Cognixia®

Software Development Life Cycle (SDLC)

Building Quality Software

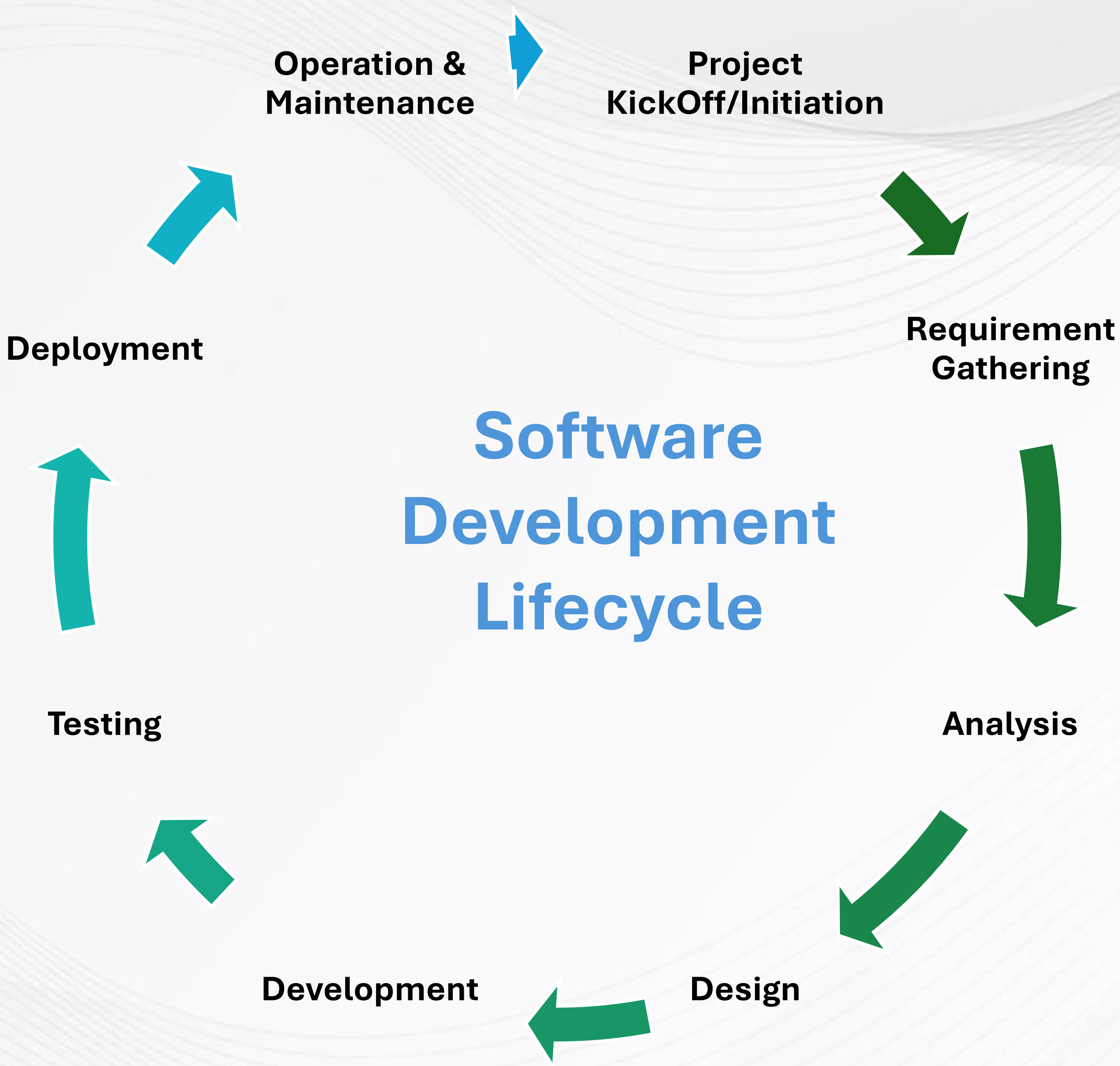
Introduction to SDLC

SDLC is a systematic approach to software development that ensures quality and correctness.

It helps organizations to:

- Create software efficiently and effectively
- Maintain control over the development process
- Deliver high-quality products that meet user expectations

Essential Phases of SDLC



Software Development Lifecycle

Project KickOff/Initiation

Requirement Gathering

Analysis

Design

Development

Testing

Deployment

Operation & Maintenance

Requirements Analysis

Foundation of any project:

- Meet with stakeholders
- Document features and scope
- Identify challenges and solutions
- Create vision of end product

Planning

Transforming requirements into action:

- Develop timelines
- Identify resources
- Create risk strategies
- Set milestones

Design

Creating the blueprint:

- Develop system architecture
- Design database structures
- Create UI mockups
- Plan for security and scalability

Implementation

Bringing design to life:

- Develop code following standards
- Create documentation
- Implement features by priority
- Perform code reviews

Testing

Ensuring quality:

Unit testing

System validation

Performance and security tests

User acceptance testing

Deployment

Moving to production:

- Prepare environment
- Plan deployment
- Train users
- Monitor performance

Maintenance

- **Keeping system running smoothly**
- Regular updates
- Bug fixes
- Performance optimization
- Security patches

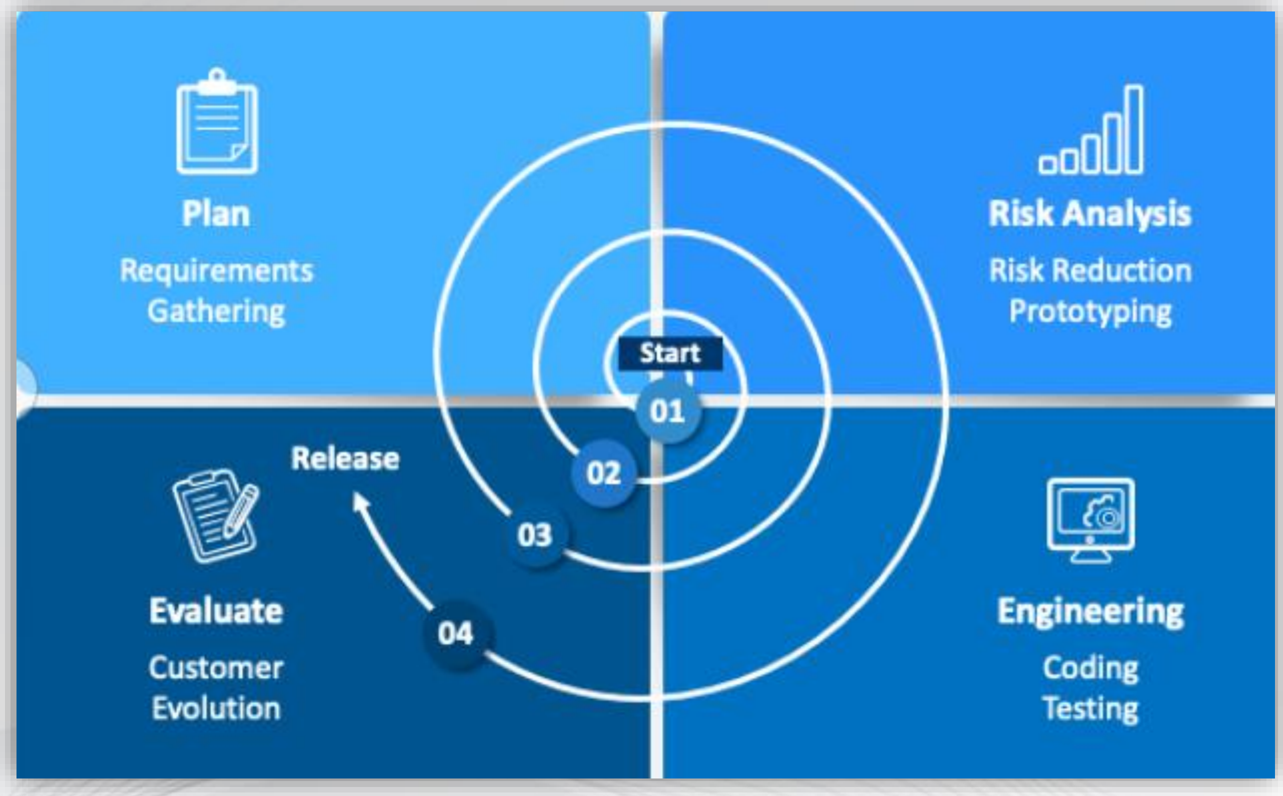
Common SDLC Models

Waterfall Model

Requirements
Design
Implementation
Verification
Maintenance

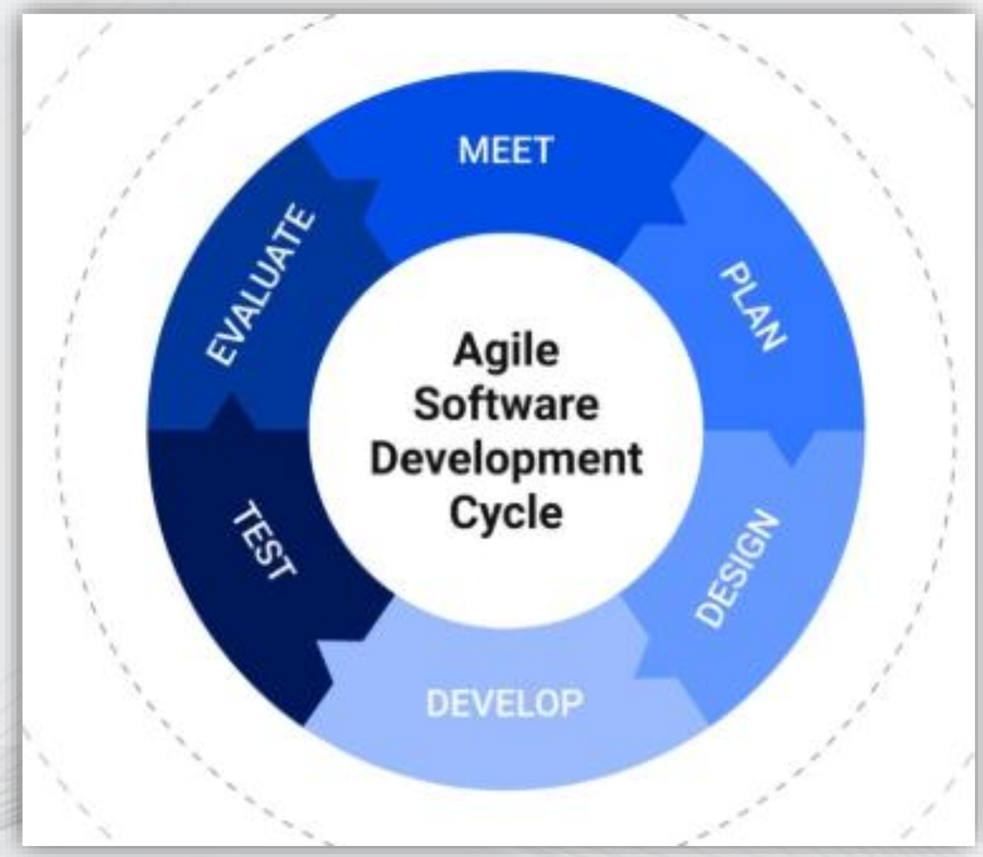
- Traditional sequential approach
- Step-by-step phases
- Each phase must be completed before next
- Clear documentation
- Best for well-defined projects

Spiral Model



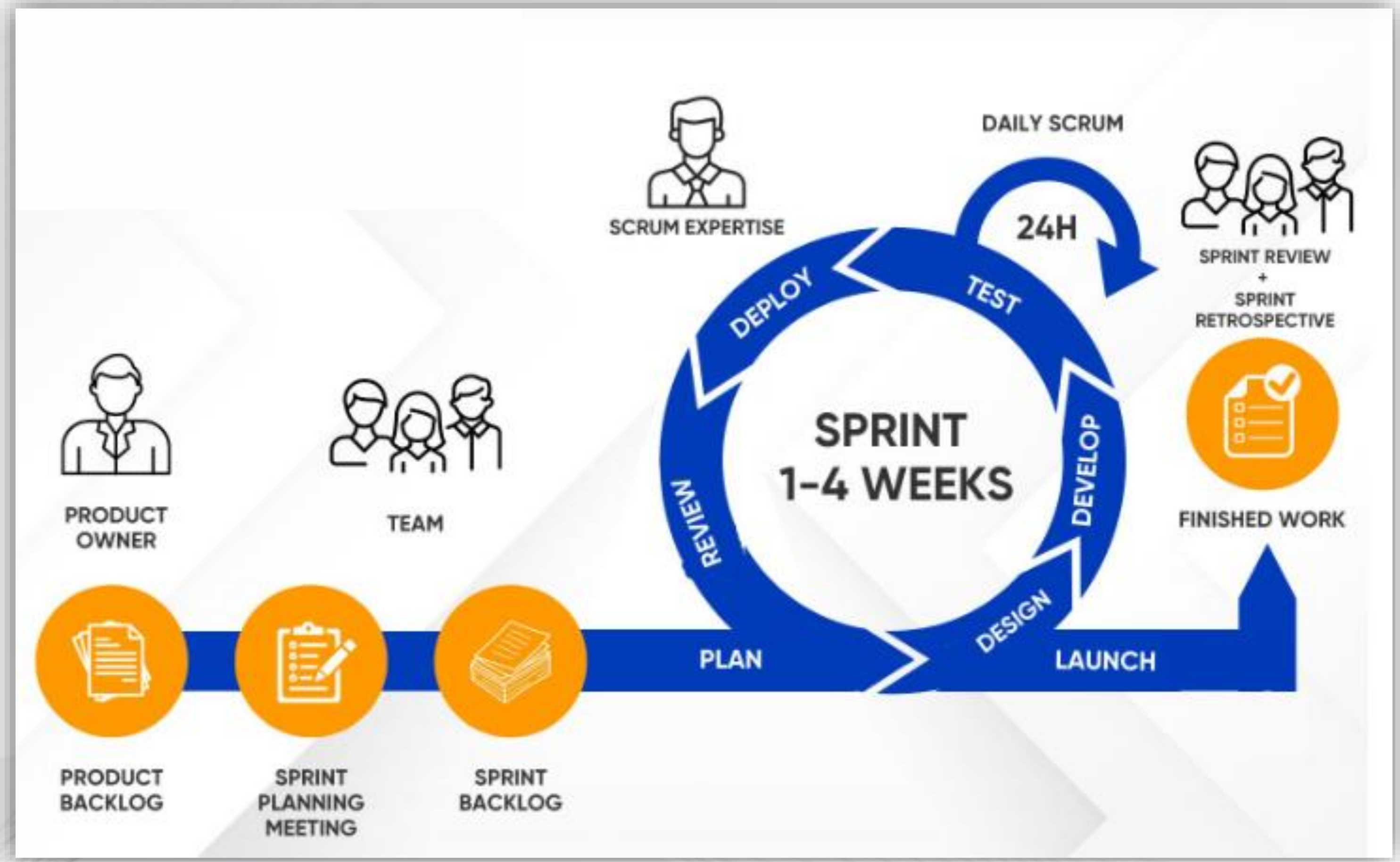
- Risk-driven development
- Combines planning and risk analysis
- Multiple development iterations
- Suitable for large, complex projects
- Regular prototyping

Agile Methodology



- Iterative and flexible:
- Breaks project into increments
- Regular feedback and adaptation
- Continuous software delivery
- Responds to change quickly

Scrum Framework

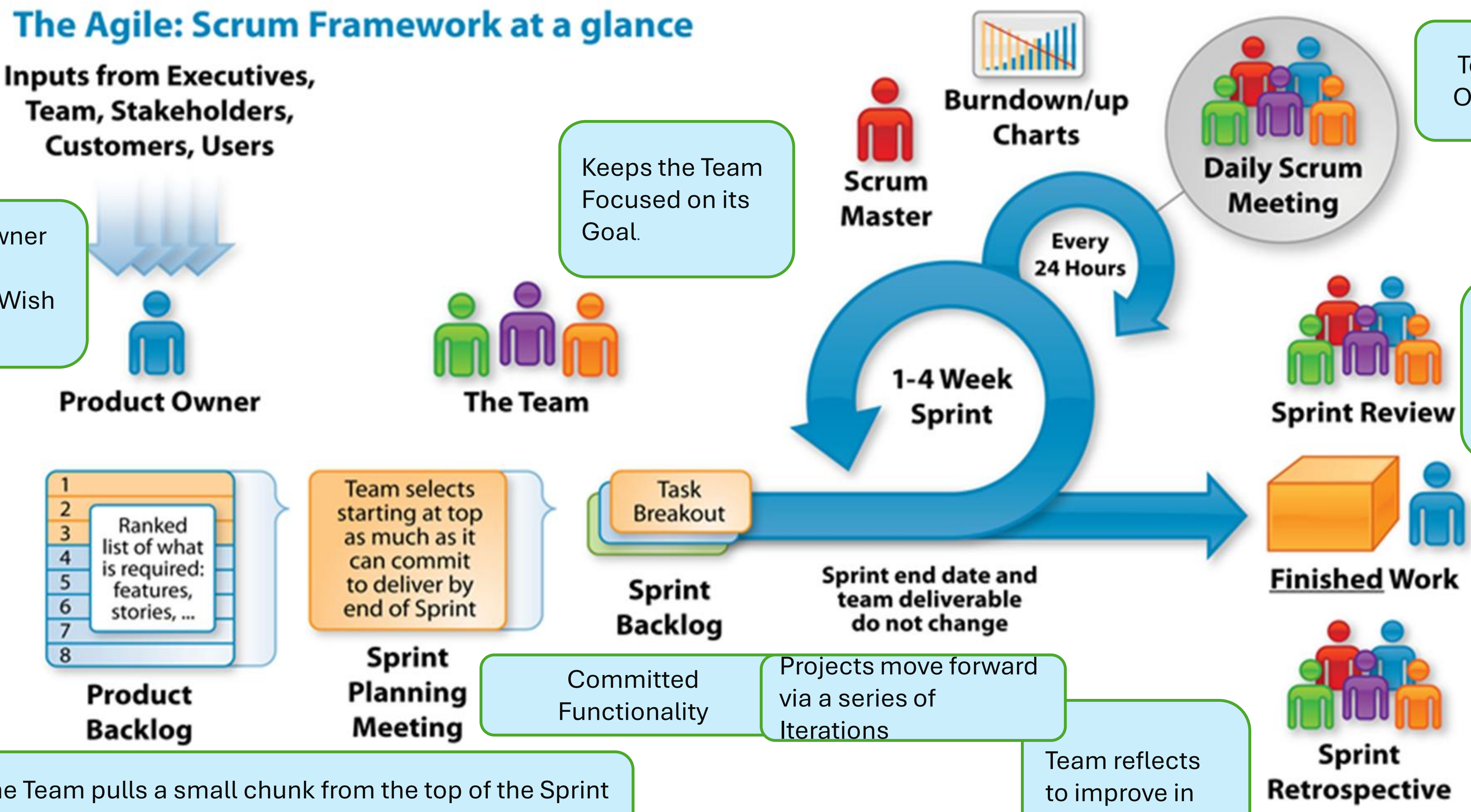


- Agile implementation:
- Short sprints (2-4 weeks)
- Daily stand-ups
- Sprint reviews and planning
- Team collaboration

The Agile Scrum Development Process

Iterative and Incremental Scrum Development Process

Amount of work remaining in a Sprint


The Agile: Scrum Framework at a glance
Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective

Team Assess Own Progress

Product Owner creates Prioritized Wish List

Keeps the Team Focused on its Goal.

Team demonstrates the new functionality

Shippable Functionality

Committed Functionality

Projects move forward via a series of Iterations

Team reflects to improve in the new Sprint.

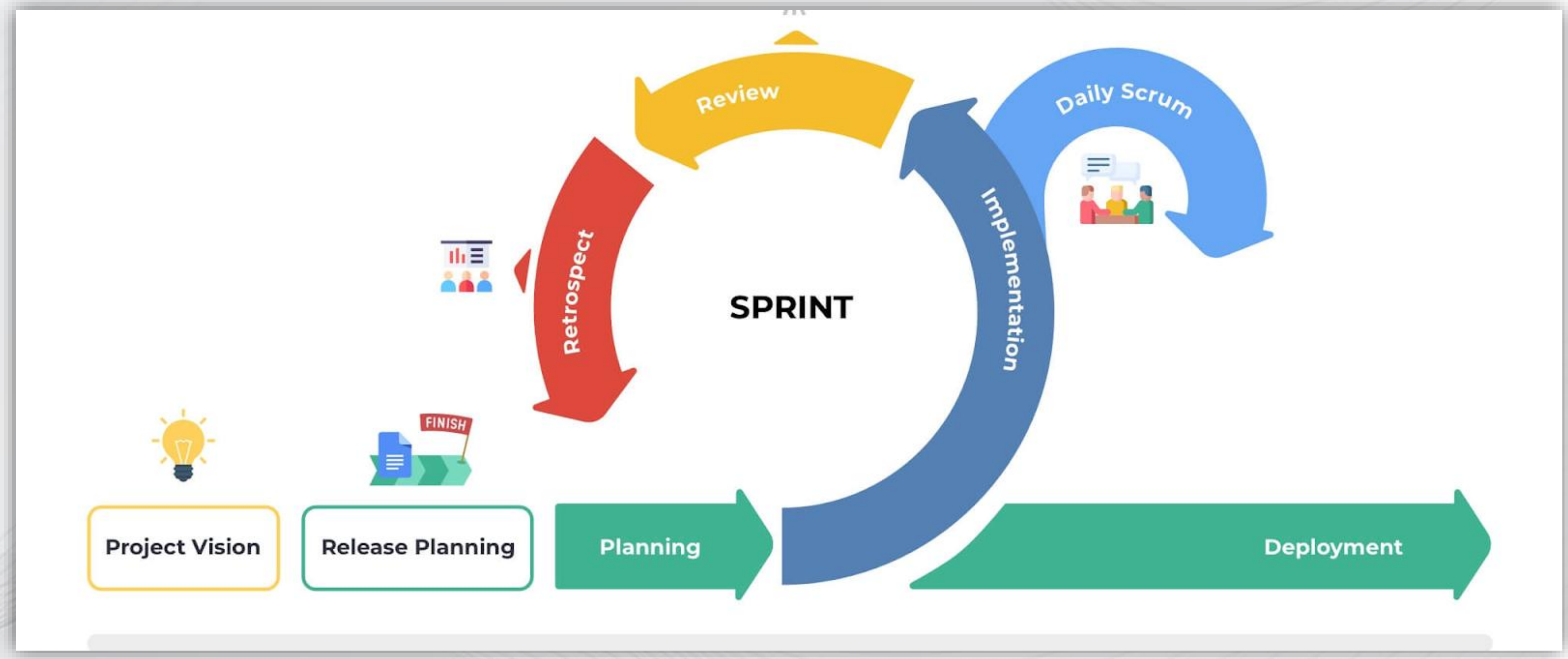
The Team pulls a small chunk from the top of the Sprint Backlog and decides how to implement those pieces.

The Scrum Methodology

Scrum Framework

- **Roles**
  - Product owner
  - Scrum Master
  - Team
- **Ceremonies**
  - Sprint planning
  - Sprint review
  - Sprint retrospective
  - Daily scrum meeting
- **Artifacts**
  - Product backlog
  - Sprint backlog
  - Burn down charts

Scrum Execution Model



Key Benefits of SDLC

Process Standardization

- □ Creating consistency:
- □ Clear development guidelines
- □ Standardized procedures
- □ Better team coordination
- □ Improved documentation

Improved Project Predictability

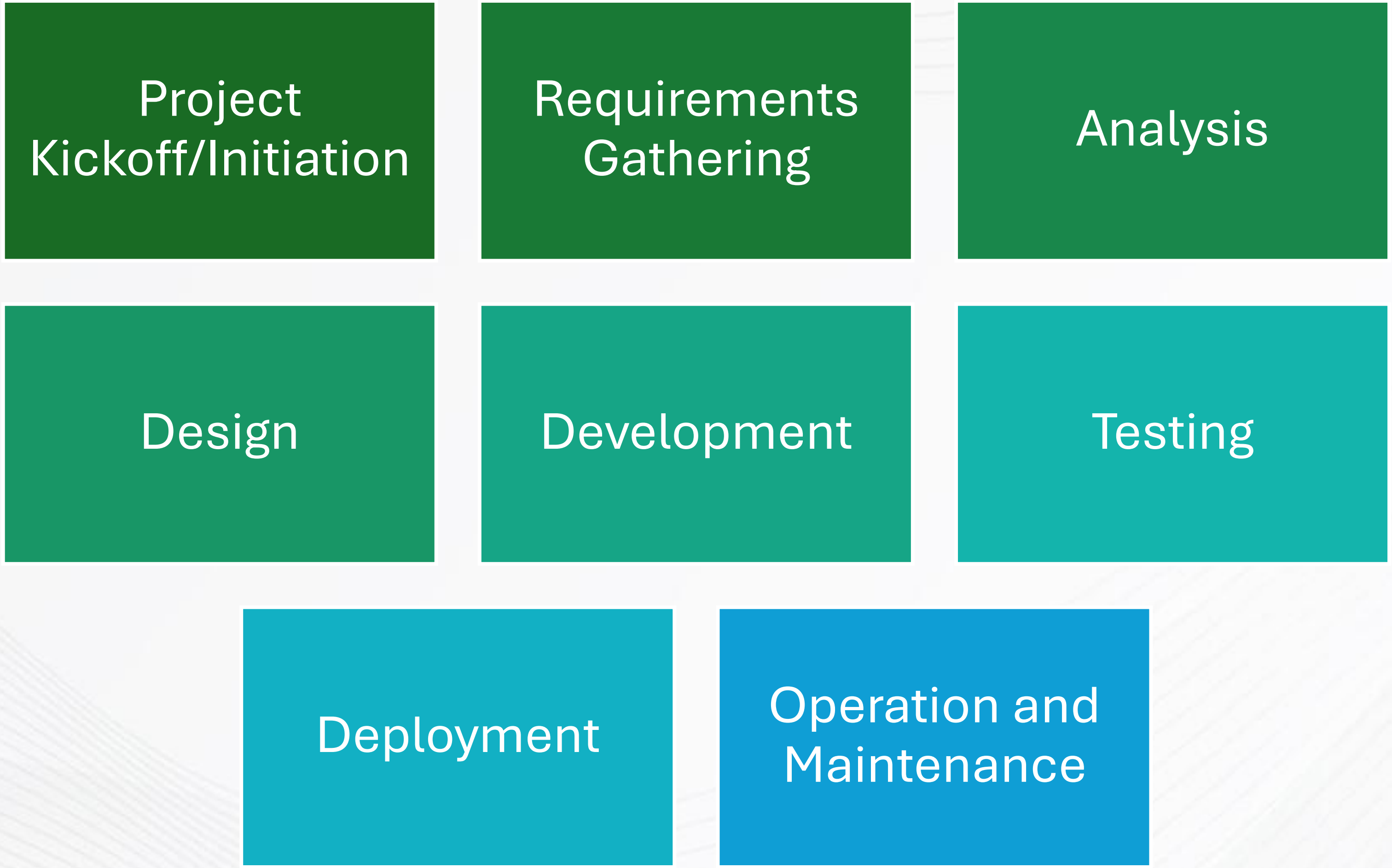
- □ Better control over outcomes:
- □ Accurate timelines
- □ Clear tracking
- □ Early risk detection
- □ Reliable delivery

Enhanced Quality Control

- □ Ensuring Excellence
- □ Regular quality checks
- □ Review processes
- □ Better product reliability
- □ Higher customer satisfaction

Software Development Life Cycle (SDLC) Variations

The phases of the Software Development Life Cycle (SDLC) can vary depending on the methodology and framework being used.Here are 8 phases:



Project Kickoff/Initiation

Requirements Gathering

Analysis

Design

Development

Testing

Deployment

Operation and Maintenance

Different Organizations and Methodologies

Different organizations and methodologies might combine or split these phases differently.

For example:

Some models combine Requirements Gathering and Analysis into one phase

Some separate Testing into multiple phases (Unit Testing, Integration Testing, System Testing)

Some combine Deployment with Operations and Maintenance

Some add additional phases like Planning or Documentation

Different Organizations and Methodologies

Note

The key is not the exact number of phases, but ensuring that all critical aspects of software development are properly addressed in a structured way.

The phases should be adapted to fit the specific needs of the project and organization while maintaining the core principles of systematic development and quality control

A Final Reflection

- SDLC provides a structured approach to software development
- Ensures quality and reliability
- Improves project management
- Enhances team collaboration
- Delivers better results

Cognixia®

Thank You !

www.cognixia.com