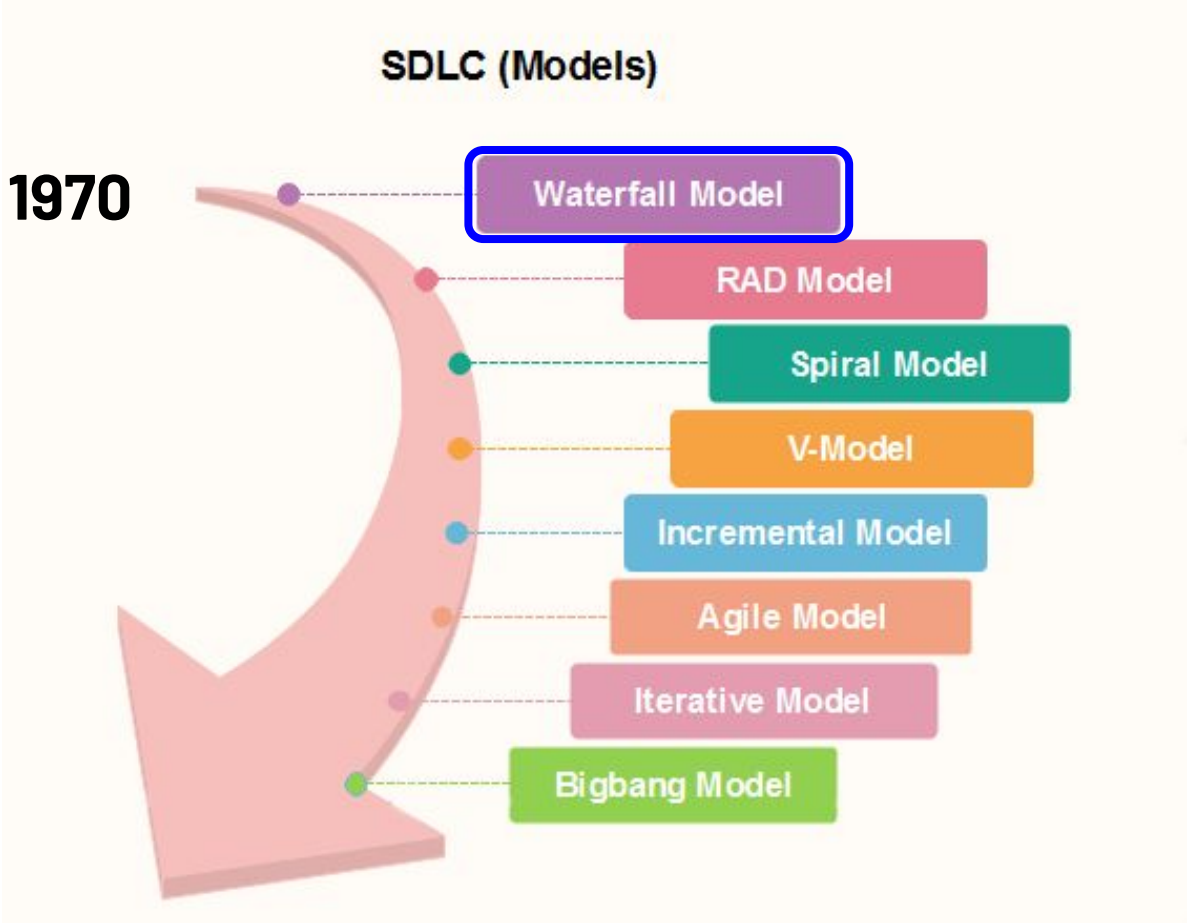
4 Waterfall Model

Waterfall Model

Waterfall Model

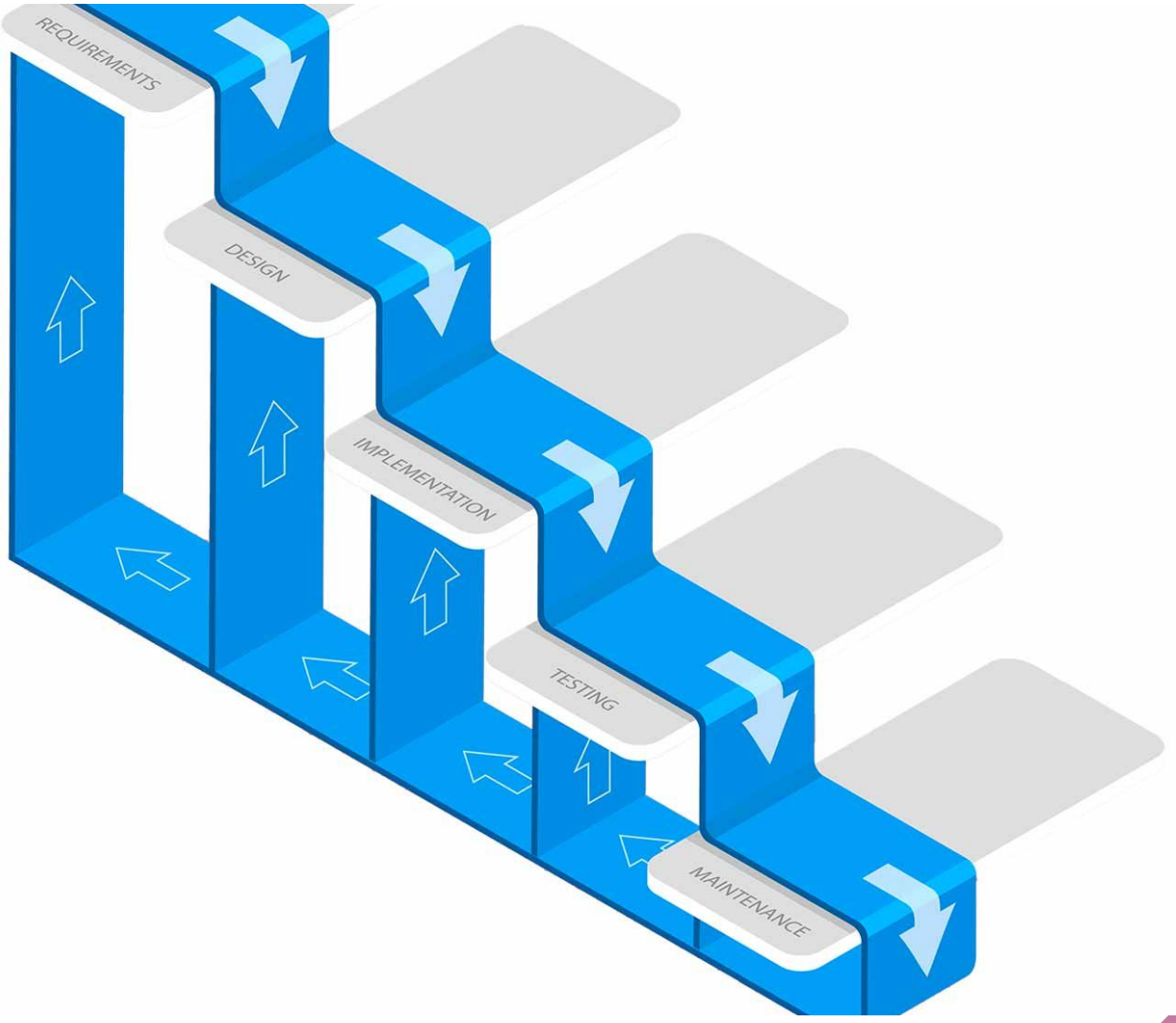
In which years did the Waterfall model appear?

Waterfall Model

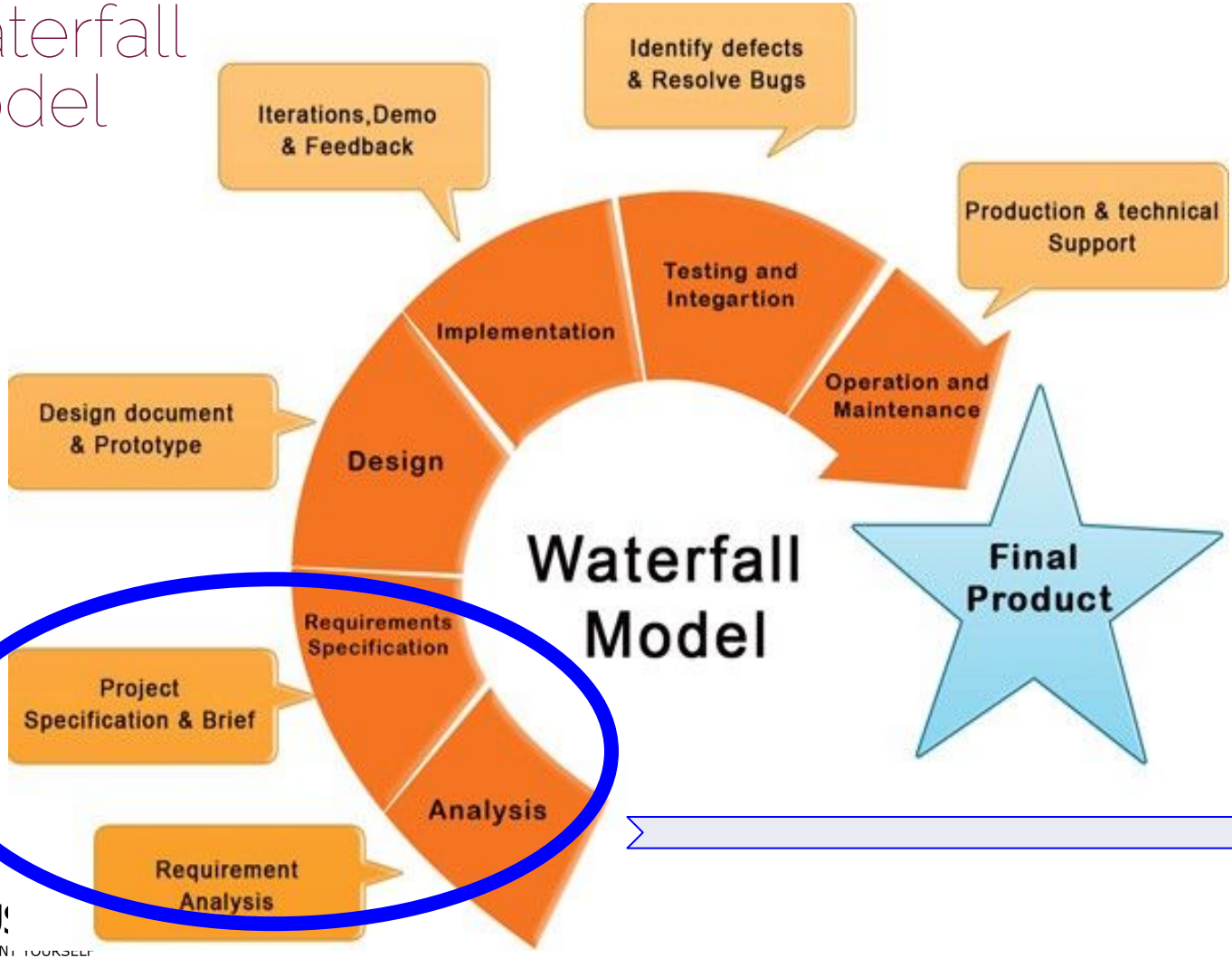


Waterfall Model

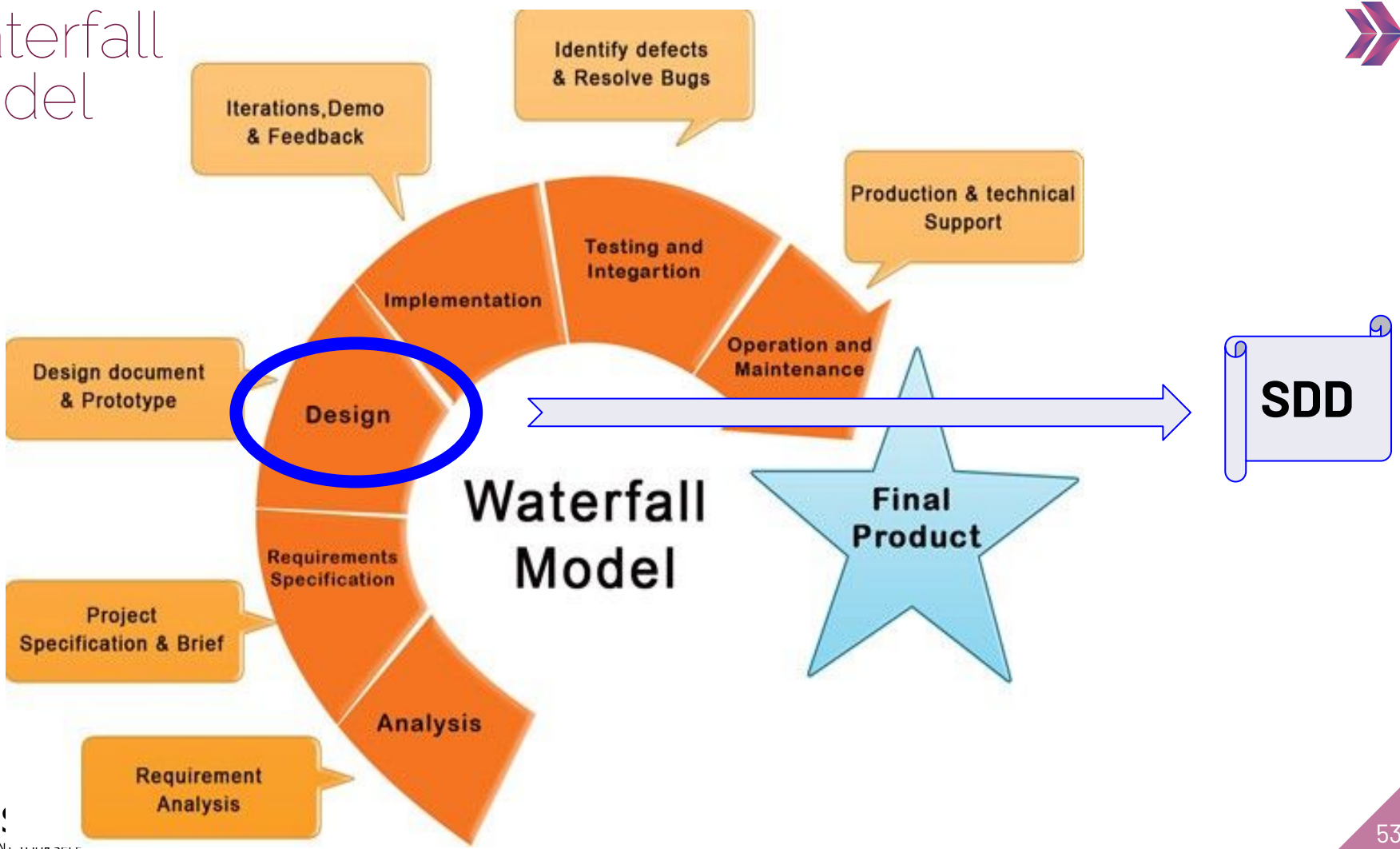
Waterfall M


REQUIREMENTS
DESIGN
IMPLEMENTATION
TESTING
MAINTENANCE

Waterfall Model



Waterfall Model

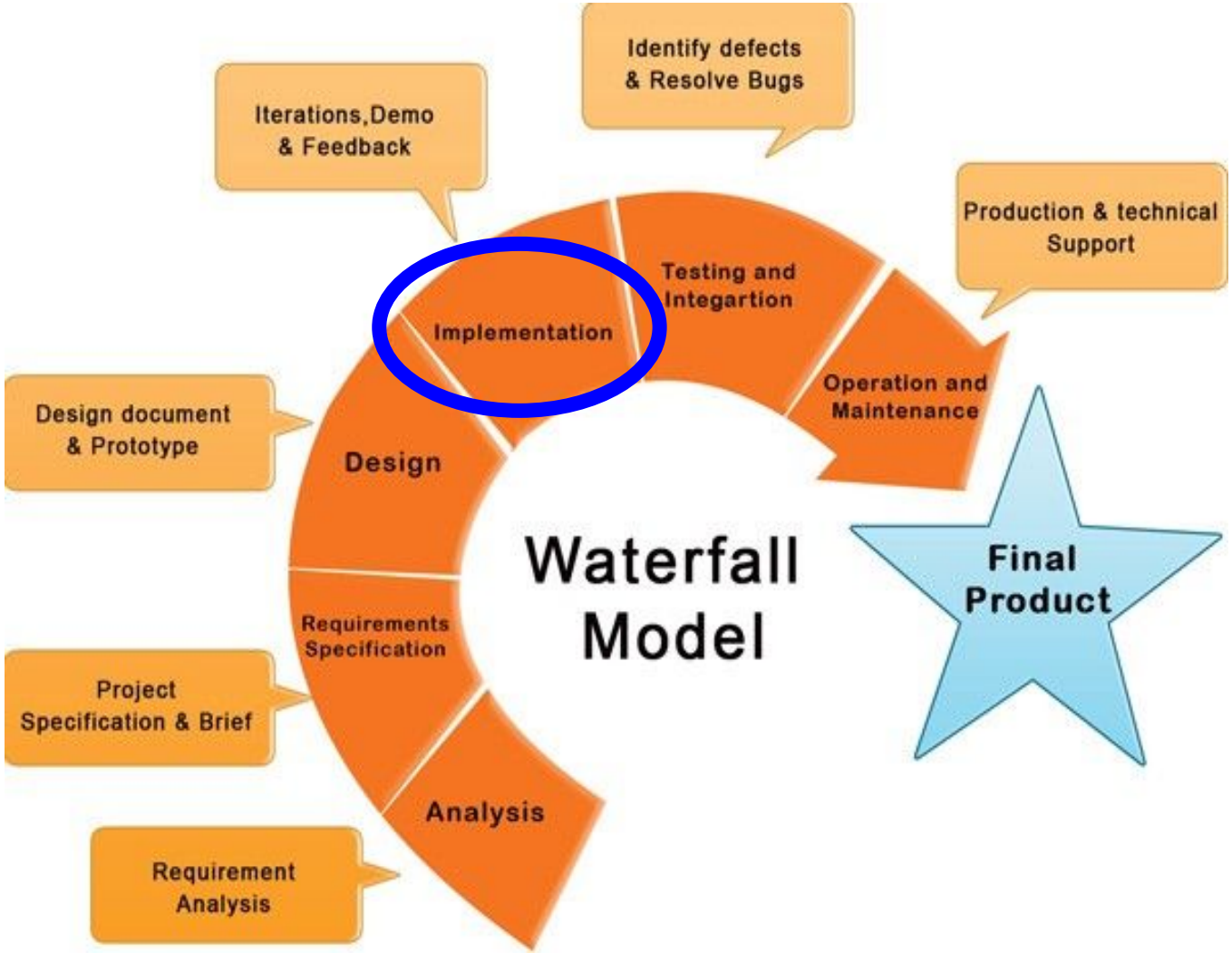


Waterfall Model

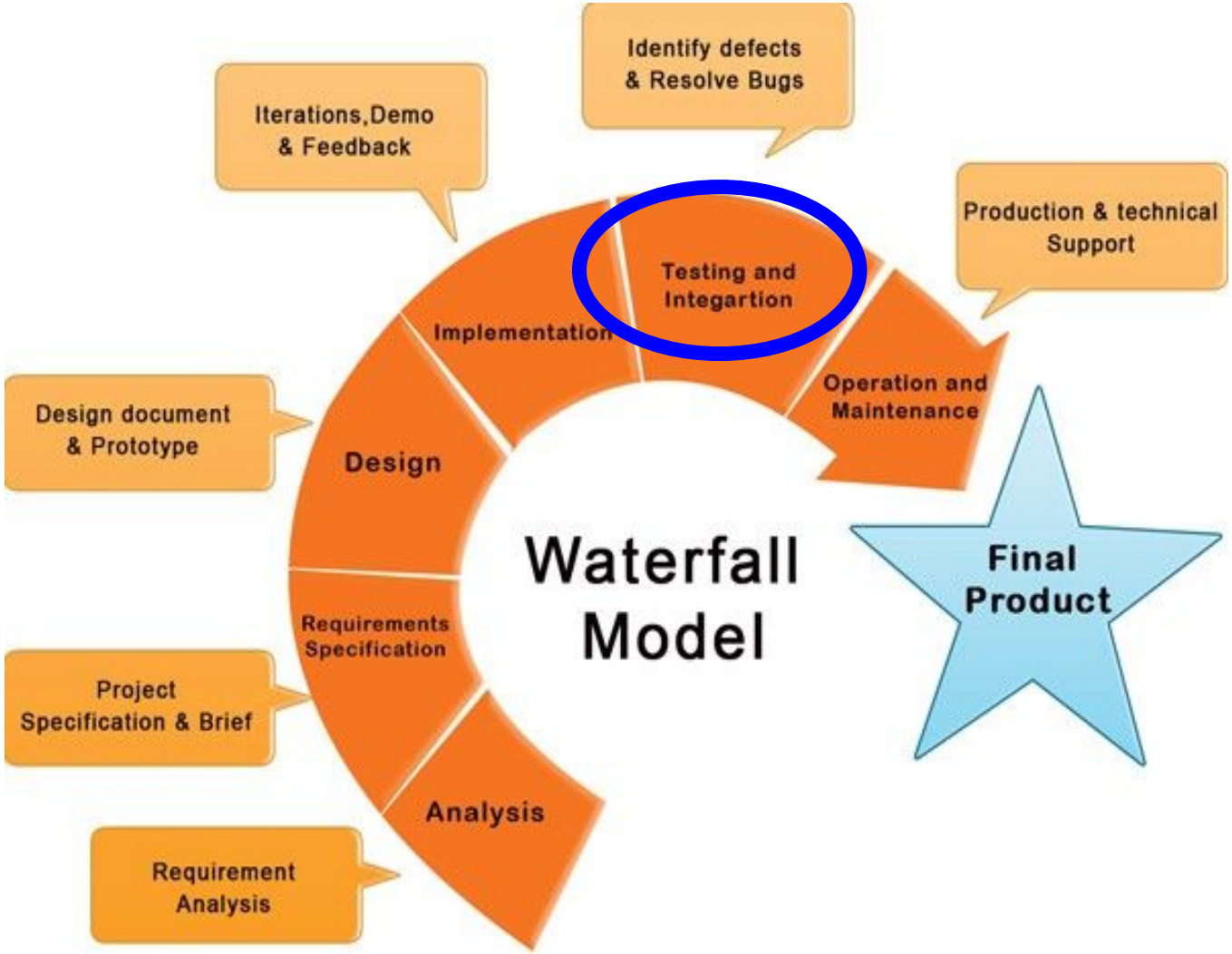
Coding is the other name for implementation/developing in SDLC.

Students choose an option

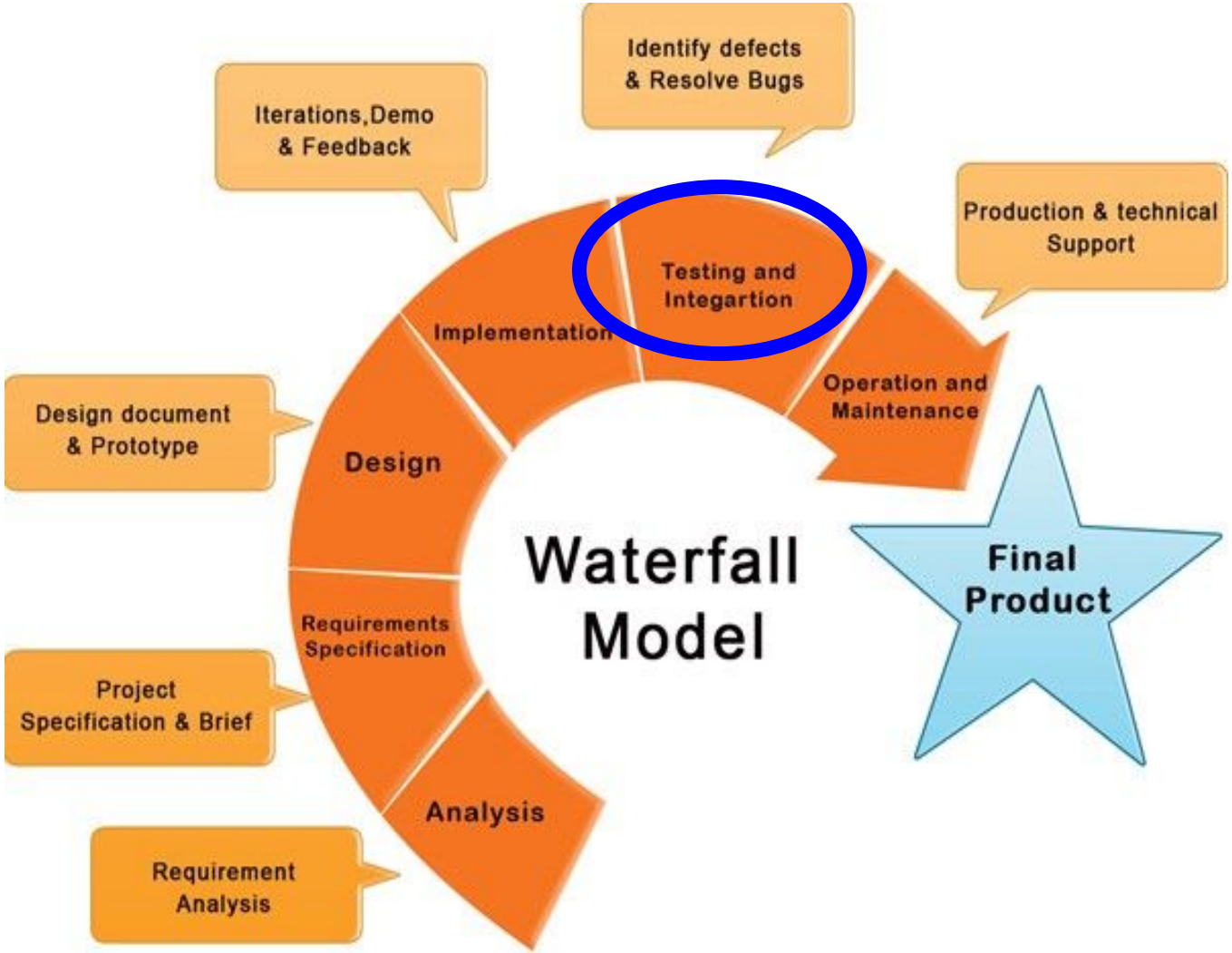
Waterfall Model



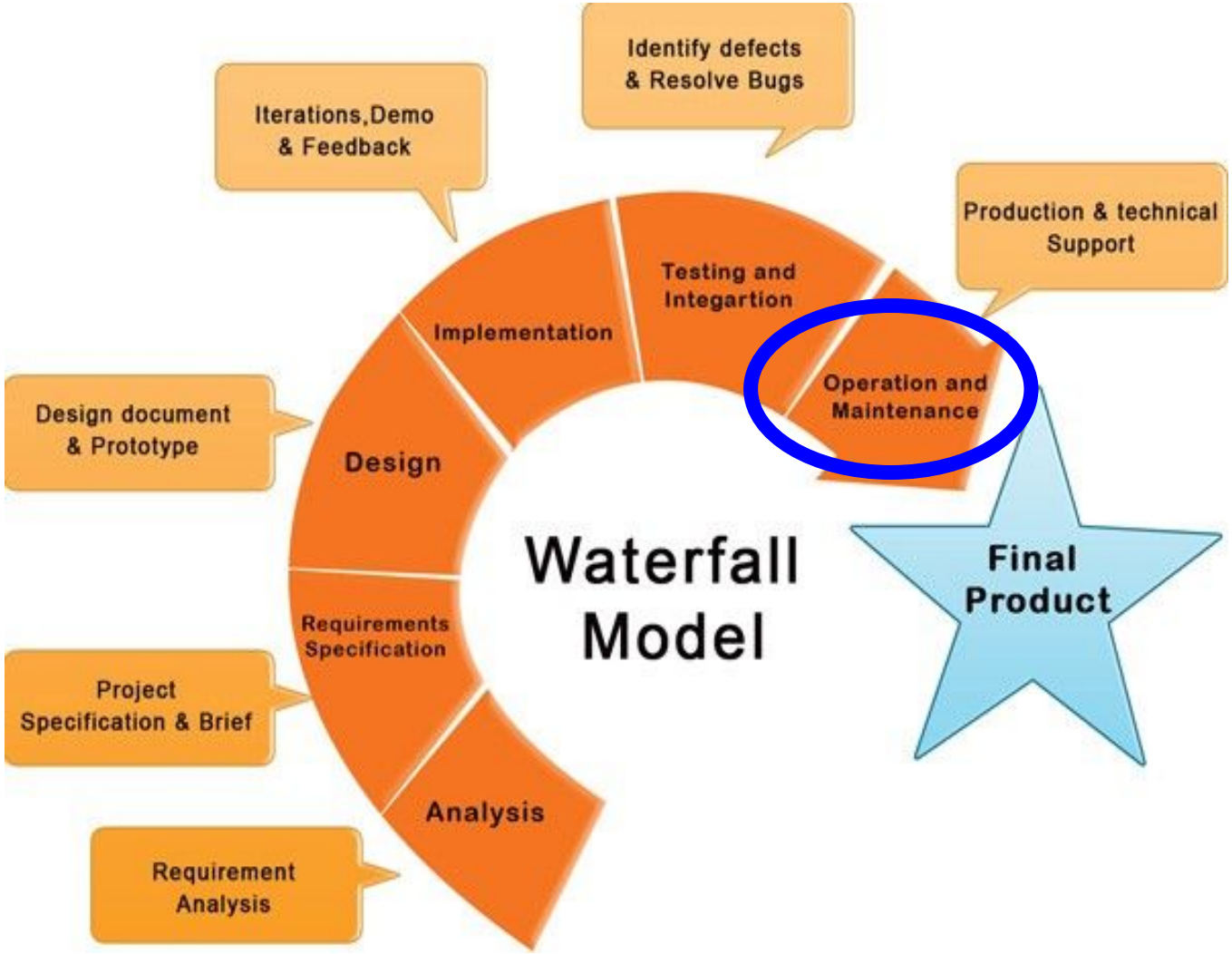
Waterfall Model



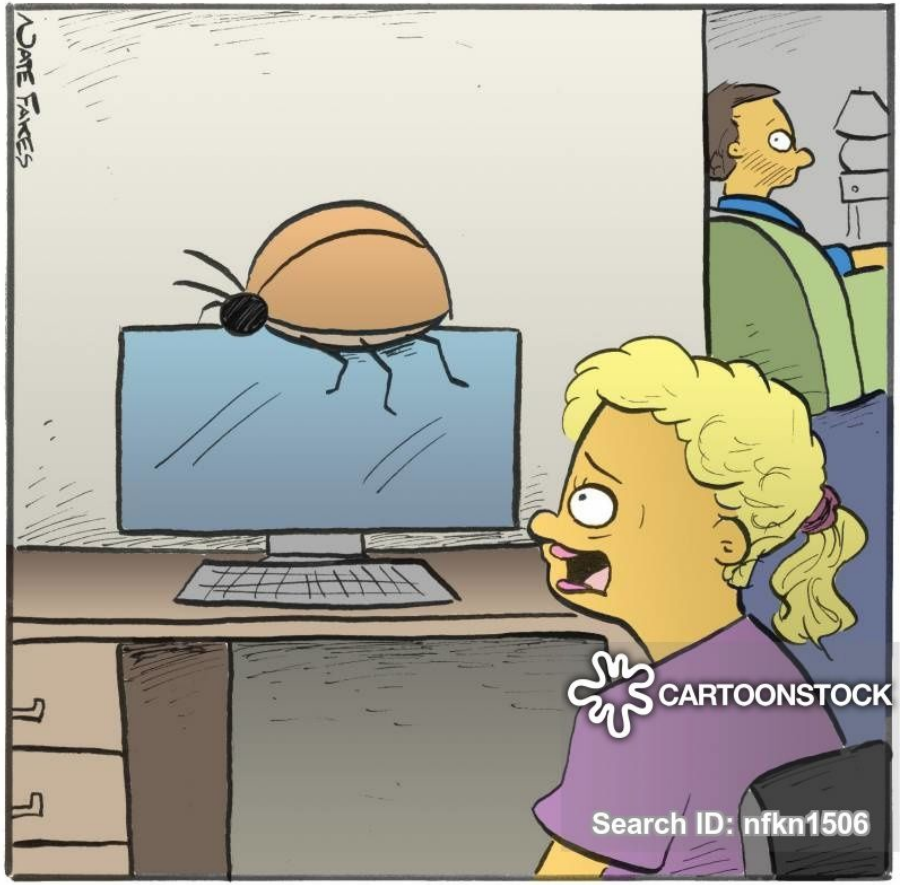
Waterfall Model



Waterfall Model



Waterfall Model

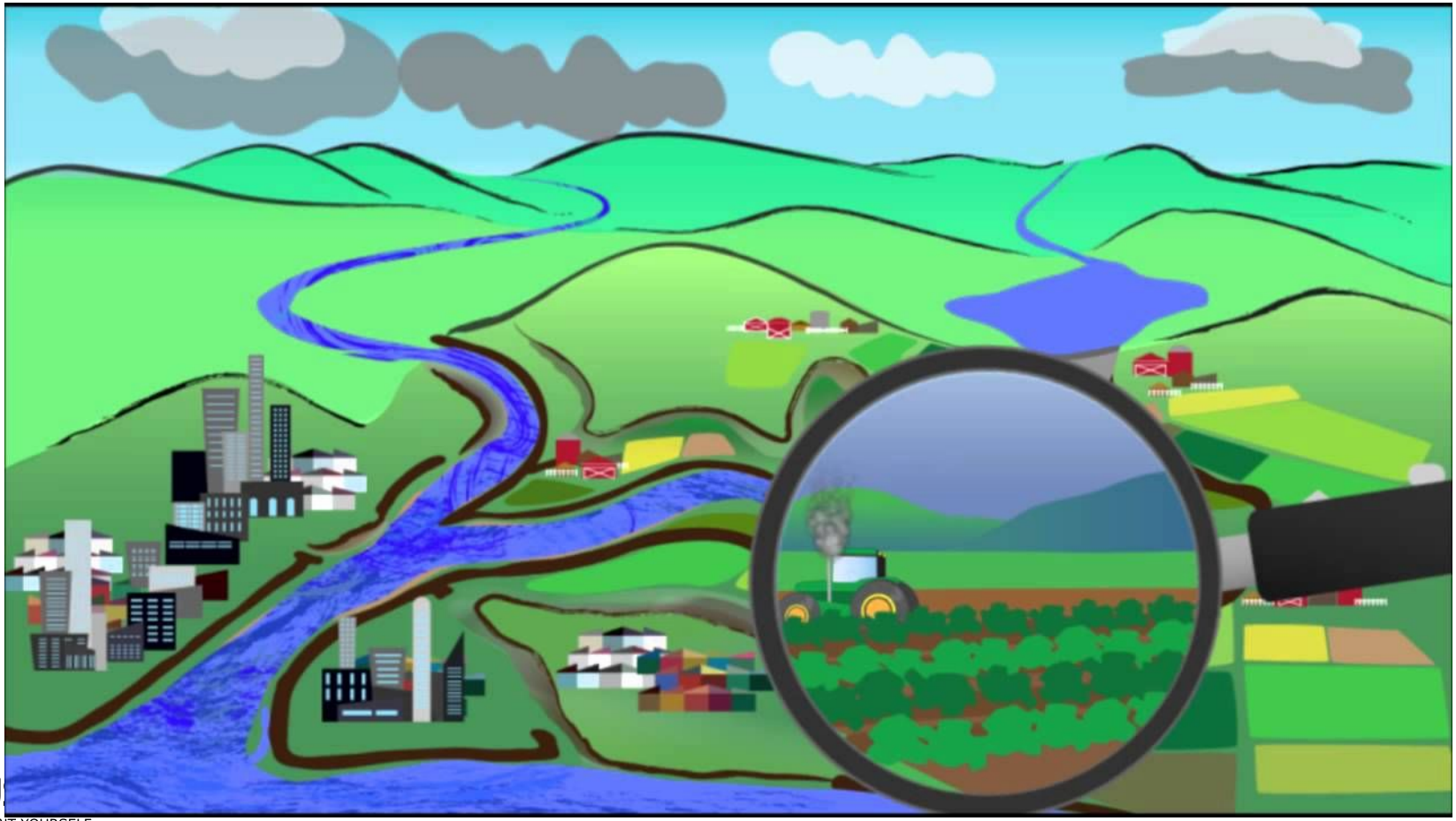


"Hon, come quick! I think we have a major computer bug."

Hayatım, çabuk geeelll!

Sanırım büyük bir bug'ımız var.

Application of the Waterfall Model



Waterfall Model

Advantages

Waterfall Model

Advantages

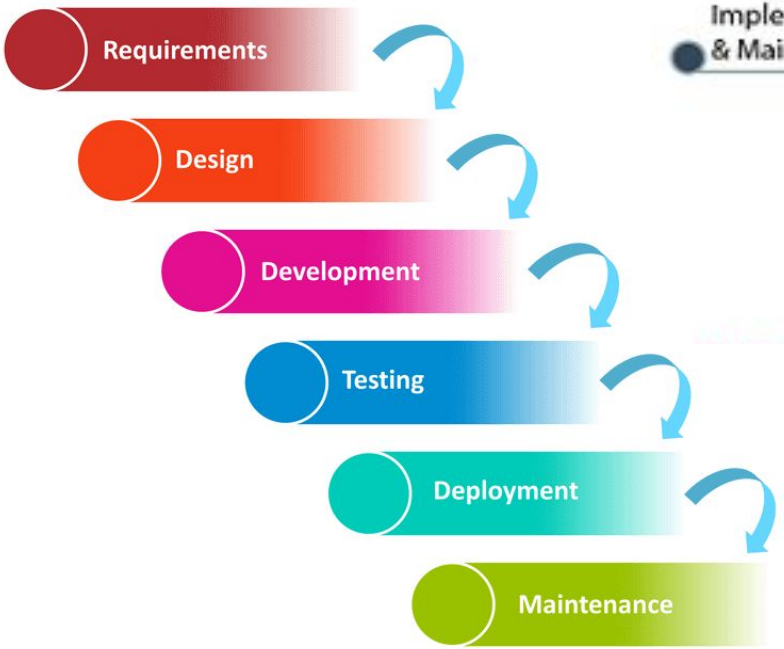
Waterfall Model

disadvantages

Waterfall Model

disadvantages

Summary


Requirements
Design
Development
Testing
Deployment
Maintenance


1
2
Requirements Gathering
Analysis
6 & 7
SDLC
3
Implementation & Maintenance
Design
5
4
Testing
Coding

C

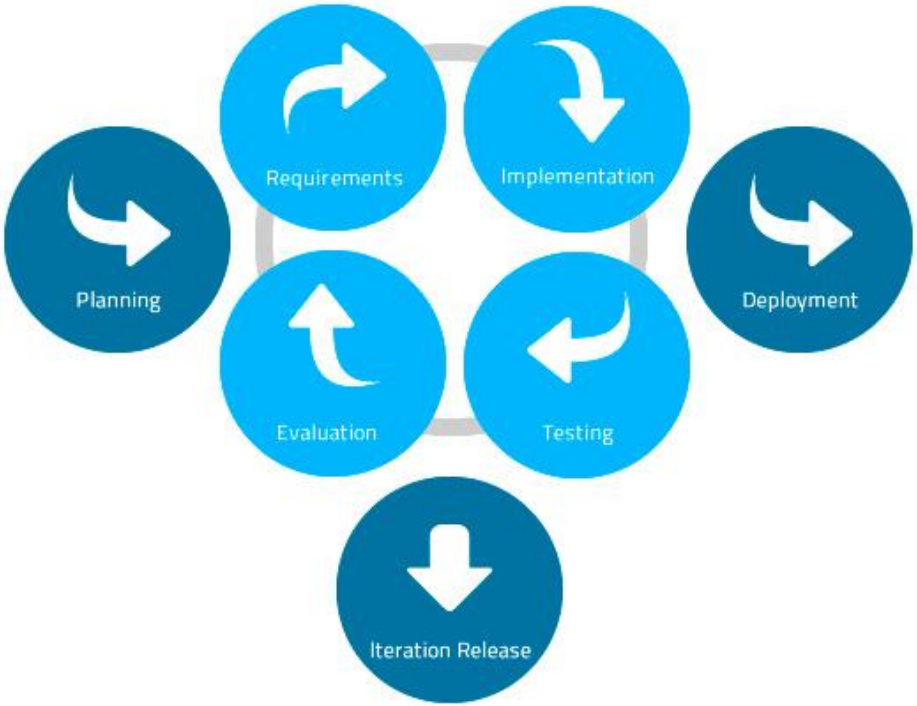

Kahoot!

THANKS!

Any questions?

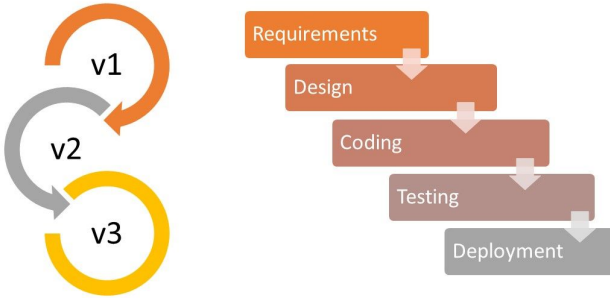
4 Iterative Model

Iterative Model



Iterative Model

Iteration vs Waterfall



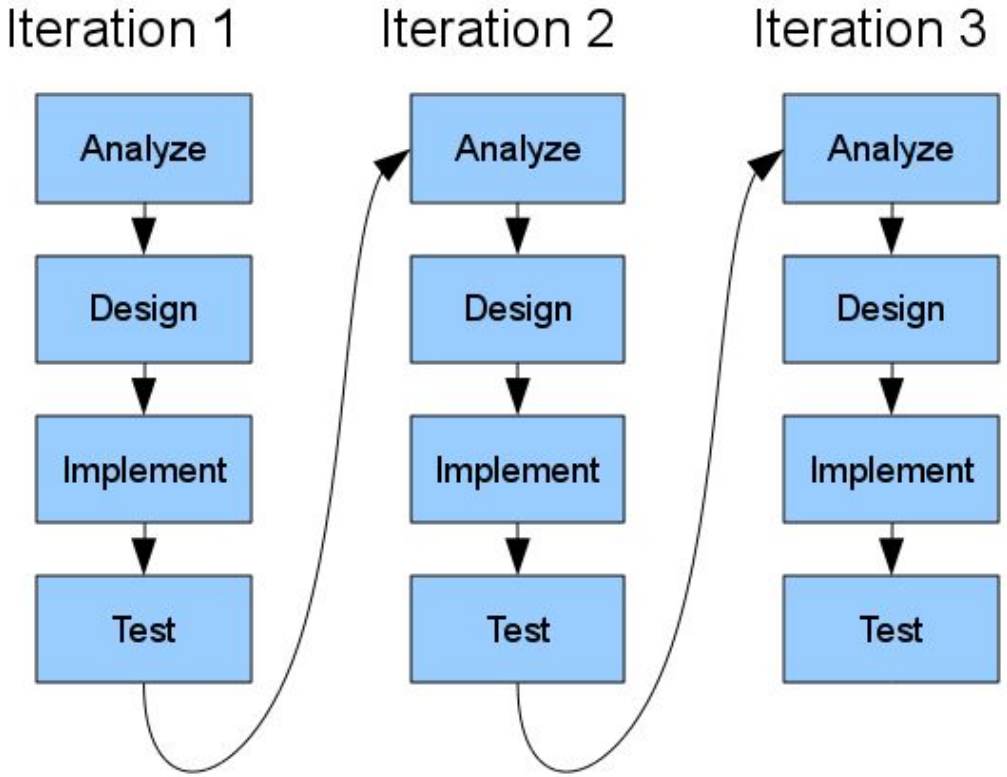
When we work **iteratively** we create rough product or product piece in one iteration

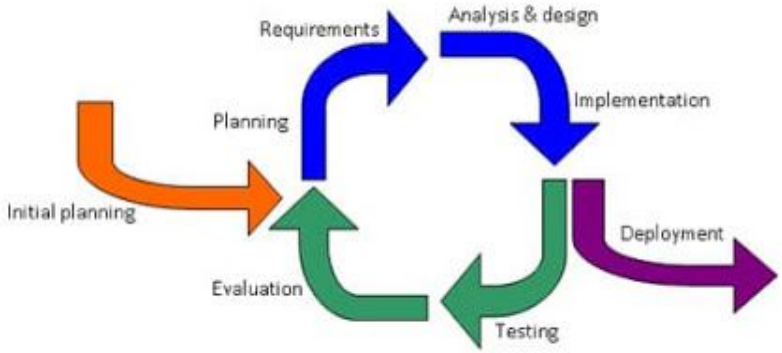
then review it and improve it in next iteration and so on until it's finished

- In the first iteration the whole painting is sketched roughly
- Then in the second iteration colors are filled
- In the third iteration finishing is done

The whole product is developed step by step

Iterative Model

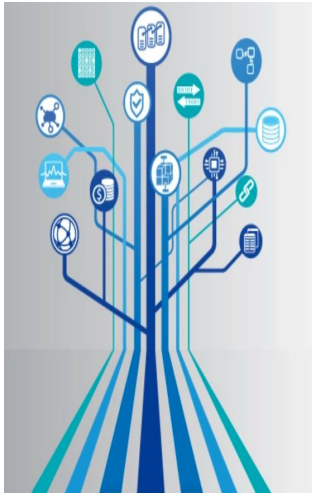




Model 1: Typical iterative development process

Iterative Model

advantages



Iterative Model

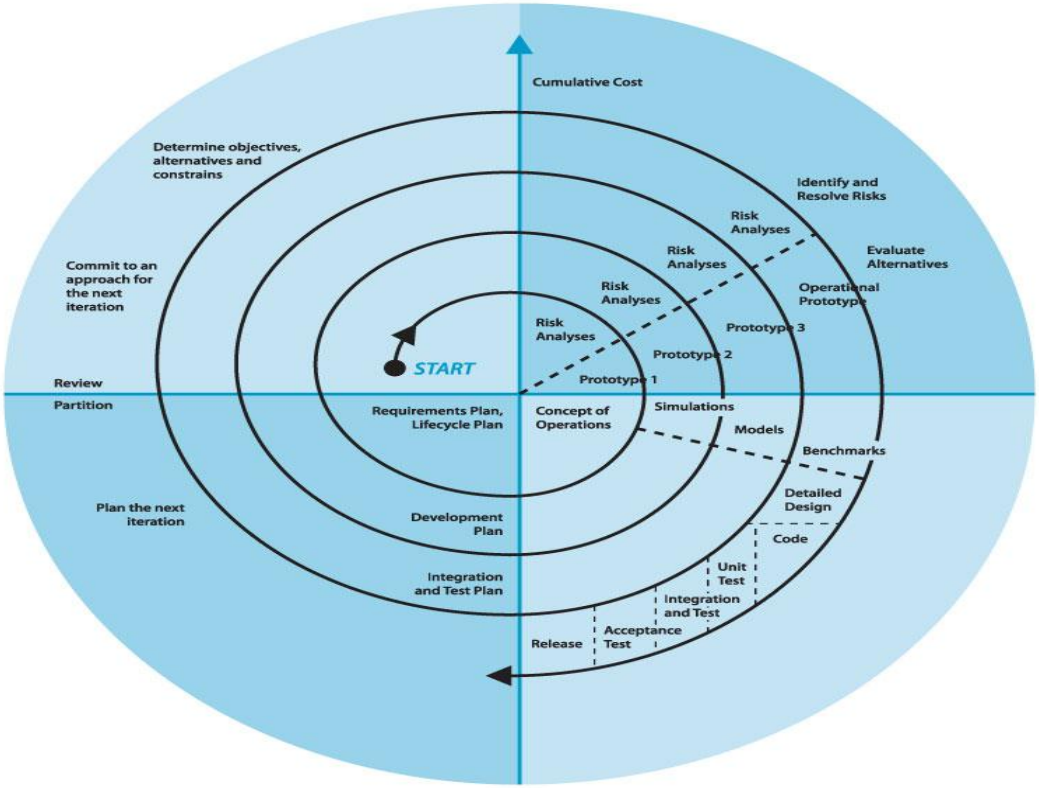
disadvantages

Iterative Model

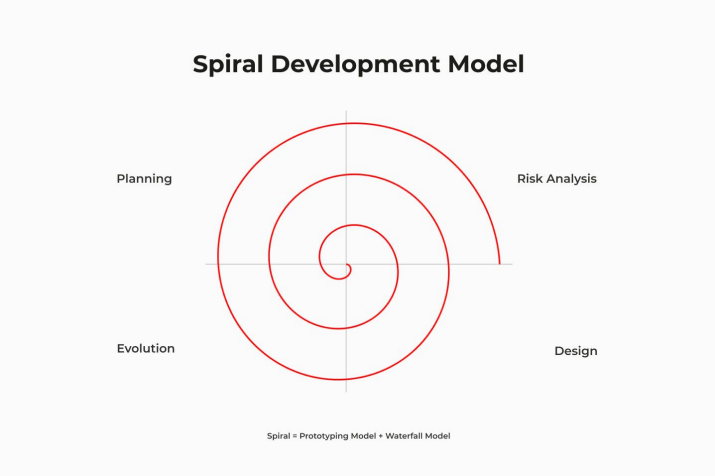
Pros	Cons
▪ Some working functionality can be developed quickly and early in the life cycle. ▪ Results are obtained early and periodically. ▪ Parallel development can be planned. ▪ Progress can be measured. ▪ Less costly to change the scope/requirements. ▪ Testing and debugging during smaller iteration is easy. ▪ Risks are identified and resolved during iteration; and each iteration is an easily managed milestone. ▪ Easier to manage risk - High risk part is done first. ▪ With every increment operational product is delivered. ▪ Issues, challenges & risks identified from each increment can be utilized/applied to the next increment. ▪ Risk analysis is better. ▪ It supports changing requirements. ▪ Initial Operating time is less. ▪ Better suited for large and mission-critical projects. ▪ During life cycle software is produced early which facilitates customer evaluation and feedback.	▪ More resources may be required. ▪ Although cost of change is lesser but it is not very suitable for changing requirements. ▪ More management attention is required. ▪ System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle. ▪ Defining increments may require definition of the complete system. ▪ Not suitable for smaller projects. ▪ Management complexity is more. ▪ End of project may not be known which is a risk. ▪ Highly skilled resources are required for risk analysis. ▪ Project's progress is highly dependent upon the risk analysis phase.

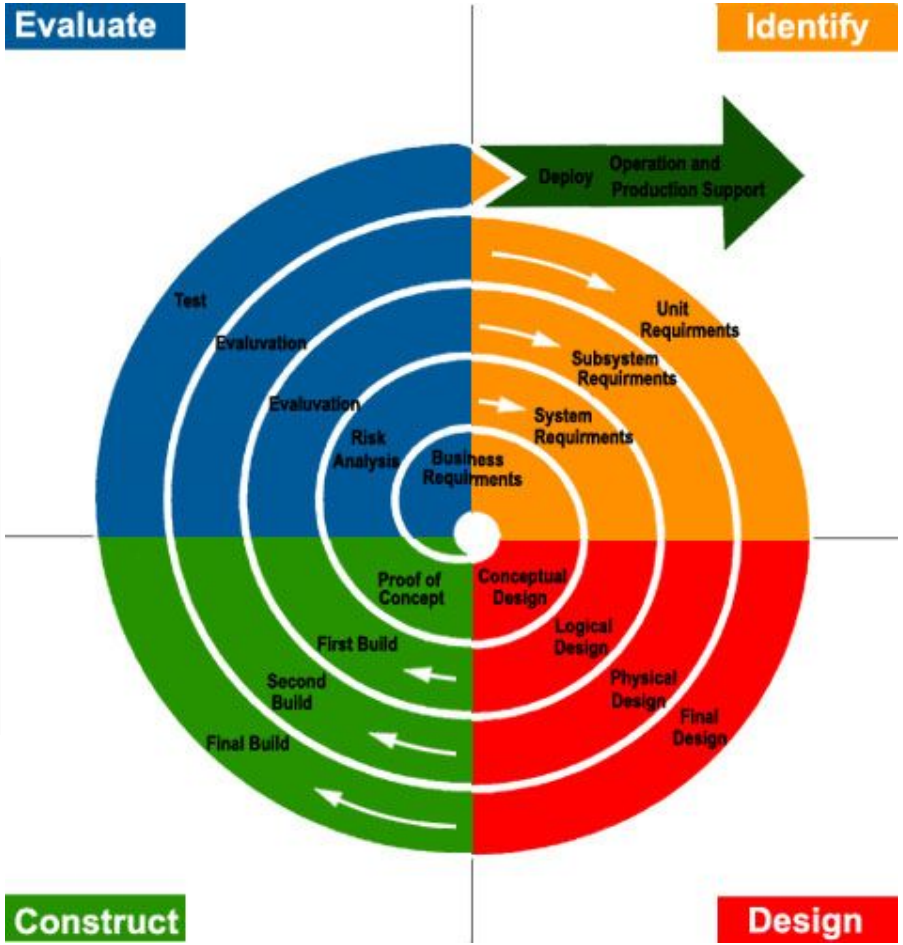
4 Spiral Model

Spiral Model



Spiral Model



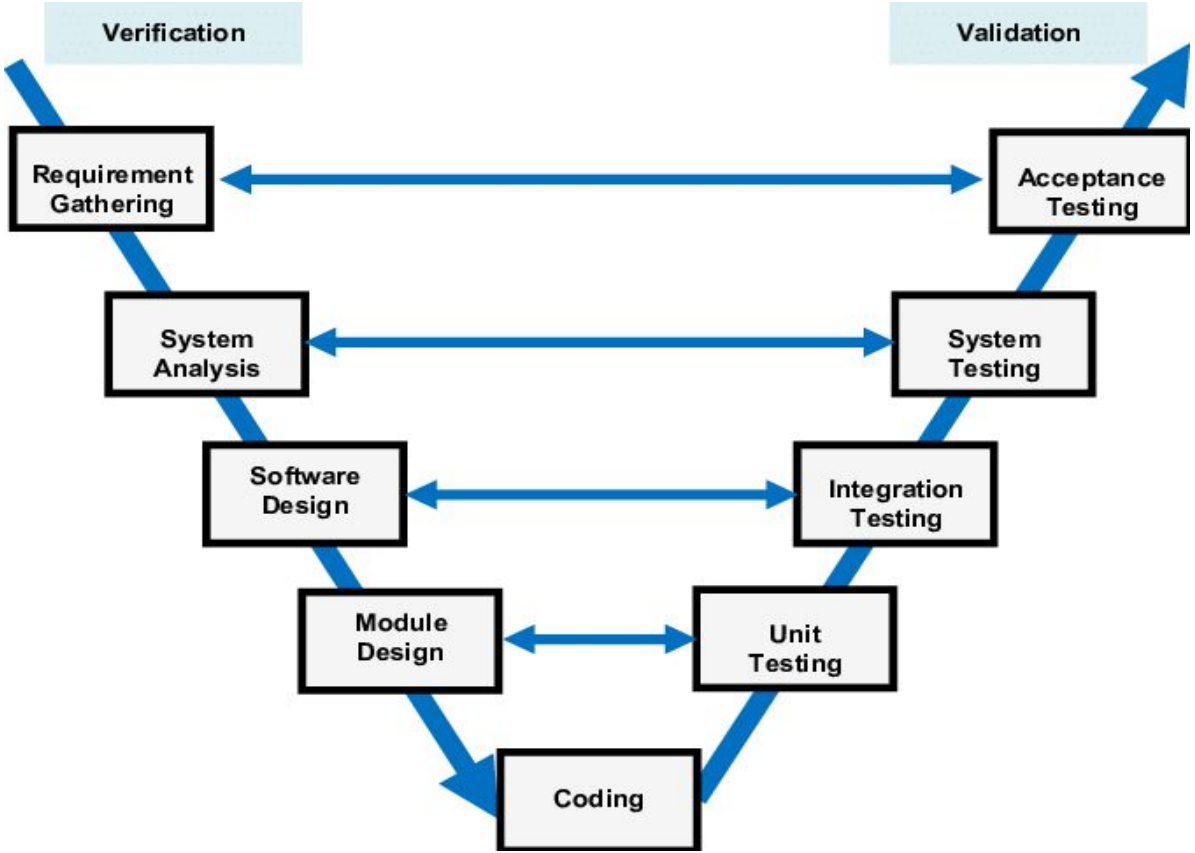


Spiral Model

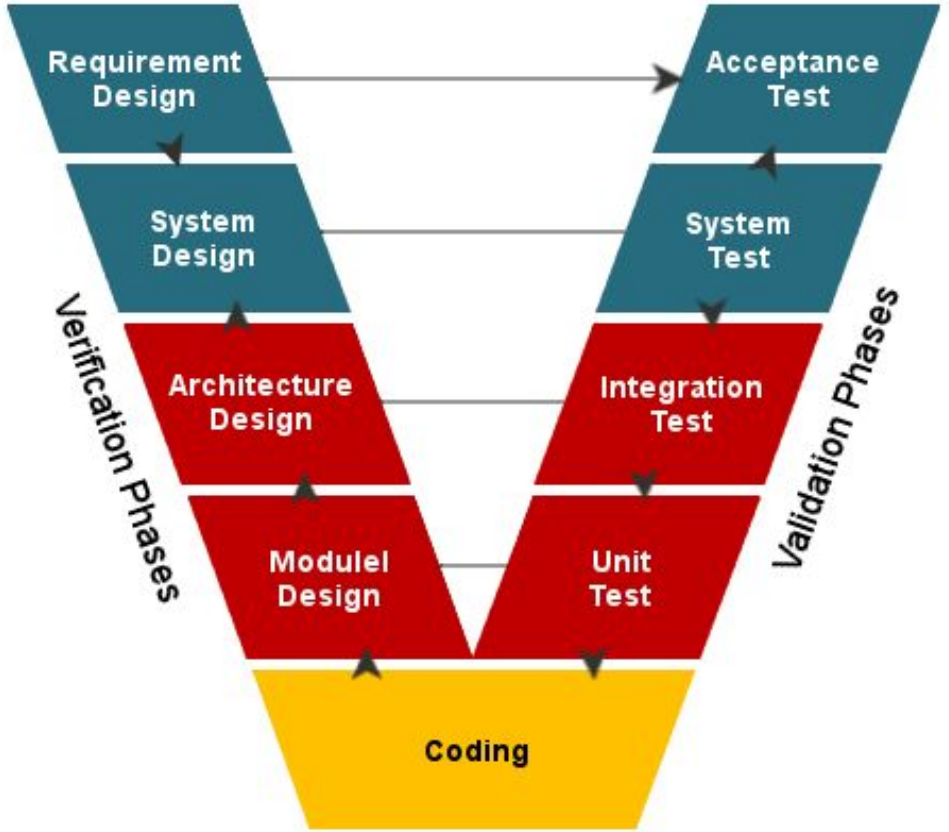
Pros	Cons
▪ Changing requirements can be accommodated. ▪ Allows for extensive use of prototypes ▪ Requirements can be captured more accurately. ▪ Users see the system early. ▪ Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management.	▪ Management is more complex. ▪ End of project may not be known early. ▪ Not suitable for small or low risk projects and could be expensive for small projects. ▪ Process is complex ▪ Spiral may go indefinitely. ▪ Large number of intermediate stages requires excessive documentation.

4 V - Model

V - Model

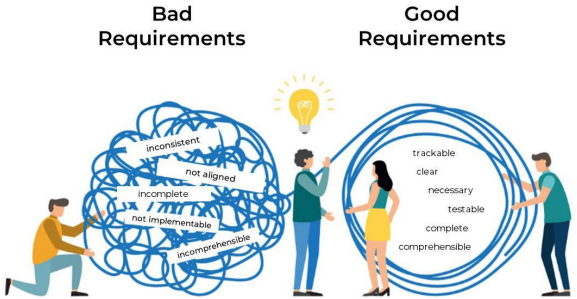

Verification
Validation
Requirement Gathering
Acceptance Testing
System Analysis
System Testing
Software Design
Integration Testing
Module Design
Unit Testing
Coding

V - Model


Requirement Design
Acceptance Test
System Design
System Test
Architecture Design
Integration Test
Modulel Design
Unit Test
Coding
Verification Phases
Validation Phases

V - Model

advantages



V - Model

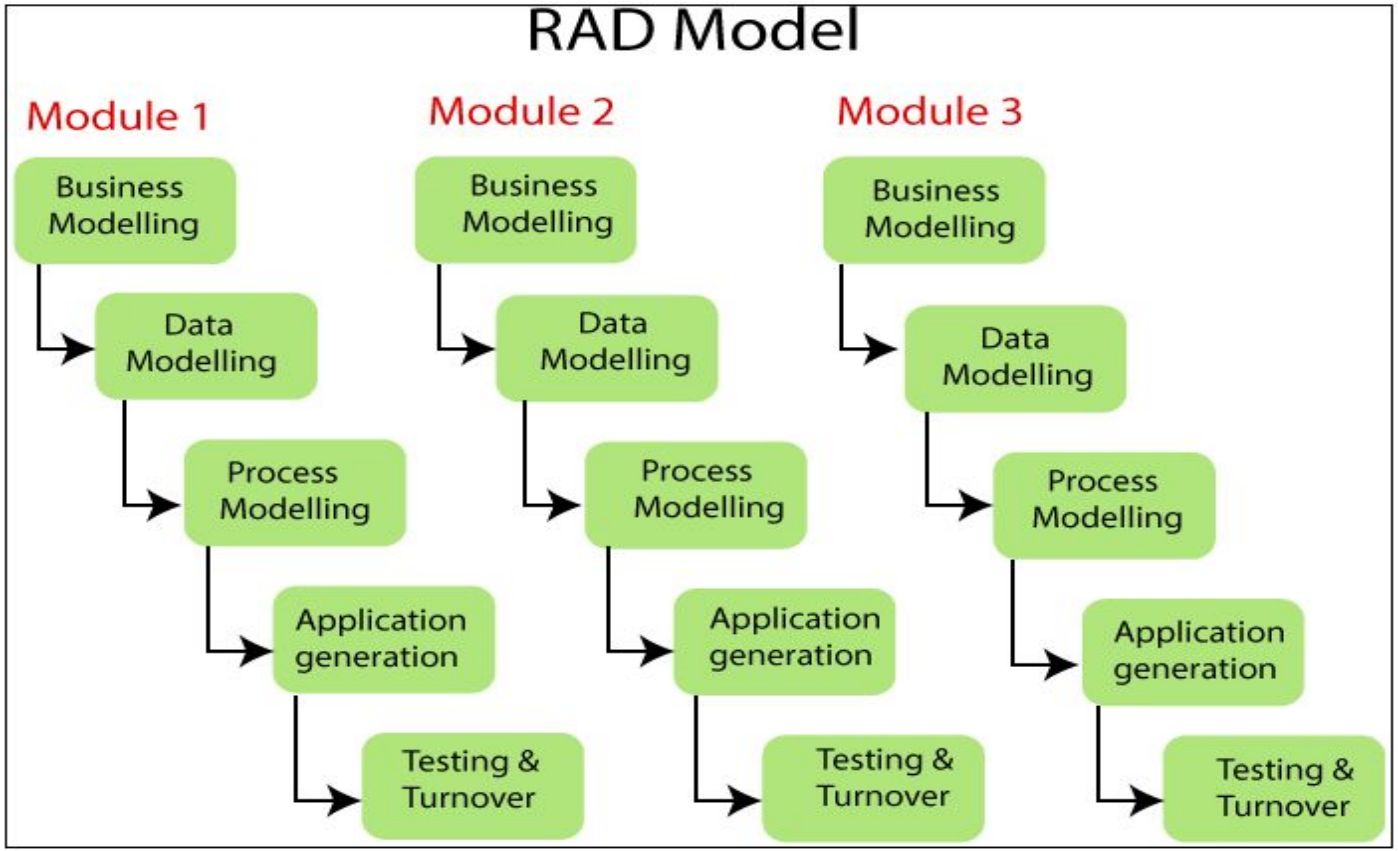
disadvantages

V - Model

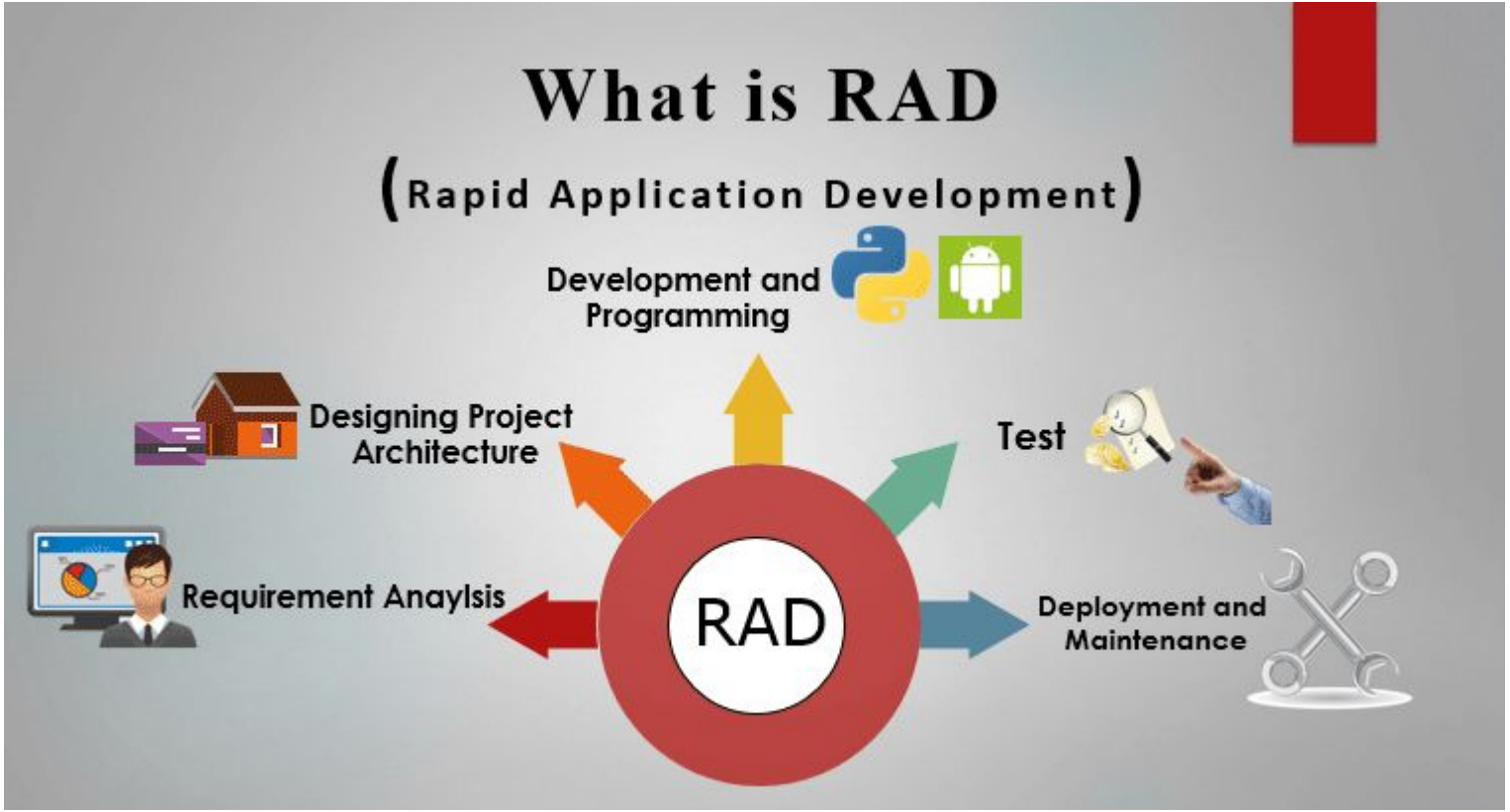
Pros	Cons
• This is a highly disciplined model and Phases are completed one at a time. • Works well for smaller projects where requirements are very well understood. • Simple and easy to understand and use. • Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.	▪ High risk and uncertainty. ▪ Not a good model for complex and object-oriented projects. ▪ Poor model for long and ongoing projects. ▪ Not suitable for the projects where requirements are at a moderate to high risk of changing.
	▪ Once an application is in the testing stage, it is difficult to go back and change a functionality ▪ No working software is produced until late during the life cycle.

4 RAD Model

RAD Model

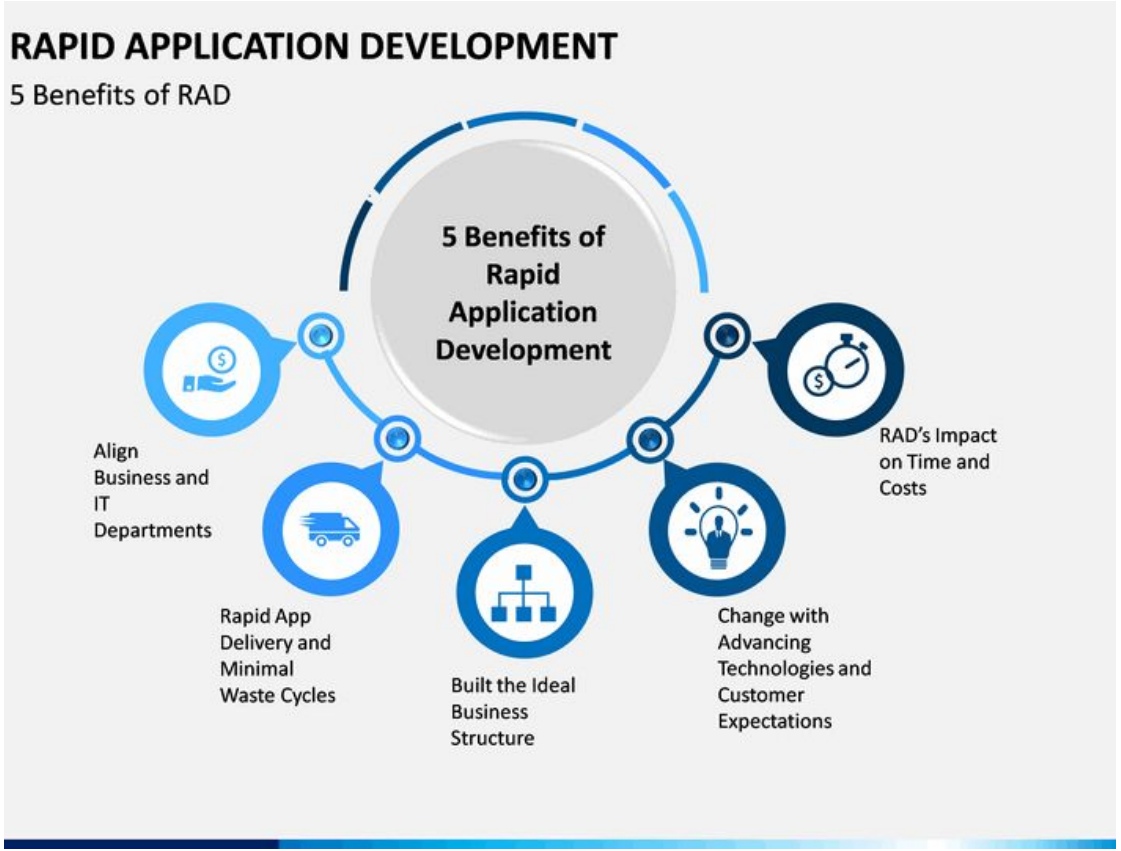

RAD Model
Module 1
Business Modelling
Data Modelling
Process Modelling
Application generation
Testing & Turnover
Module 2
Business Modelling
Data Modelling
Process Modelling
Application generation
Testing & Turnover
Module 3
Business Modelling
Data Modelling
Process Modelling
Application generation
Testing & Turnover

RAD Model

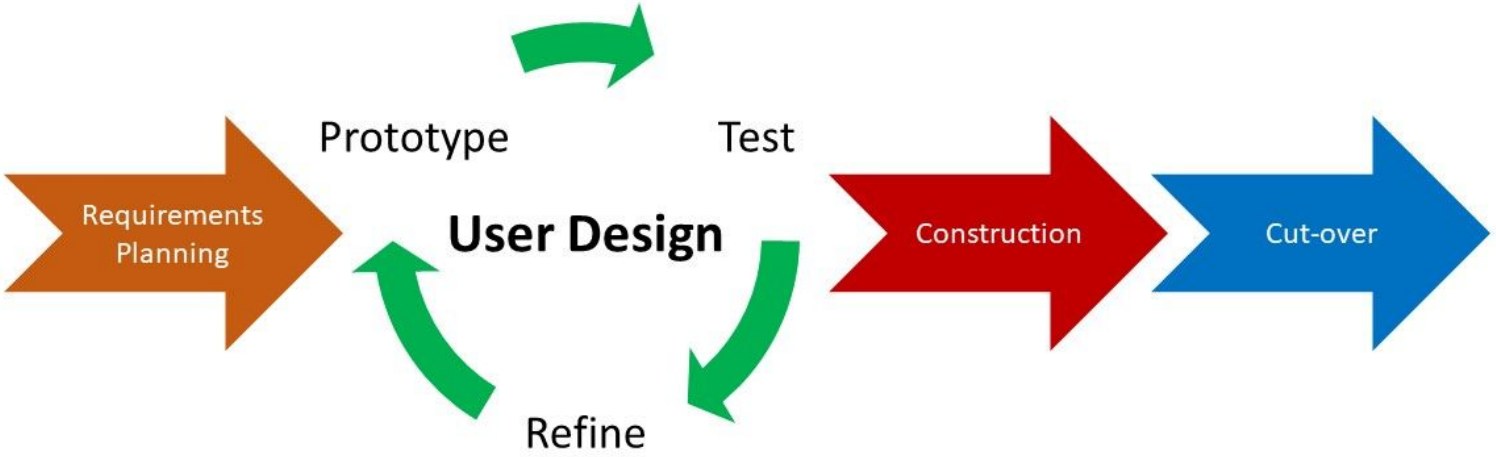

What is RAD
(Rapid Application Development)
Development and Programming
Designing Project Architecture
Test
Requirement Anaylsis
RAD
Deployment and Maintenance

RAD Model


RAD
Requirements Planning
ProtoTyping
Testing
Cut Over


RAPID APPLICATION DEVELOPMENT
5 Benefits of RAD
5 Benefits of Rapid Application Development
Align Business and IT Departments
Rapid App Delivery and Minimal Waste Cycles
Built the Ideal Business Structure
Change with Advancing Technologies and Customer Expectations
RAD's Impact on Time and Costs

RAD Model

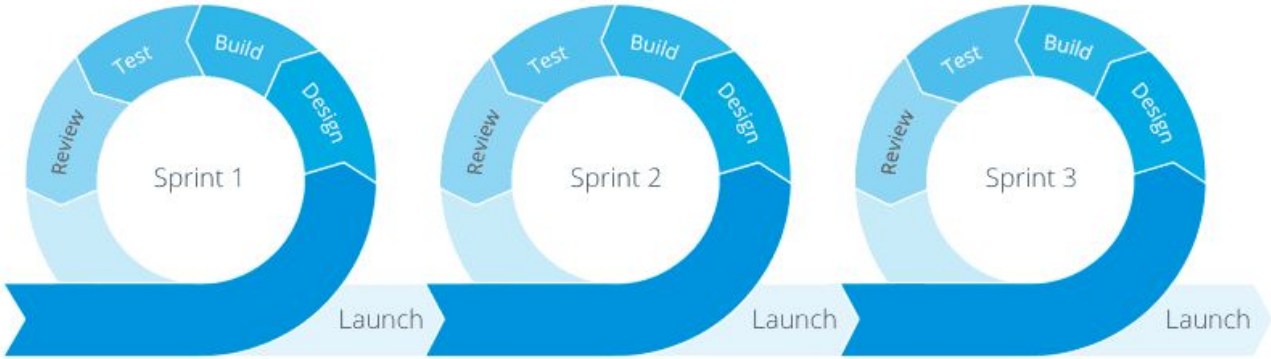

Prototype
Test
Requirements Planning
User Design
Construction
Cut-over
Refine

RAD Model

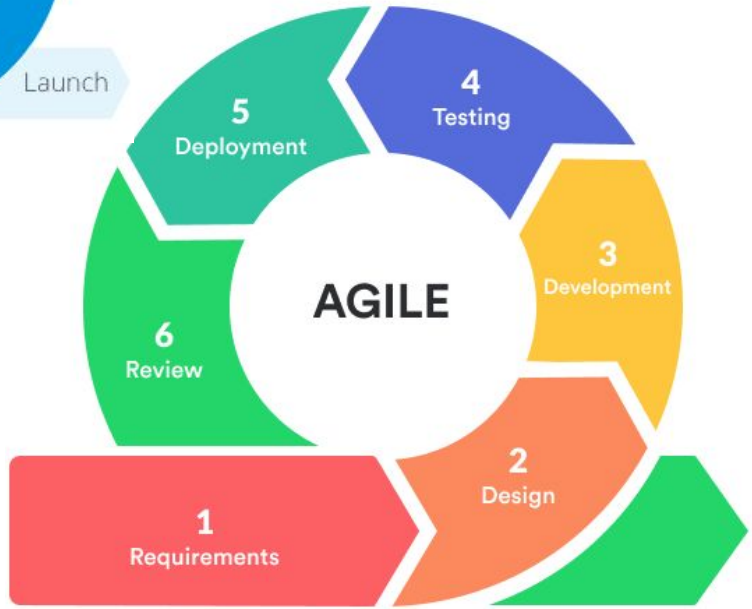
Pros	Cons
▪ Changing requirements can be accommodated. ▪ Progress can be measured. ▪ Iteration time can be short with use of powerful RAD tools. ▪ Productivity with fewer people in short time. ▪ Reduced development time. ▪ Increases reusability of components	▪ Dependency on technically strong team members for identifying business requirements. ▪ Only system that can be modularized can be built using RAD ▪ Requires highly skilled developers/designers. ▪ High dependency on modeling skills ▪ Inapplicable to cheaper projects as cost
▪ Quick initial reviews occur ▪ Encourages customer feedback ▪ Integration from very beginning solves a lot of integration issues.	of modeling and automated code generation is very high. ▪ Management complexity is more. ▪ Suitable for systems that are component based and scalable. ▪ Requires user involvement throughout the life cycle. ▪ Suitable for project requiring shorter development times.

Agile Model

Agile Model



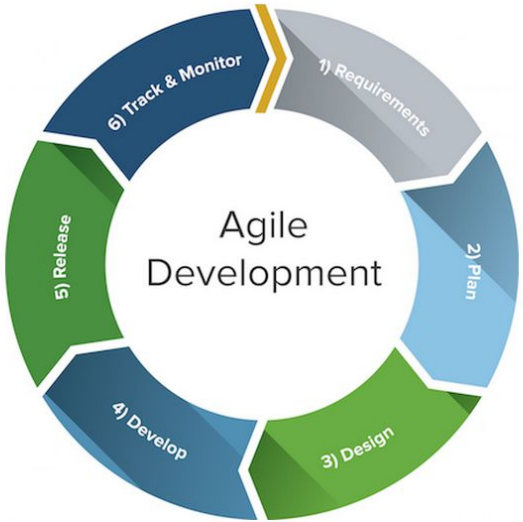
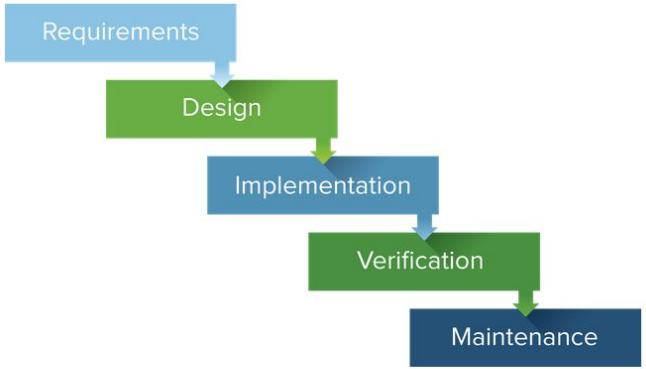
Each iteration lasts from 1 - 3 weeks



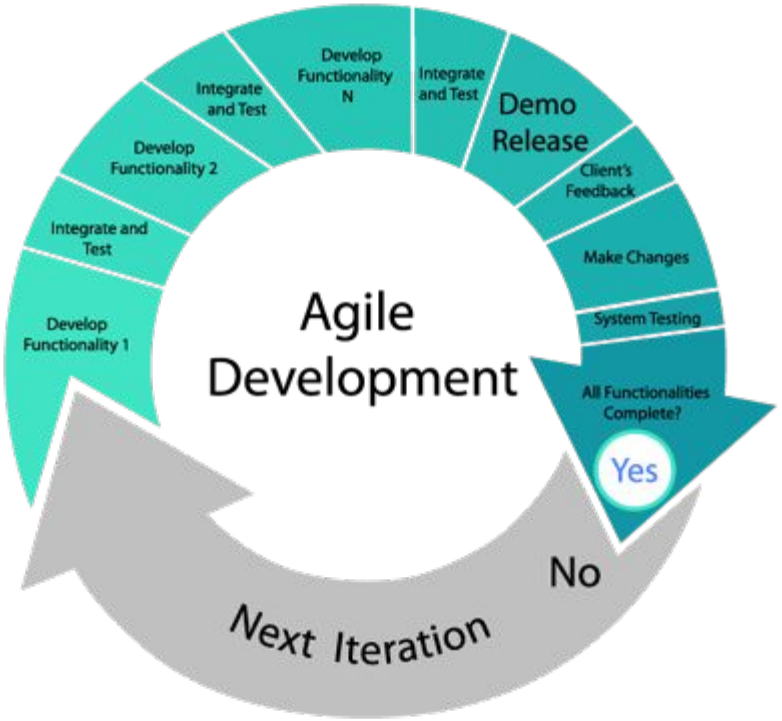
Agile Model

AGILE SOFTWARE DEVELOPMENT

Agile Vs Traditional SDLC Models

Agile	Waterfall
	
• Continuous cycles • Small, high-functioning, collaborative teams • Multiple methodologies • Flexible/continuous evolution • Customer involvement	• Sequential/linear stages • Upfront planning and in-depth documentation • Contract negotiation • Best for simple, unchanging projects • Close project manager involvement

Agile Vs Traditional SDLC Models



THANKS!

Any questions?