

JANUARY 2021

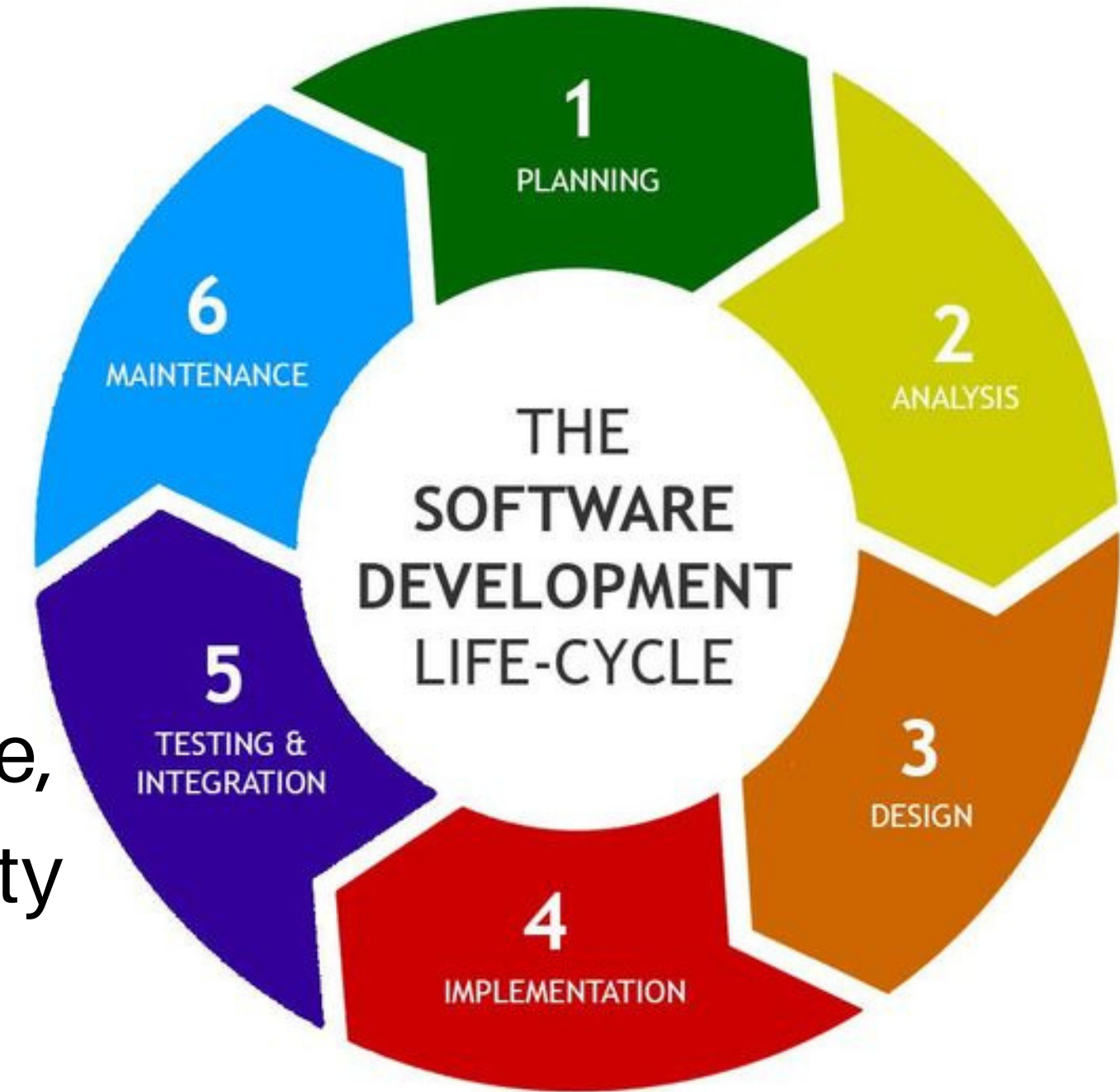
Agile Framework - Software development

prepared by Trueye Zerihun



Software Development Life Cycle

1. **Planning:** This includes calculating labor and material costs, creating a timetable with target goals, and creating the project's teams and leadership structure, feedback from stakeholders
2. **Define Requirements:** what the application is supposed to do and its requirements. For example, a social media application would require the ability to connect with a friend





Software Development Life Cycle

3.Design and Prototyping

Architecture – Specifies programming language, overall design, and use of any templates or boilerplate

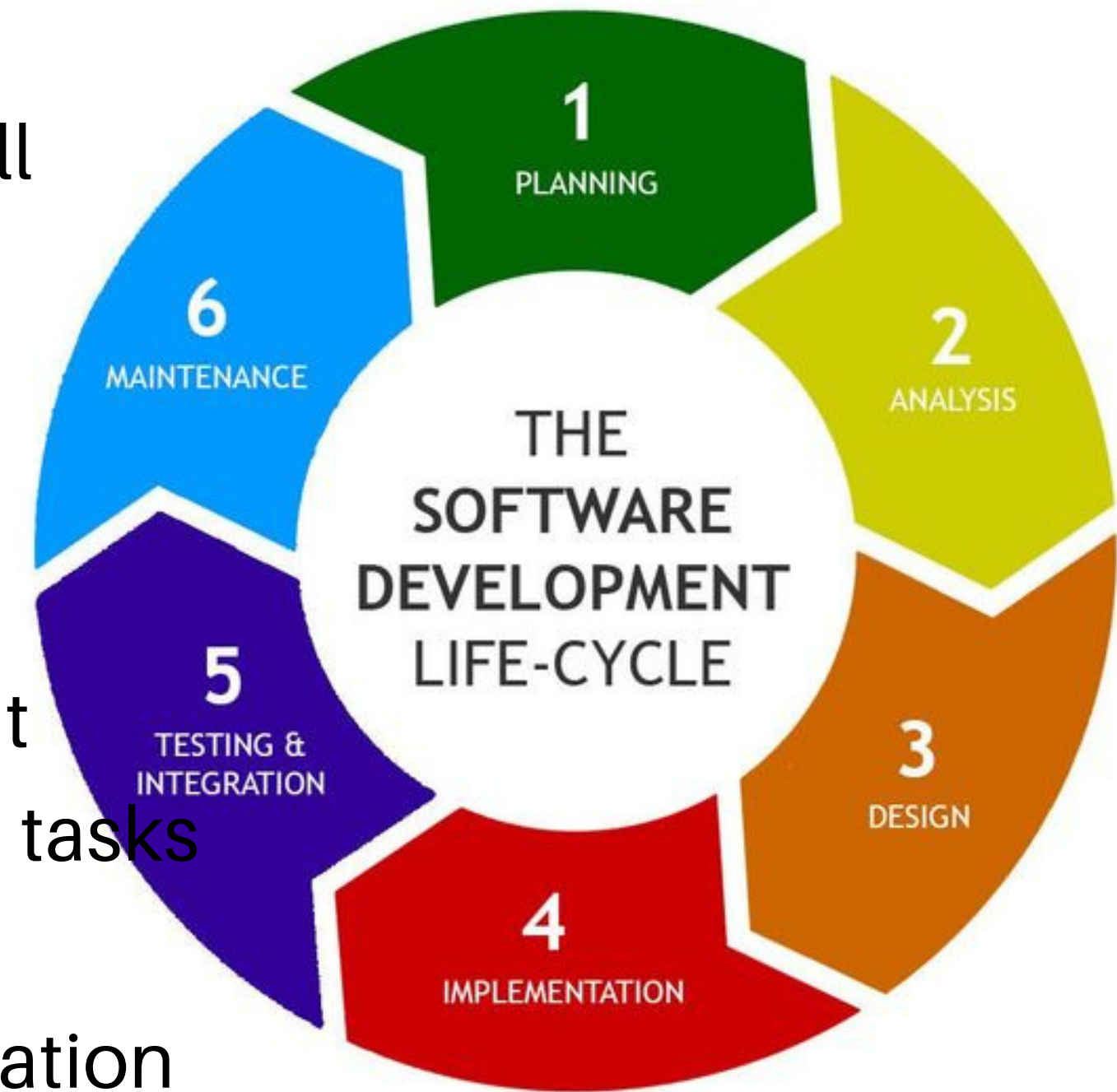
User Interface – Defines the ways customers interact

Platforms – software will run, such as Apple, Android, Windows

Programming – Not just the programming language, but including methods of solving problems and performing tasks in the application

Communications – central server or instances of application

Security





Software Development Life Cycle

4. Software Development

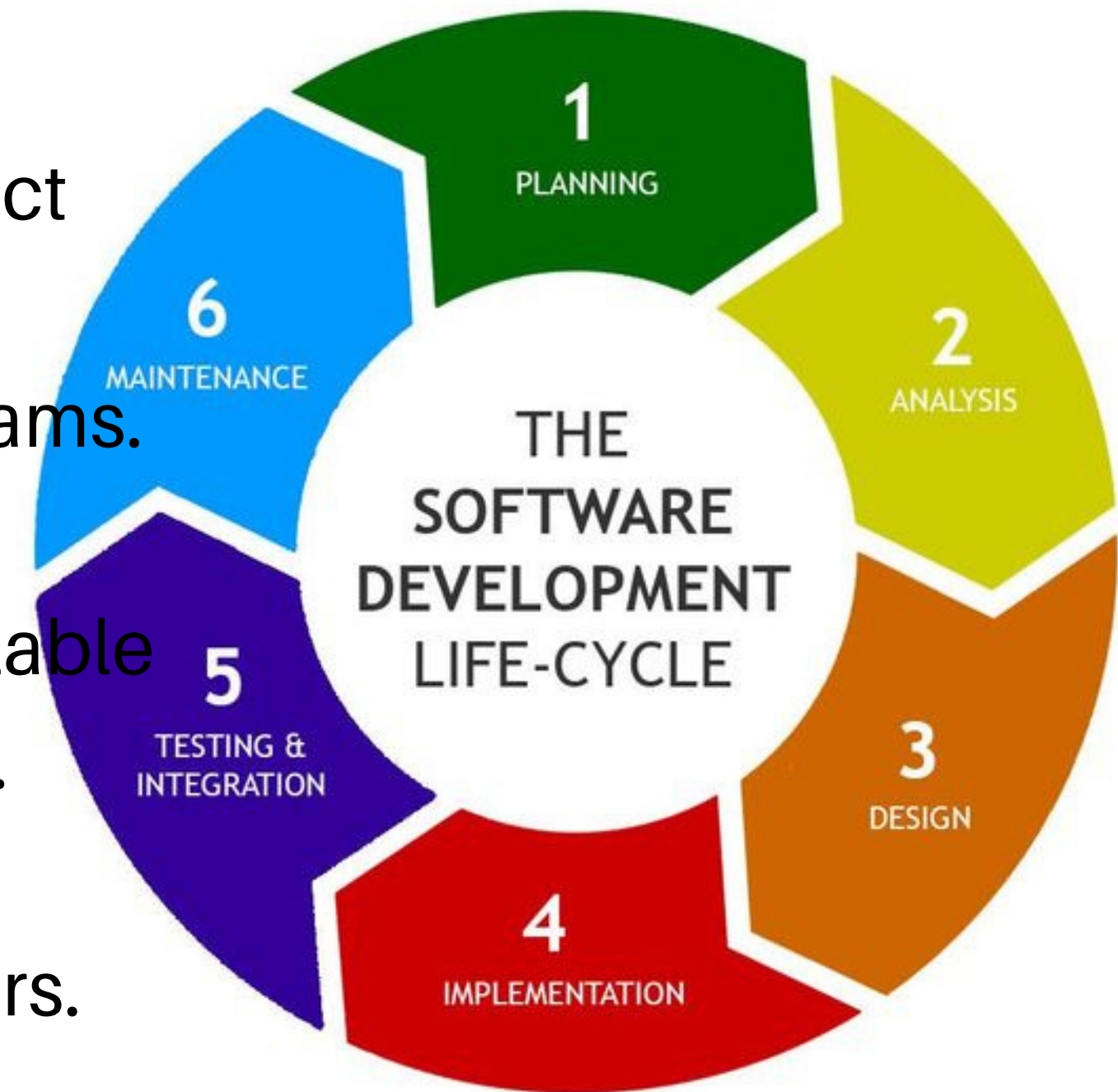
This is the actual writing of the program. A small project might be written by a single developer, while a large project might be broken up and worked by several teams.

5. Testing

It's critical to test an application before making it available to users. like manual, automated, like security testing.

6. Deployment

The phase where application is made available to users.

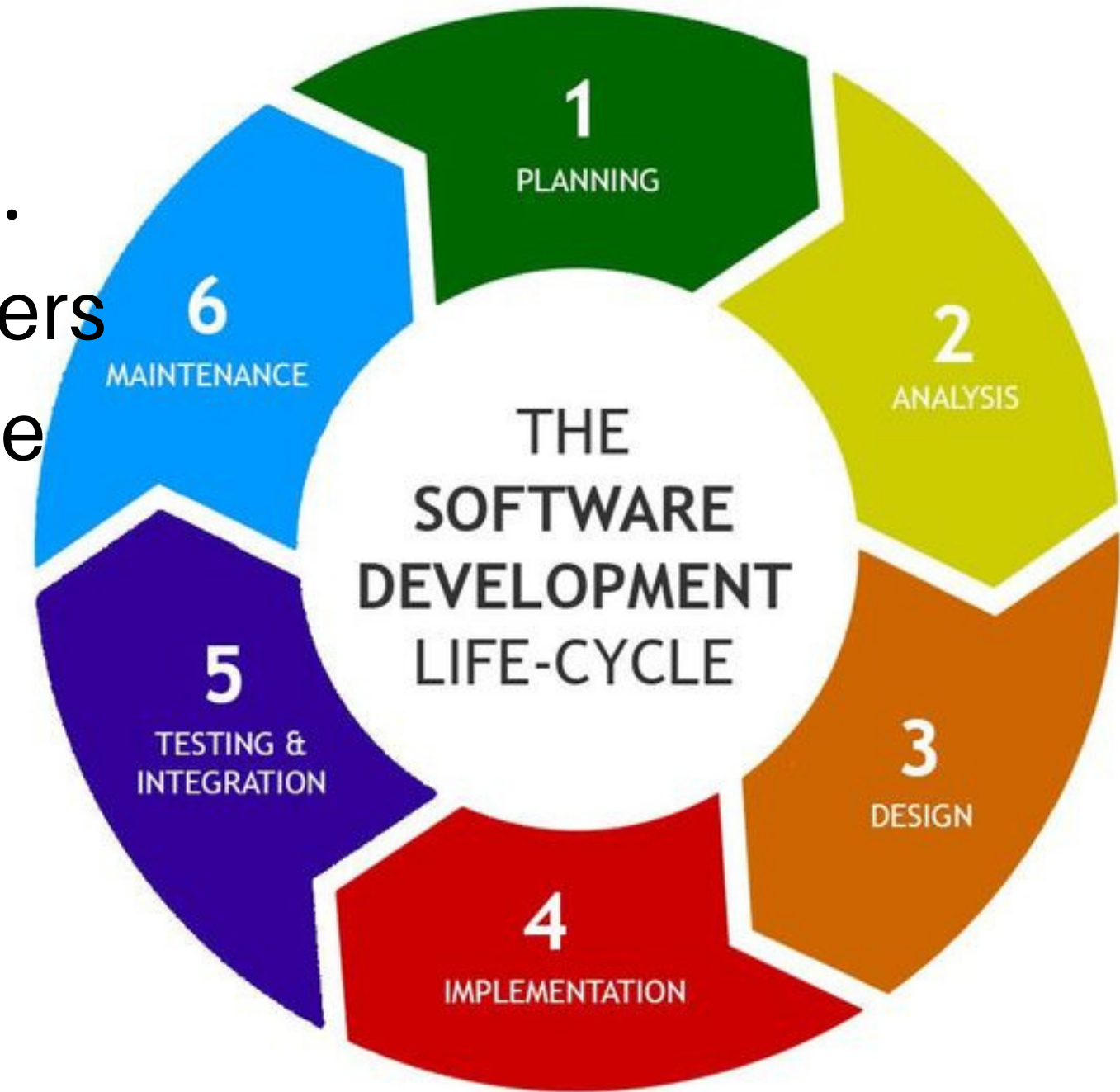




Software Development Life Cycle

7. Operations and Maintenance

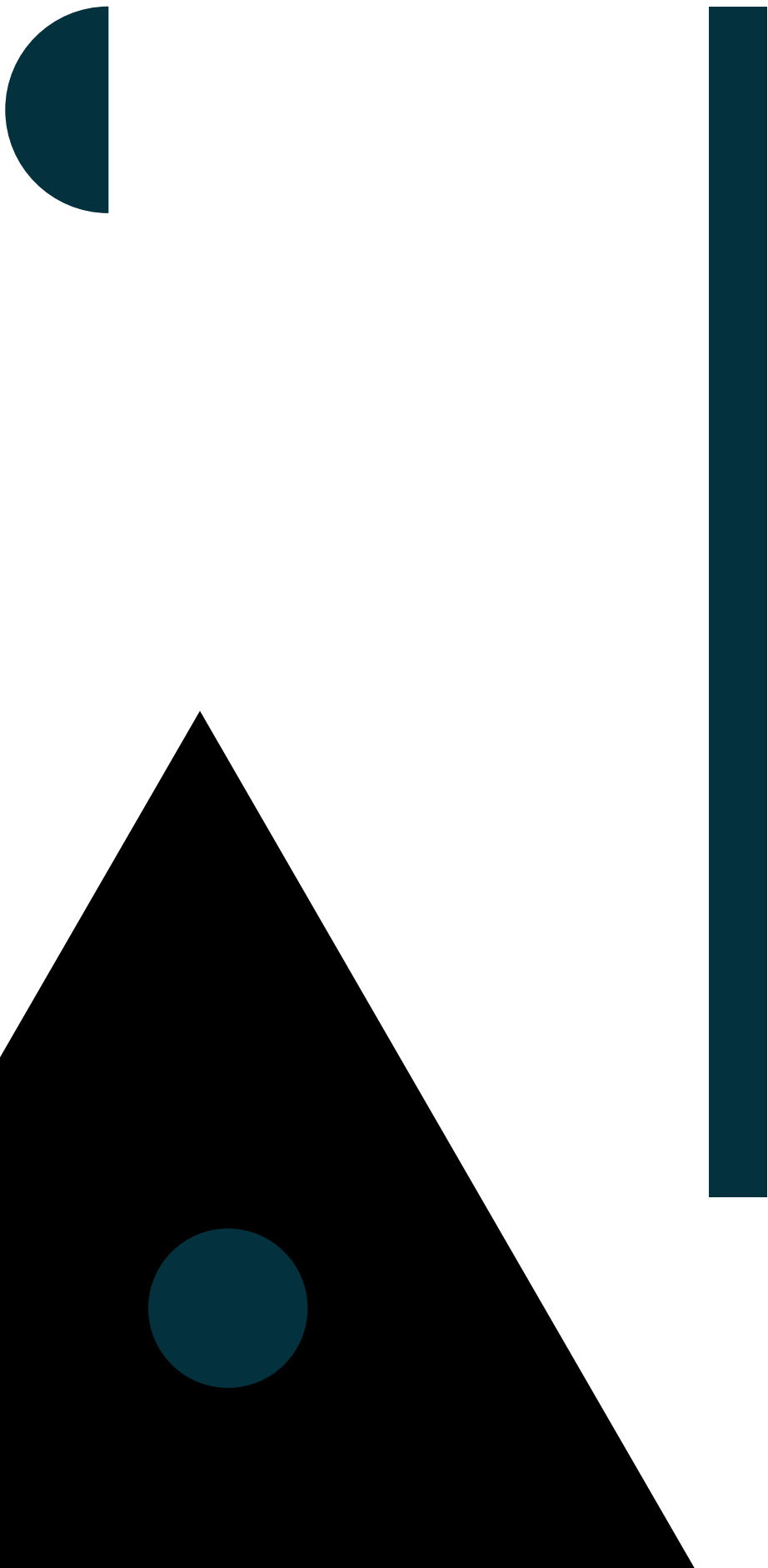
At this point, the development cycle is almost finished. The application is done and being used in the field. users discover bugs that weren't found during testing. These errors need to be resolved, which can spawn new development cycles.



SDLC Models & Methodologies



- *Waterfall*
- *Iterative*
- *Agile*
- *DevOps*



What is Agile

Agile is a term used to describe approaches to software development emphasizing incremental delivery, team collaboration, continual planning, and continual learning.

It's not a specific set of tools or practices, but rather a planning mindset that is always open to change and compromise.

Agile methods (often called frameworks) are comprehensive approaches to phases of the software development lifecycle.

Agile is not cowboy coding

Agile requires both a Definition of Done and explicit value delivered to customers in every sprint.

Agile is a development with specification

- ensuring specs are right-sized, and reflect appropriately how the team will sequence and deliver work.

Agile holds planning

continual planning throughout the project, not just planning up front. Continual planning ensures the team can learn from the work they're executing

Agile excuse using roadmap

Organizations and teams following an Agile approach absolutely know where they're going and the results they want to achieve

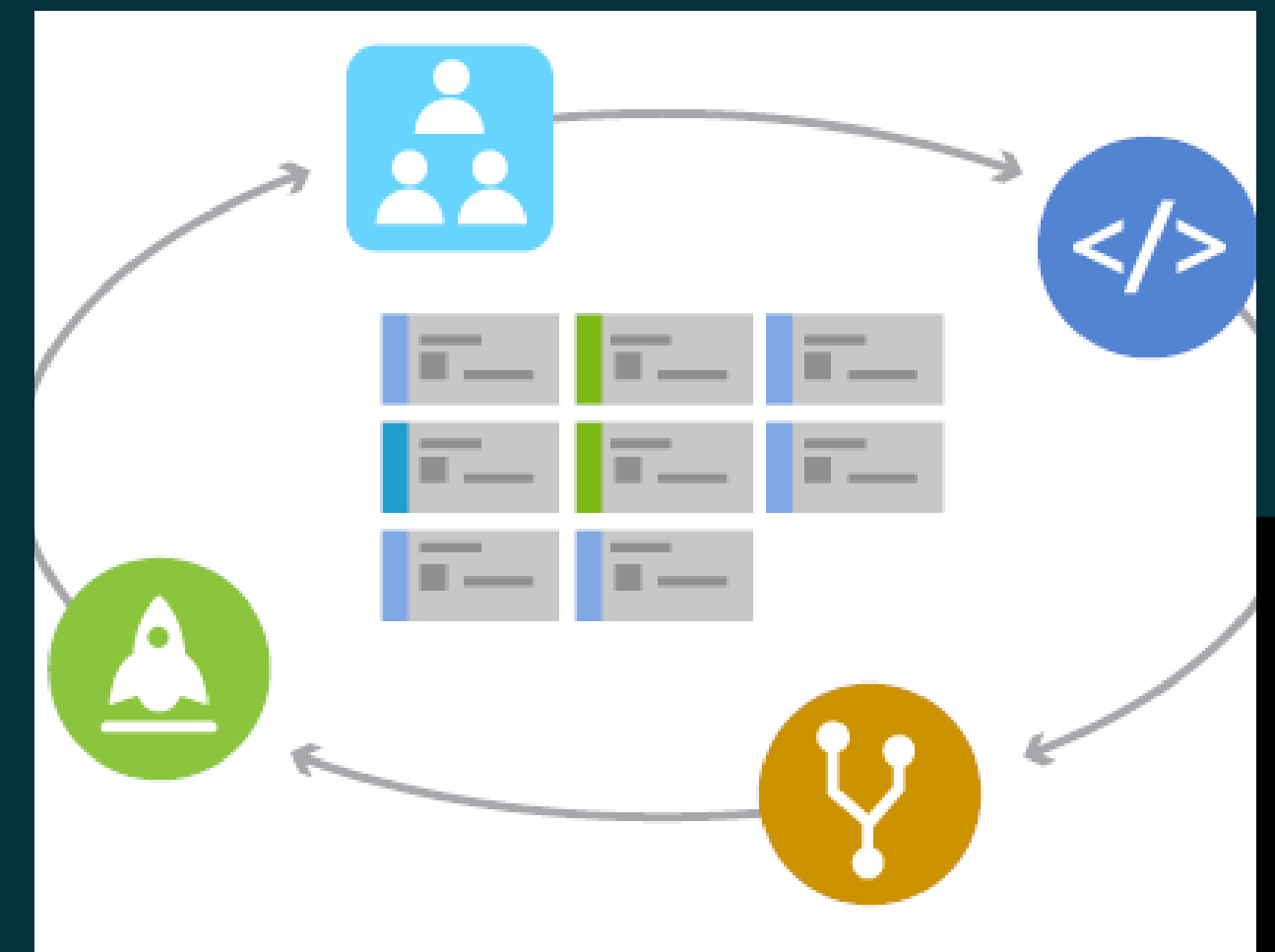
Agile methods and practices

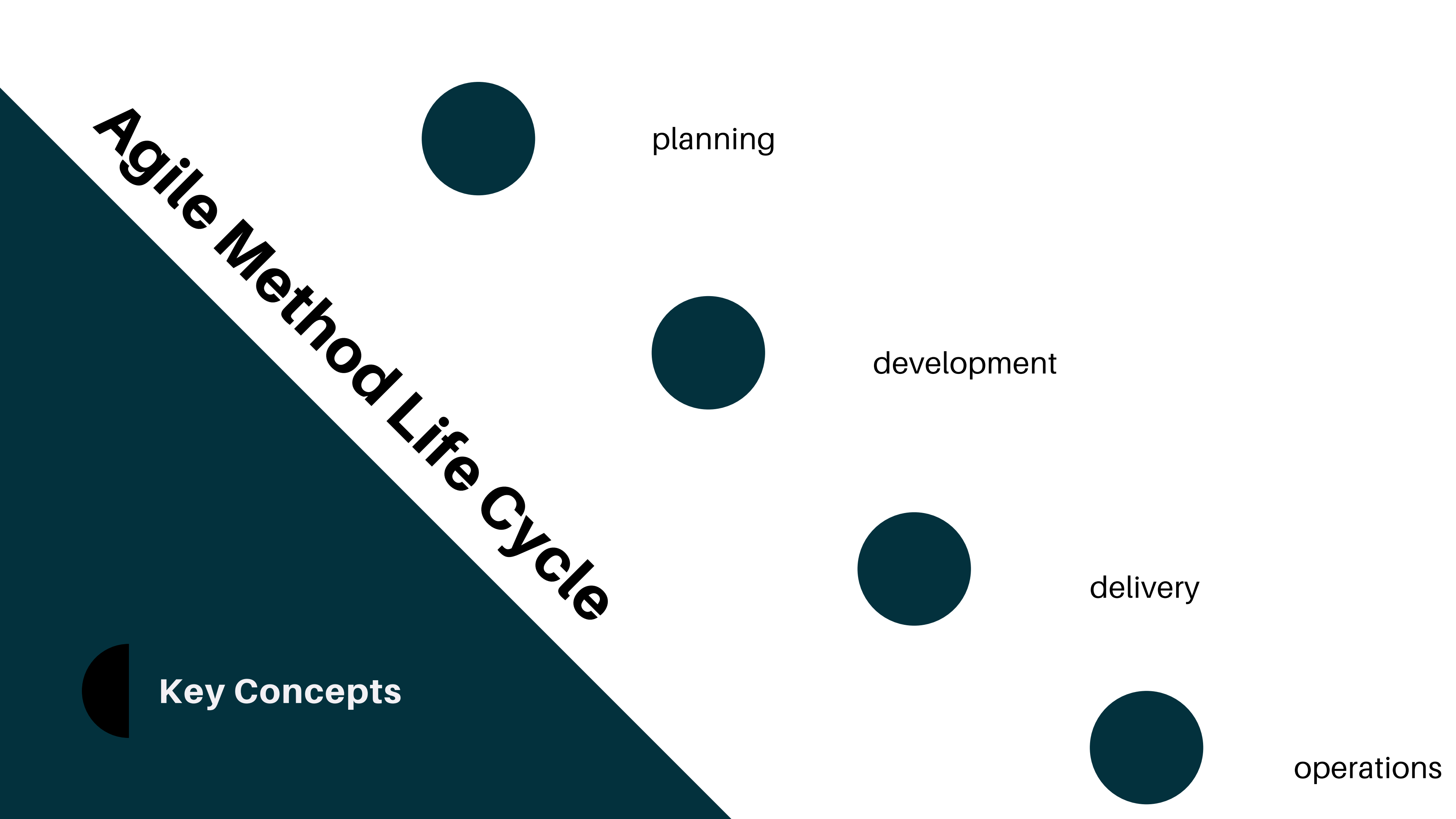
Agile is not a thing; you don't do Agile. Rather, Agile is a mindset that drives an approach to software development.

Agile is not a thing; you don't do Agile. Rather, Agile is a mindset that drives an approach to software development

As there is no single approach that works for all situations, the term Agile has come to represent a variety of methods and practices

A Legacy of Growth





Agile Method Life Cycle

Key Concepts

planning


development

delivery

operations



Building Agile teams



Agile method can greatly help with team work as it introduces more transparency into Development Operations

Any one can benefit from building productive, customer focused teams.

What is Agile Development

Agile development is a term used to describe iterative software development. Iterative software development shortens the Development and Operation lifecycle

Agile development is often contrasted with traditional or waterfall development, where larger projects are planned up front and executed against that plan

sprints is work in smaller increments, usually 1-4 weeks long.

Diligent backlog refinement

Integrate early and often

Minimize technical debt

**key success factors for
Agile development teams**

Diligent backlog refinement

The backlog is prioritized so the most important user stories The product owner owns the backlog and adds, changes, and reprioritizes user stories based on the customer's needs

One of the biggest drags on an Agile team's productivity is a poorly defined backlog. A team cannot be expected to consistently deliver high quality software each sprint unless they have clearly defined requirements.

When refining the backlog, key considerations to remember

- 1. Refining user stories is often a long-lead activity.**
- 2. A user story is not refined unless the team says it is**

Integrate early and often

Continuous integration and continuous delivery (CI/CD) sets your team up for the fast pace of Agile development. automate the build, test, and deployment pipelines. This should be one of the first things a team sets up when starting a new project.

When implementing a difficult-to-deploy feature, teams become aware immediately as the build and deployments fail. CI/CD forces a team to fix deployment

key CI/CD activities that are critically important to effective Agile development.

1. Unit testing
2. Build automation
3. Deploy to an environment

Minimize technical debt

Technical debt includes anything the team must do to deploy production quality code and keep it running in production

Examples are bugs, performance issues, operational issues, accessibility, and others.

There are many pressures to delay fixing bugs. It feels good to work on features and ignore debt. Unfortunately, somebody must pay the technical debt sooner or later

product owner works with their team to ensure there is time to pay off technical debt every sprint



Scrum

Scrum is not a methodology. Scrum replaces a programmed algorithmic approach with a heuristic one, with respect for people and self-organization to deal with unpredictability and solving complex problems

help with manufacturing, software development teams share many of the same goals, including wanting to increase their flow

Typical column names include To-do, Doing, and Done, but teams can customize this to suit the states in their workflow.

kanaba

