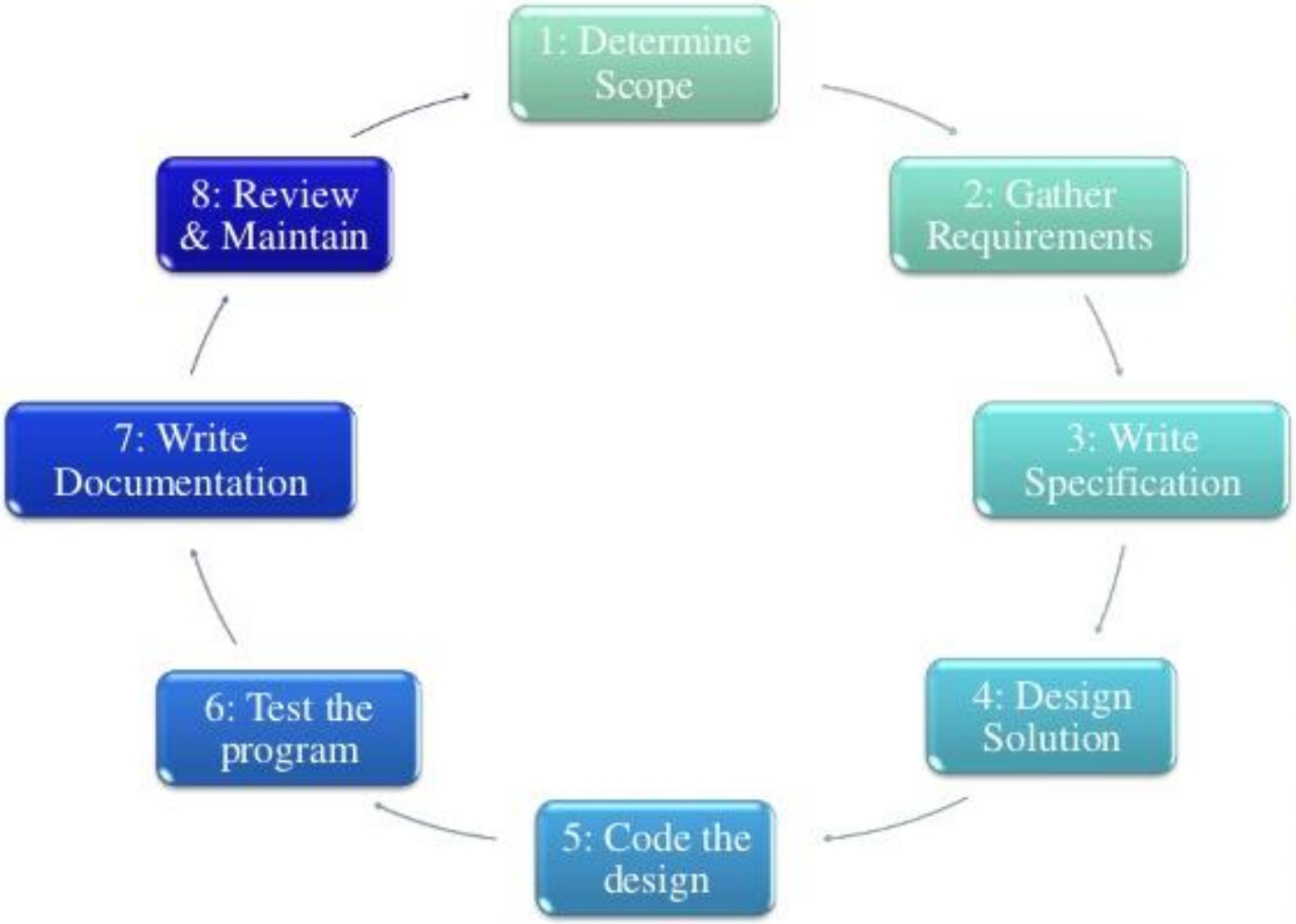
Software Development Process Life Cycles

Senior-lecturer
Nazgul R.K.

OUTLINE OF TALK

1. SW Development Process Stages
2. SW Development Process Stages and Artifacts
3. SW Development Process:
   Waterfall and Iterative

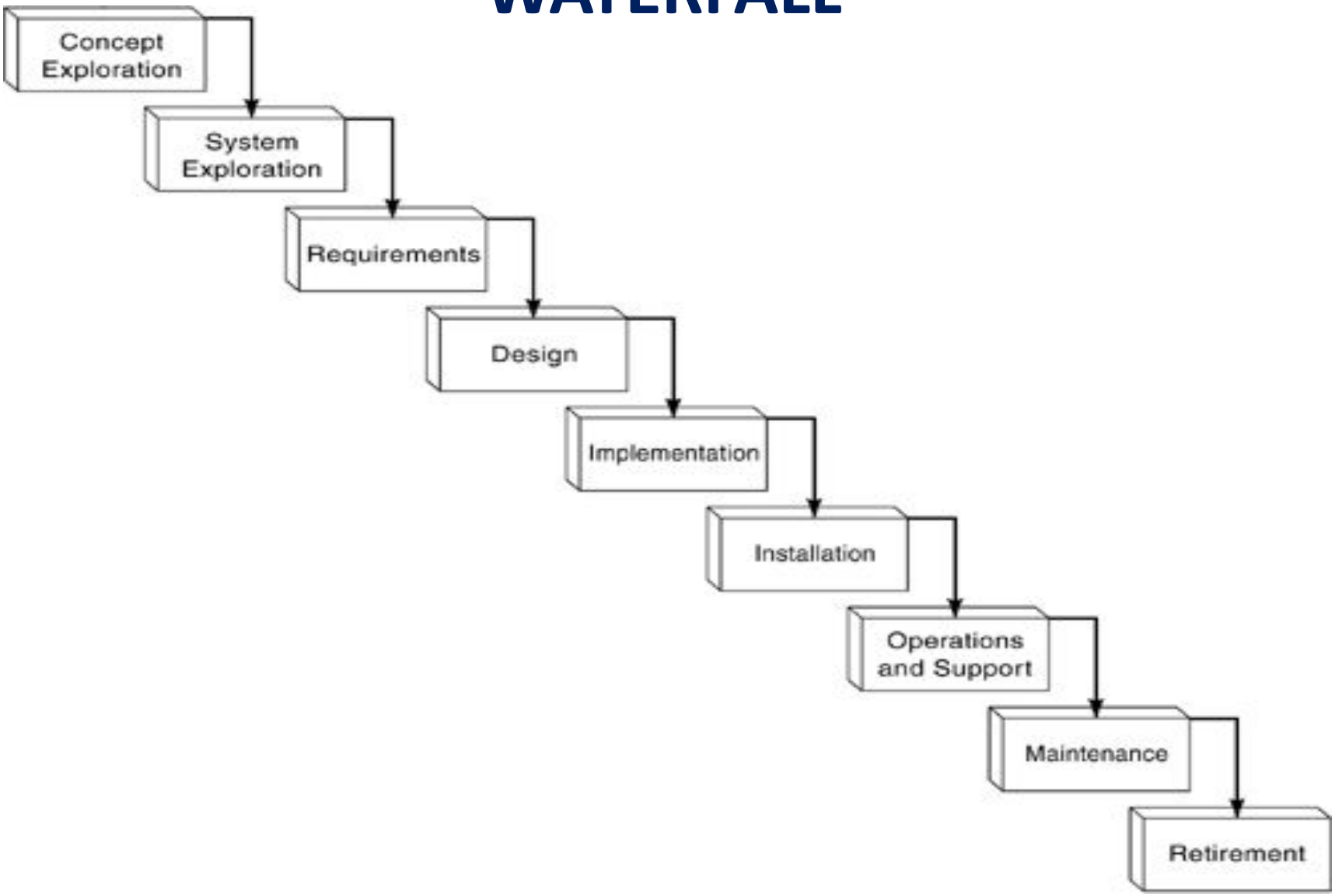
SDLC Stages


1: Determine Scope
2: Gather Requirements
3: Write Specification
4: Design Solution
5: Code the design
6: Test the program
7: Write Documentation
8: Review & Maintain

SOFTWARE DEVELOPMENT METODOLOGIES

- Waterfall
- Agile
- V-Model
- Iterative
- Incremental
- Rapid Application Development (RAD)
- Spiral

WATERFALL


Concept Exploration
System Exploration
Requirements
Design
Implementation
Installation
Operations and Support
Maintenance
Retirement

WATERFALL

It is also referred to as a **linear-sequential life cycle model**.

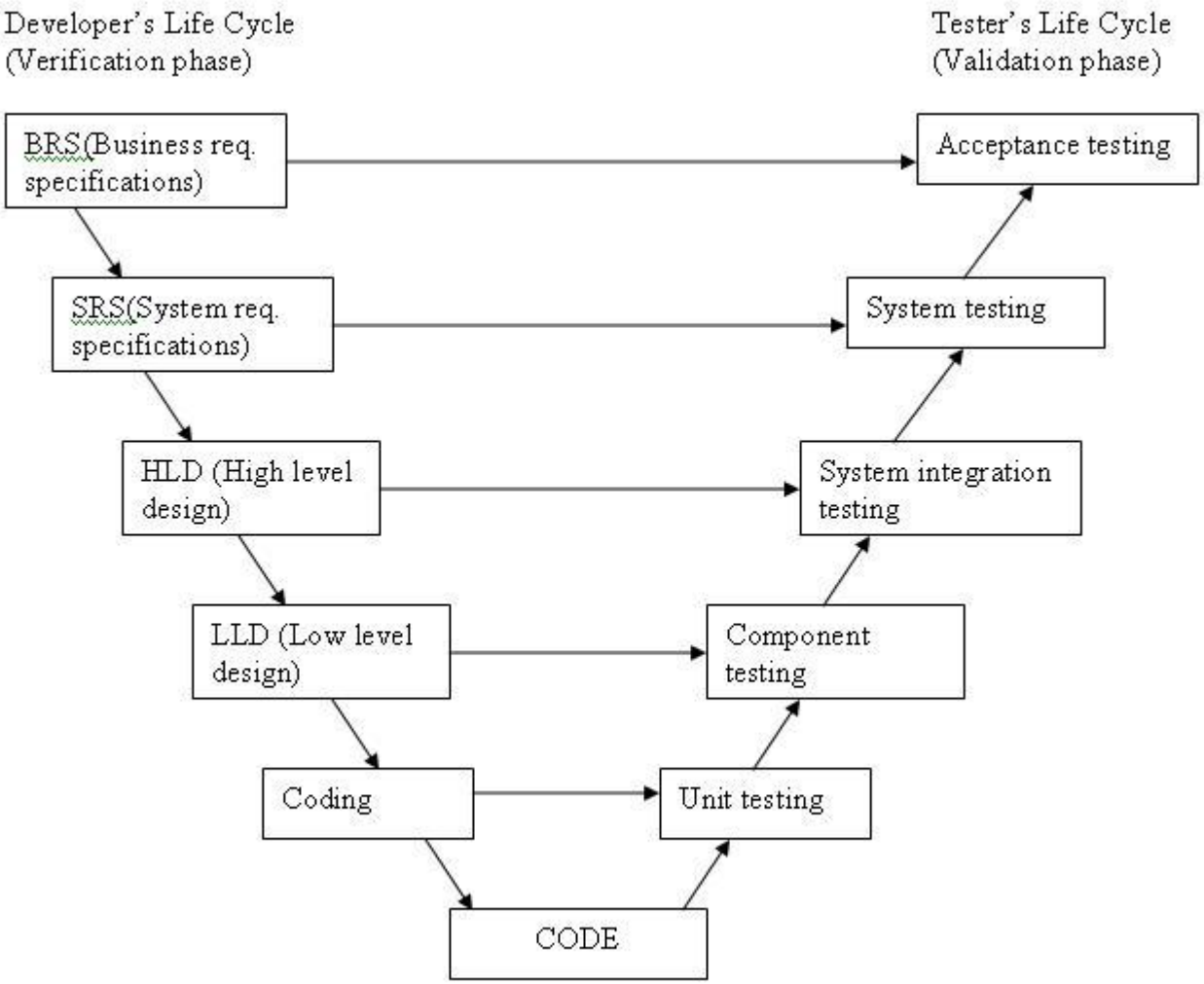
In a waterfall model, each phase must be completed fully before the next phase can begin.

This type of model is basically used for the for the project which is small and there are no uncertain requirements.

In this model the testing starts only after the development is complete.

In **waterfall model phases** do not overlap.

V- model


Developer's Life Cycle
(Verification phase)
Tester's Life Cycle
(Validation phase)
BRS(Business req. specifications)
Acceptance testing
SRS(System req. specifications)
System testing
HLD (High level design)
System integration testing
LLD (Low level design)
Component testing
Coding
Unit testing
CODE

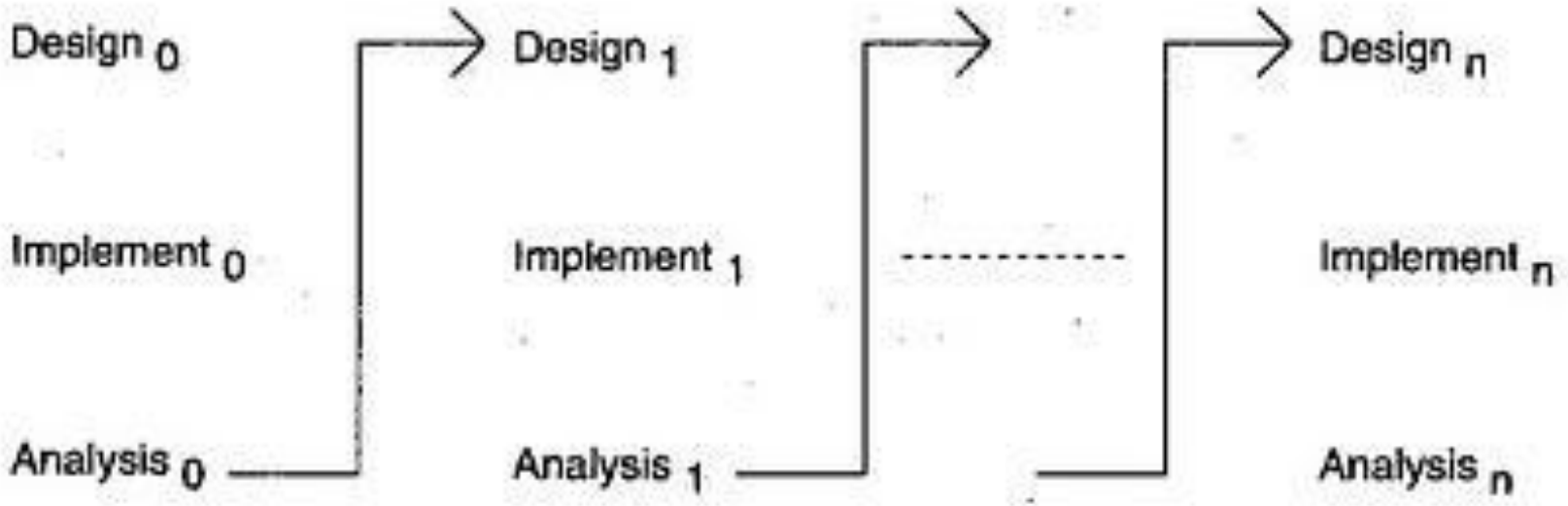
V-model

Verification and Validation model.

Testing of the product is planned in parallel with a corresponding phase of development in **V-model**.

Before development is started, a **system test** plan is created. The test plan focuses on meeting the functionality specified in the requirements gathering.

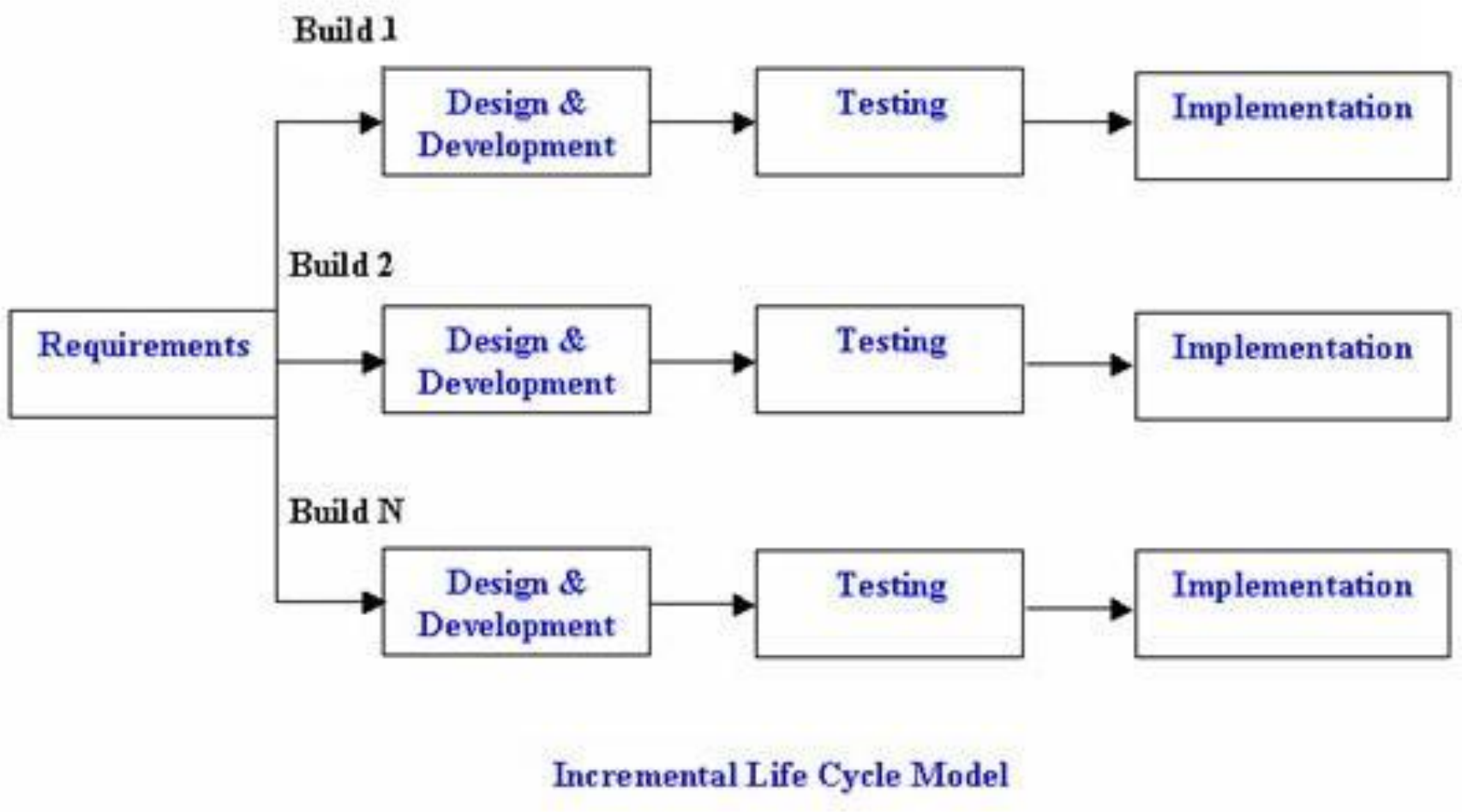
ITERATIVE


Design 0
Implement 0
Analysis 0
Design 1
Implement 1
Analysis 1
Design n
Implement n
Analysis n

ITERATIVE

Development begins by specifying and implementing **just part of the software**, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

INCREMENTAL



Incremental Life Cycle Model

INCREMENTAL

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

Incremental model

1 

2 

3 

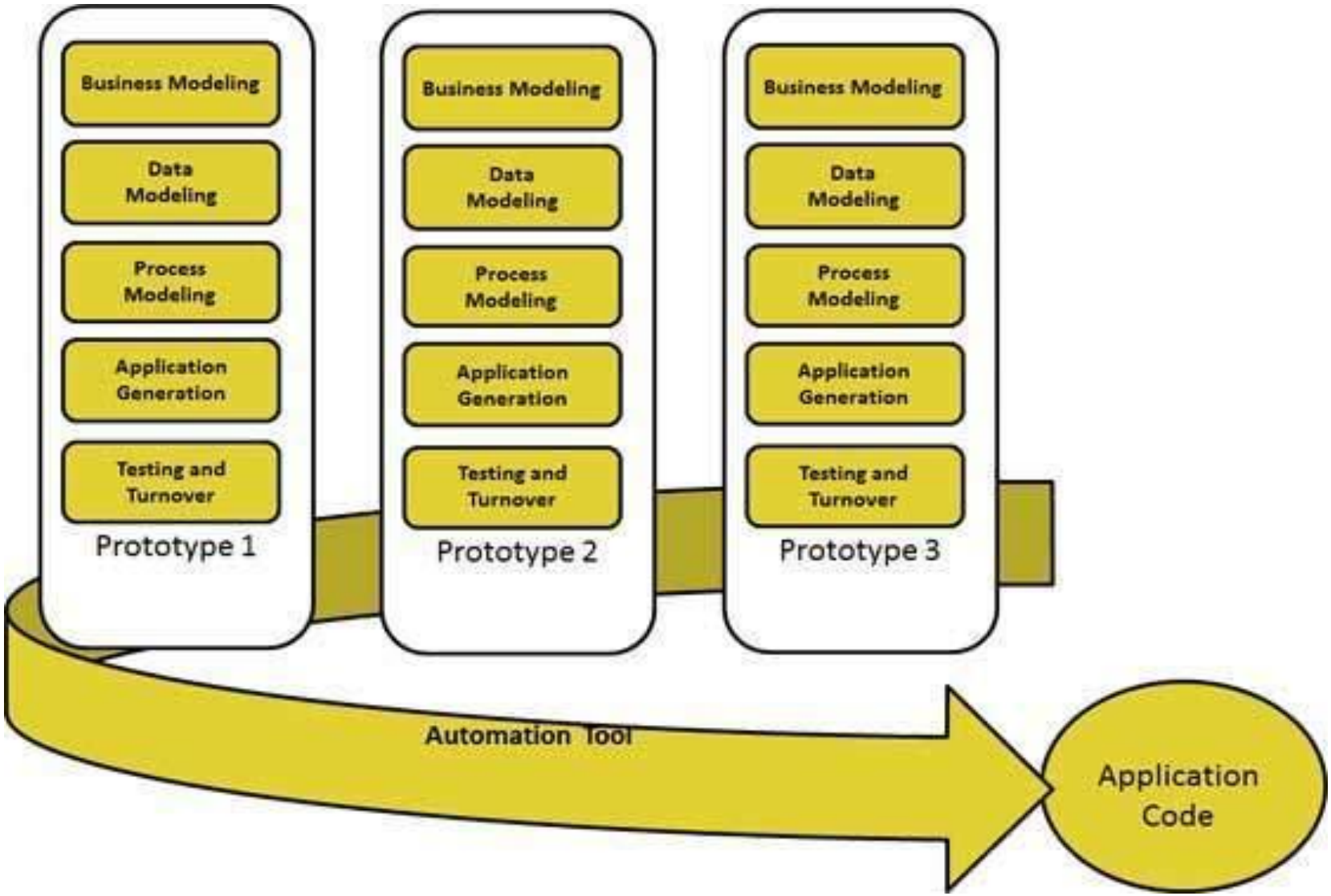
Iterative model

1 

2 

3 

RAD


Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing and Turnover
Prototype 1
Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing and Turnover
Prototype 2
Business Modeling
Data Modeling
Process Modeling
Application Generation
Testing and Turnover
Prototype 3
Automation Tool
Application Code

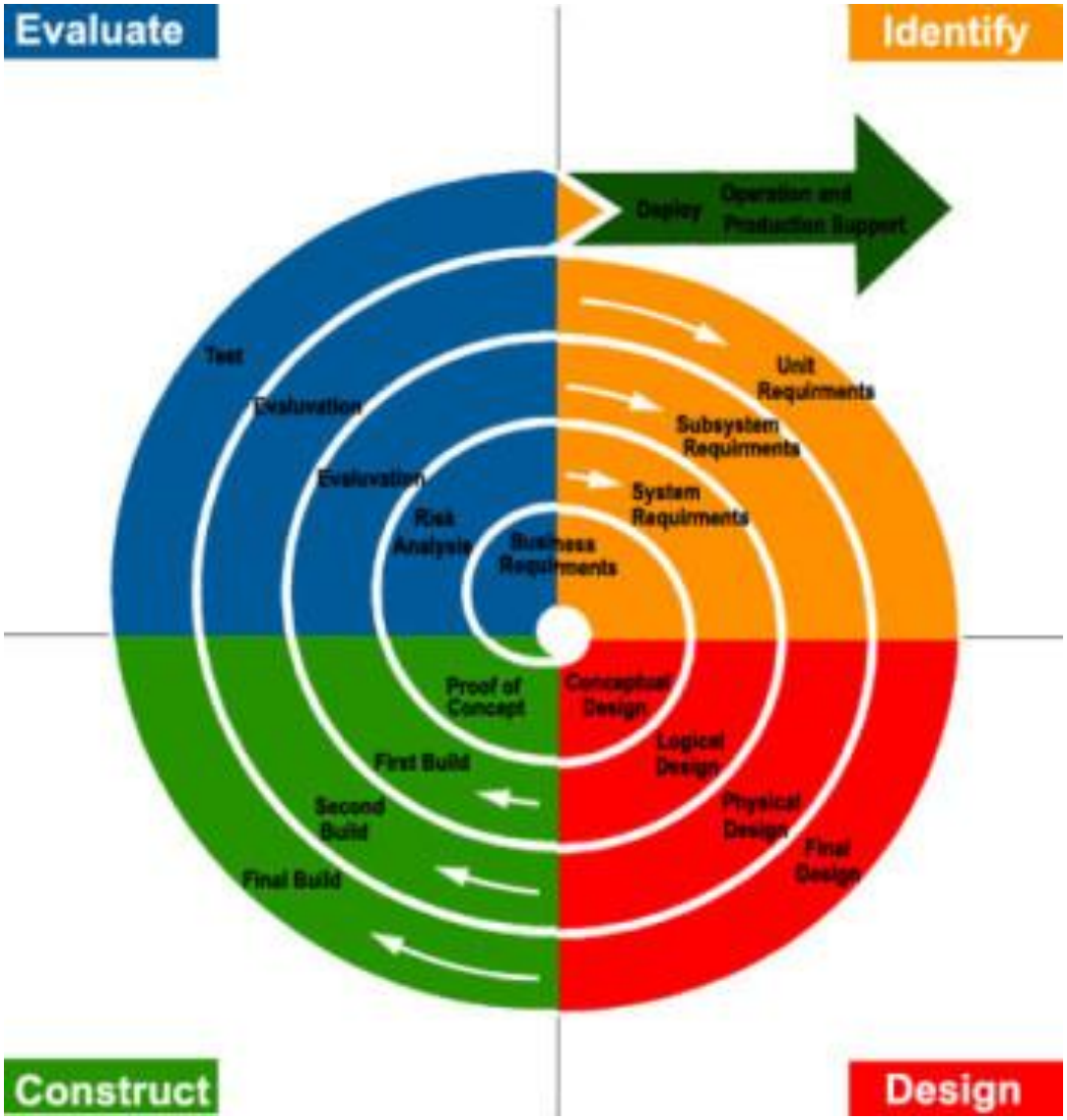
RAD

It is a type of **incremental model**. In RAD model the components or functions are developed in parallel as if they were mini projects.

The developments are time boxed, delivered and then assembled into a working prototype.

This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

SPIRAL


Evaluate
Identify
Deploy
Operation and Production Support
Test
Evaluation
Evaluation
Risk Analysis
Business Requirements
Unit Requirments
Subsystem Requirments
System Requirments
Proof of Concept
Conceptual Design
Logical Design
Physical Design
Final Design
First Build
Second Build
Final Build
Construct
Design

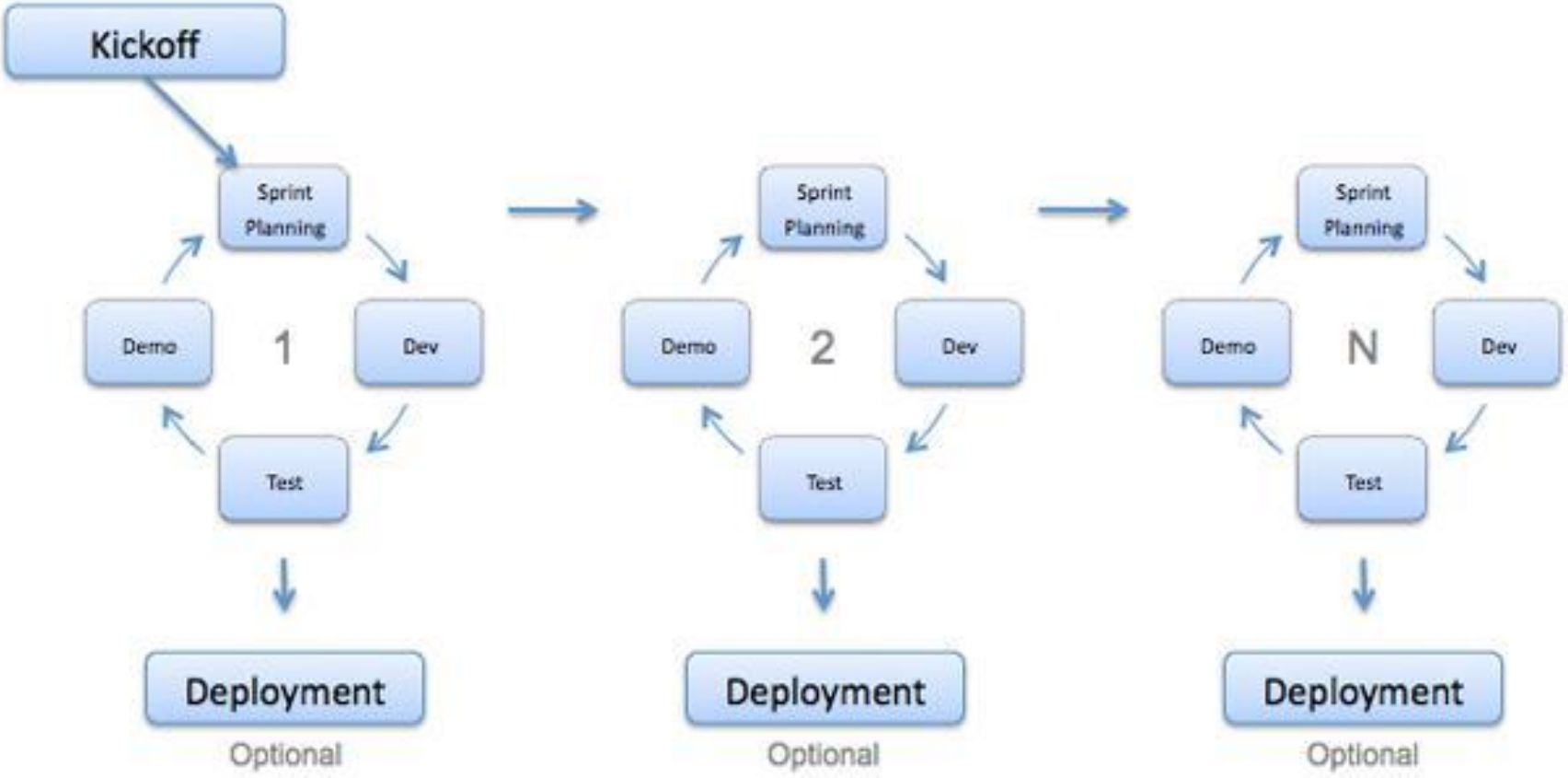
SPIRAL

The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation.

A software project repeatedly passes through these phases in iterations (called Spirals in this model).

The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral

AGILE


Kickoff
Sprint Planning
Demo
1
Dev
Test
Deployment
Optional
Sprint Planning
Demo
2
Dev
Test
Deployment
Optional
Sprint Planning
Demo
N
Dev
Test
Deployment
Optional

AGILE

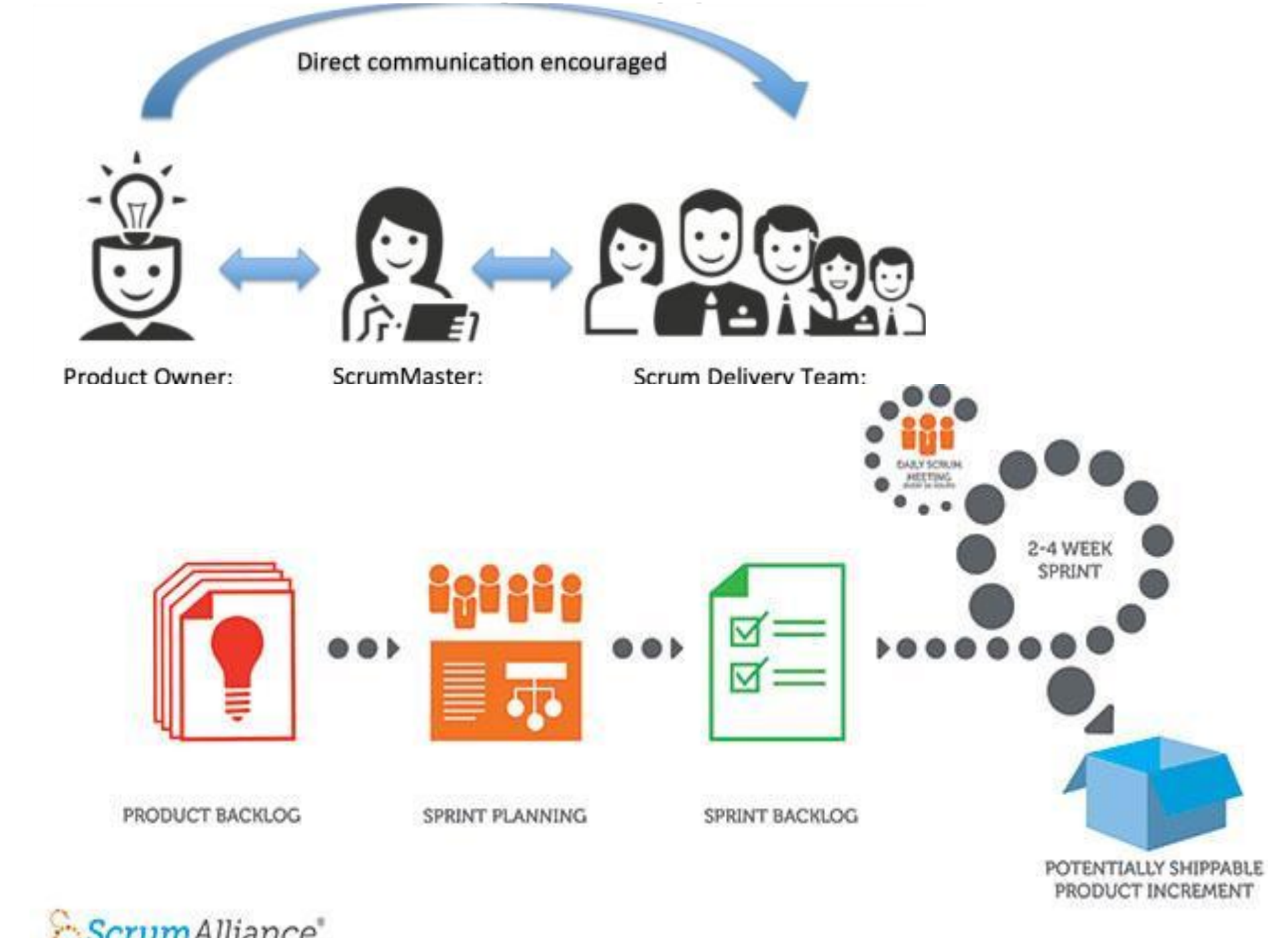
ASD - is an umbrella term for a set of methods and practices based on the **values** and **principles** expressed in the Agile Manifesto.

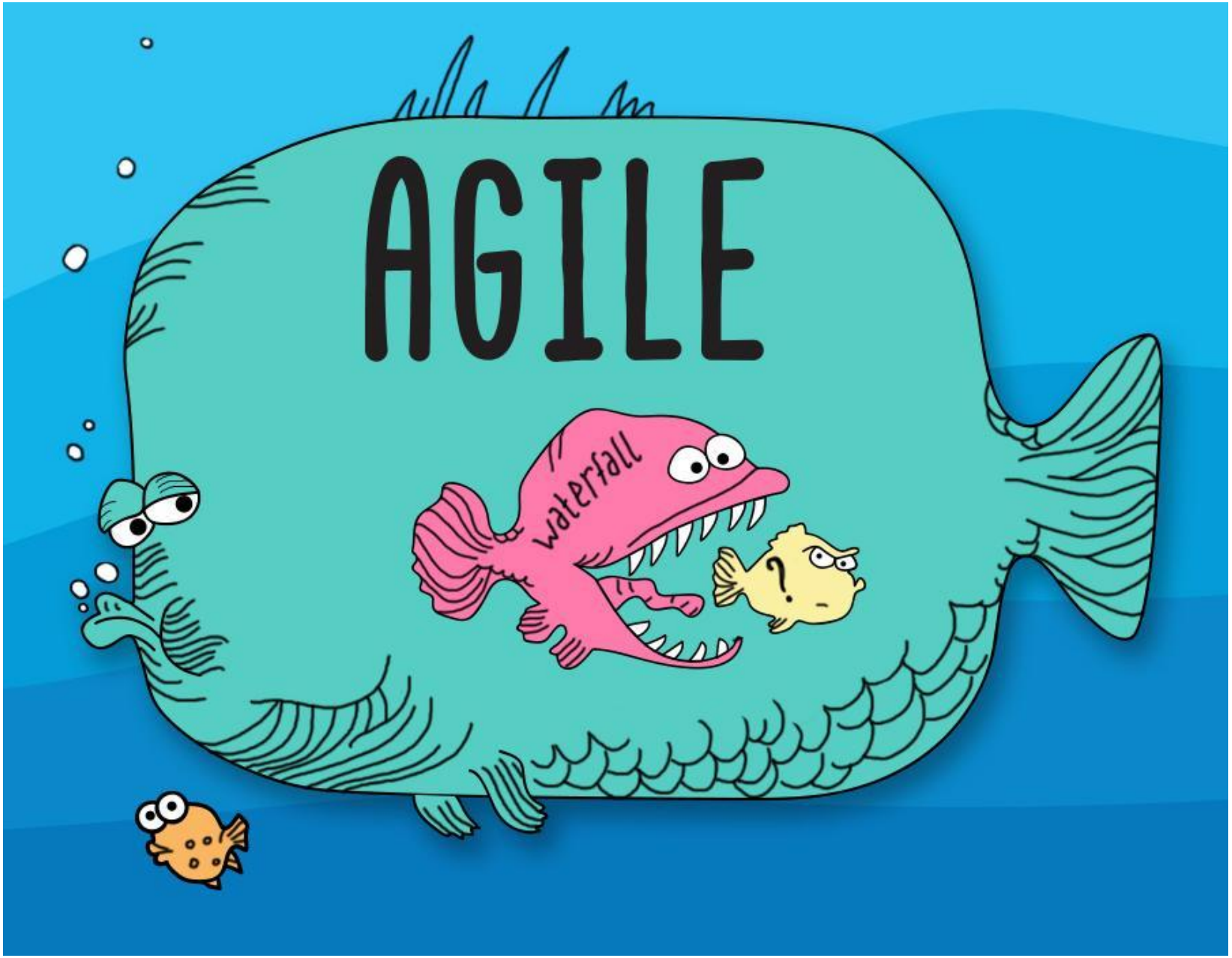
- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

AGILE

- *Extreme Programming*
- *Dynamic Systems Development Method*
- ***Scrum***

Roles in SCRUM




AGILE
waterfall

Summary

- How could software development methodologies been grouped?
- What is advantage and disadvantage for each of them?