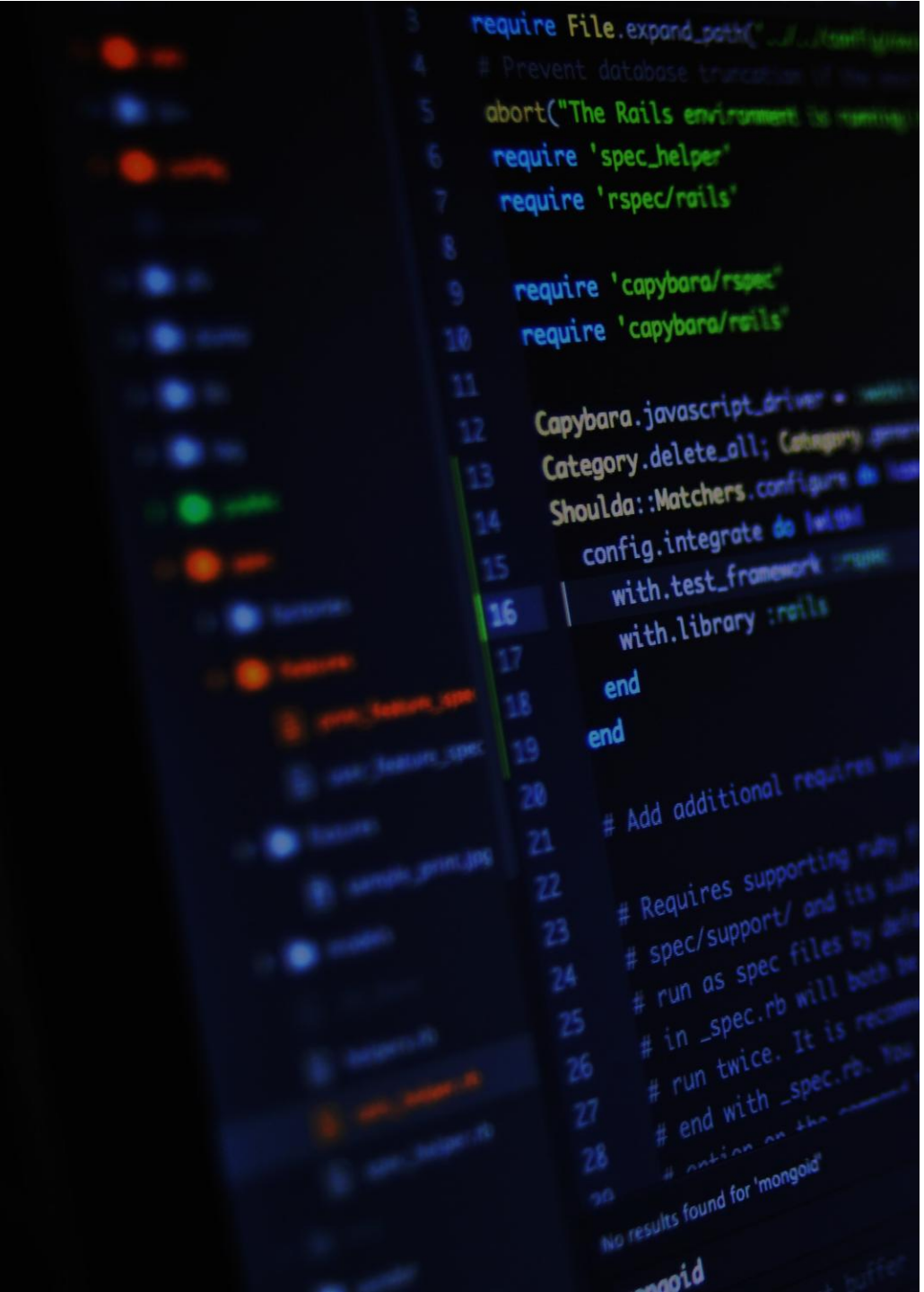


LECTURE 1

Introduction to Software Engineering

WEB DEVELOPMENT 2

2025/2026 2ND TERM

ASSOC. PROF. DR. ASHRAF ABDELAZIZ TAHA

1 Why Software Engineering?

- Software development is hard.
- Important to distinguish “easy” systems (one developer, one user, experimental use only) from “hard” systems (multiple developers, multiple users, products).
- Experience with “easy” systems is False.

2 Goals of Software Development

- Satisfy users requirements.
- High reliability.
- Low maintenance costs.
- Delivery on time.
- Low production costs.
- High performance.
- Ease of reuse.

2.1 Satisfying User Requirements

- Many programs simply don’t do what end users want.
- Typical percentages for large-scale ordered systems:
  - 45% delivered but not used
  - 27% paid for but not delivered
  - 17% ignored
  - 6% used after changes
  - 5% used as delivered

- Users find it hard to describe what they want.
- Developers find it hard to understand what users say!

2.2 High Reliability

- Mistakes in programs are generically known as bugs.
- Bugs can be expensive, in terms of. . .
  – Human lives: in safety critical systems, e.g., nuclear reactor control and fly-by-wire aircraft
  – Money: software bug in failed Ariane 5 launch cost US$500 Million
  – Poor customer relations: Microsoft problems with original Windows release caused the company huge problems.

2.3 Low Maintenance Costs

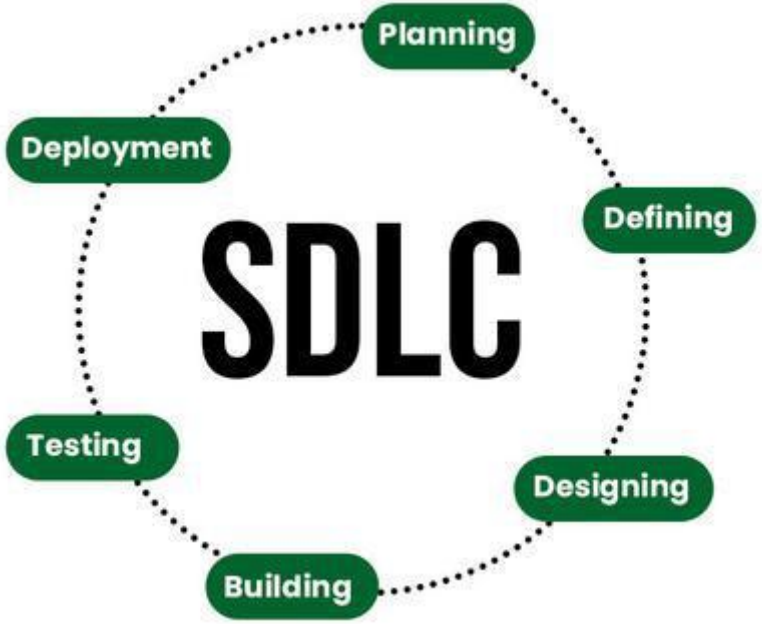
- Maintenance is what is done to software after it starts being used.
- Maintenance may be:
  - Corrective — fixing bugs (21%);
  - Adaptive — altering software to fit changing software (25%);
  - Perfective — to meet new requirements (50%);
  - Preventative — to reduce further maintenance (4%).
- Maintenance typically accounts for 65% of overall project costs.

3 The Software Process

Software development life cycle (SDLC):

A structured process used for design, develop, and test good-quality software.

SDLC, is a methodology that defines the entire procedure of software development step-by-step.



Stages of the Software Development Life Cycle

SDLC specifies the task(s) to be performed at various stages by a software engineer or developer.

It ensures that the product is able to meet the customer's expectations within the overall budget.

Hence, it's vital for a software developer to have prior knowledge of this software development process.

stages

Stage-1: Planning and Requirement Analysis

- Planning is important, like making a roadmap.
- Requirement analysis helps in understanding what's needed for the project.
- This information is used to create the basic plan for the project.

Stage-2: Defining Requirements

- We list down everything the software needs to do.
- People like customers and stakeholders approve these requirements.
- We use a special document (SRS) to keep everything clear.

Stage-3: Designing Architecture

- We use the SRS to decide how the software will be built.
- Different ideas for building the software are put in a document called DDS.
- We pick the best idea from DDS for moving forward.

Stage-4: Developing Product

- Now, we start building the actual software based on the chosen design.
- We use specific computer languages like C/C++, Python, or Java.
- Following the rules is very important during this stage.

Stage-5: Product Testing and Integration

- After building, we check the software thoroughly to find and fix any mistakes.
- This is like testing a car before selling it.
- making documents (manual) to help people use the software and offer support.

Stage-6: Deployment and Maintenance of Products

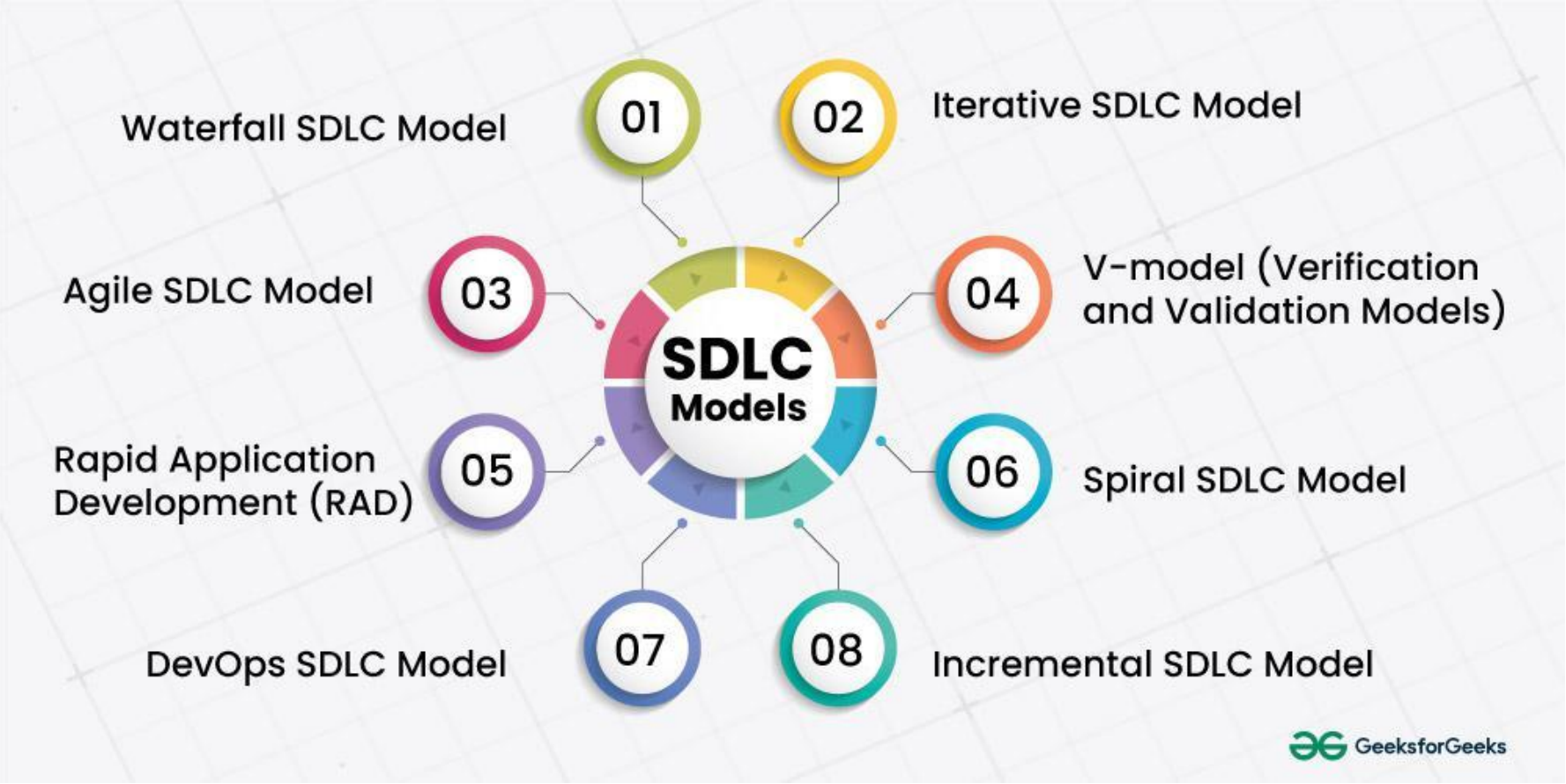
- We release the software in parts based on a plan.
- We check how well it works in the real world.
- We keep an eye on it even after release to make sure it works well. If people suggest improvements, we make them.

3.1 Software Development Models

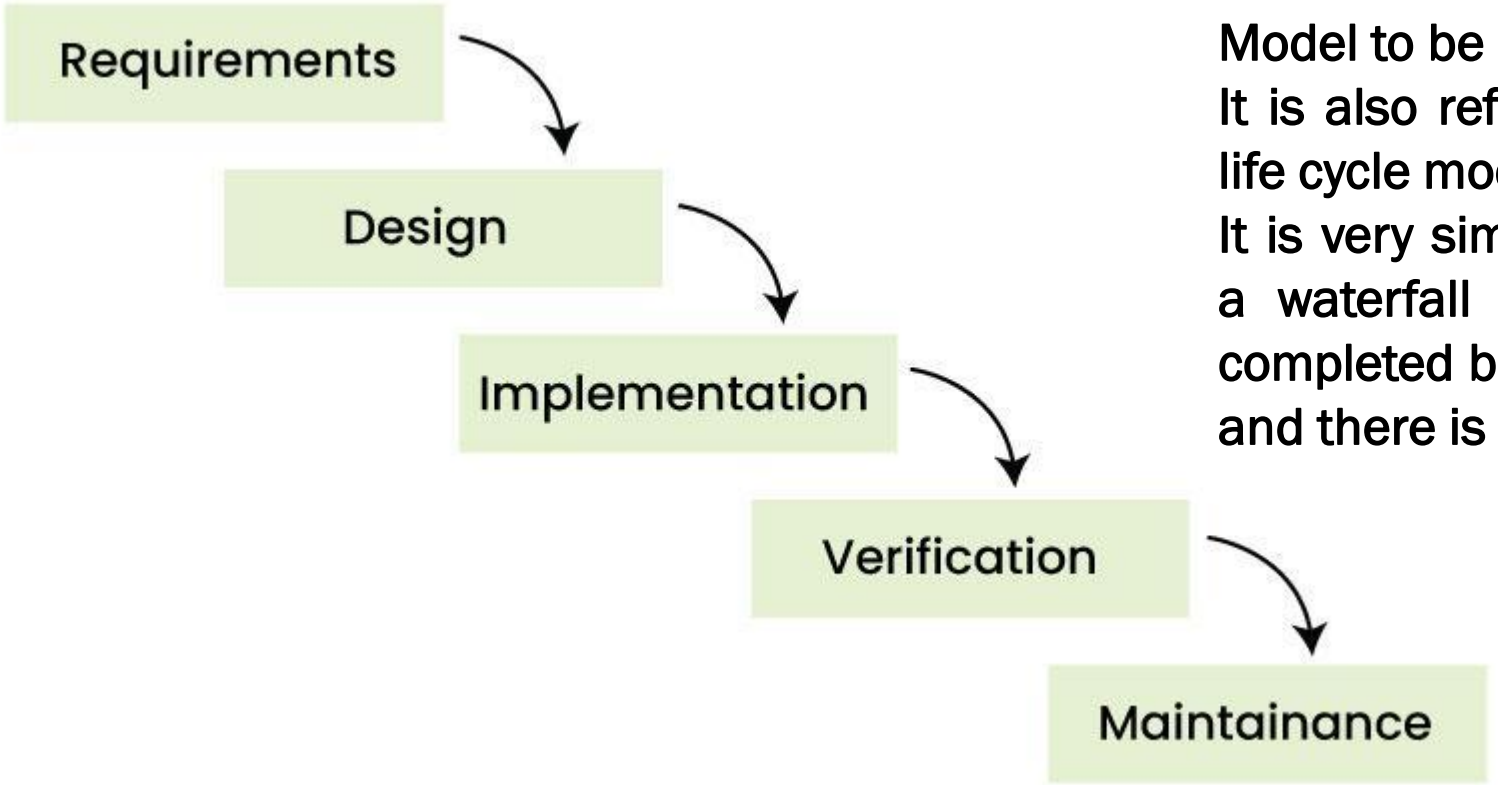
Various processes or methods that are chosen for project development depending on the objectives and goals of the project.

Many development life cycle models have been developed to achieve various essential objectives. Models specify the various steps of the process and the order in which they are executed.

3.1 Software Development Models



Waterfall Model



The Waterfall Model was the first Process Model to be introduced.
It is also referred to as a linear-sequential life cycle model.
It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

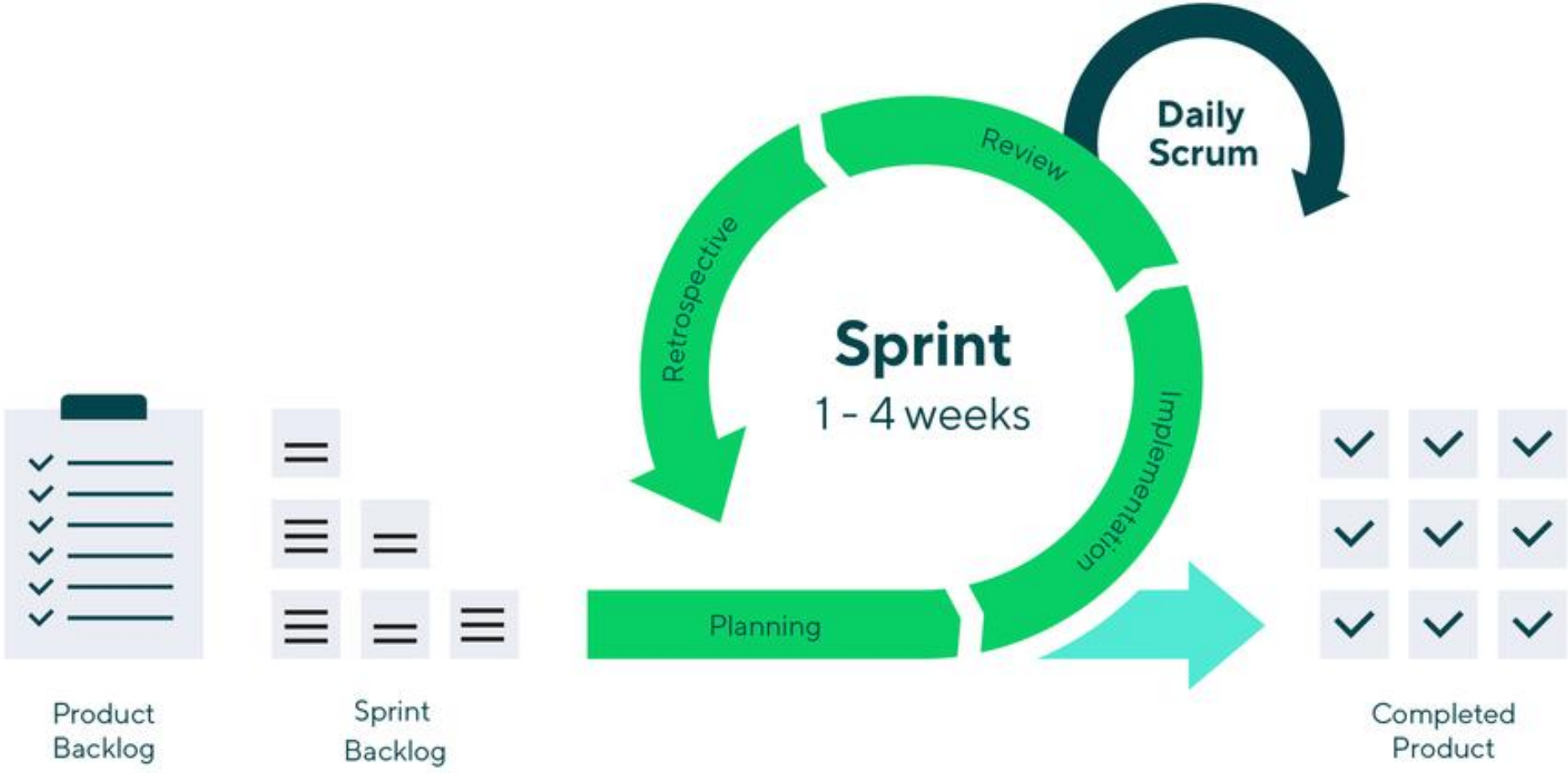
Waterfall Model Advantages:

1. **Simplicity:** Easy to understand with a straightforward sequential approach.
2. **Well-Defined Phases:** Clear and distinct phases with specific goals.
3. **Clear Milestones:** Provides checkpoints for monitoring progress.
4. **Stable Requirements:** Works well when project requirements are stable.

Waterfall Model Disadvantages:

1. **Inflexibility to Changes:** Difficult to make changes once a phase is completed.
2. **Limited Client Involvement:** Client input mainly at the beginning and end, restricting feedback.
3. **Long Delivery Time:** Sequential nature may result in a longer time to deliver a functional product.
4. **Risk of Uncertainty:** May not adapt well to uncertain or changing requirements.

2. Agile SDLC Models (Software Development Models):




Thank
you!