CLEMSON UNIVERSITY

CPSC 3720
Lesson 4

Connie Taylor
Professor of Practice

School of COMPUTING

Today's Objectives

- Continue discussion of software lifecycle and processes
  - Major Phases Overview
  - Process Overview

The Tar Pit – Complexity of a Program vs. Product

Single program *Couple devs in a garage – used by the devs*	**Programming System** *Dependencies/ integration, performance testing*
Programming <u>Product</u> *General usage, testing, doc*	**Programming Systems <u>Product</u>** *Product+ Systems needs*

3x

3x

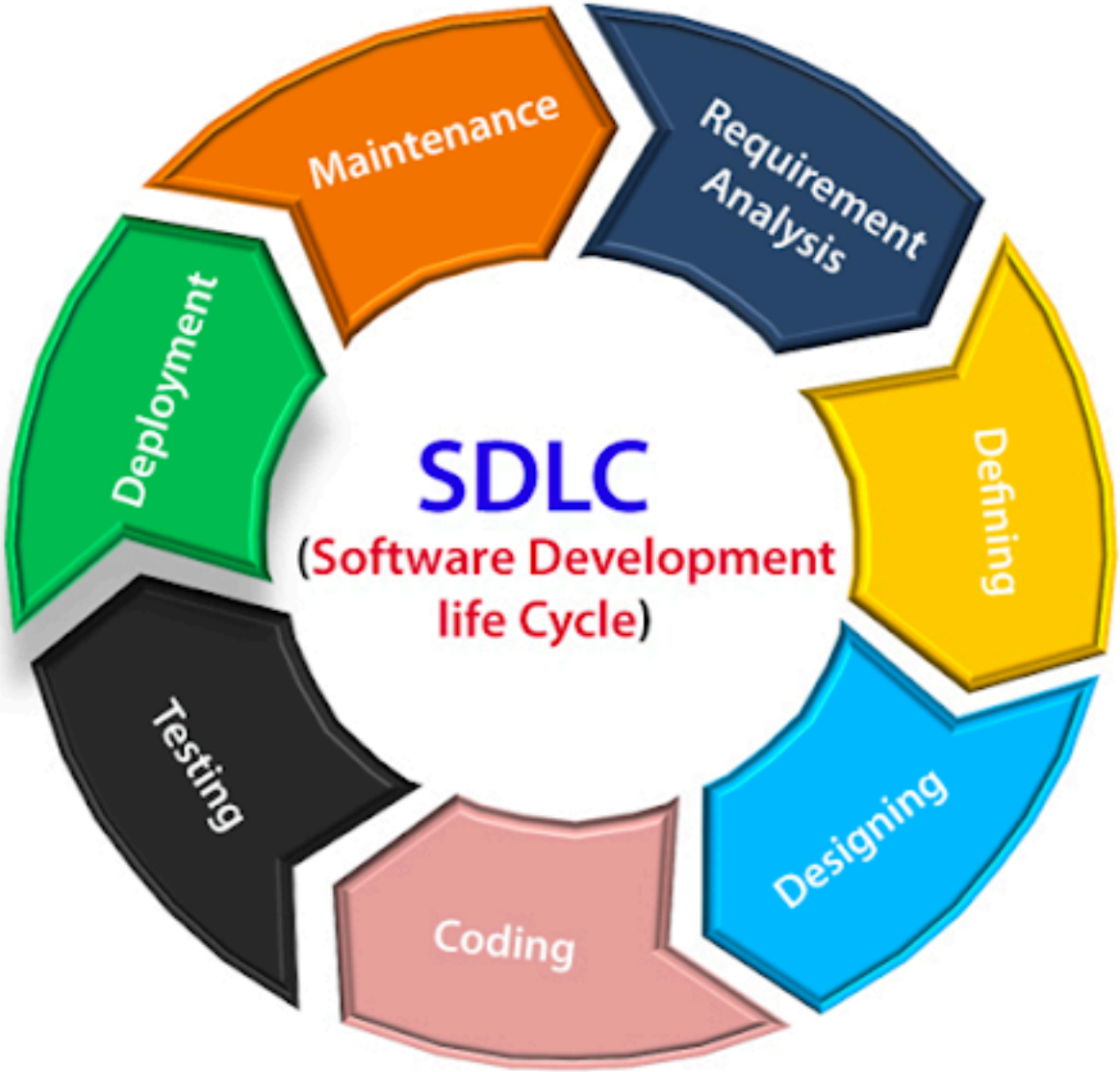
9x

How do we manage this complexity??

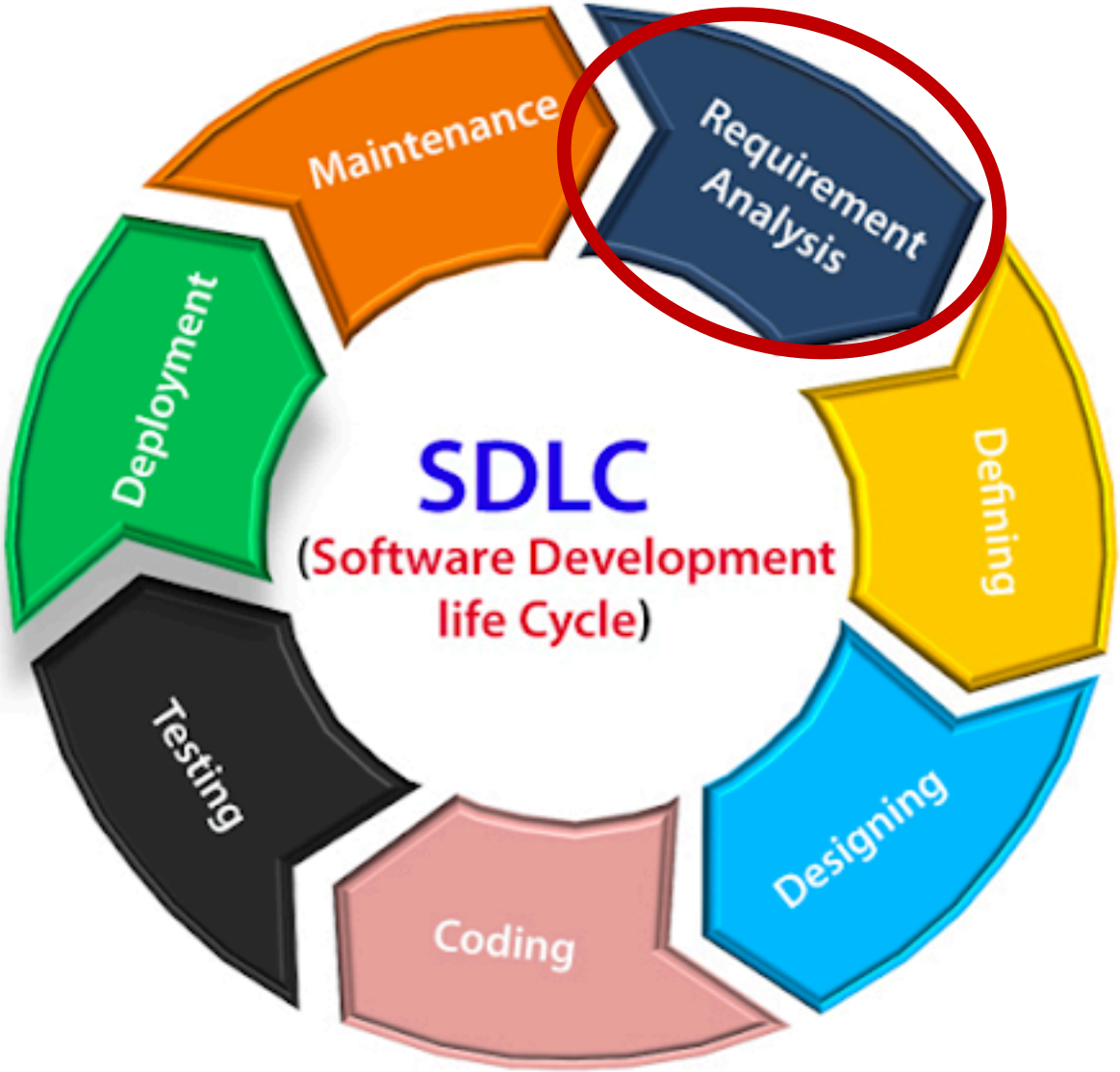
Software Development Process

Software Process: a way of breaking down the overall software development work into manageable sub-tasks; systematic and somewhat formal

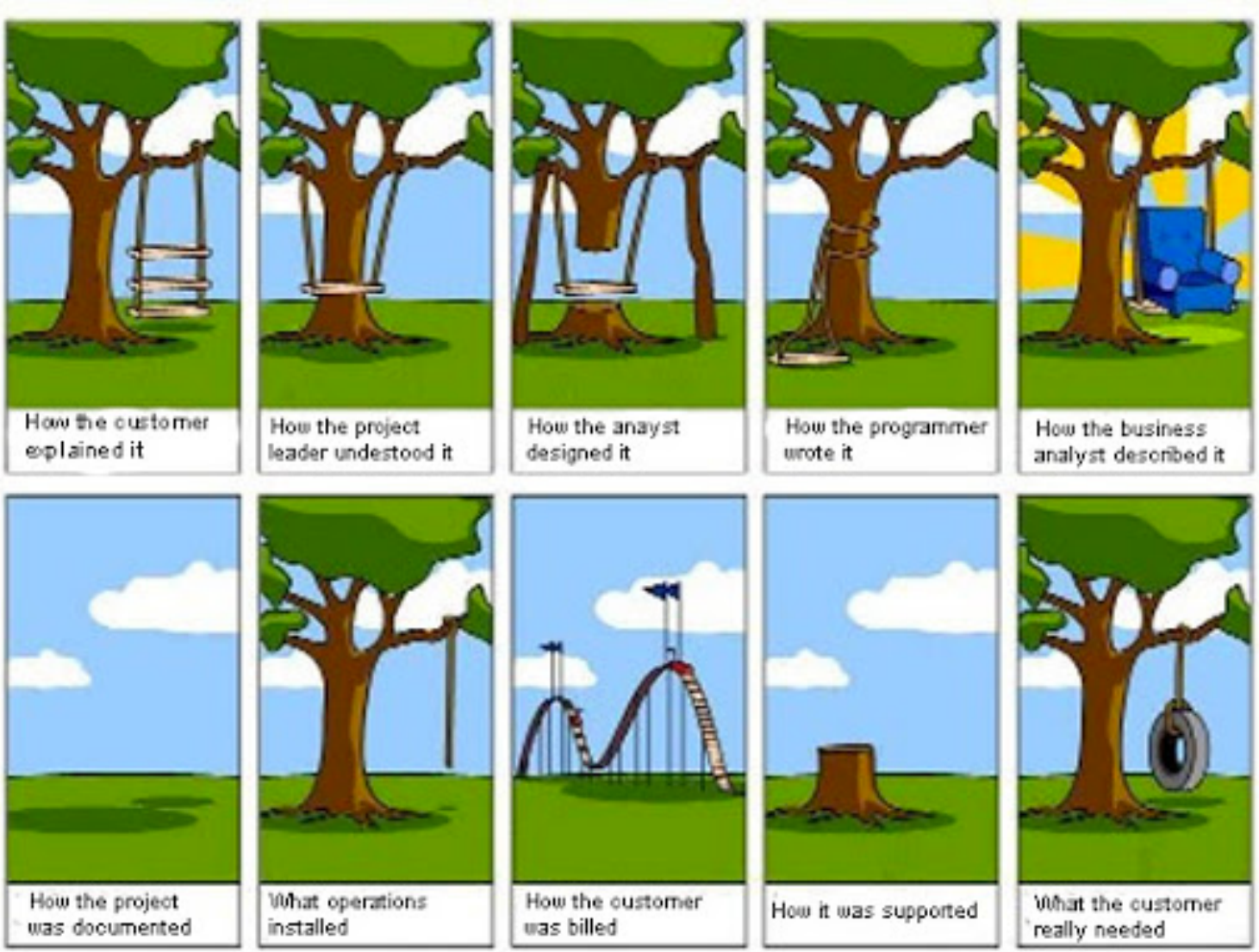
Software Development Process Steps



Requirements Analysis



Requirements Analysis



Requirements Analysis

- The **WHAT**? and the **WHY**?
- Understanding what the customer wants or what they "think" they want (Ask WHY?)
- Focus on the business problem you are trying to solve
- Understand what is most important to the customer to enable prioritization
- Also, ensure you understand non-functional requirements
- Can be documented in various ways depending on the process:
  - Formal requirements specifications
  - Wireframes
  - Use case documents
  - Prototypes

So How Do We Know We Got it Right?

Requirements Analysis

How do we validate the requirements?

Avoid producing a good apple when an orange is required

Validation vs. Verification

- Project Management book of knowledge (IEEE standard) defines these terms:

- **Validation**: The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers.

- **Verification**: The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process.

Validation vs. Verification

- **Validation**: Are we producing the Right product?

- **Verification**: Are we producing the product Right?

Discussion:

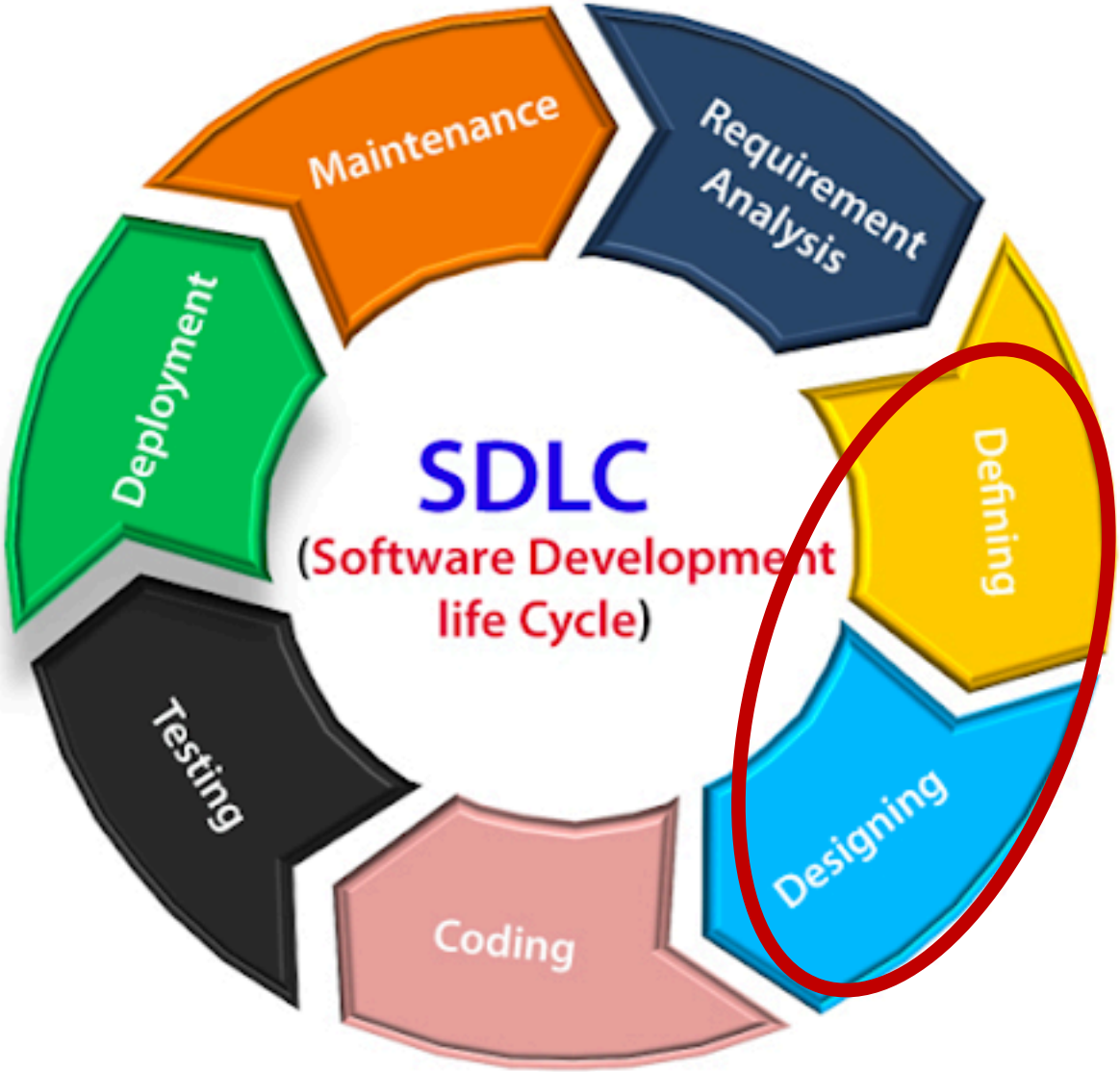
Requirements Analysis

Which is more important?

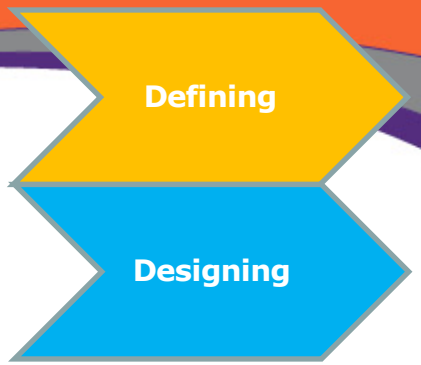
Validation (producing the Right product) –or-
Verification (producing the product Right)

"Software engineering IS requirements discovery"

Define and Design

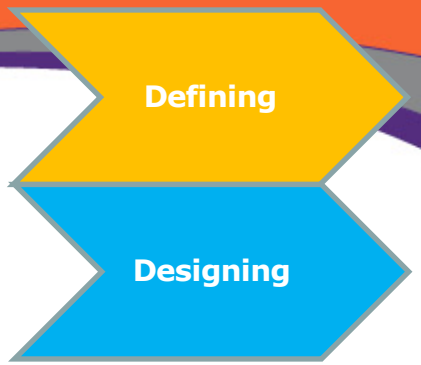


Define and Design



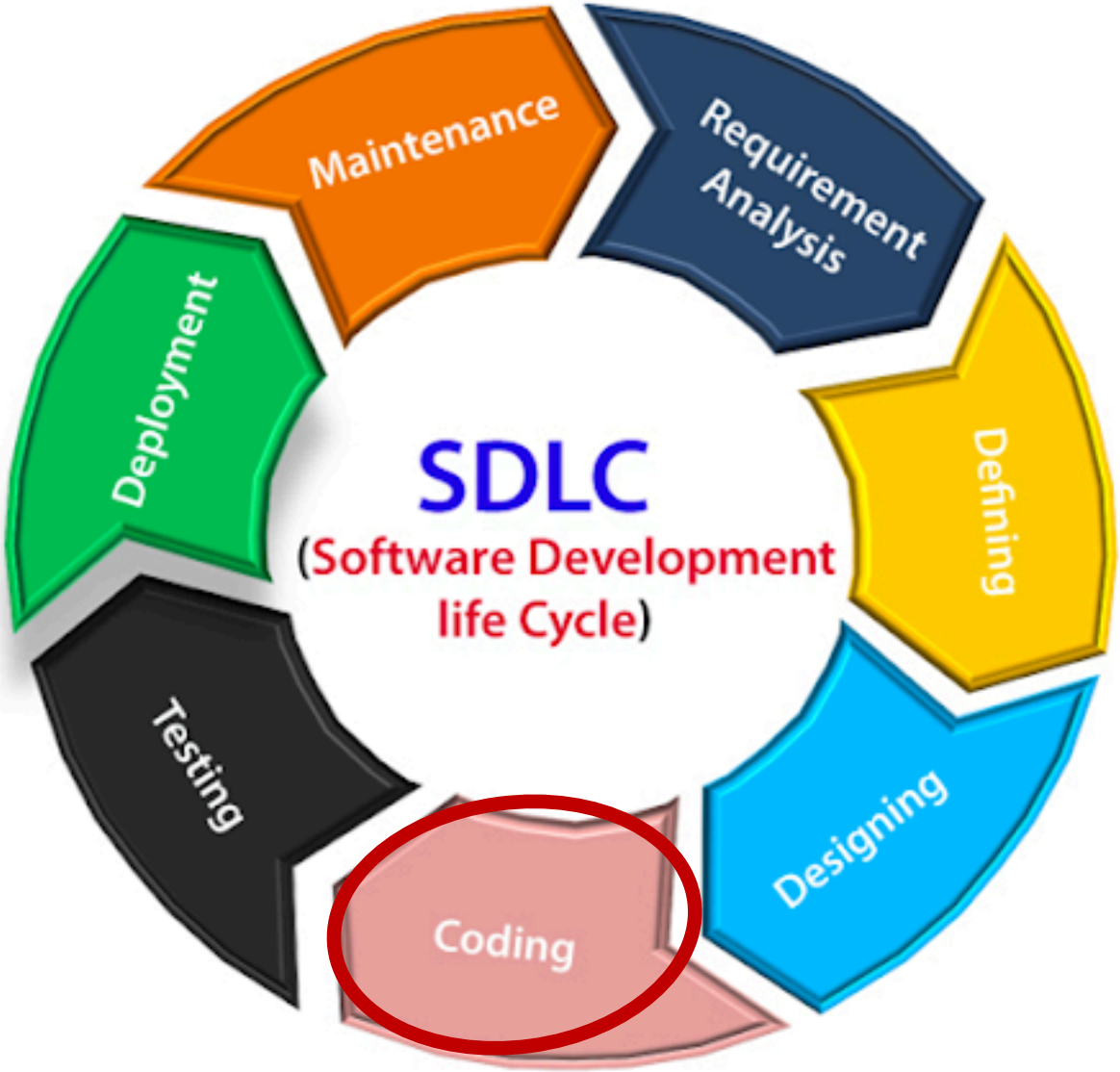
- The **HOW**?
- Need to understand both the business and non-functional requirements
- Depending on the type and size of system you will have various layers of design:
  - System architecture
  - Deployment architecture
  - Object/Class Designs
  - Database designs
  - UI designs

Define and Design



- Can be documented in various ways depending on the process and the type of system:
  - Formal design specifications
  - UML
  - State and Transition diagrams
  - Wiki documents
  - Contract documentation (for APIs)
  - ERDs (Entity Relationship Diagrams)
  - Whiteboarding sessions and photos!

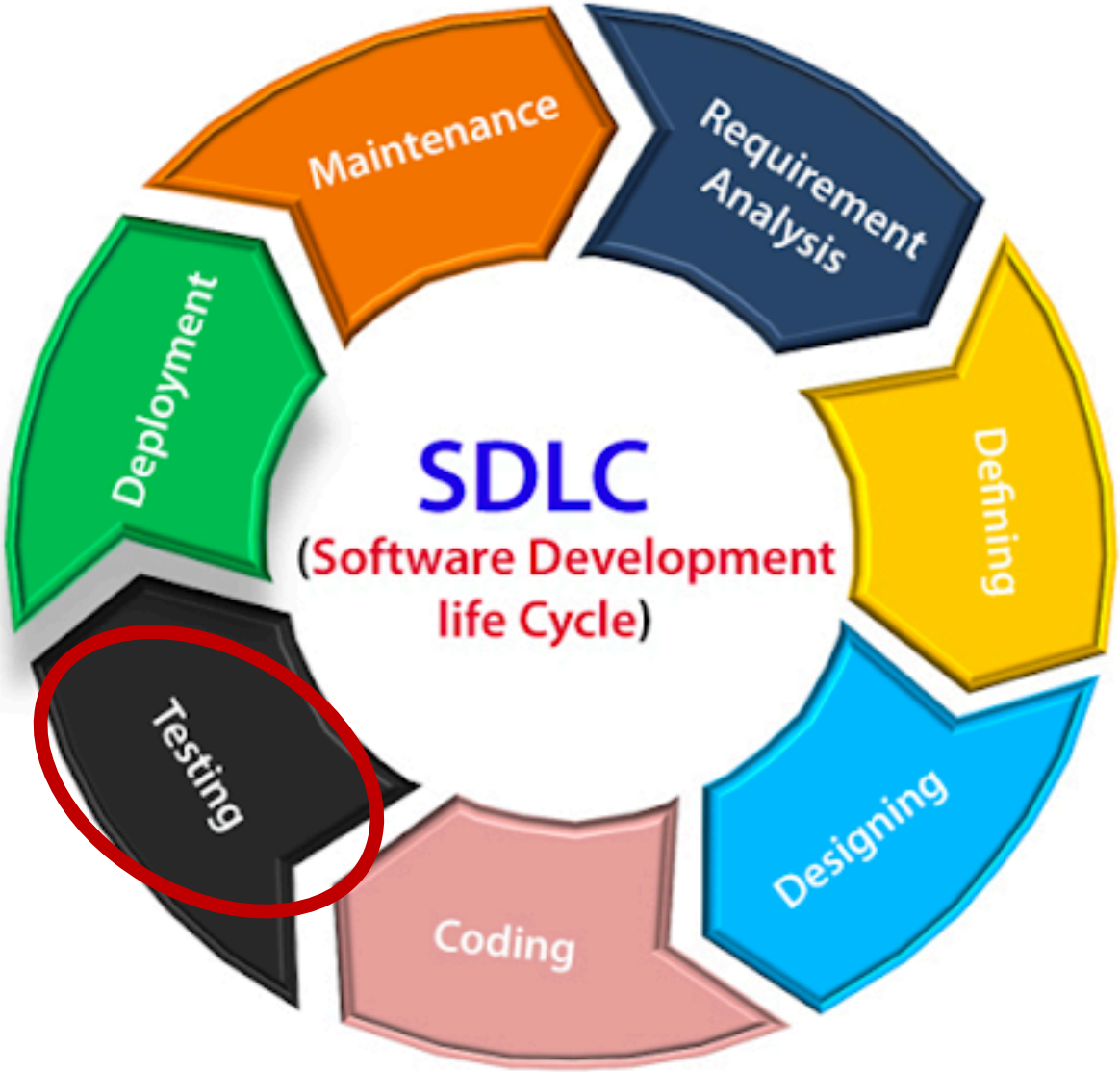
Coding


Maintenance
Requirement Analysis
Deployment
Defining
SDLC
(Software Development life Cycle)
Testing
Designing
Coding

Coding

- AKA Implementation
- Depending on organization you could have different standards and methods:
  - Language requirements
  - Pair programming
  - Code standards
  - Code reviews
  - Tools usage (configuration management, IDEs, 3rd party tools, open software rules)
  - Internal and external frameworks
  - Unit testing requirements

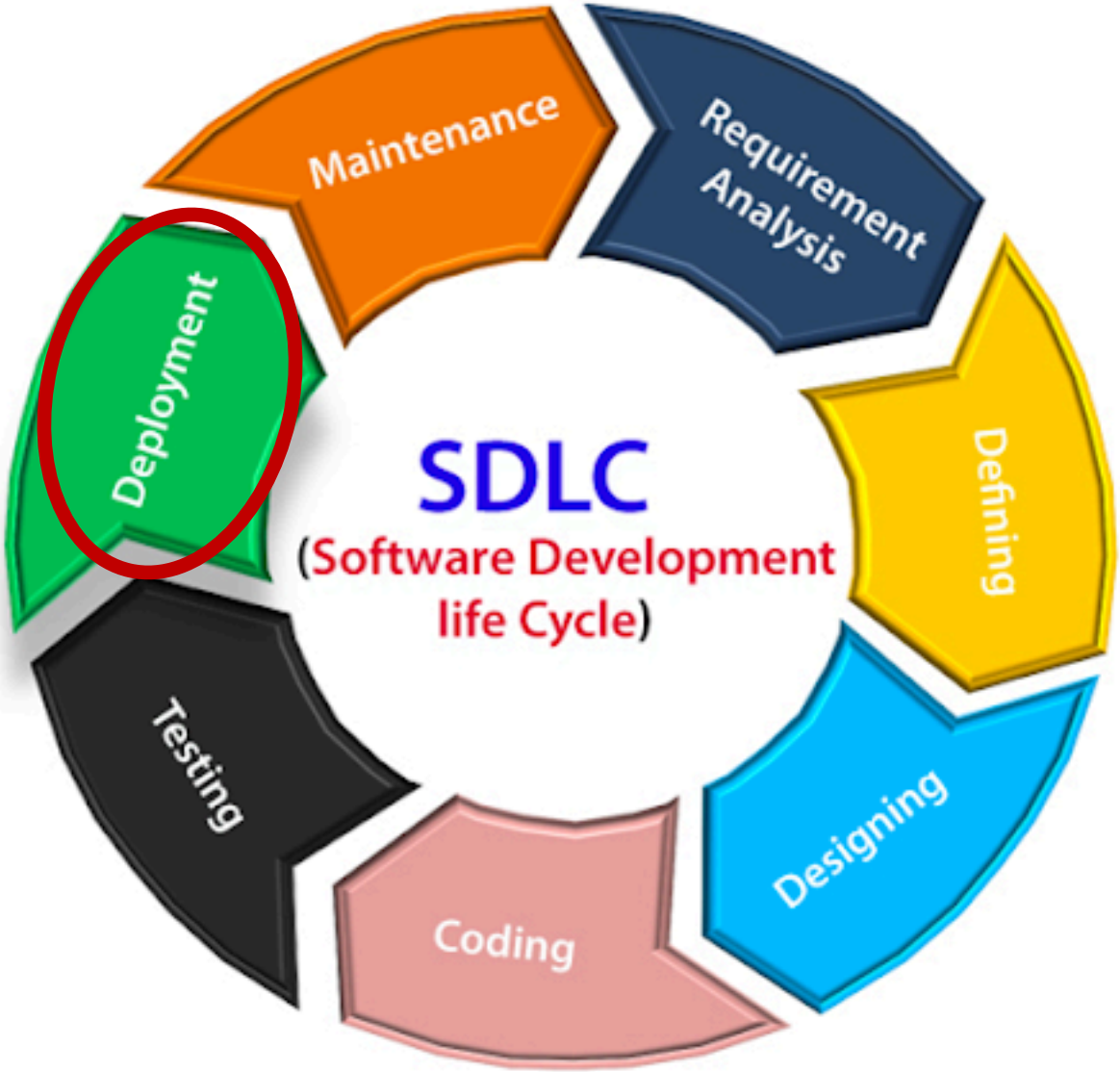
Testing


Maintenance
Requirement Analysis
Deployment
Defining
SDLC
(Software Development life Cycle)
Testing
Designing
Coding

Testing

- AKA Quality Assurance – the validation AND verification
- Depending on organization and process you will have different approaches:
  - When you test:
    - Unit Testing
    - Integration Testing
    - System Testing
    - Performance Testing
    - Regression Testing
  - How you Test
    - Test Driven Development
    - Continuous Testing
    - Automated vs. Manual

Deployment

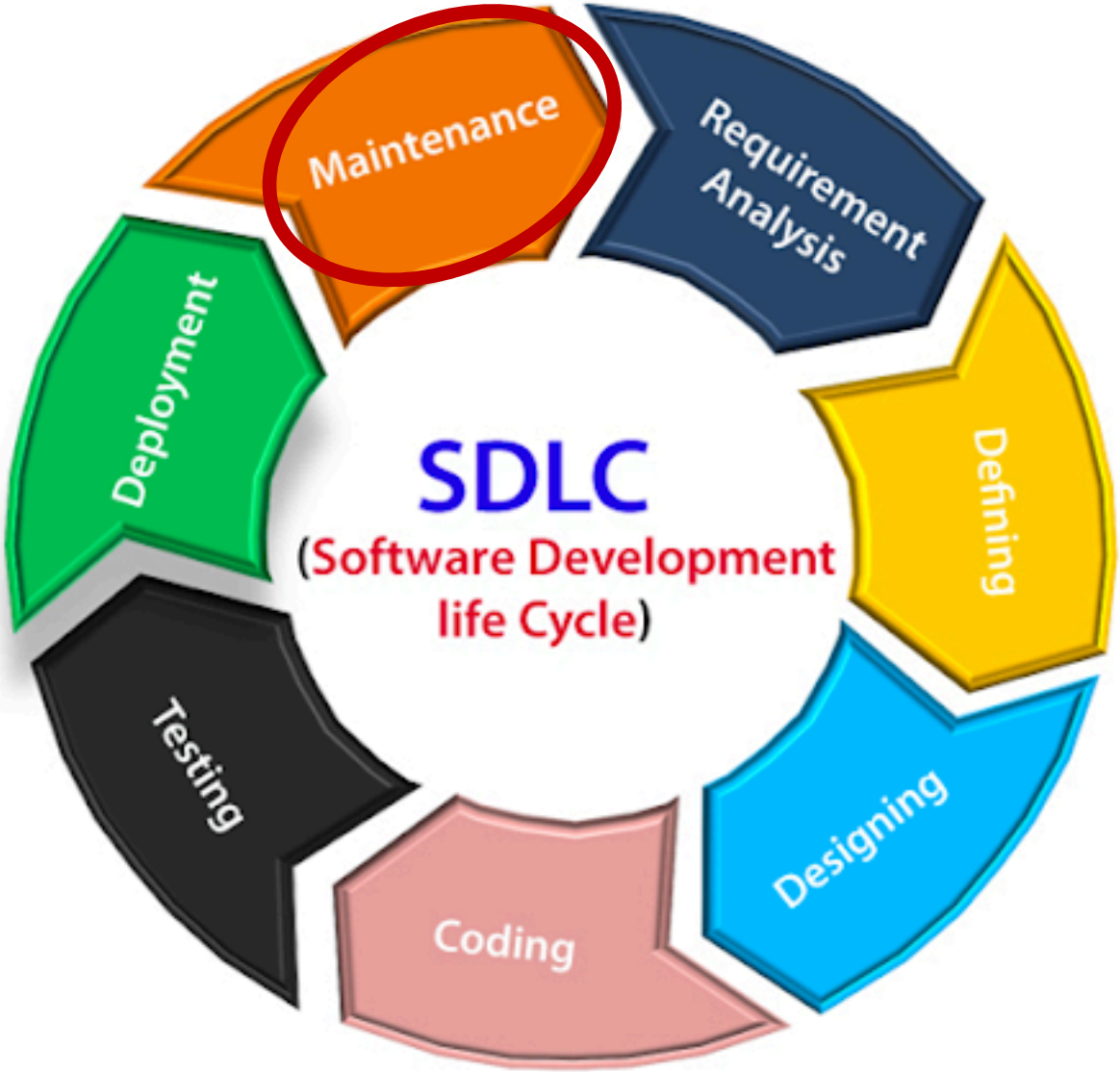

Maintenance
Requirement Analysis
Deployment
Defining
SDLC
(Software Development life Cycle)
Testing
Designing
Coding

Deployment

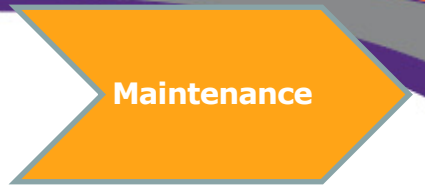
- Deployment is the process of putting your software into production for use by the customer

- Deployment will vary based on type of software being developed:
  - **Commercial "on-premise"** – Customer IT will deploy the software
  - **Commercial SaaS** – Company dev/operations will deploy the software
  - **Internal** – Company dev/operations will deploy the software

Maintenance



Maintenance



- Fixing bugs in production that are found by the customer
- Who "owns" the deployment will dictate how the maintenance is delivered
    - Emergency Patch fixes
    - Maintenance Releases
    - Major Releases

-OR-

    - Continual Deployment (more about this later)

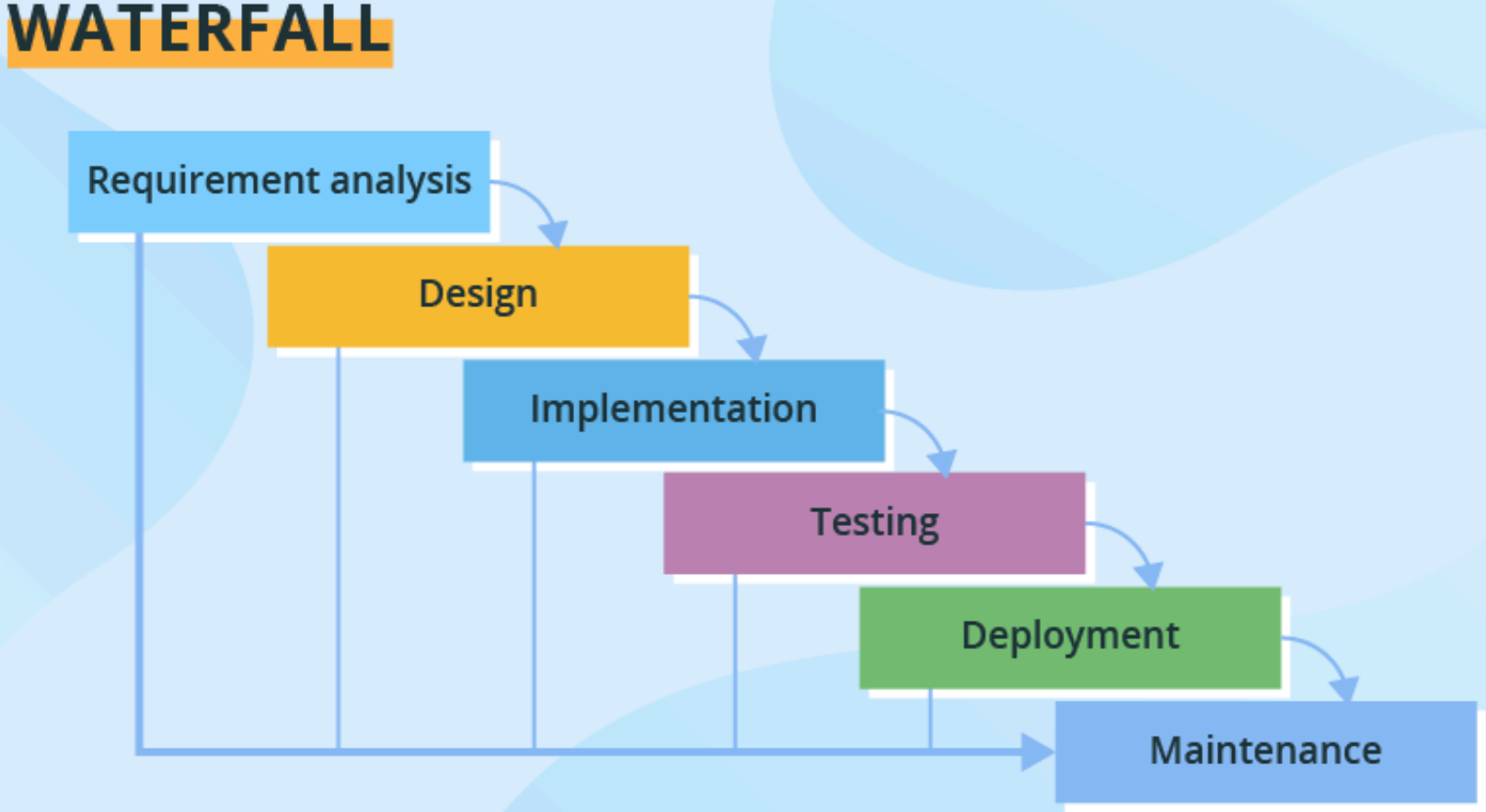
Putting it all Together

- The steps described happen for all software engineering projects but can be assembled differently
- The particular SDLC in use will vary company to company based on their business needs and culture; not a “one size fits all”
- Companies will tailor the process model as well
- Will use this source for following slides:
  - https://www.scnsoft.com/blog/software-development-models

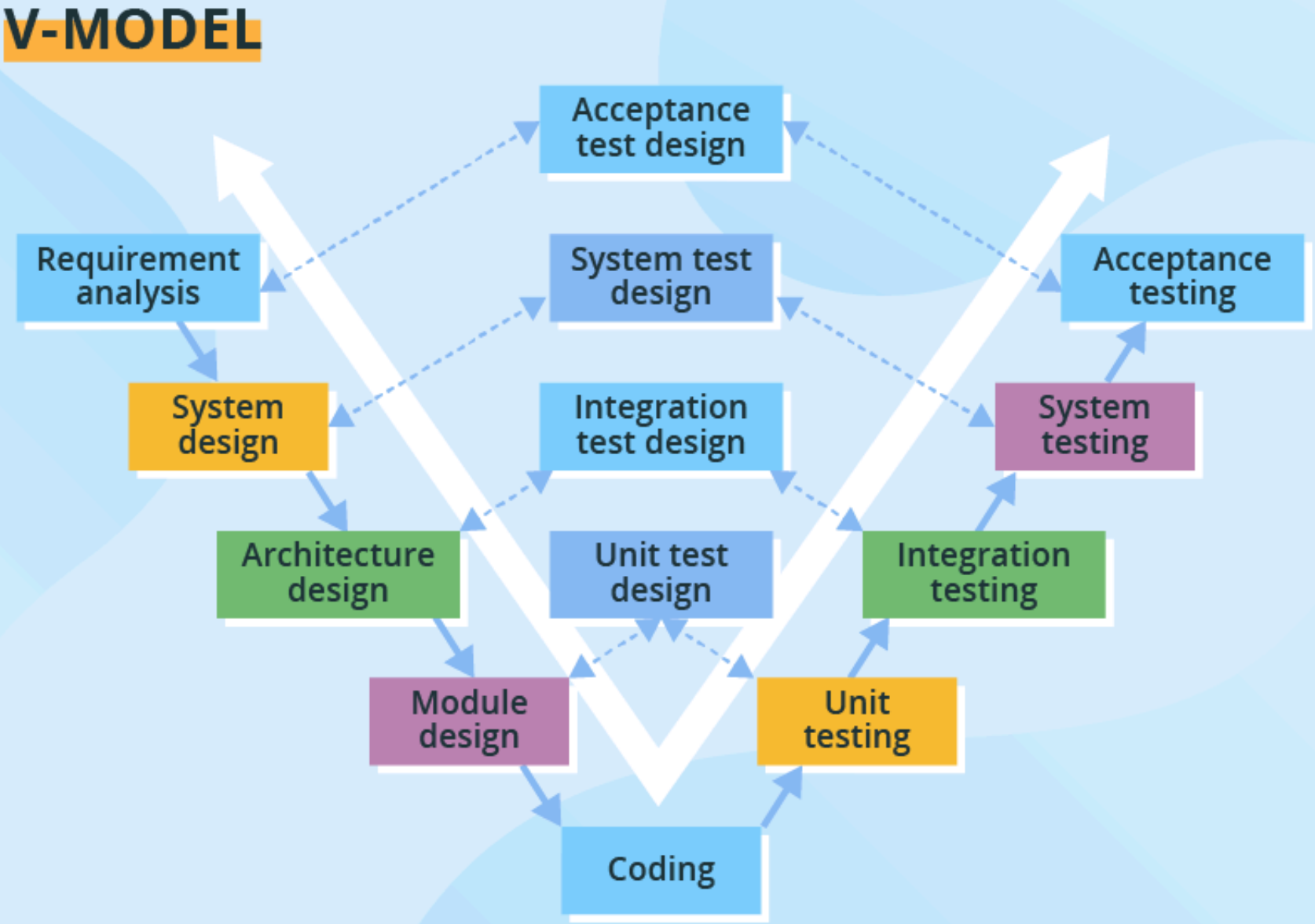
Putting it all Together
Process Models

- Waterfall
- V-Model
- Incremental
- Iterative
- Spiral Model
- RUP
- Agile:
  - Scrum
  - XP
  - Kanban

Putting it all Together
Process Models

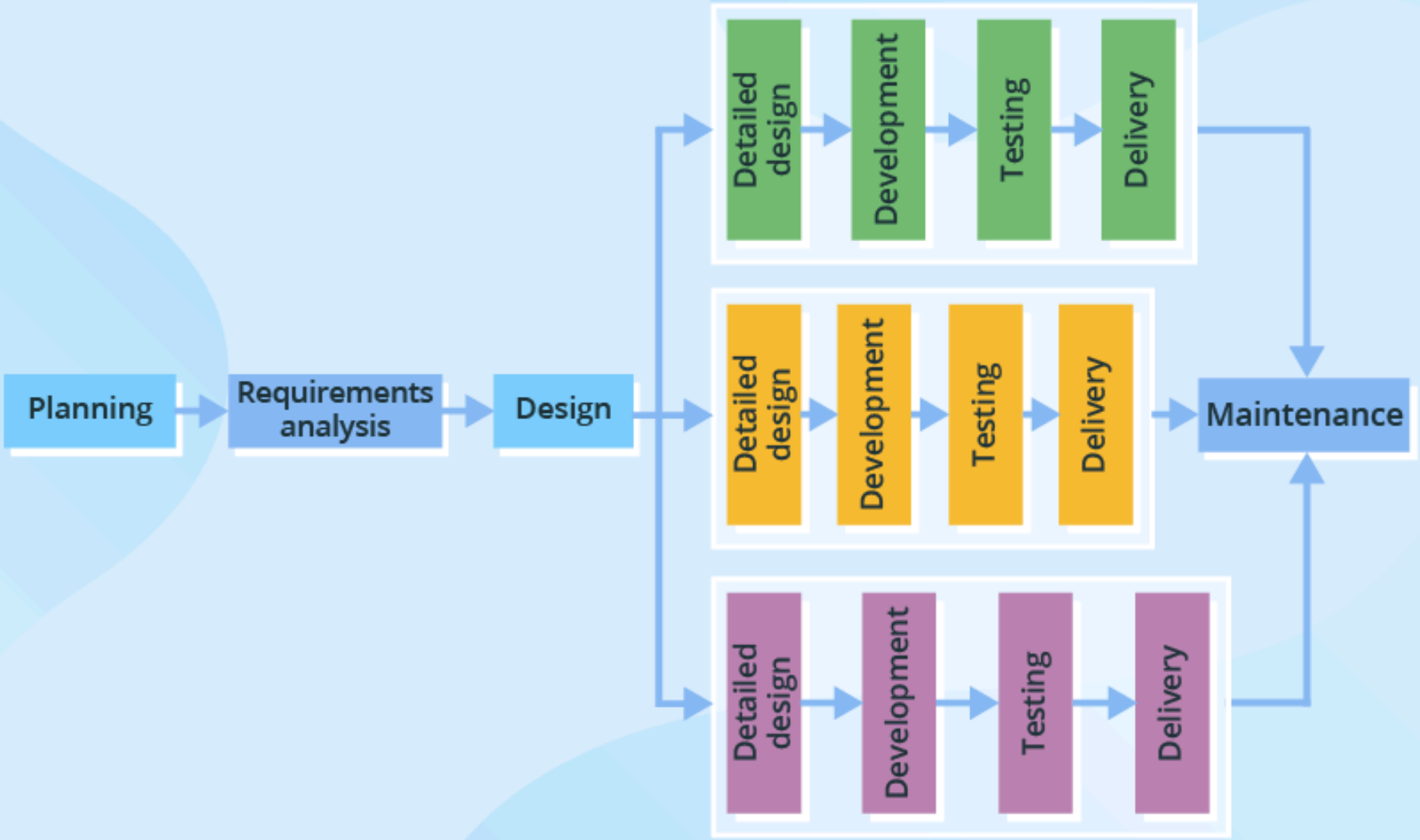


Putting it all Together
Process Models



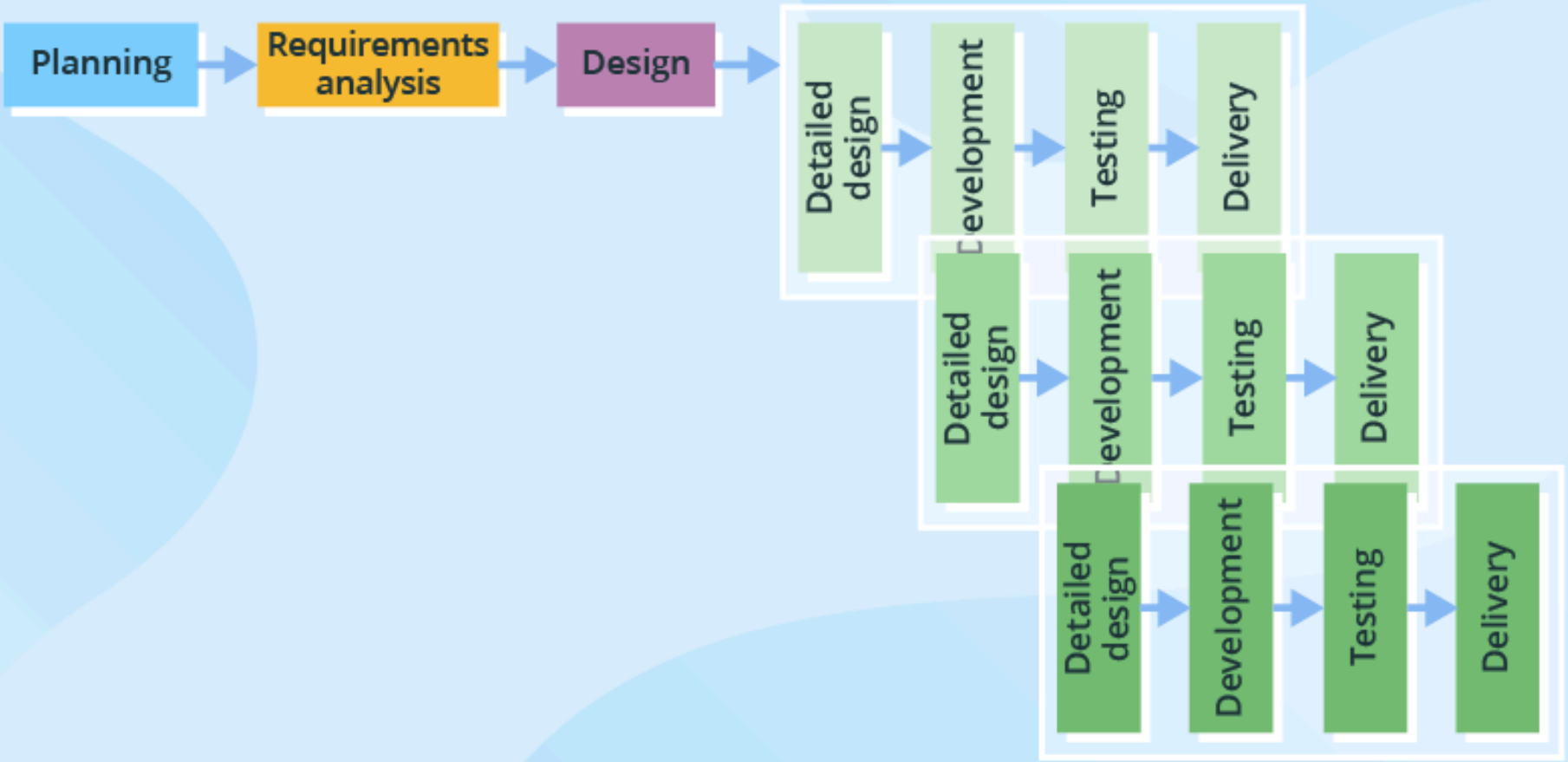
Putting it all Together
Process Models

INCREMENTAL MODEL

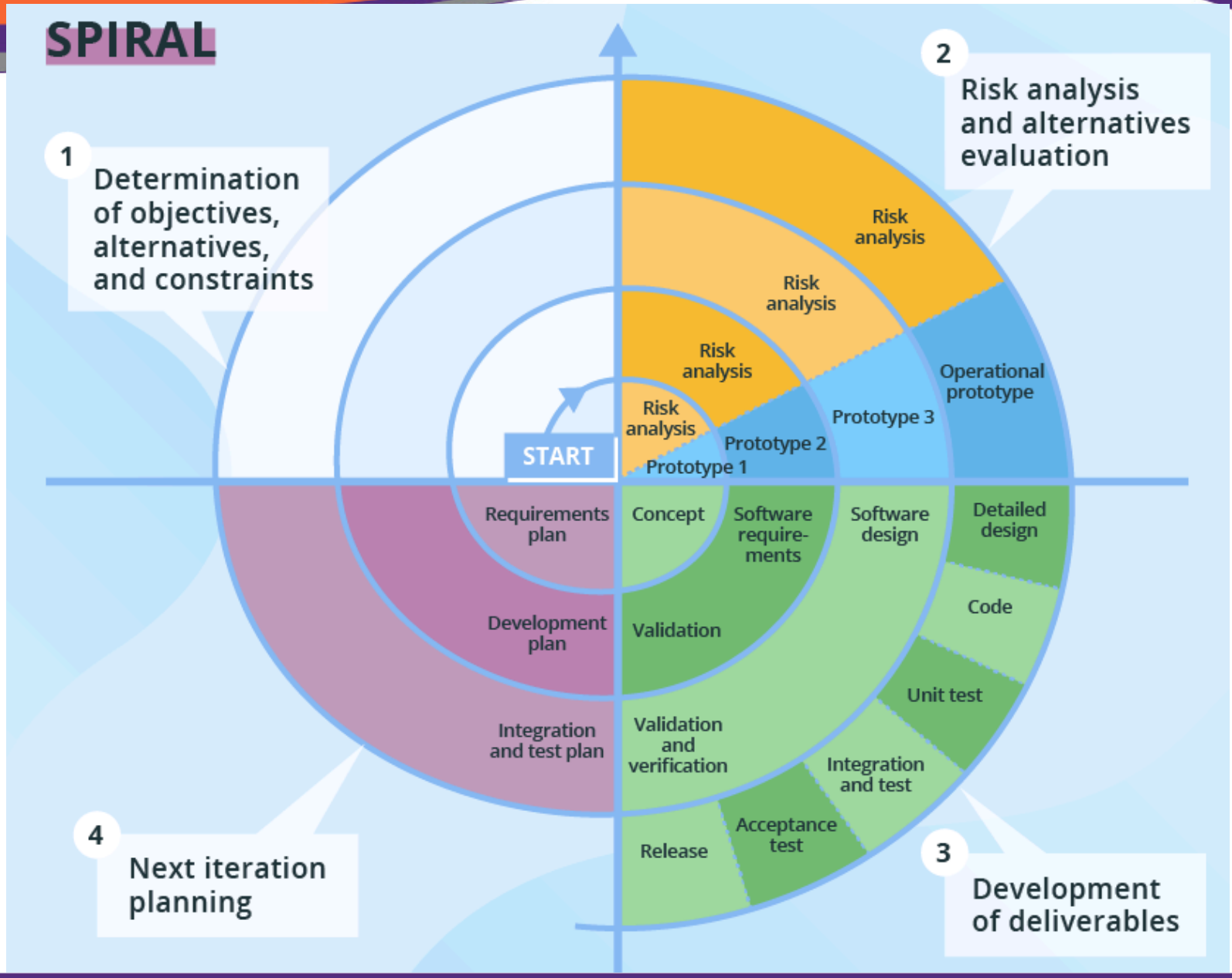


Putting it all Together
Process Models

ITERATIVE MODEL

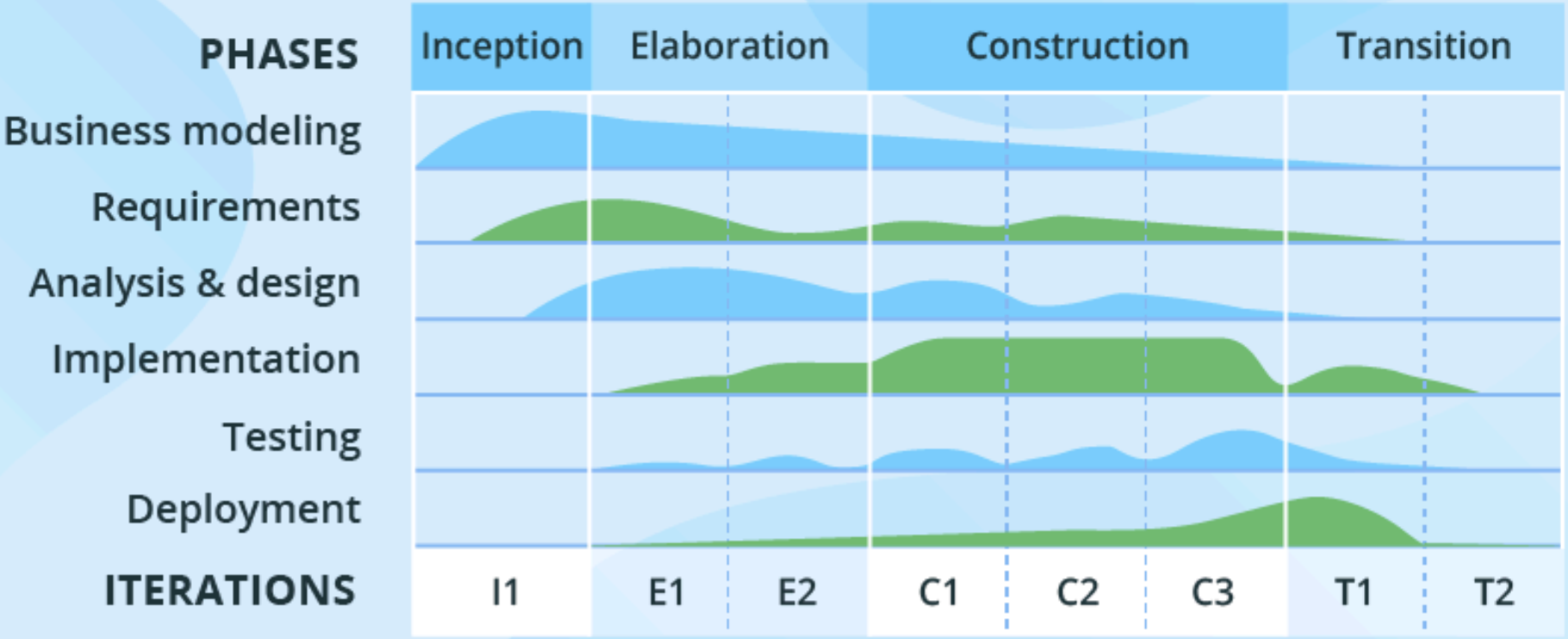


Putting it all Together
Process Models

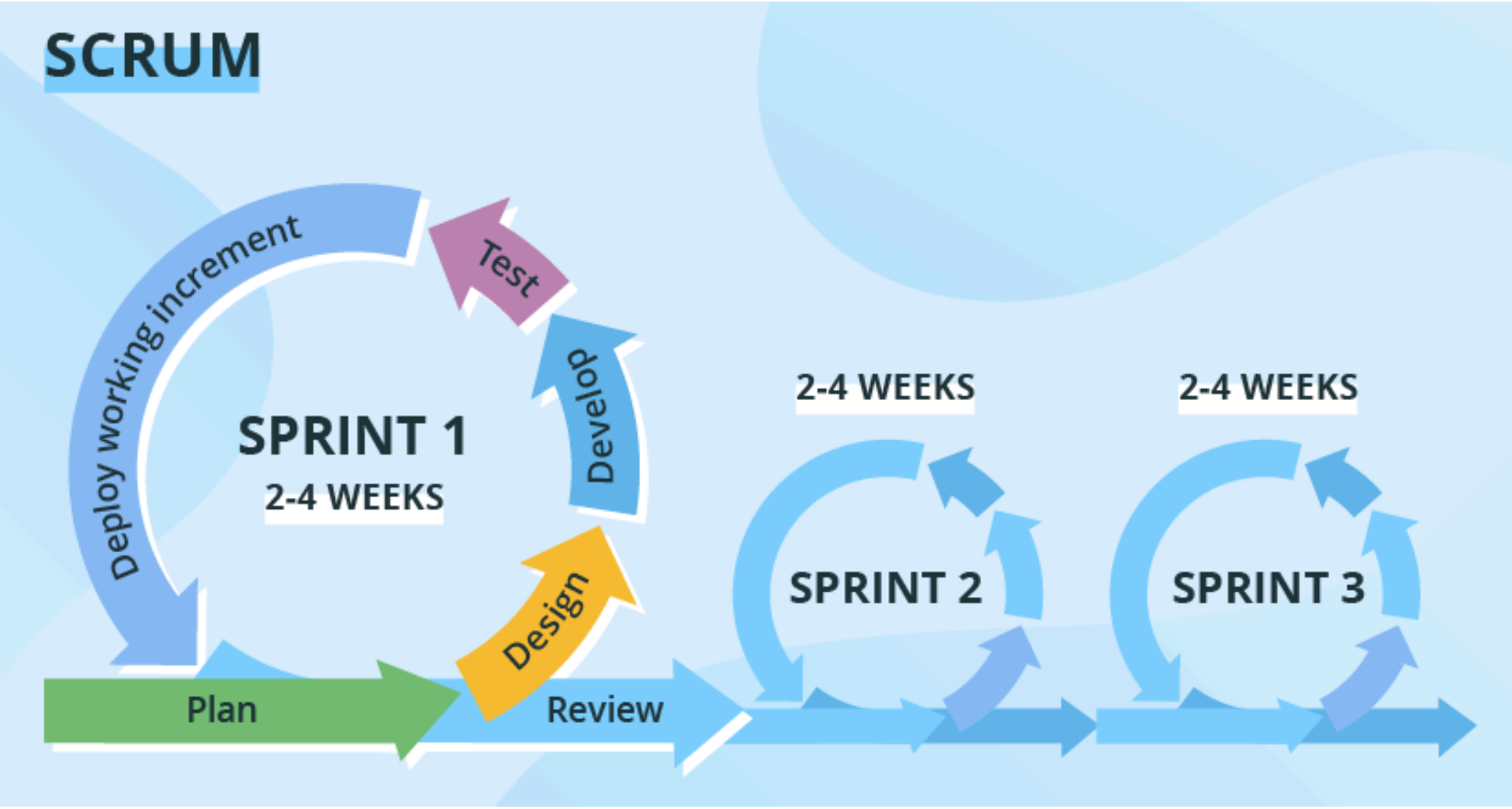


Putting it all Together
Process Models

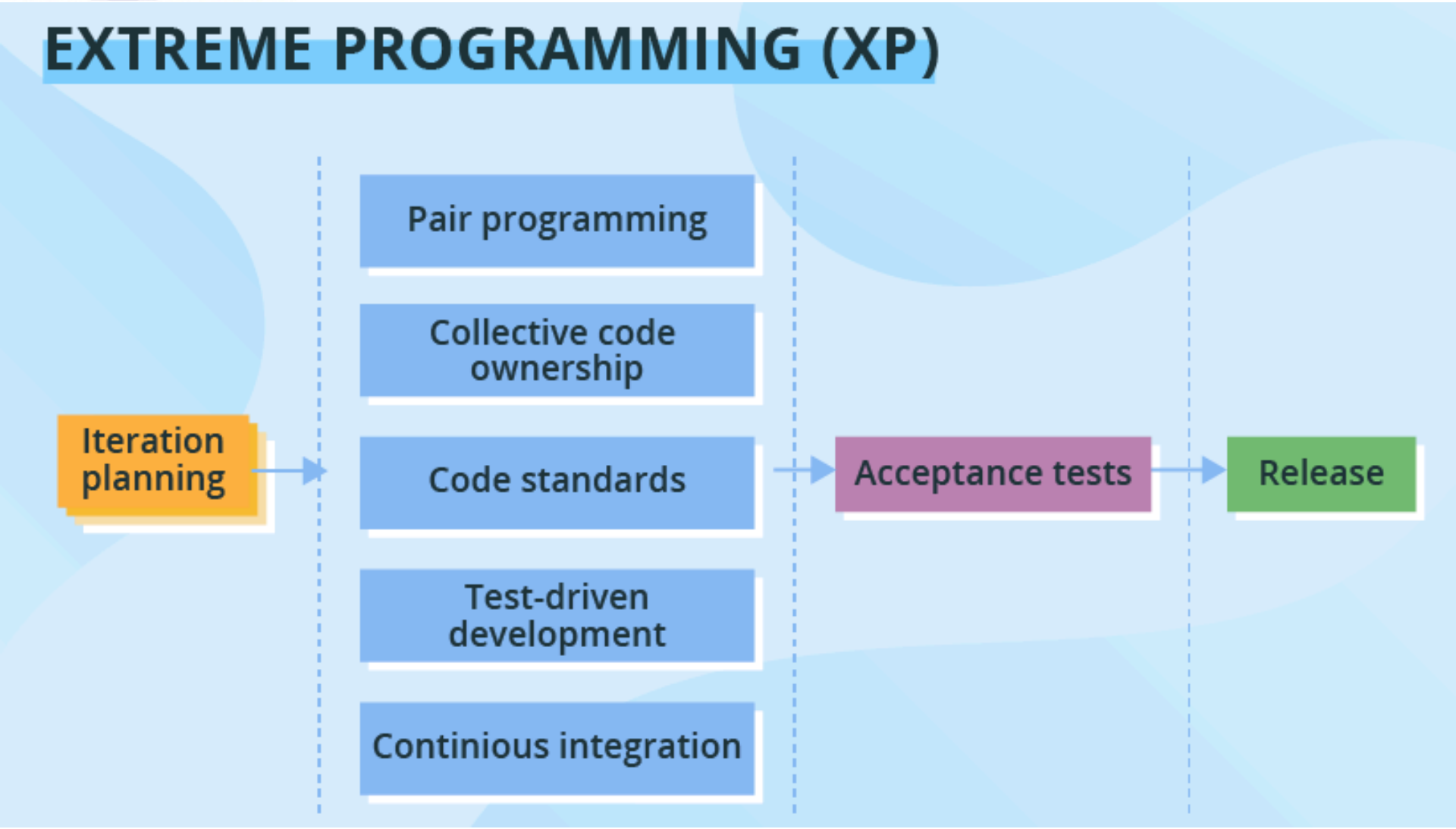
THE RATIONAL UNIFIED PROCESS (RUP)



Putting it all Together
Agile Process Models

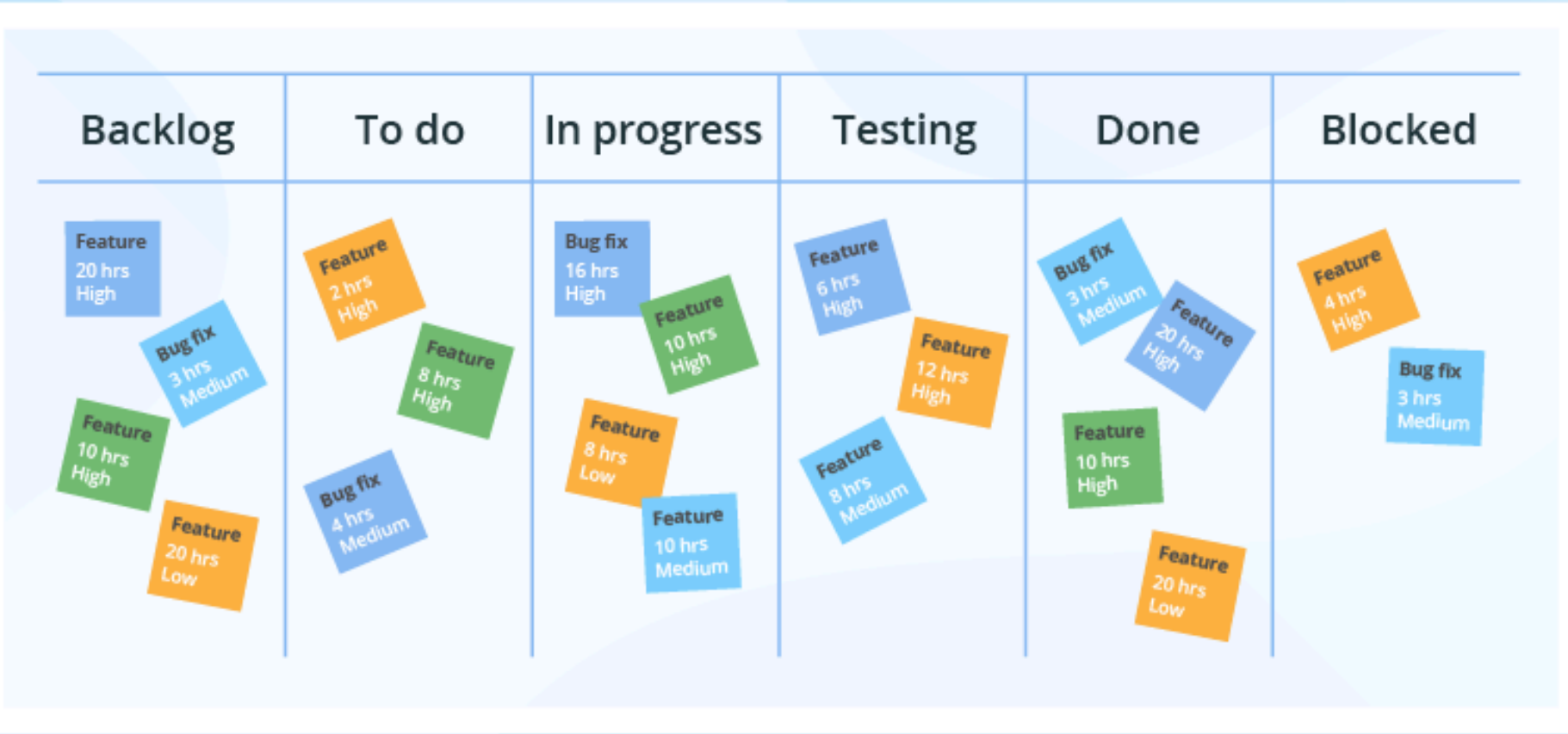


Putting it all Together
Agile Process Models

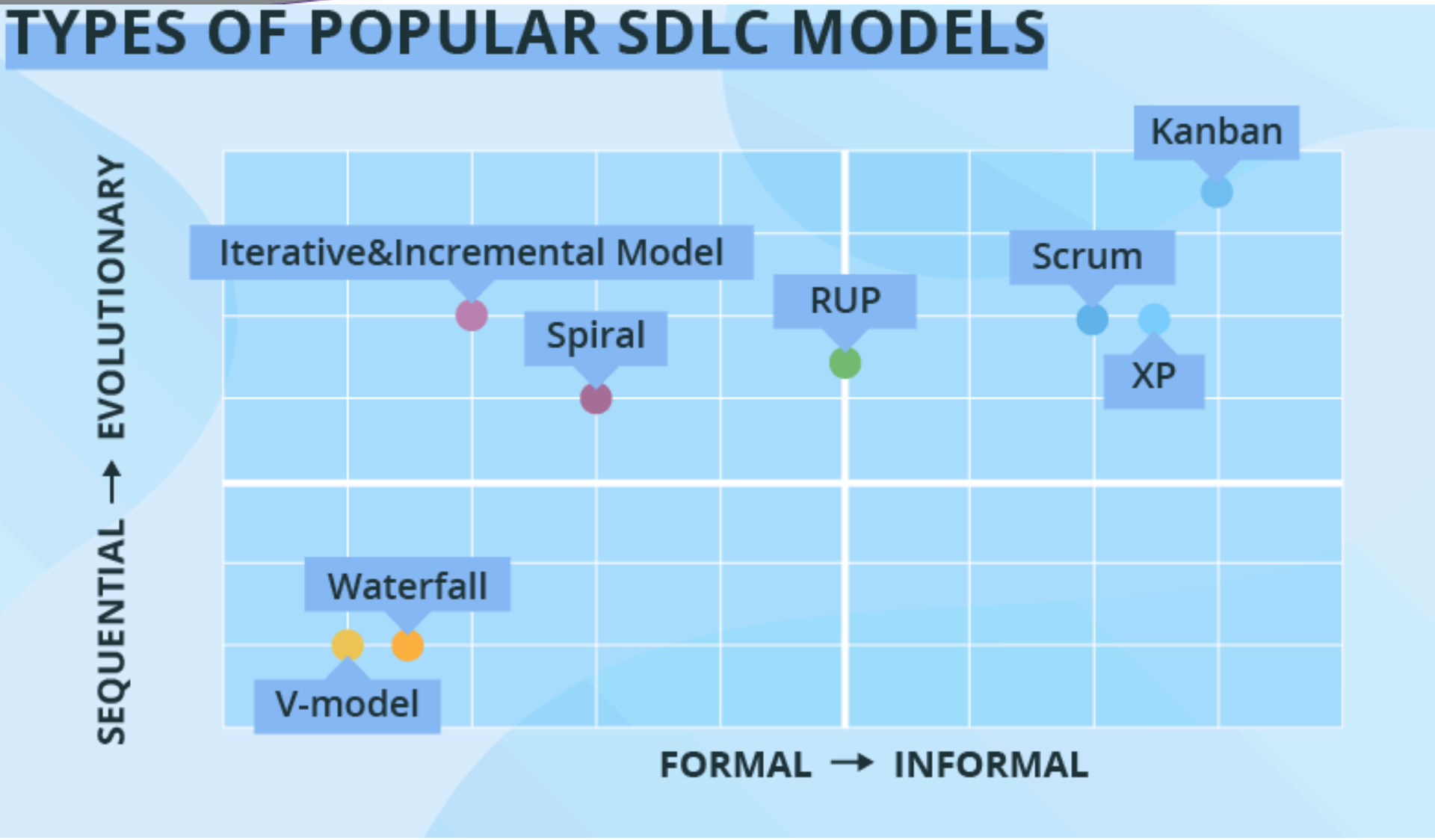


Putting it all Together
Agile Process Models

KANBAN



Putting it all Together
Which One????



CLEMSON UNIVERSITY

Before Next Class

- Reading: Agile Manifesto and Principles -
  - https://agilemanifesto.org/
  - https://agilemanifesto.org/principles.html

  - The Scrum Guide
  - https://www.scrumguides.org/scrum-guide.html

PSA

A group of students are starting a **Google Developers Student Club** at Clemson University. They will be hosting an info session at a soon to be determined time next week. If you are interested, please sign up using the link below.

https://docs.google.com/forms/d/e/1FAIpQLSfJLxR1FEAh1nq71V7IPYMNevdFUs6Ic3nalvrz76Ej2am55g/viewform?usp=sf_link