

Software Development Life Cycle

Development Processes and Software Development Methodologies

SoftUni Team
Technical Trainers



SoftUni



Software University

<https://about.softuni.bg>

You Have Questions?

sli.do

#IT-BA

facebook.com

/groups/ITBusinessAnalysisAugust2023/

1. Software Development Life Cycle Basics

- Stages and Processes

2. SDLC Methodologies

- Basics
- Pros and Cons
- When to Use?





Software Development Life Cycle

Development Processes and Software Development
Methodologies

```
        object to mirror_ob
        mirror_object = mirror_ob

    def operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    def operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    def operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

    #selection at the end -add back the deselected
    mirror_ob.select= 1
    mirror_ob.select=1
    for context.scene.objects.active = modifier_ob
    mirror_ob.select = 0
    for key, context.selected_objects[0]
    for one, context.objects[one.name].select = 1

    print("please select exactly two objects,")
```

OPERATOR CLASSES

```
class MirrorOperator(bpy.types.Operator):
    """Mirror object to the selected object"""
    bl_idname = "mirror_mirror_x"
    bl_label = "Mirror X"
```

```
    def execute(self, context):
        if context.selected_objects is not None:
```

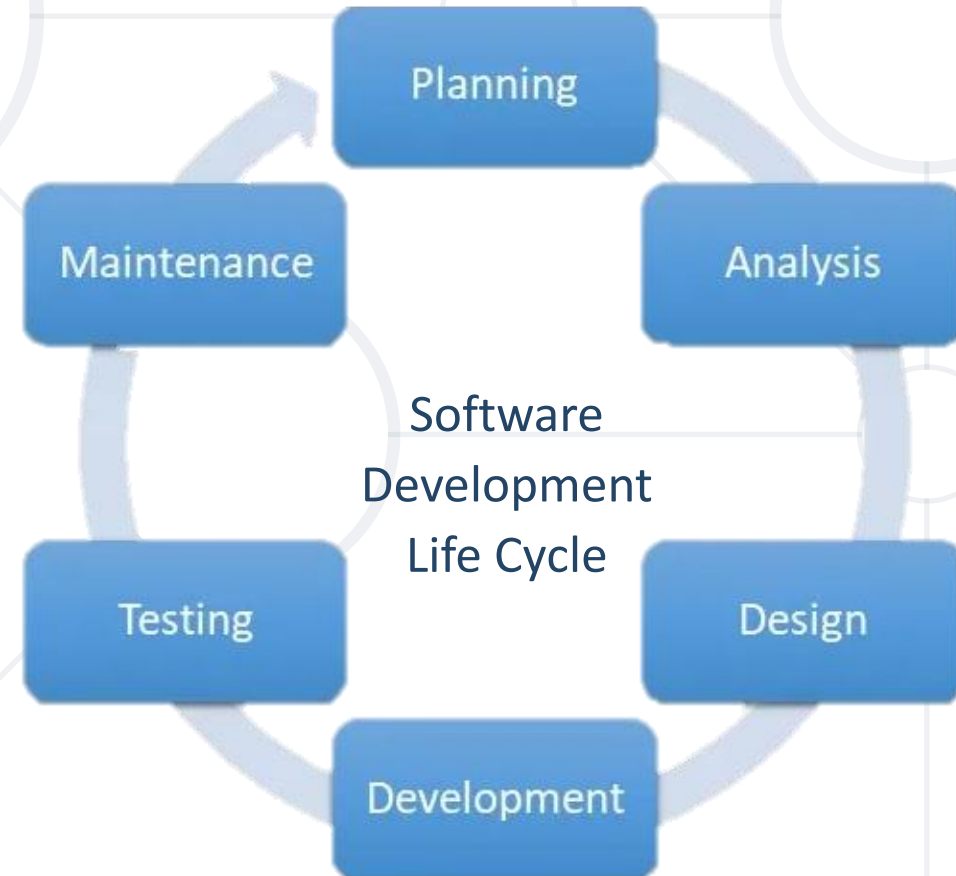
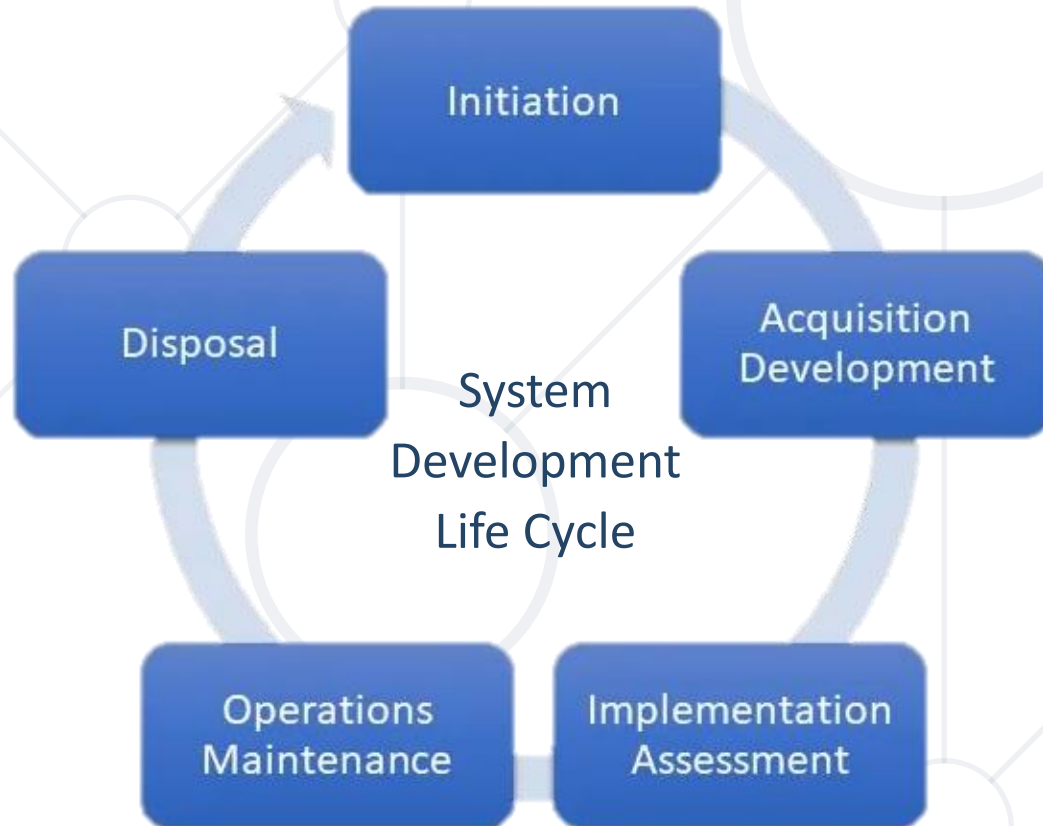


Let's Start With...

System Development Life Cycle (SDLC)

vs.

Software Development Life Cycle (SDLC)



SDLC & SDLC

System Development Life Cycle

- People
- Process
- Physical Architecture
- Training
- Organizational Announcements

Software development Life Cycle

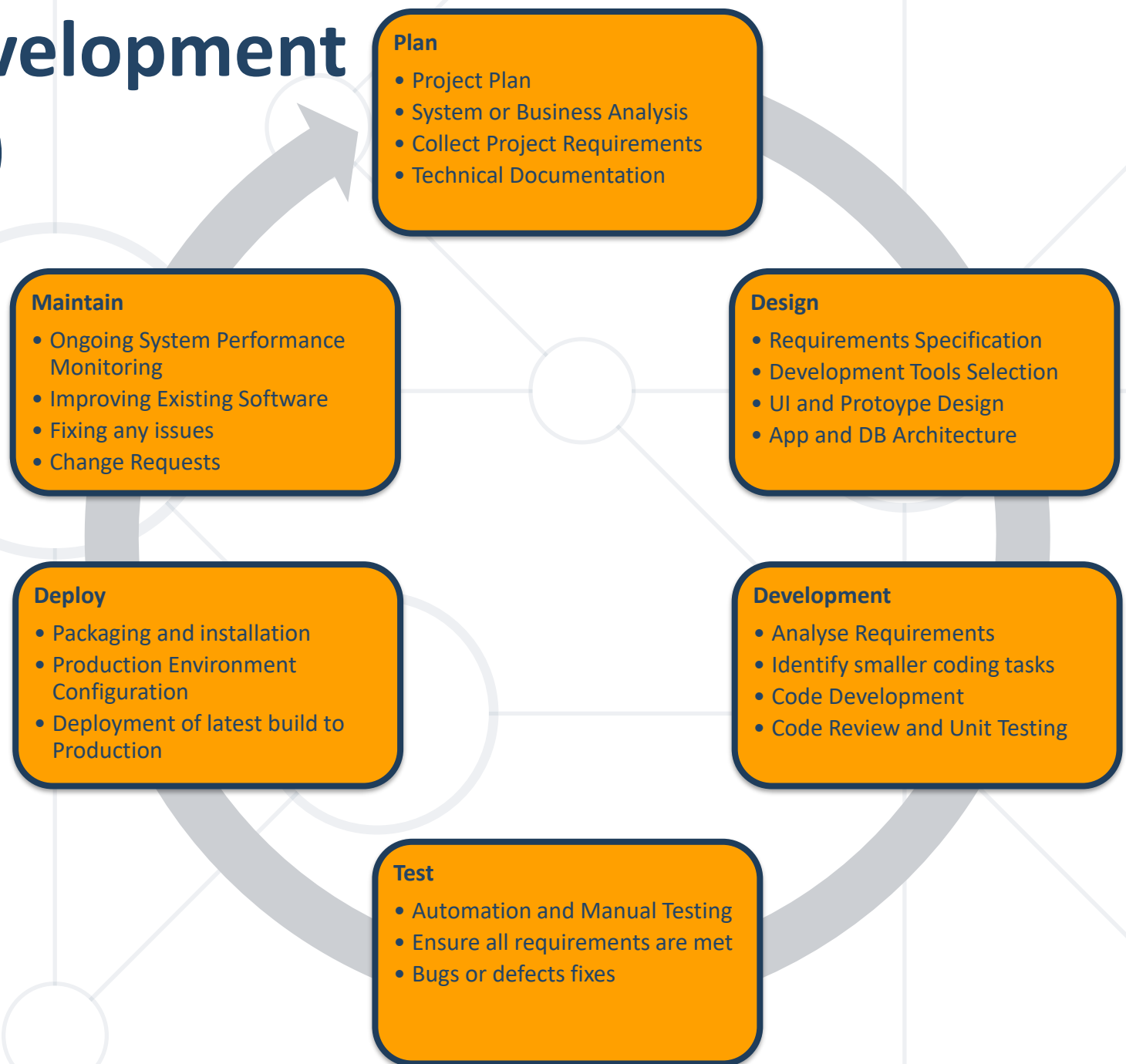
- Requirements
- Design Software Solution
- Development
- Testing
- Deployment
- Maintain Software Solution



What's Software Development Life Cycle (SDLC)?

*Software development processes can be divided into tasks that can be **assigned, completed, and measured** following the **Software Development Lifecycle (SDLC)** - a cost-effective and time-efficient methodology consisting of six stages*

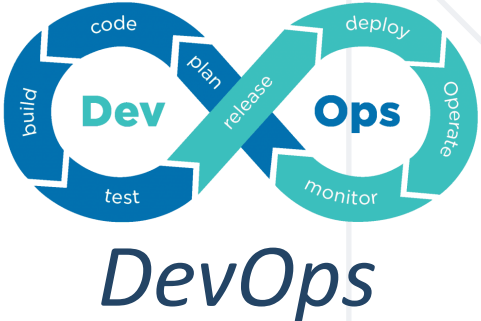
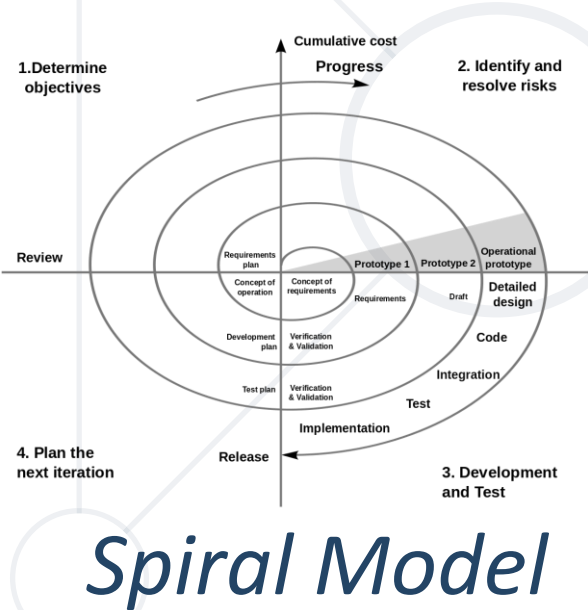
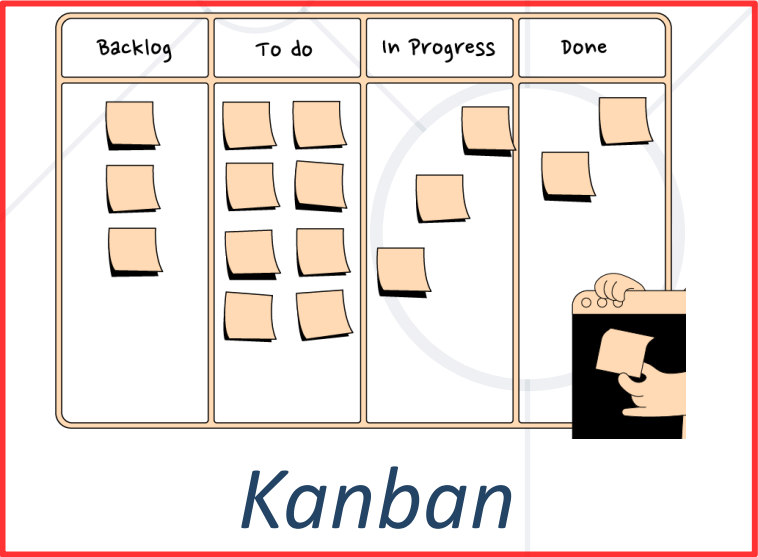
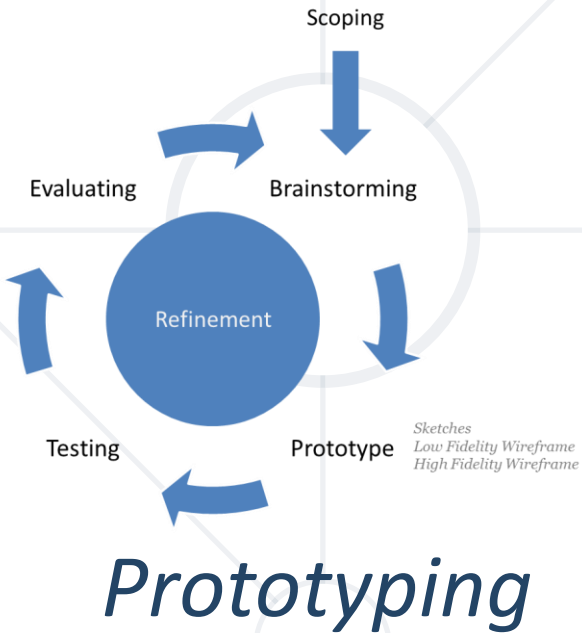
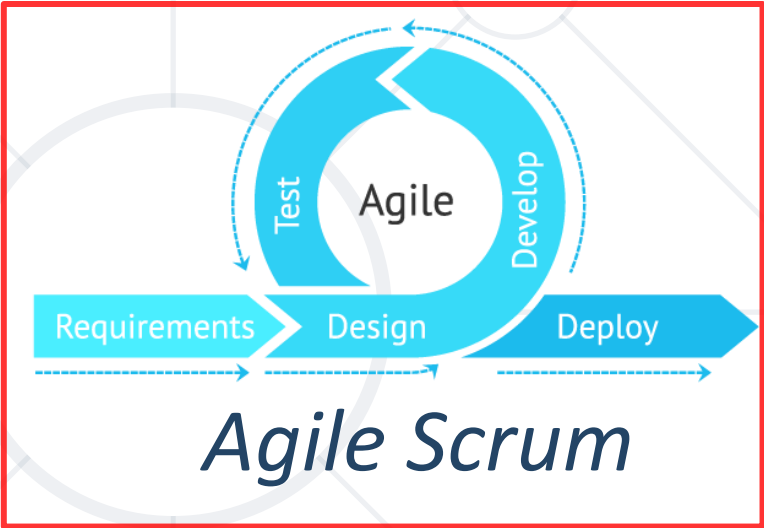
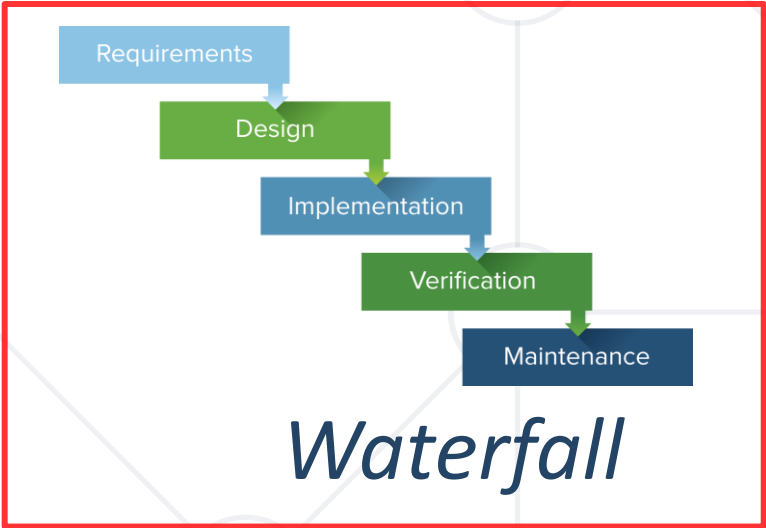
What's Software Development Life Cycle (SDLC)? (2)



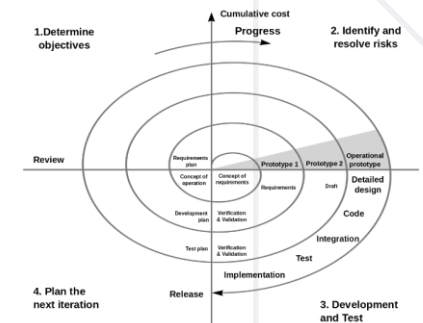


SDLC Methodologies

Methodologies

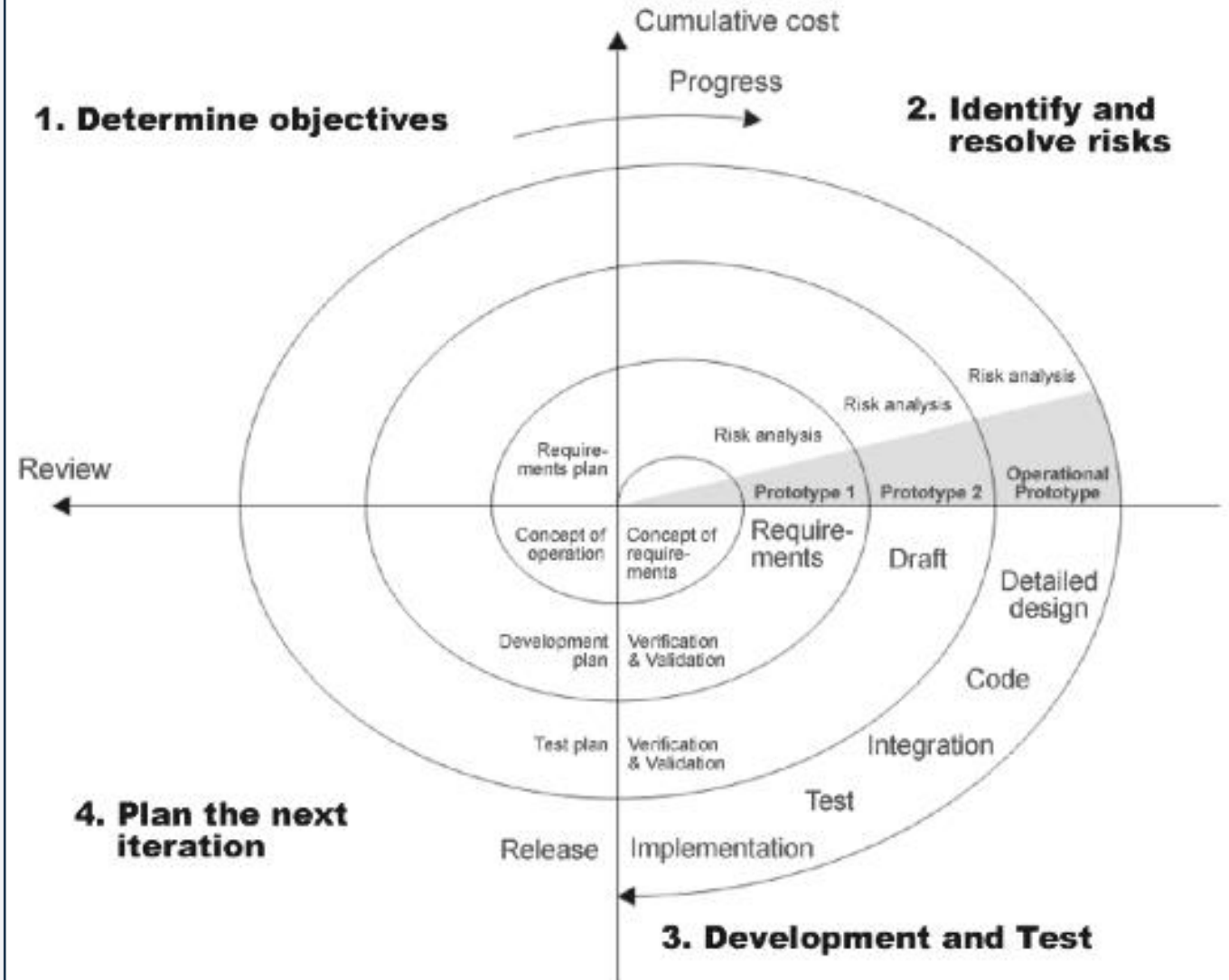


*In the **Spiral Model**, the software development process is represented as a spiral, with **each loop of the spiral representing a phase in the development lifecycle**. It is **Risk Driven** as it provides **support for Risk Handling**.*



Spiral Methodology

- Combination of **linear and iterative**
- Focuses heavily on **Risk Assessment**
- Minimized risk by breaking a project into **smaller segments**
- Each cycle **begins** with **identifying stakeholders and what success looks like**
- Each cycle **ends** with **a review and a commitment**



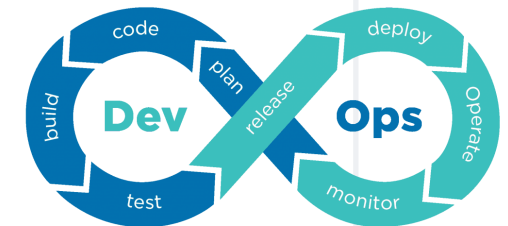
Spiral Model Pros and Cons

Advantages	Disadvantages
Accurately manages and mitigates risks	Long time to get to finished product
Issues are identified and solved early	High cost and Long Time
Estimates near the end of the project are very accurate	The success of the Spiral Model relies heavily on accurate risk analysis
Early involvement from developers increases successful design	Very customized solution limits reusability
Supports the development of prototypes early in the process.	More complex to manage compared to linear methodologies like the Waterfall Model

When to Use Spiral?

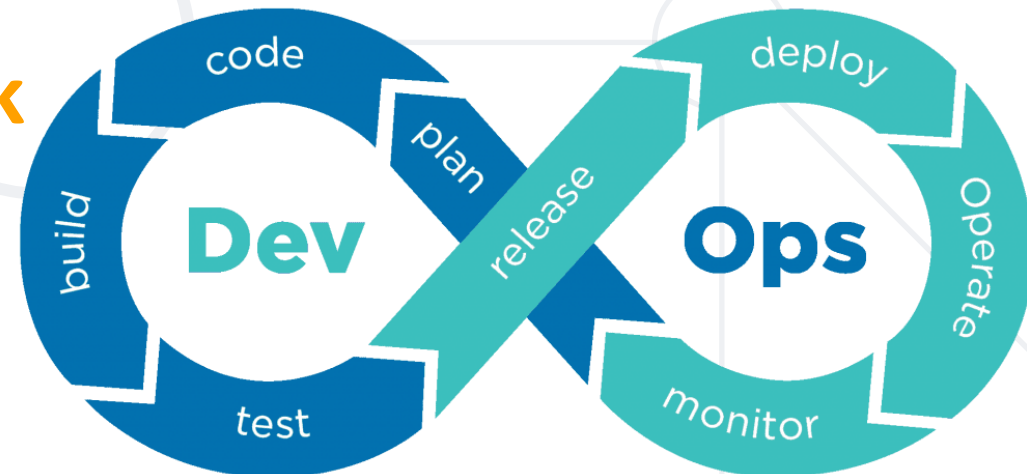
- Risk identification and mitigation are extremely important
- Medium to High-risk projects
- Users are unsure of their needs
- Prototypes are needed
- Requirements are complex
- Time and cost are not as important
- Little to no experience in project's area

*DevOps is a set of practices, principles, and cultural philosophies that aim to **improve collaboration, communication, and integration** between **development (Dev)** and **IT operations (Ops)** teams in the software development and deployment process.*



DevOps Methodology (2)

- **Automates repetitive tasks**, such as build, testing, deployment, and monitoring
- **Streamlines** the development and delivery process
- Emphasizes **Continuous Integration** and **Continuous Delivery (CI/CD)**
- **Continuous monitoring and feedback** of applications and infrastructure



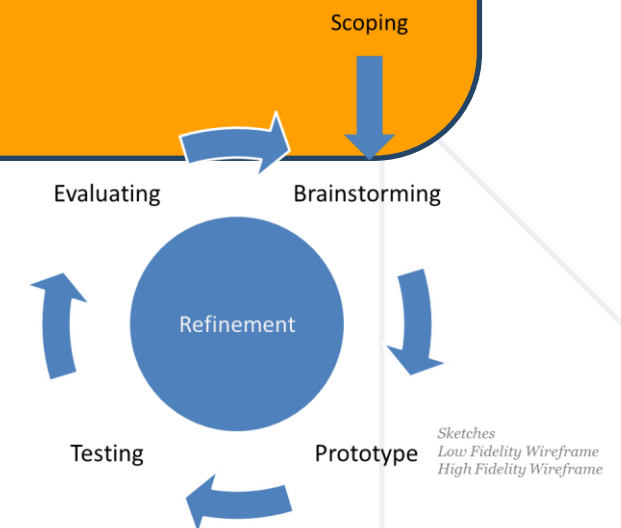
DevOps Pros and Cons

Advantages	Disadvantages
Enables rapid and frequent software releases	May require significant initial effort, especially in organizations with traditional development and operations silos
Fosters better communication and collaboration between development, operations, and other teams	Could increase complexity in the development pipeline
Automation reduces the risk of human errors, making deployments more reliable and consistent	Maintaining and updating automation scripts and tools
Enables organizations to scale their software development and deployment processes efficiently	Cultural shift

When to Use DevOps?

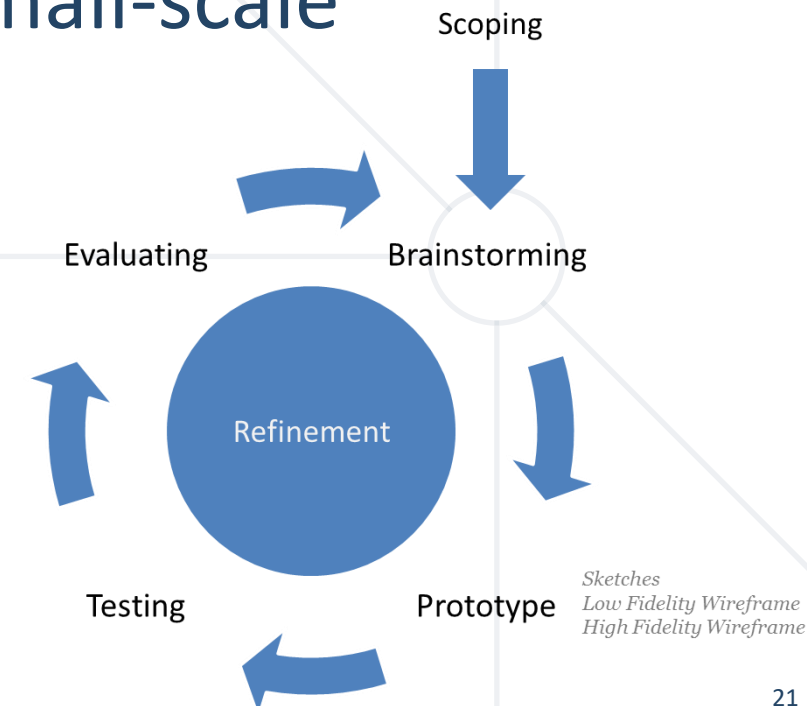
- If the organization needs to **release software updates or new features frequently**
- When you have **multiple teams working on a project**, such as development, testing, operations, and security
- DevOps **complements cloud-based architectures** and **microservices**
- If the development process involves **many repetitive tasks**

*The **Prototyping** model is a software development method in which a **prototype** is built, tested and then **reworked** as necessary until an acceptable outcome is achieved from which the **complete system or product** can be developed.*



Prototyping Methodology (2)

- Iterative progression
- Used in conjunction with Spiral, Rapid Application Development (RAD), and Incremental models
- Breaks project into segments and creates small-scale mock-ups of the system, called prototypes
- Prototypes are iterated until they meet the requirements
- Final prototype usually discarded
- Business user is involved in the full process



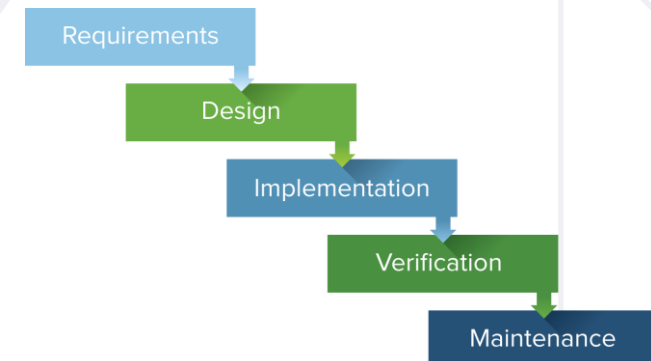
Prototyping Pros and Cons

Advantages	Disadvantages
Gives business users a visual of their wants and needs	Very loose approval and control process
Promotes user participation and communication	Non-functional requirements are tough to identify
Allows progress even when unclear requirements	Can lead to poorly designed systems
Encourages innovation	False expectations thinking the system is complete from prototype

When to use Prototyping?

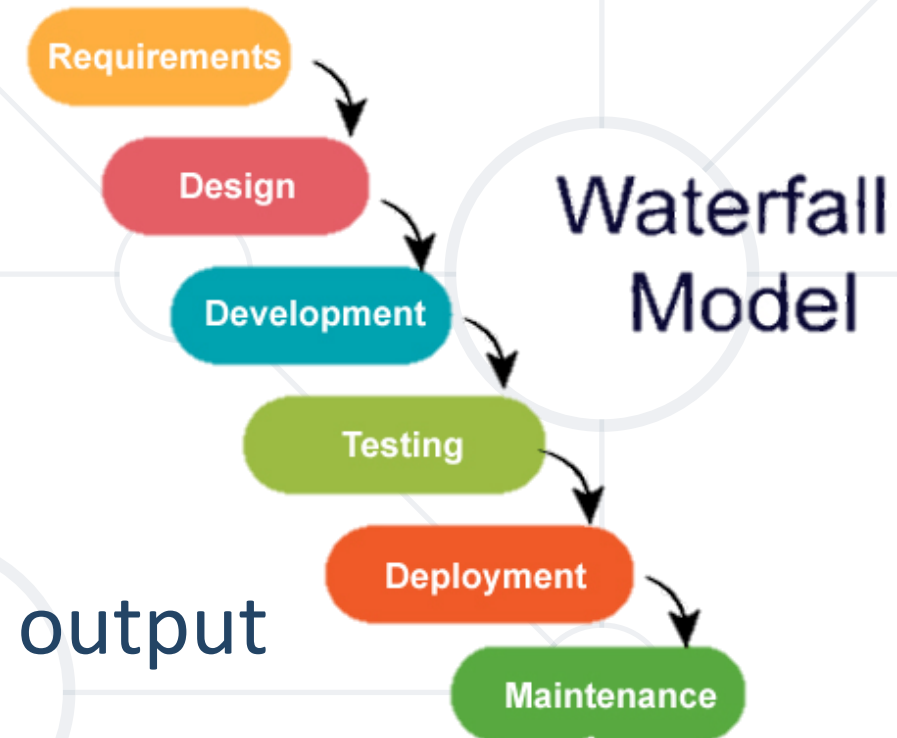
- Project objectives are unclear
- High pressure to implement "something"
- Frequent requirement changes
- Minimal resource constraints
- No strict approval processes are needed
- Innovative, flexible designs that need to accommodate future changes

*The Waterfall Methodology is a **sequential development process** that flows like a waterfall through all phases of a project, with **each phase completely wrapping up before the next phase begins.***

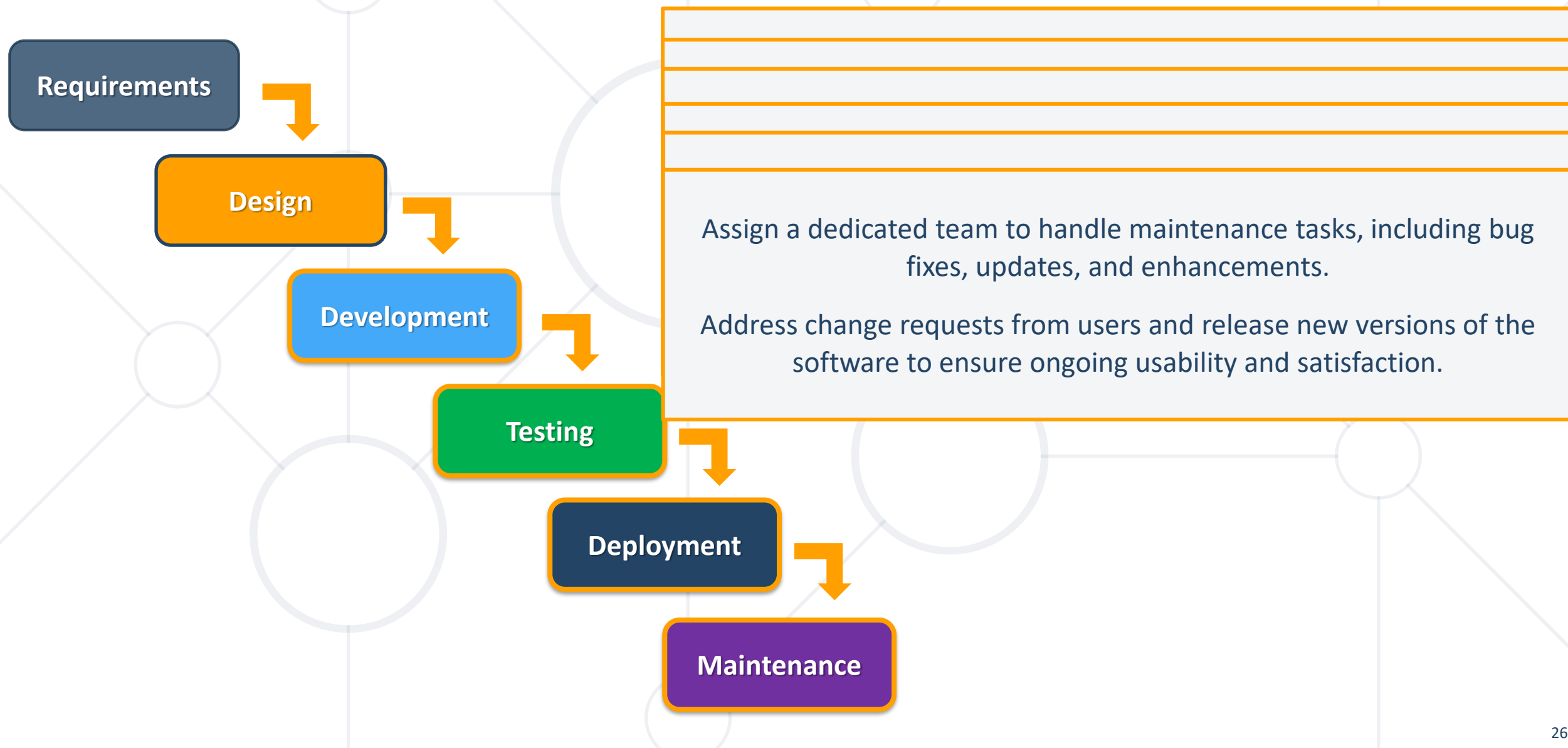


Waterfall Methodology (2)

- Linear progression
- "Traditional" model
- Next phase begins once the previous phase is complete
- Each phase begins with previous phase's output
- No process to go back to the previous phase
- Fully planned, including time schedule and budget
- Extensive written documentation and formal approval process



Waterfall Methodology (3)



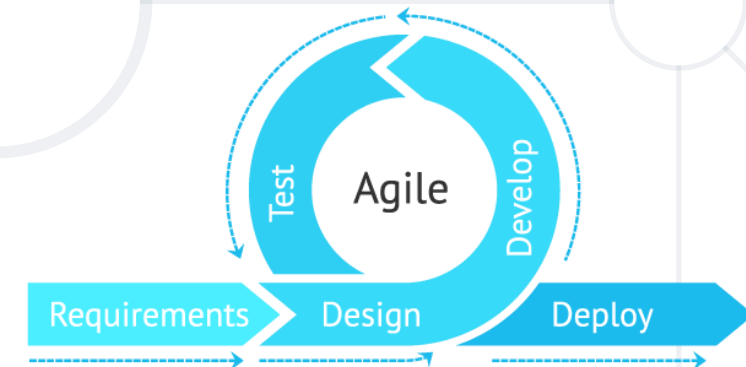
Waterfall Pros and Cons

Advantages	Disadvantages
Clear expectations on schedule, budget, and resourcing needs	Inflexible and cumbersome
Extensive documentation helps ensure quality, reliability, and maintainability	Long pole from project start to something tangible
Progress is easily measured	Problems are discovered at user testing
	Written documentation is never kept up-to-date, loses usefulness
	Promotes gap between the business users and the development team

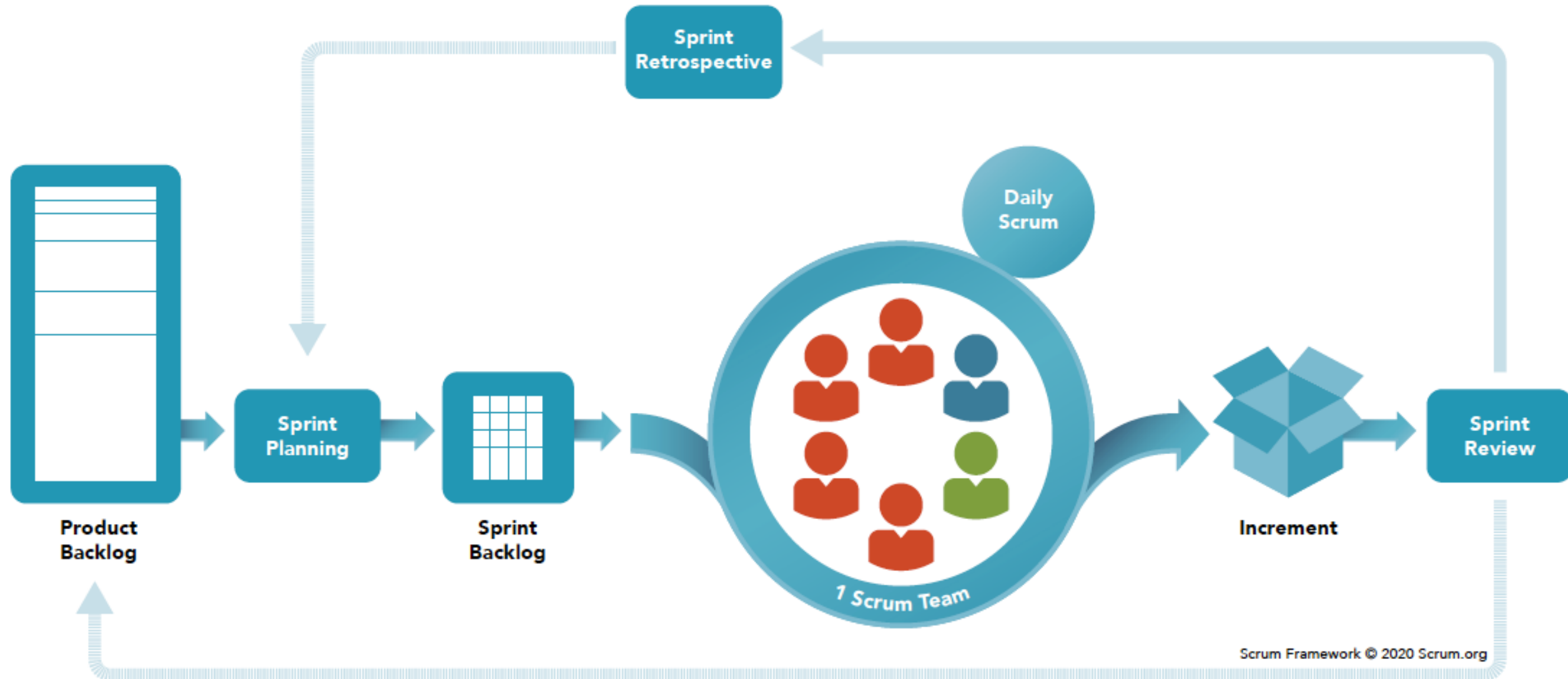
When to Use Waterfall?

- Clear objectives and solution
- Large, complicated, and expensive
- No pressure for immediate return on investment (ROI) from implementation
- Project team is fully knowledgeable about the solution application
- Requirements are stable and will be unchanged through development
- Resource constraints
- Strict requirement for formal approvals at milestones

*Agile methodology is a project management framework that breaks projects down into **several dynamic phases**, commonly known as **sprints**. The Agile framework is an **iterative methodology**.*



Agile Scrum (1)

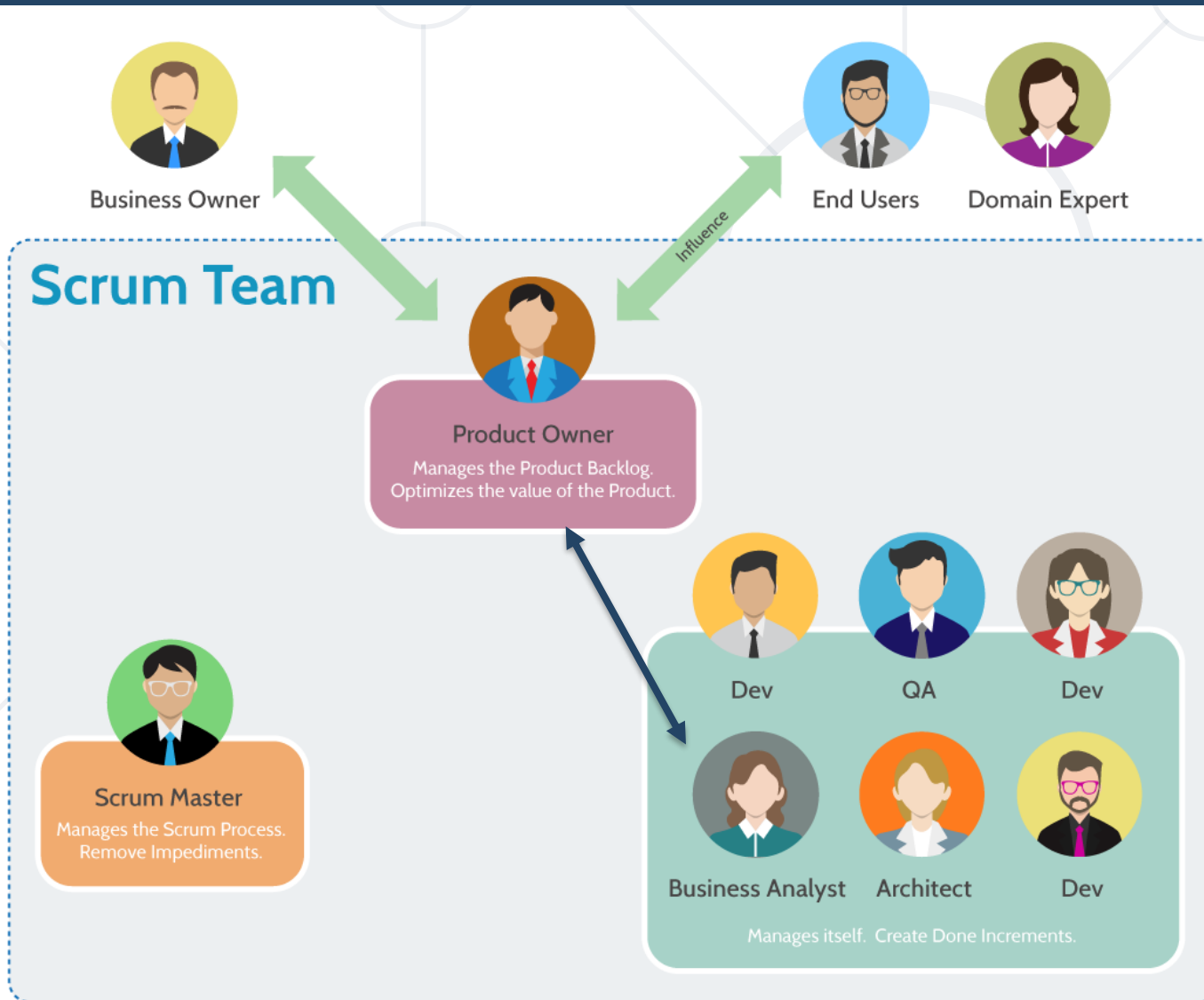


Agile Scrum (2)

- Uses iterative approach called Sprints
- Flexible and adaptable, great for the unknown
- Values individuals and interactions over processes and tools
- Values working software over comprehensive documentation
- Values customer collaboration over contract negotiation
- Values responding to change over following a plan



Agile Scrum Team



Note: Nearly 95% of the time the BA and Product Owner is one person in Scrum.

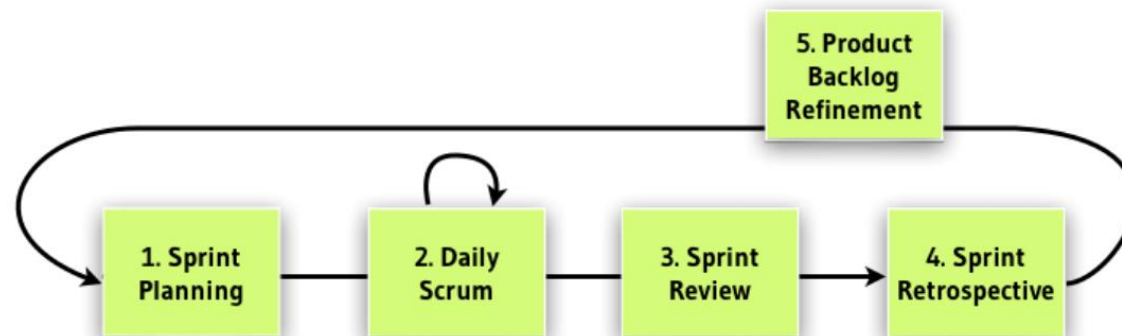
Agile Scum Ceremonies

Sprint Planning: At the beginning of each sprint, the team collaboratively selects a subset of items from the product backlog to work on during that sprint. They plan how to complete these items and define the sprint goal.

Daily Stand-ups: Short daily 10-15 minutes meetings are held, where team members share updates on what they accomplished the previous day, what they plan to do that day, and any impediments they are facing.

Sprint Review: At the end of each sprint, the team demonstrates the completed work to stakeholders and receives feedback. This meeting helps the team gather insights and make improvements.

Sprint Retrospective: Following the sprint review, the team holds a retrospective to evaluate their performance during the sprint. They identify what went well, what could be improved, and action items to enhance their processes.



Agile Scrum Artifacts are information that a scrum team and stakeholders use to detail the **product being developed, actions to produce it, and the actions performed during the project.**

Scrum artifacts

Product backlog

Sprint backlog

Increment

Definition of done

Burndown chart

The **Product Backlog** is a **list of new features, enhancements, bug fixes, tasks, or work requirements** needed to build a product. It's compiled from input sources like customer support, competitor analysis, market demands, and general business analysis.



Prioritized list of items



Dynamic and Evolving

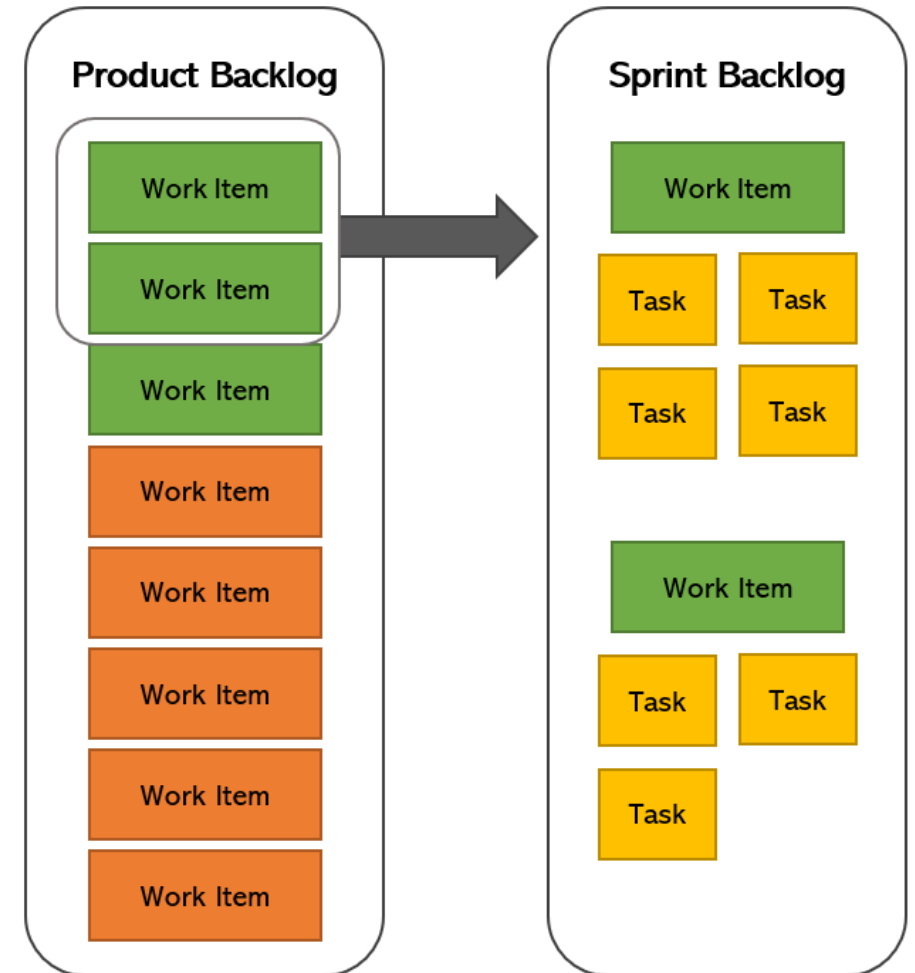


User-Centric

Agile Scrum Artifacts – **Sprint Backlog**

The **Sprint Backlog** is a set of **product backlog tasks** that have been **promoted to be developed during the next product increment**.

They are created by selecting a task from the product backlog and **breaking that task into smaller, actionable sprint items**.



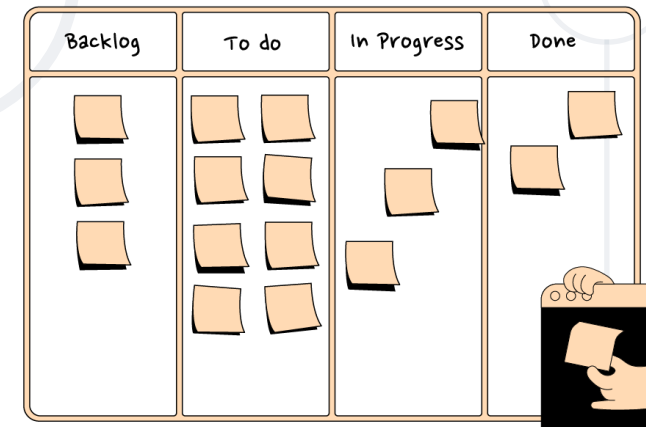
Agile Scrum Pros and Cons

Advantages	Disadvantages
Flexible and adaptable to changing requirements	Leads to scope creep due to no defined end date
Gets workable product in-front of the business stakeholders quicker	Relies on commitment of all team members
Promotes collaboration between business and development teams	Challenging to initially adopt and train in an organization
Daily feedback on progress and roadblocks which prevent from completion of tasks	Changing or leaving team members can have a drastically negative effect

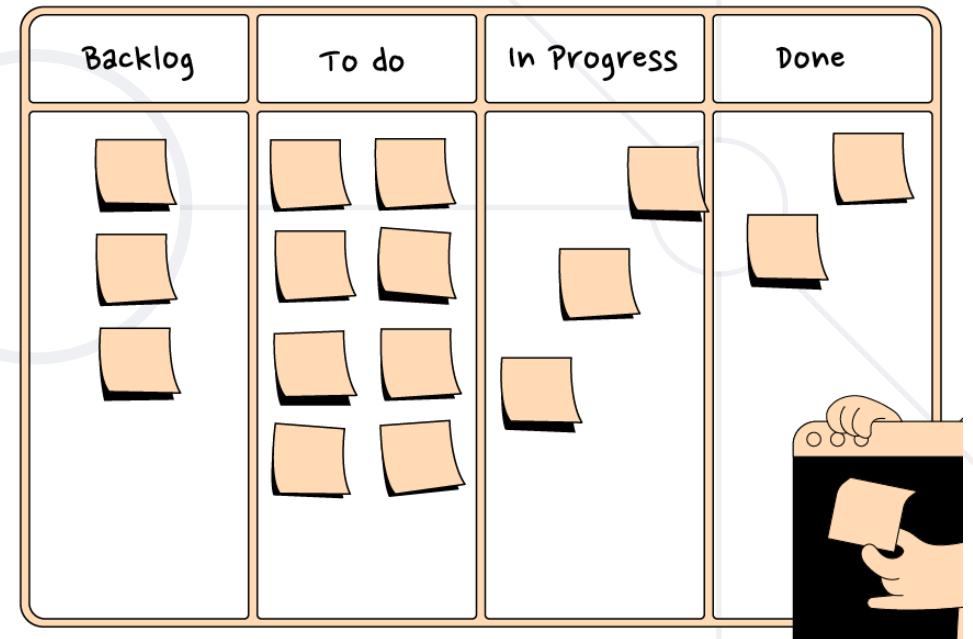
When to Use Agile Scrum?

- The project is unpredictable and will have changing requirements
- Using or creating leading edge technology
- Organization as an experienced Scrum Master
- Business has experienced resource that can dedicate time to the project
- Pressure to produce a tangible product quickly
- Little to no concerns on length of project or budget
- Development team doesn't have resource constraints

*Kanban provides a **visual representation of work items, tasks, or user stories** on a physical or digital board. The board typically consists of **columns representing different stages of the workflow**, and each work item is represented by a card that moves across the columns as it progresses through the development process.*



- Visual Representation as it utilized form of a board
- Transparency
- Work In Progress Limits for not overloading team members
- Easy management of workflow and work items
- Ability to be adopted with all other SDLC methodologies



How to Choose SDLC Methodology?



- Most times this is already set by the business
- There are different **factors** that are **considered** (project size, time constraints, client involvement, etc.)
 - The methodology **needs to fit the particular business**, the **project** and the **project team**
- Sometimes it is most appropriate to use a combination of **SDLC methodologies** (e.g., Agile and Waterfall)

- There is a difference between **System Development Life Cycle** and **Software Development Life Cycle**
- The most adopted SDLC methodologies are **Agile Scrum** and **Waterfall**
- All methodologies have Pros and Cons, depends on the kind of project and organization's culture which to use
- IT BA's responsibilities are almost the same as the Product Owner's ones in Agile Scrum



SoftUni Diamond Partners

**SUPER
HOSTING
.BG**



**Coca-Cola HBC
Bulgaria**



POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers



AMBITIONED

 **DRAFT
KINGS**



**SOFTWARE
GROUP**

createX



Postbank

Решения за твоето утре

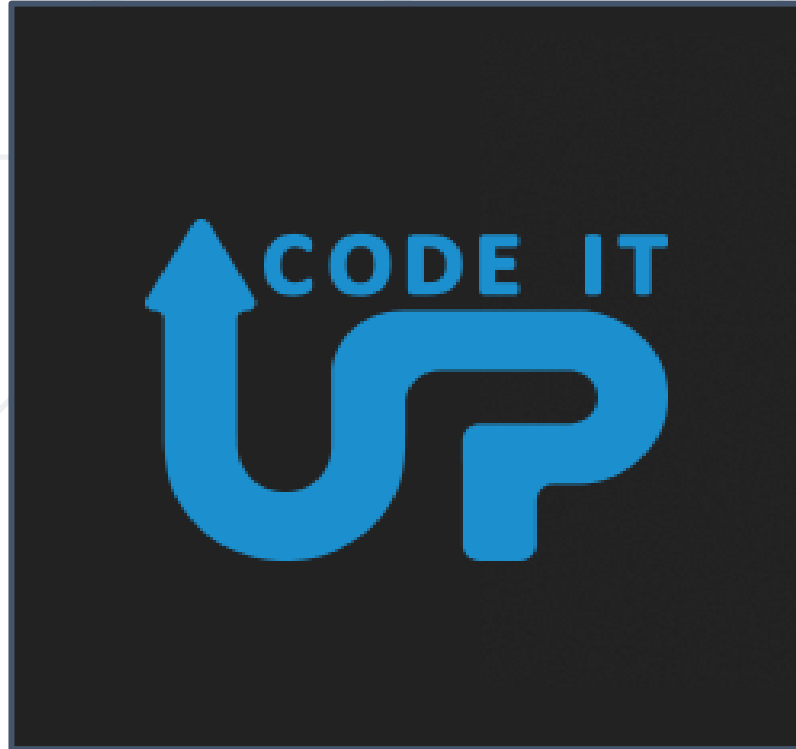


BOSCH

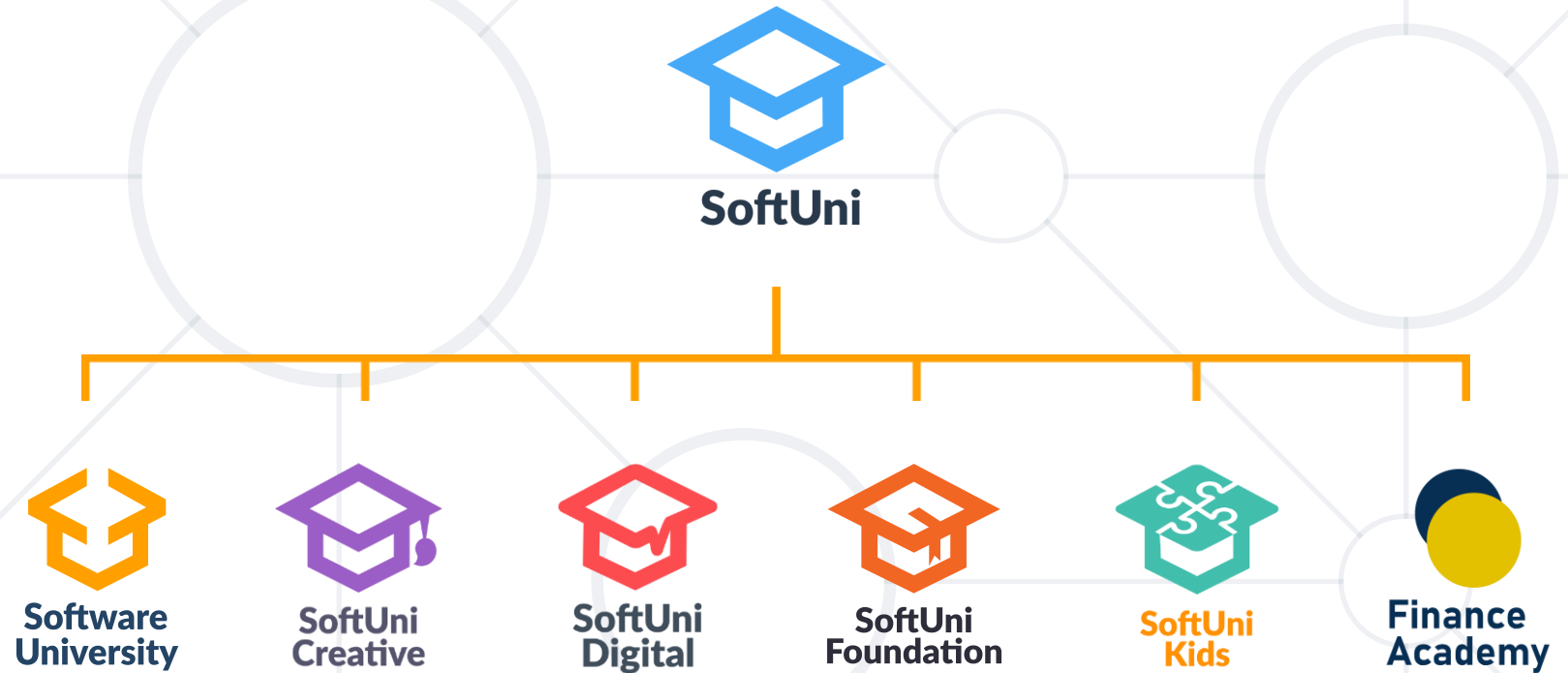
DXC
TECHNOLOGY



SmartIT



Questions?



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers

- softuni.bg, about.softuni.bg

- Software University Foundation

- softuni.foundation

- Software University @ Facebook

- facebook.com/SoftwareUniversity

- Software University Forums

- forum.softuni.bg

