CSE 3421

Software Development Methodologies

SPRING 2022

MD. RAFI-UR-RASHID

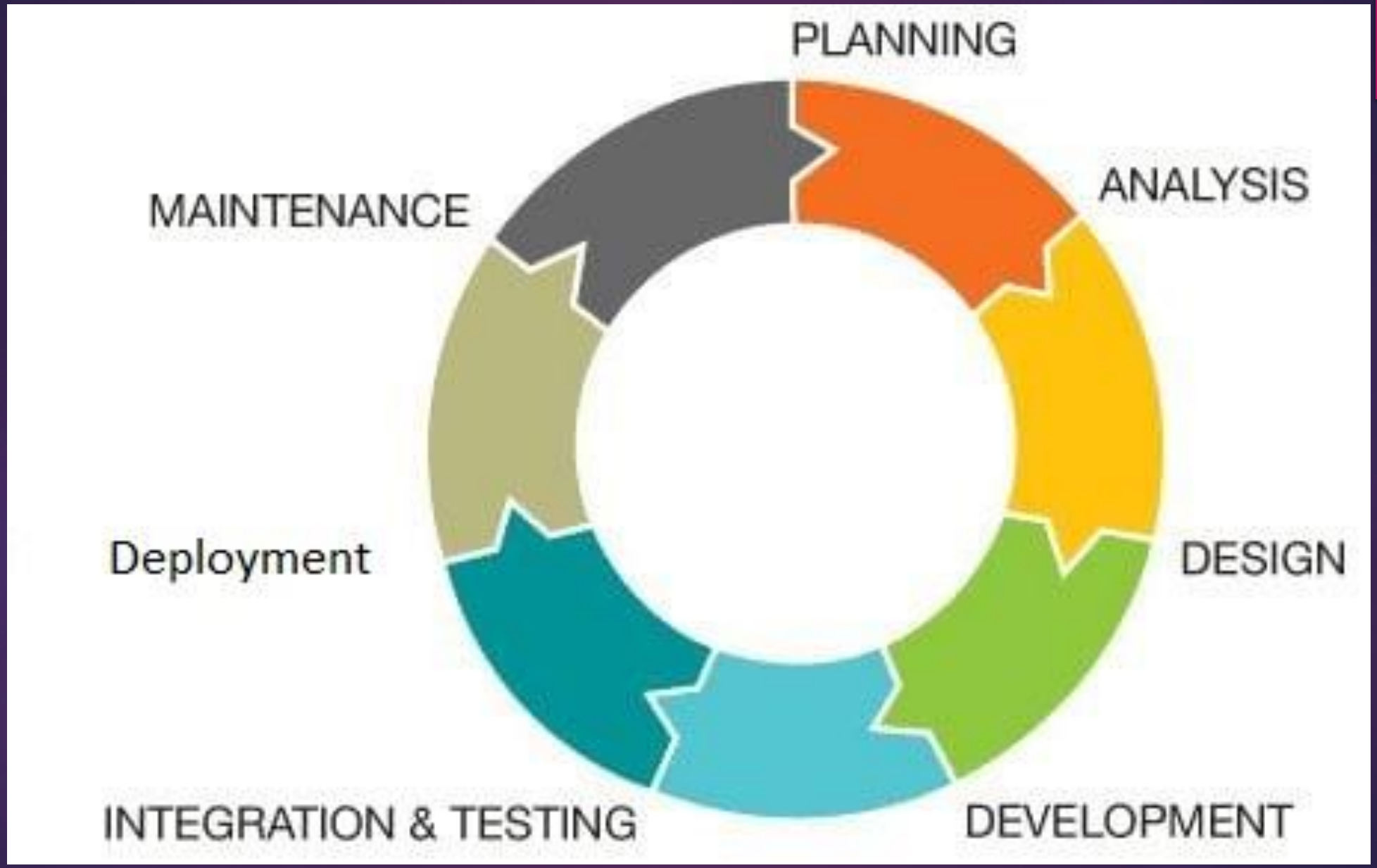
LECTURER, DEPT. OF CSE, UIU

Why This Lesson?

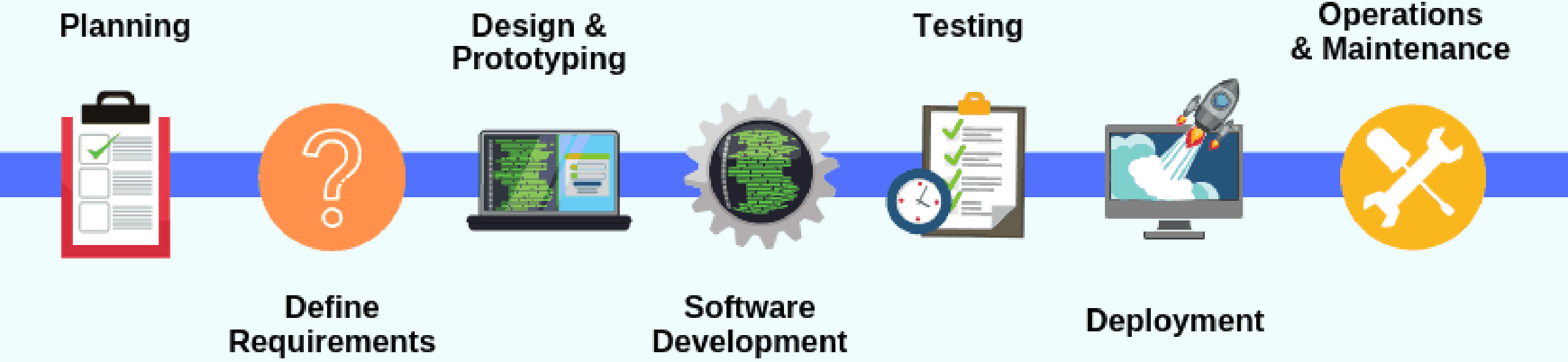
We are going to do something. Which is better- With or Without Plan?

A structured management of the workflow during a project

Ways of Implementing Software Development Life-Cycle (SDLC)


PLANNING
ANALYSIS
DESIGN
DEVELOPMENT
INTEGRATION & TESTING
Deployment
MAINTENANCE

Seven Phases of Software Development Life Cycle



phoenixNAP GLOBAL IT SERVICES

Planning

- Defining Scope
- Feasibility Analysis
- Benchmark Study

Requirement Analysis

- Analyzing the Resources:
  - Hardware
  - Software
  - Skilled People
  - Time

Designing & Prototyping

- Visual Representation of the project :
  - Wireframe Modeling

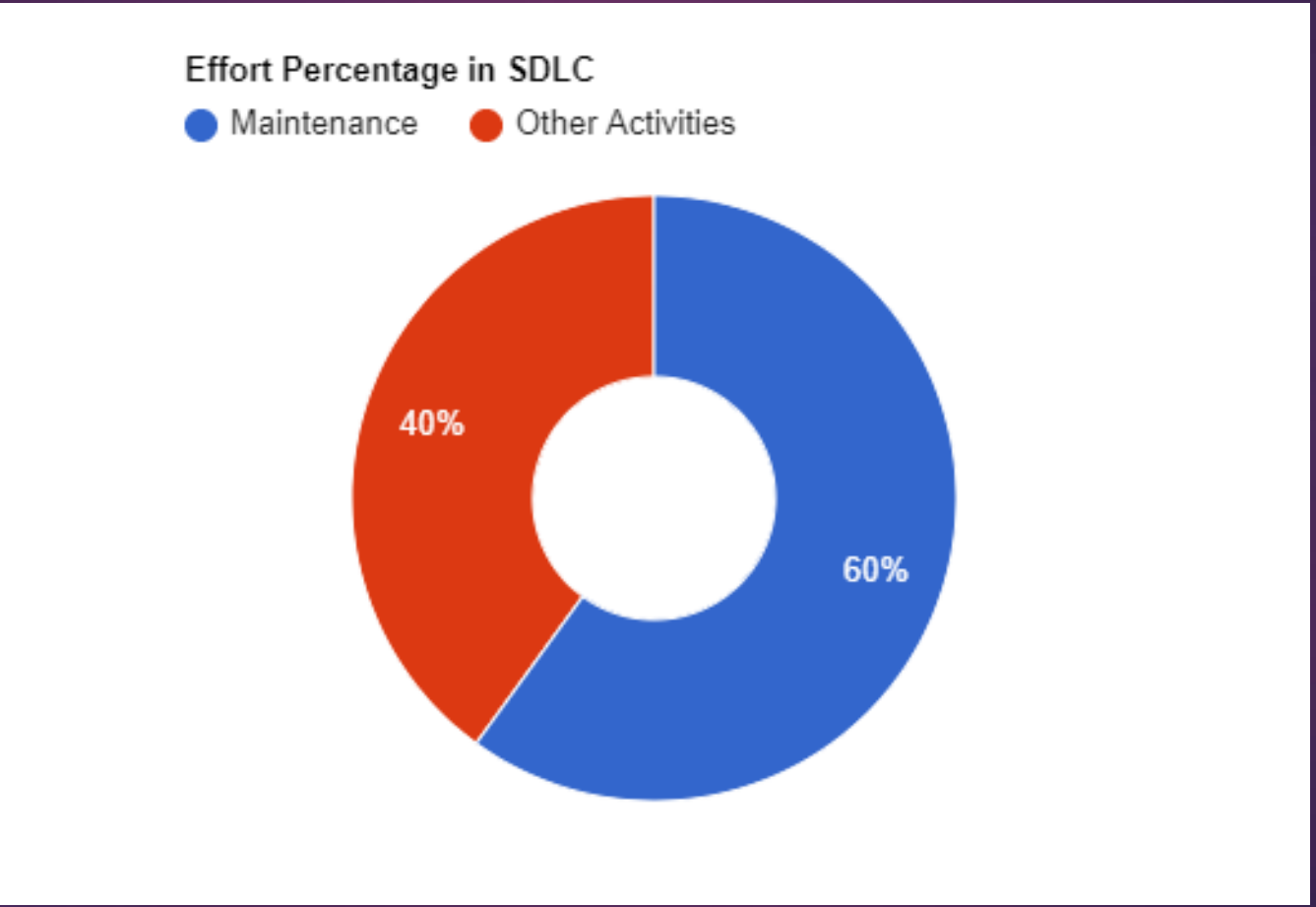
- Prototype: A prototype is an early sample, model, or release of a product
  - Manual/ Digital Prototyping

Development/ Implementation

Testing

Deployment/ Release

Maintenance



Maintenance

- Post-release bug fixing
- Check for optimization & improvement scope
- Deletion of outdated features/ data
- Enhancement of existing features
- Migration
- Release newer versions

A typical Software Team Structure

- Sales/Marketing/Business team
- Product team
- QA/Testing team
- Design team
- DevOps team
- Developer team

https://tsh.io/blog/software-development-team-structure/

Project scales

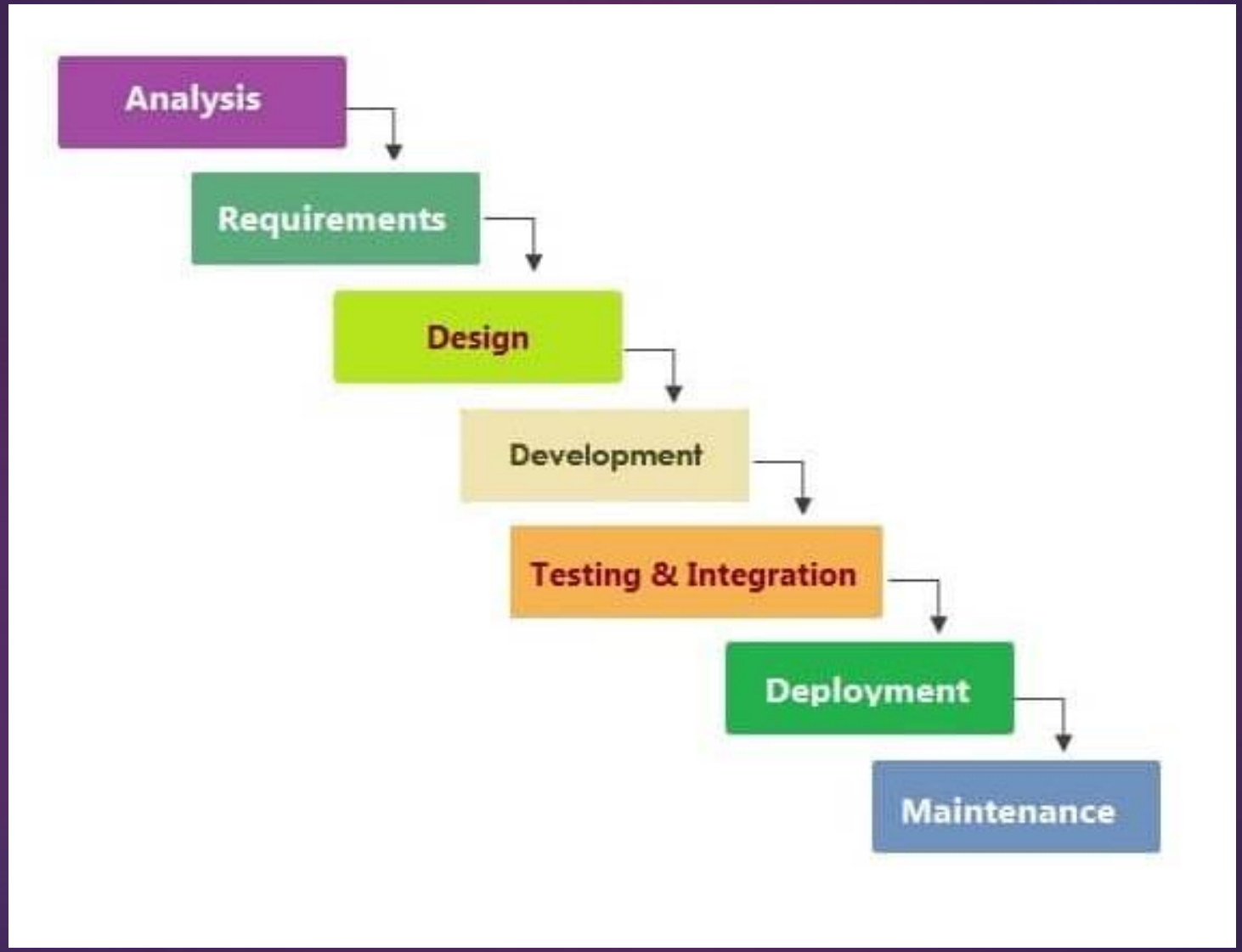
- Deciding factors: Duration, Finance, Resource

- Small scale projects: some weeks
- Mid scale projects: some months
- Large scale projects: 1+ years

Major Types of Models

- Linear/ Sequential
- Iterative
- Incremental
- Adaptive

Waterfall Model


Analysis
Requirements
Design
Development
Testing & Integration
Deployment
Maintenance

Key Idea

- A linear-sequential life cycle model.
- Each phase must be completed before the next phase begins
- There is no overlapping in the phases.

https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

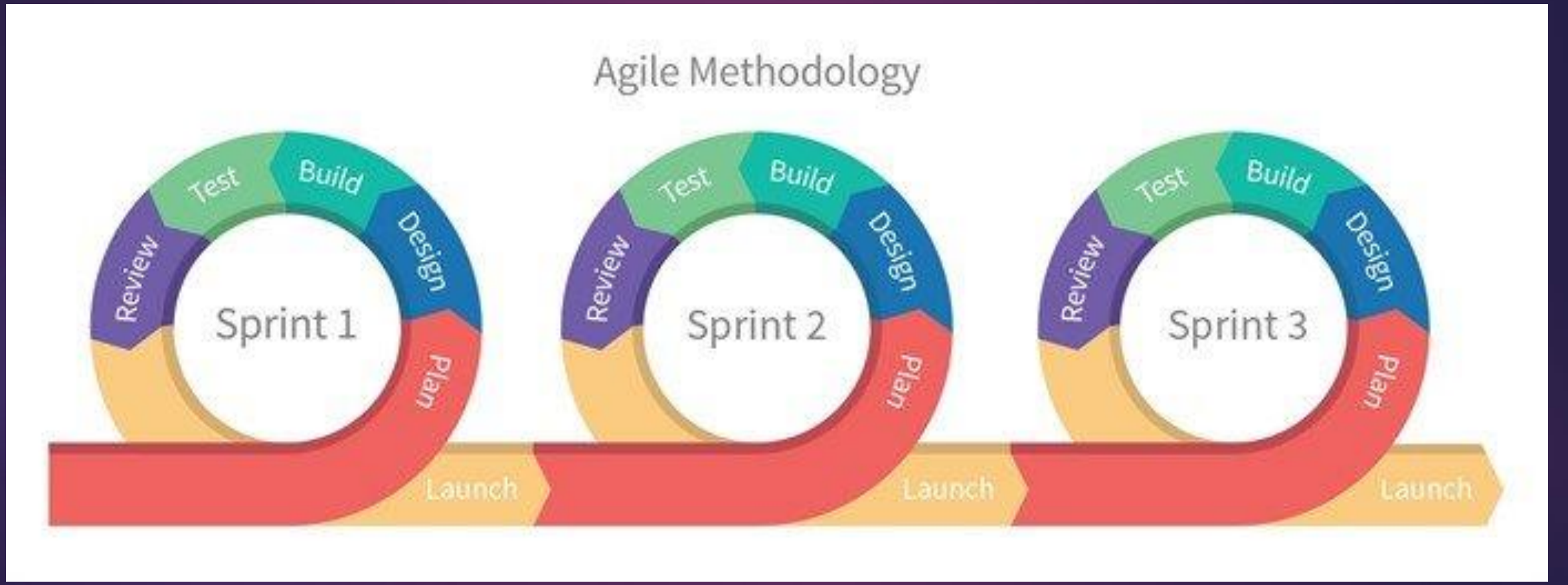
Advantages

- Simple and easy to understand and use
- Clearly defined stages.
- Easy to arrange tasks.
- Nice documentation is possible
- Good for small scale projects

Disadvantages

- Very rigid structure and thus slow
- No working software is produced until the end
- Cannot accommodate changing requirements.
- It is difficult to measure progress within stages.
- High amounts of risk and uncertainty.
- Not a good model for complex or large scale projects

Agile Model


Agile Methodology
Review
Test
Build
Design
Plan
Sprint 1
Launch
Review
Test
Build
Design
Plan
Sprint 2
Launch
Review
Test
Build
Design
Plan
Sprint 3
Launch

Key Idea

- An adaptive software development methodology
- Break the project into small incremental builds, also called sprint
- Incremental delivery of working software rather than documentation
- Interaction and collaboration of individuals

https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm

Advantages

- Enables concurrent development and delivery
- Allow users to realize software benefits by delivering partial working solutions.
- Easy to measure progress by stages
- Promotes teamwork and cross training.
- Accommodate changing requirements
- Improve working efficiency with team collaboration
- Good for large scale projects
- Lower risk factors

Disadvantages

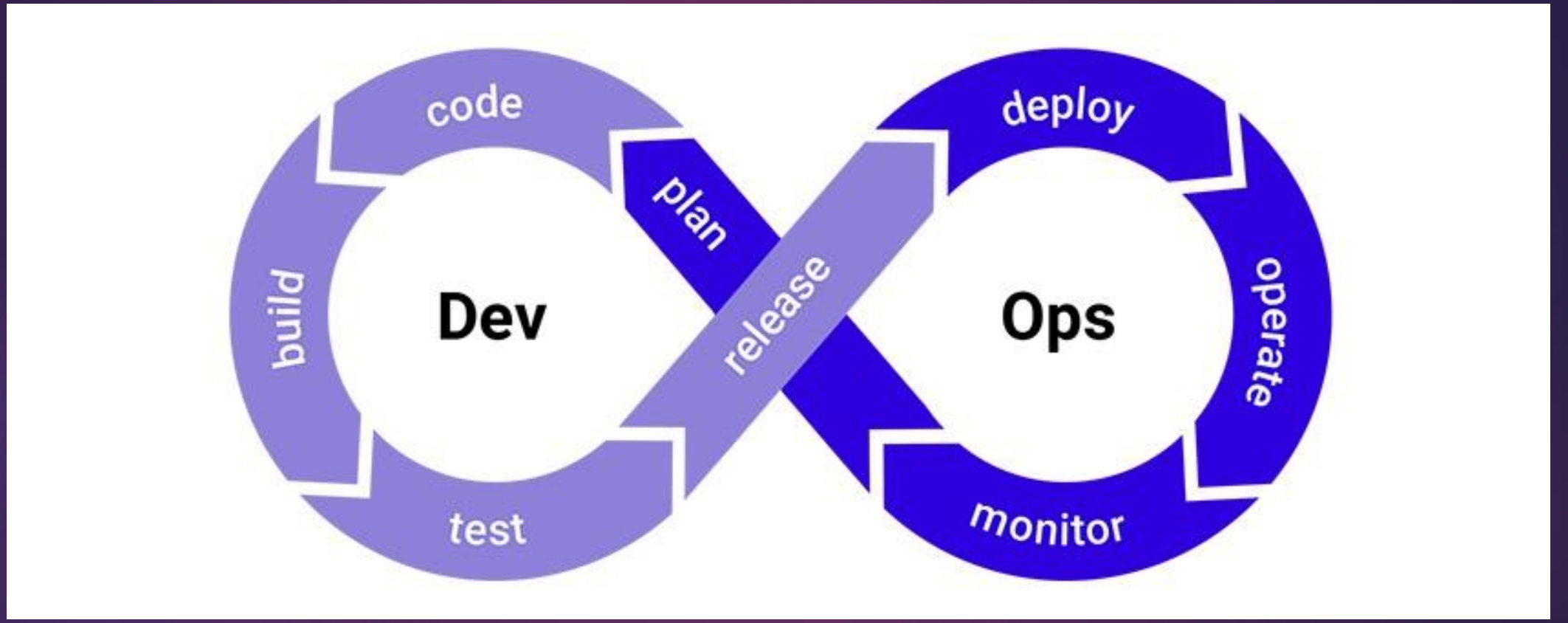
- High maintainability is a must
- Lack of documentation
- Difficult for a new comer to cope up
- Depends heavily on customer interaction

Different Forms of Agile

- Extreme Programming (XP)
- Scrum
- Crystal
- Feature Driven Development

DevOps

DevOps is a set of practices that bridges the gap between software development and IT operations.


code
plan
build
Dev
release
test
deploy
operate
Ops
monitor

Who is a DevOps engineer?

- *DevOps Engineer is somebody who understands the Software Development Lifecycle and has the outright understanding of various automation tools for developing CI/ CD pipelines.*

- So they need to take care of automating-
  - Building
  - Integration
  - Testing
  - Deployment
  - Issue tracking
  - Configuration management

DevOps Tools

- **GitHub Actions–** Source control https://github.com/features/actions
- **Jenkins –** Automation server, with plugins built for developing CI/ CD pipelines. https://www.jenkins.io/
- **Selenium –** Automation testing https://www.selenium.dev/
- **Docker, Kubernetes –** Software Containerization Platform
- **Jira, Trello** – Issue and project tracking software
- **Chef** – Configuration Management and Deployment https://docs.chef.io/

Thank You