

Agile / Scrum / Kanban intro

Agile

Portnov Computer School

Topics

- ▶ **SDLC**
- ▶ **SDLC and SQA**
- ▶ **Project Management and Software Development methodologies**
- ▶ **Waterfall overview**
- ▶ **Agile overview**

SDLC

The background features a series of overlapping, semi-transparent geometric shapes in various shades of blue and teal. These shapes are primarily located on the right side of the image, creating a modern, layered effect. The colors range from light sky blue to deep forest green. The overall composition is clean and minimalist.

SDLC

- ▶ Planning
- ▶ Requirement Analysis
- ▶ Design
- ▶ Implementation
- ▶ Testing
- ▶ Deployment
- ▶ Maintenance

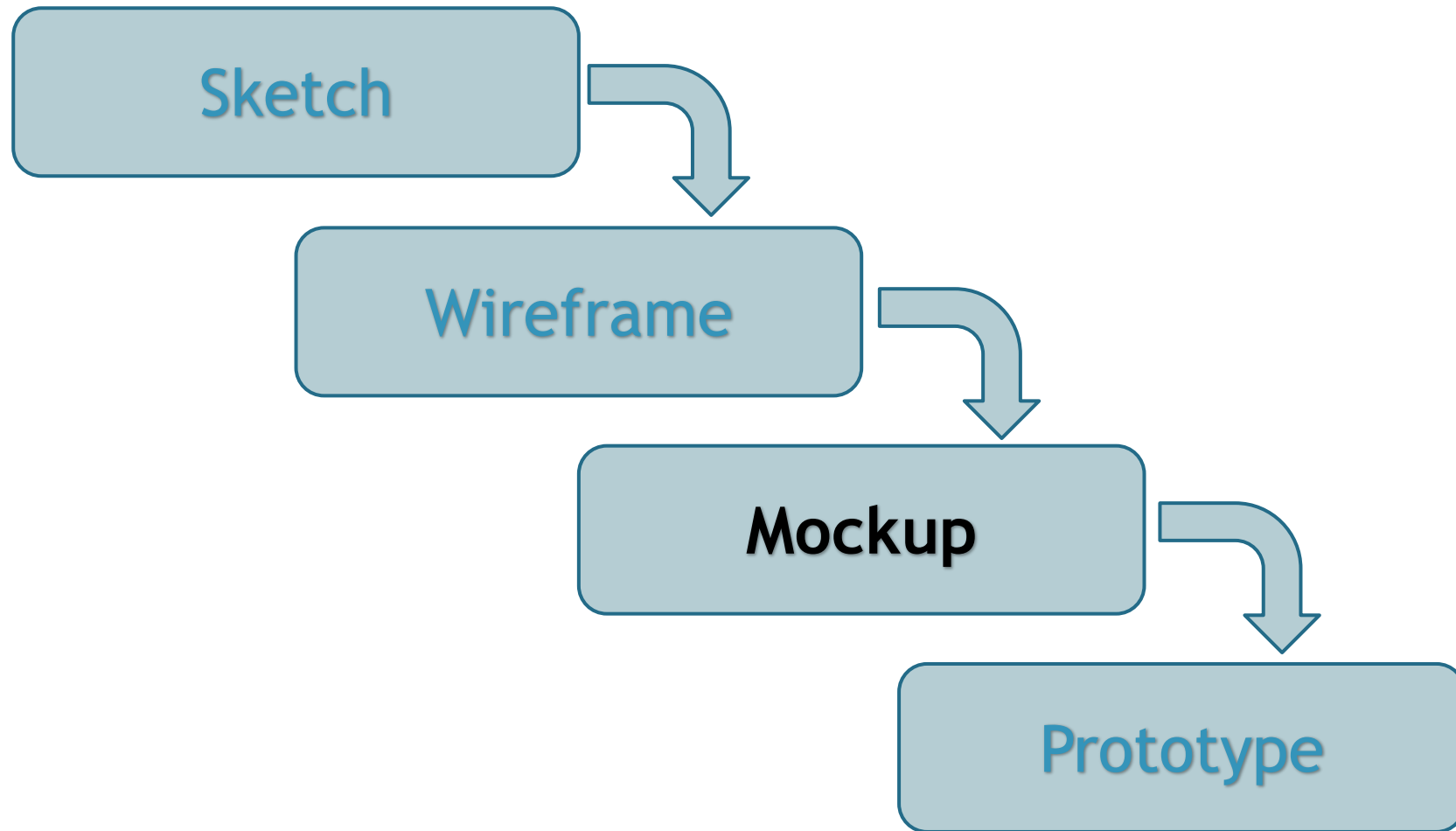
SDLC and SQA

- ▶ **To Build Better Products, Start Quality Early**
- ▶ **How early?**

QA: Planning and Requirement Analysis

- ▶ Product knowledge. Be an advocate and expert on user experience.
- ▶ Identify logical conflicts and discrepancies. Weigh in on new features.
- ▶ Identify poorly defined or unaddressed elements. Point out potential issues.
- ▶ Provide feedback and ask questions!
- ▶ Having a preview can help planning for future tests.

Application Design



QA: Application Design

- ▶ Can verify **Human Interface Guidelines**
- ▶ Can identify aspects of the design that might cause problems
- ▶ Will the application be likable? Boring? Annoying?
- ▶ Can reduce time in development

QA: Implementation and Testing

- ▶ Involved if issues were brought up during development
- ▶ Keeping track of all updates
- ▶ Test cases/test plans
- ▶ Bug reports
- ▶ Test reports
- ▶ Prioritizing tests

QA: Deployment and Maintenance

- ▶ On-call during releases.
- ▶ Post deployment testing.
- ▶ Hot-fixes testing.
- ▶ Upgrades testing.

Project Management and Software Development

The background features a series of overlapping, semi-transparent geometric shapes in various shades of blue and teal. These shapes are primarily located on the right side of the frame, creating a modern, layered effect. The colors range from light sky blue to deep forest green, with some shapes appearing as thin lines or faint washes.

Project Management and Software Development Methodologies

- ▶ Why do we need to learn about it?
- ▶ Benefits:
 - ▶ Make better business decisions
 - ▶ Improve internal communications
 - ▶ Save money
 - ▶ Iterate on your success

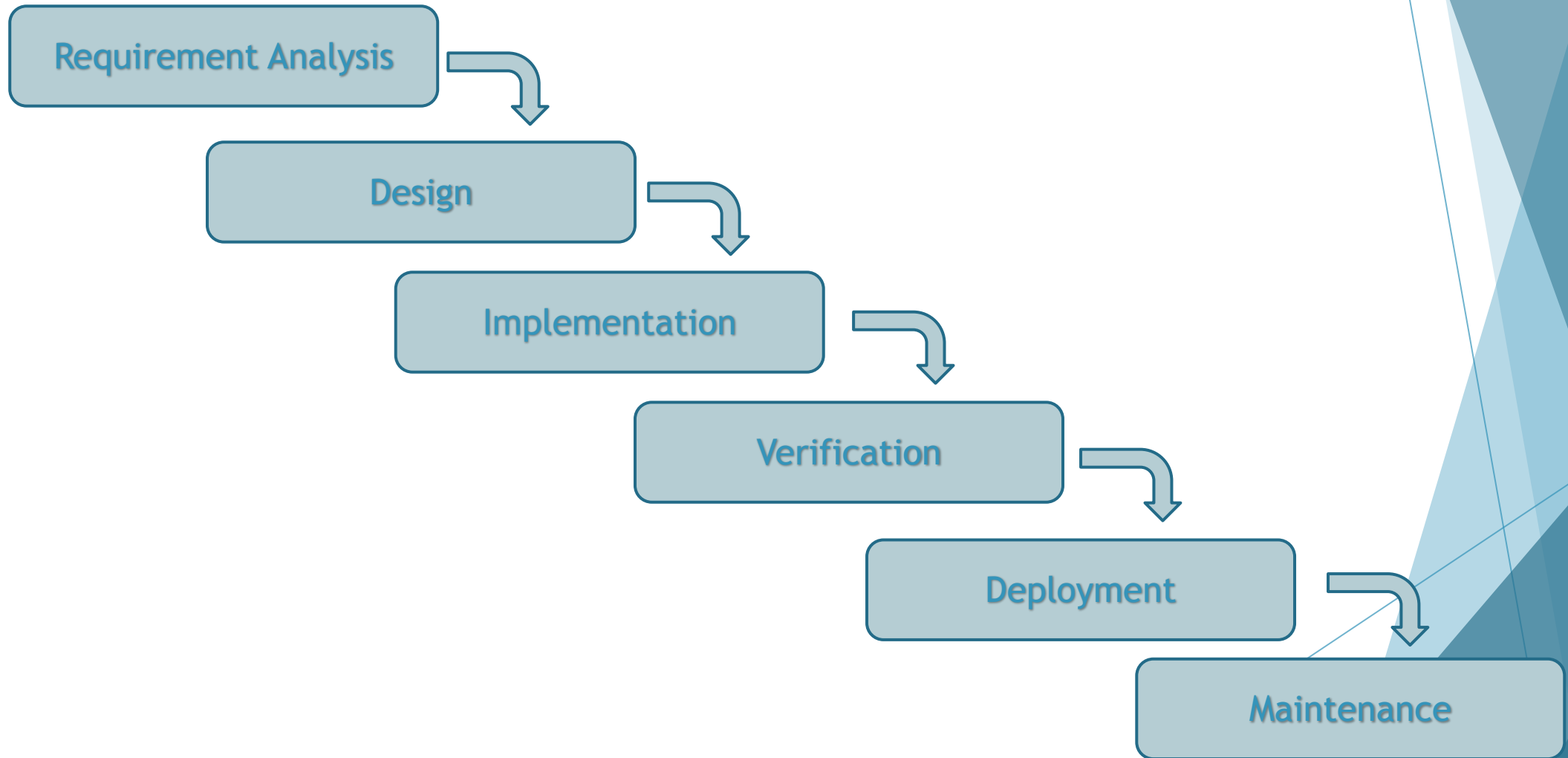
Project Management and Software Development Methodologies

- ▶ Waterfall development
- ▶ Agile development
- ▶ DevOps deployment
- ▶ Rapid application development
- ▶ etc...

Waterfall



Waterfall



Waterfall

- ▶ Waterfall methodology follows a sequential, linear process
- ▶ The Waterfall model originated in the manufacturing and construction industries, both highly structured environment where changes can be too expensive or sometimes impossible
- ▶ Each phase must be completed before the next phase can begin and there is no overlapping in the phases

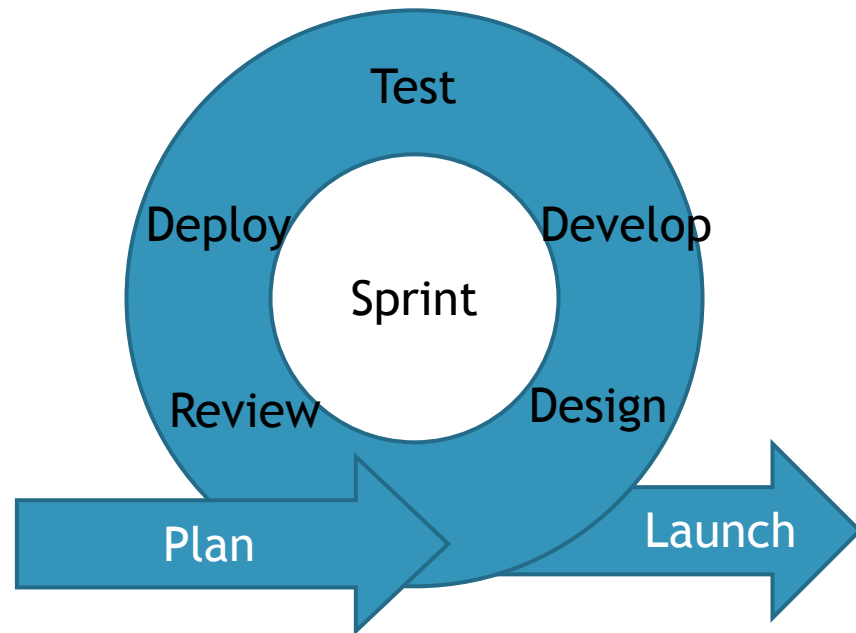
Advantages and Disadvantages

- ▶ Simple and easy to understand, use, and manage
- ▶ Phases are processed and completed one at a time
- ▶ Defined team roles
- ▶ Clearly defined stages and milestones
- ▶ Easy to arrange tasks
- ▶ Process and results are well documented
- ▶ No working software is produced until late during the life cycle
- ▶ Testing is done only at the latter phases of the project
- ▶ Doesn't allow to identify challenges at early stages of the project
- ▶ No allowance for uncertainty
- ▶ Cannot accommodate changing requirements
- ▶ Adjusting scope during the life cycle can end a project
- ▶ Integration is done at the very end, doesn't allow to identify challenges at early stages of the project

Agile

Manifesto for Agile Software Development

Agile



Agile

- ▶ Agile - collaborating to iteratively deliver whatever works
- ▶ Agile is an incremental and iterative approach

Values

- ▶ **Individuals and interactions** over processes and tools
- ▶ **Working software** over comprehensive documentation
- ▶ **Customer collaboration** over contract negotiation
- ▶ **Responding to change** over following a plan

12 Principles

- ▶ Customer satisfaction through early and continuous software delivery
- ▶ Accommodate changing requirements throughout the development process
- ▶ Frequent delivery of working software
- ▶ Collaboration between the business stakeholders and developers throughout the project
- ▶ Support, trust, and motivate the people involved
- ▶ Enable face-to-face interactions
- ▶ Working software is the primary measure of progress
- ▶ Agile processes to support a consistent development pace
- ▶ Attention to technical detail and design enhances agility
- ▶ Simplicity
- ▶ Self-organizing teams encourage great architectures, requirements, and designs
- ▶ Regular reflections on how to become more effective

Advantages and Disadvantages

- ▶ Continuous quality assurance
- ▶ Faster software development life cycle
- ▶ Customer-focused, increased customer satisfaction
- ▶ Flexible in accepting changes
- ▶ Empowers teams to manage projects
- ▶ Promotes efficient communications
- ▶ Ideal for projects with non-fixed funding
- ▶ Requires a high degree of customer involvement
- ▶ Assumes every team member is completely dedicated to the project
- ▶ Limited documentation
- ▶ Difficult planning at early stages
- ▶ Lack of long-term planning
- ▶ Agile recommends co-location for efficient communication

Agile vs Waterfall

- ▶ Incremental and iterative approach
 - ▶ Separates a project into sprints
 - ▶ Focus on customer satisfaction
 - ▶ Requirements can be revised, and changes can be made any time
 - ▶ Team manages the project
 - ▶ **Testing is performed concurrently**
- ▶ Linear and sequential approach
 - ▶ Divides a project into phases
 - ▶ Focus on successful project delivery
 - ▶ Requirements are prepared once at the start, no scope changes
 - ▶ Project Manager's role is essential
 - ▶ **Testing phase comes only after the build phase**

Agile? Scrum? Kanban?

- ▶ Agile tends to be used as an umbrella term used for flavors of Agile including Scrum, eXtreme Programming (XP), Kanban, and Scrumban
- ▶ Scrum and Kanban are subsets of Agile
- ▶ They are lightweight process frameworks for agile development