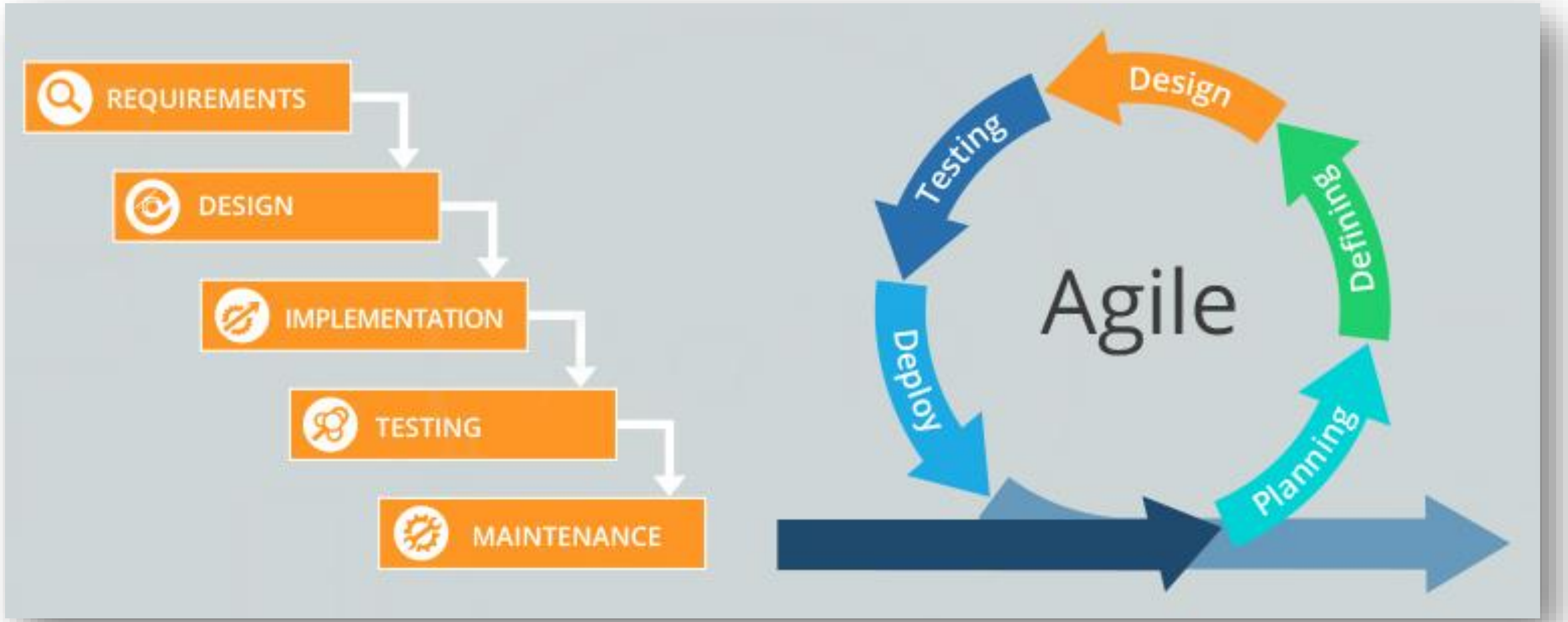
Beyond Scrum



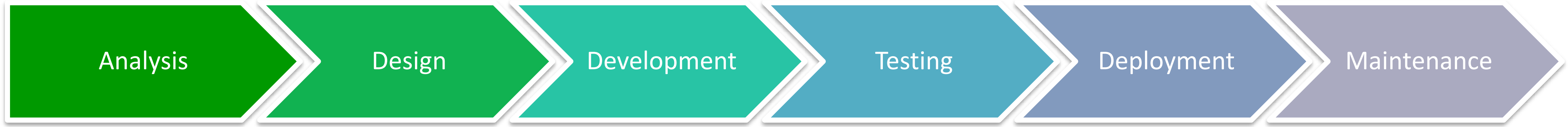
SWEN-261
Introduction to Software Engineering

Department of Software Engineering
Rochester Institute of Technology

Software Engineering
Rochester Institute of Technology

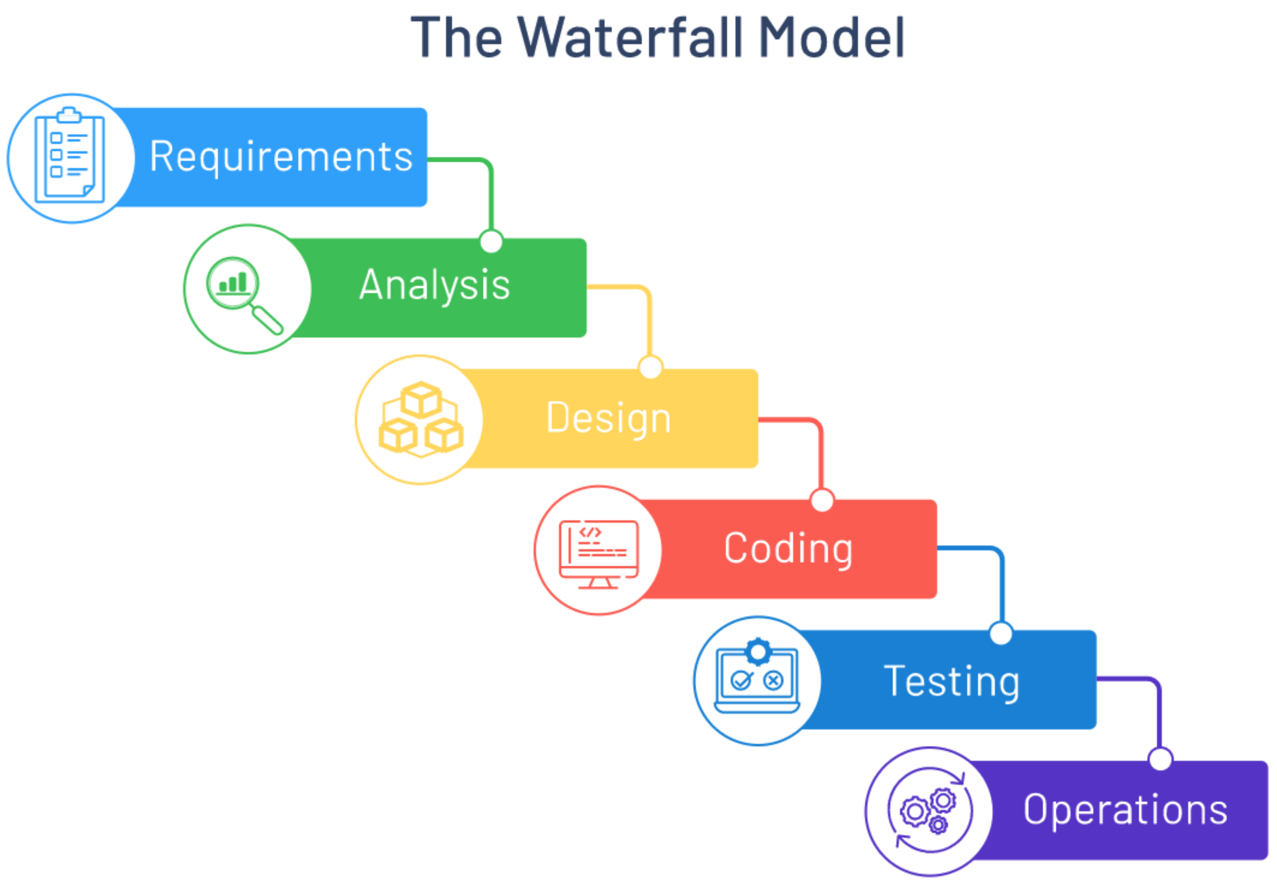
Picking a Development Methodology

- At the beginning of any software project, teams and organizations will have to first deal with the question of **Agile** vs. **Waterfall**.
- Software projects follow a methodology of clearly defined processes or software development life cycle (SDLC) to ensure the end product is of high quality.
- An SDLC identifies phases and the structured flow from one phase to another phase.
- Typically, there are six to seven phases. **Agile** and **Waterfall** are two popular, but very different, development processes.



Waterfall

- Waterfall project management is a traditional model for developing engineering systems and is originally based on manufacturing and construction industry projects.
- When applied to software development, specialized tasks completed in one phase need to be reviewed and verified before moving to the next phase. It is a linear and sequential approach, where phases flow downward (waterfalls) to the next.



Source: https://www.actitime.com/wp-content/uploads/2021/02/The-Waterfall-Model.svg

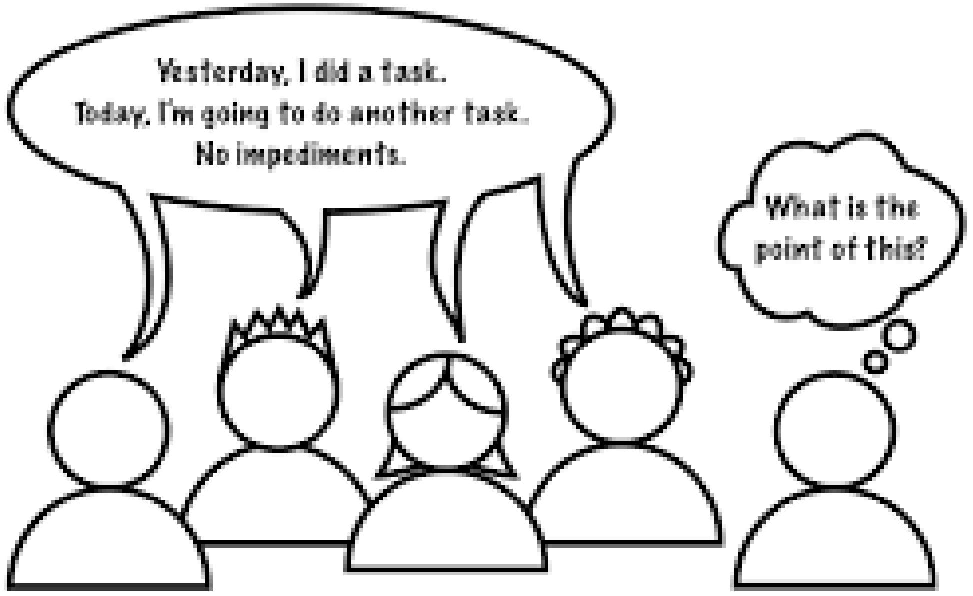
Waterfall Pros/Cons

- **Pros**
  - ***Easy to explain to users***
  - ***Structured approach***
  - ***Stages and activities are well-defined***
  - ***Helps to plan and schedule the project***
  - ***Verification at each stage ensures early detection of errors/misunderstanding***
  - ***Each phase has specific deliverables***
- **Cons**
  - ***Assumes that the requirements of a system will not change after initial agreement***
  - ***Very difficult to go back to any stage after it is finished***
  - ***Adjusting scope can be difficult and expensive***
  - ***Costly and requires more time, in addition to the detailed plan***

Source: https://dev.to/hansikaherath/sdlc-methods-and-their-advantages-disadvantages-15d5

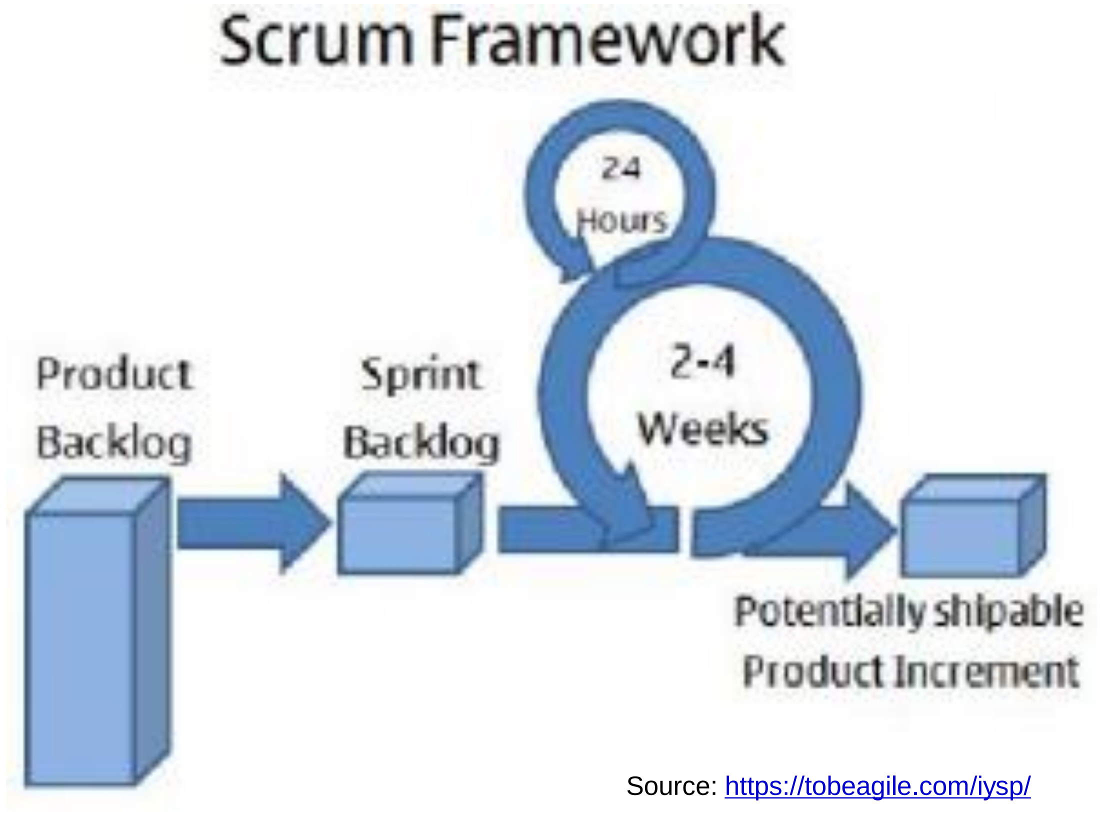
Agile Development Methods

- Agile development methods are incremental development methods that focus on rapid software development, frequent releases of the software, reducing process overheads by minimizing documentation and producing high-quality code.
- Agile development practices include
  - *User stories for system specification*
  - *Frequent releases of the software*
  - *Continuous software improvement*
  - *Test-first development*
  - *Customer participation in the development team*
- We followed a **Scrum** process for Agile
  - *What did you like or dislike about Scrum?*


Yesterday, I did a task.
Today, I'm going to do another task.
No impediments.
What is the point of this?

Agile - Scrum Review

- Scrum is an agile method that provides a project management framework.
  - ***It is centred round a set of sprints, which are fixed time periods when a system increment is developed.***



Source: https://tobeagile.com/iysp/

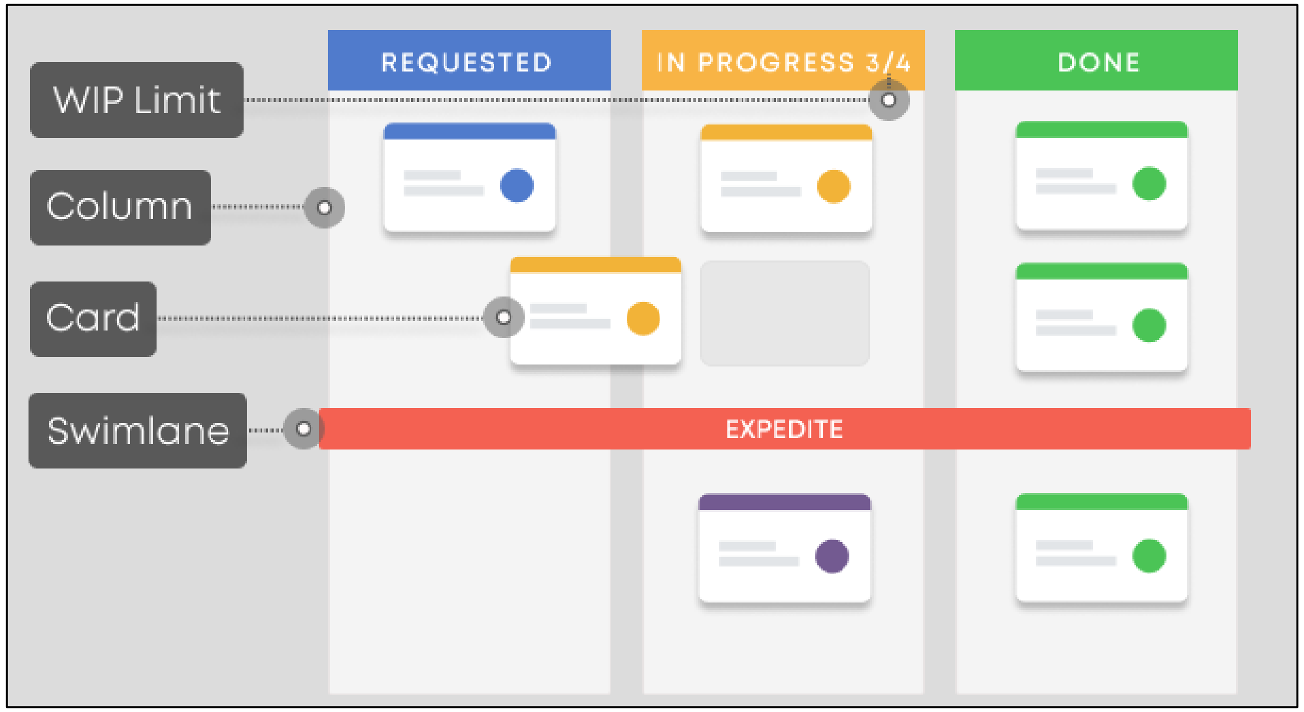
- **Pros**
  - *Product owner sets priorities*
  - *Team owns decision making*
  - *Documentation is lightweight*
  - *Supports frequent updating*
- **Cons**
  - *Difficult to control the cost of changes*
  - *May not be suitable for large teams*

Kanban

- Kanban is a prevailing system used to execute DevOps and Agile software development.
  - ***Kanban boards display work items to let the team members see the work they need to do whenever they want.***
- Kanban is a system that provides visual means of managing work when it moves from one level to another in a given process.
  - ***It visualizes the workflow and the actual work passing through.***
- The one thing that distinguishes Kanban from Scrum is that it is <u>not</u> iterative.
  - ***In contrast, Scrum processes have short iterations, which clones a project lifecycle on a smaller scale with a clear beginning and ending for each iteration.***
  - ***Kanban is an agile methodology that allows software development in one considerable cycle time.***

Kanban Details

- Visualizing workflow using a Kanban board.
  - ***WIP (Work in Progress) Limits the amount of work in progress at any given time.***
  - ***Swimlanes are horizontal lanes on your board that help to separate and further define your workflow.***
- Focuses on continuous improvement by creating feedback loops where changes are introduced.
- Make process changes collaboratively and involve all stakeholders as needed.



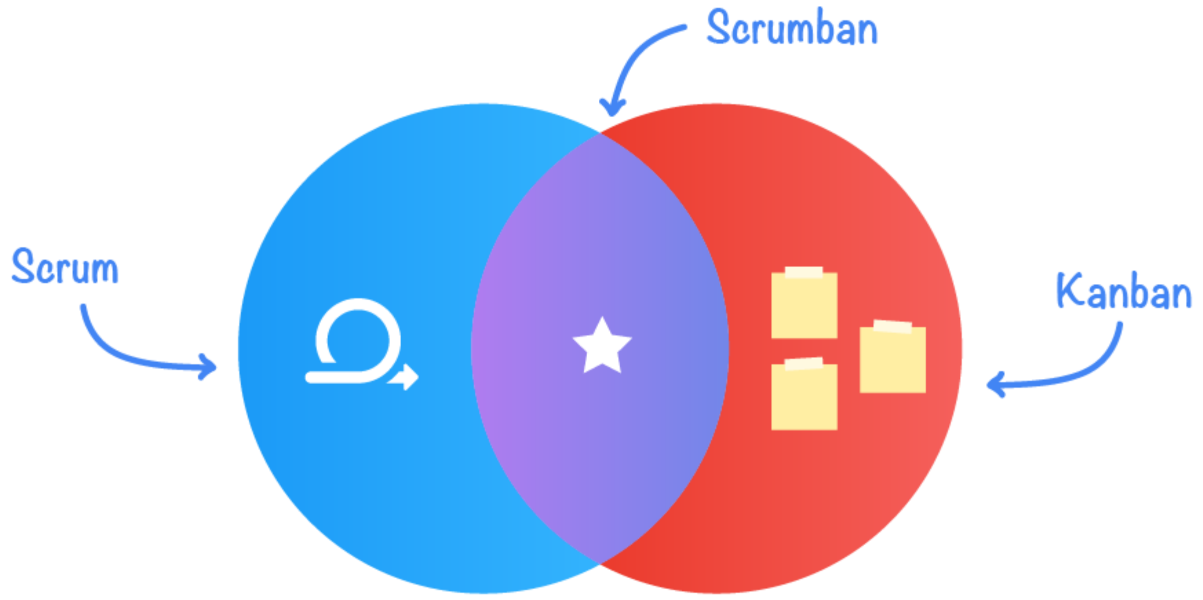
Source: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban

Kanban Pros/Cons

- **Pros**
  - ***Lower budget and time requirements***
  - ***Allows early product delivery***
  - ***Less process heavy, more flexible process***
  - ***Continuous process improvement***
- **Cons**
  - ***Harder to plan out and project large release initiatives***
  - ***Team collaboration skills determine success***
  - ***Poor business analysis can doom the project***
  - ***Flexibility can cause developers to lose focus***

Scrumban = Scrum + Kanban

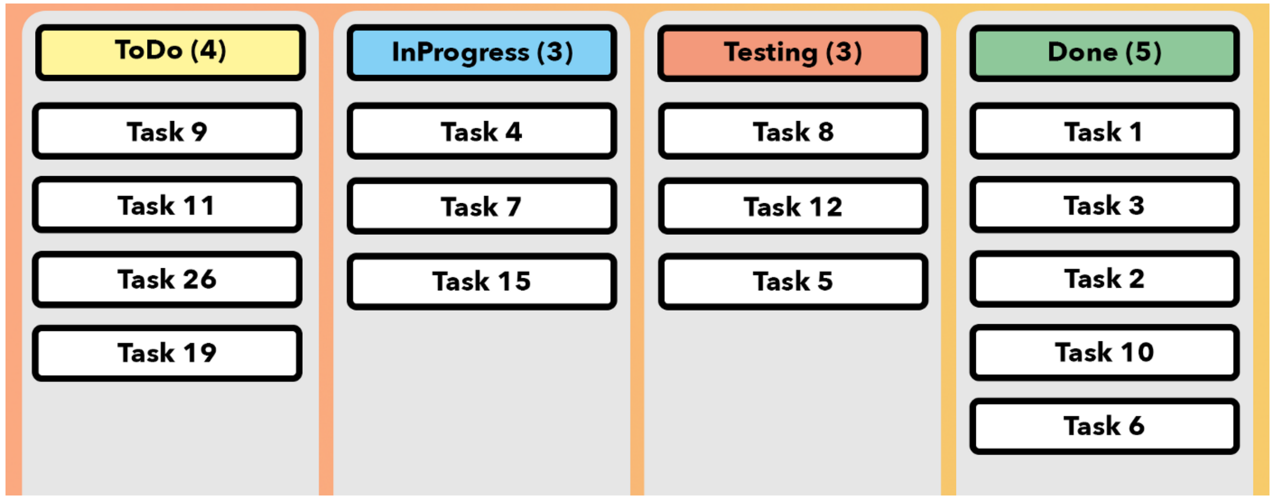
- **Scrumban** is an Agile project management methodology that is a hybrid of Scrum and Kanban
  - ***It combines the structure of Scrum with the flow-based methods and visualization of Kanban***



- Scrumban was developed to make it easier for existing Scrum teams to transition to Kanban and explore lean methodologies

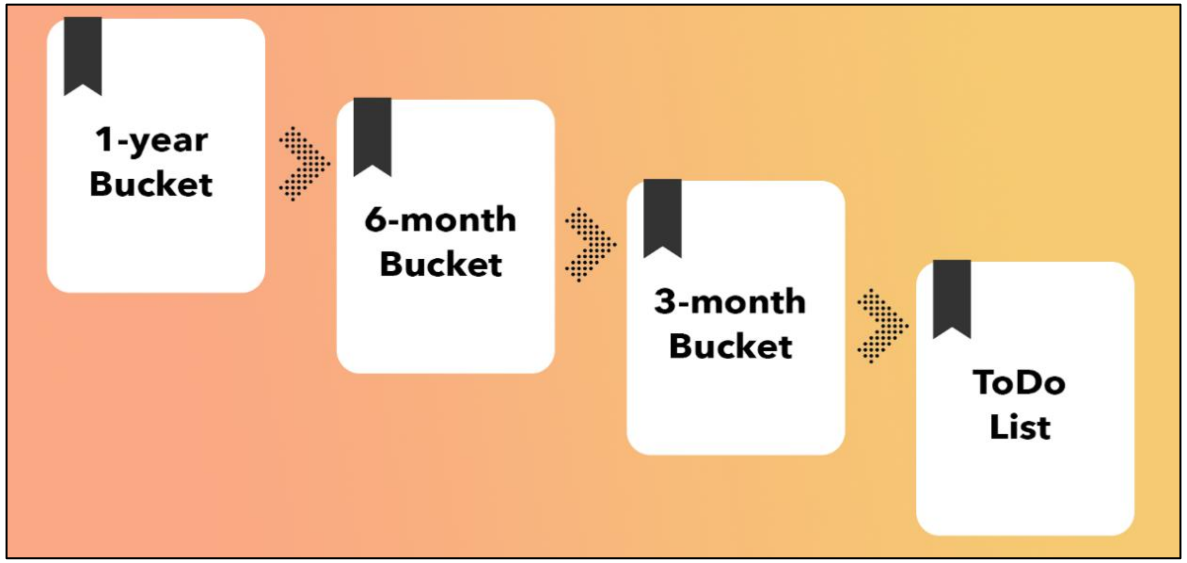
Scrumban Details

- The goal is to deliver working results after the completion of the cycle.
- Iterations look like sprints in the Scrum methodology, with some major differences:
  - ***Two weeks is the maximum limit of a Scrumban iteration***
  - ***Tasks are not assigned to the team members but rather choose the task themselves***
  - ***To prevent overburdening the team, a WIP limit is set in the to-do column.***
- Scrumban builds on Kanban's flexibility in the planning process by introducing the concept of need-based or on-demand planning
- **Bucket size planning** brings the possibility of long-term planning to Scrumban.
  - ***These planning sessions are integrated with the condition of the work-in-progress column.***



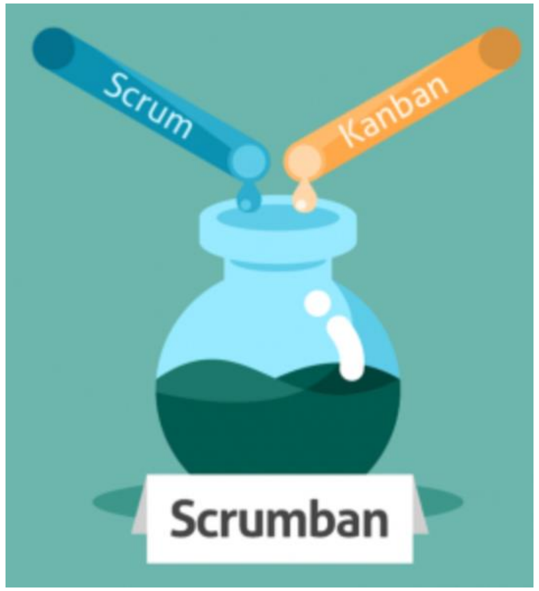
Scrumban - Bucket size planning

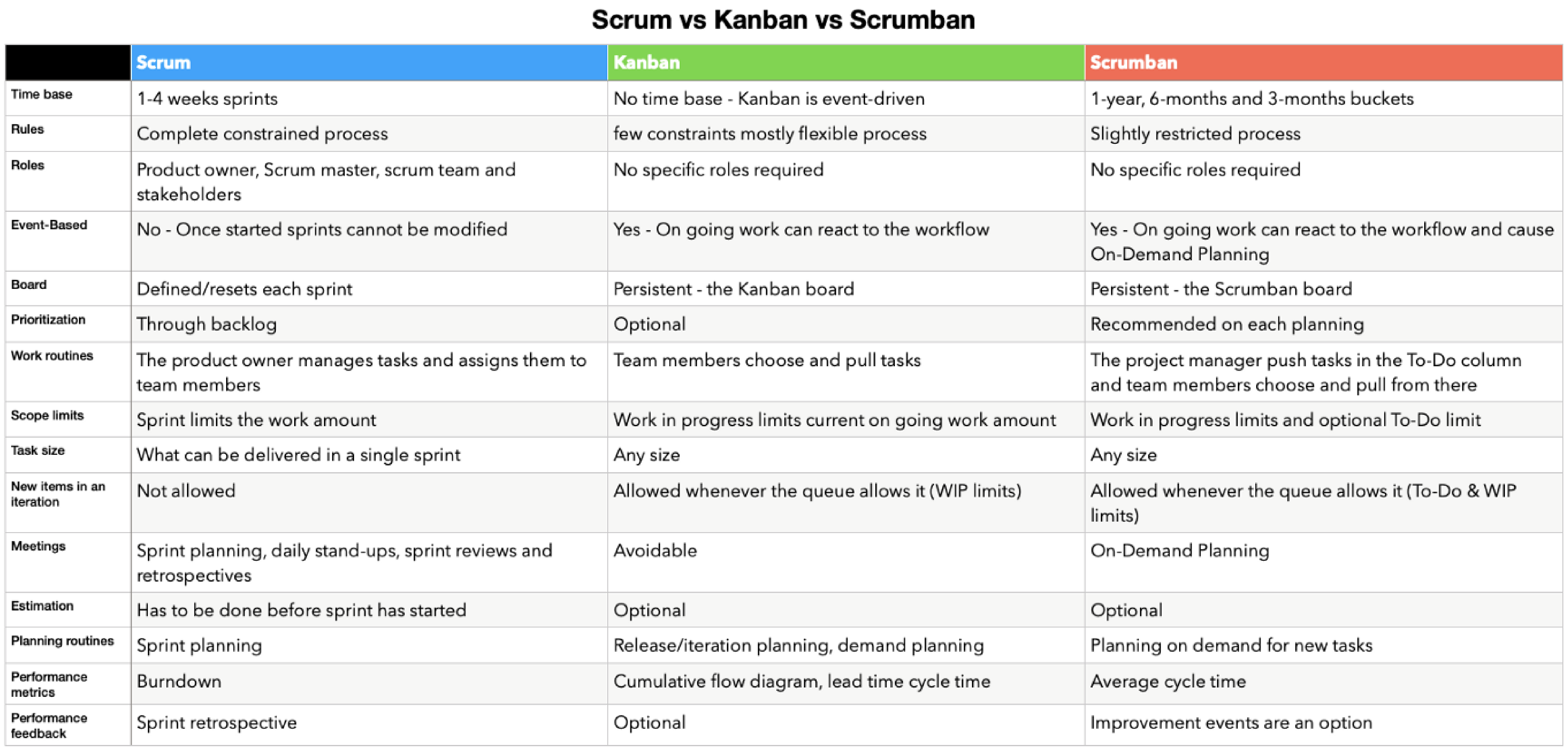
- It is based on the system of three buckets that the work items need to go through before making it on the Scrumban board.
  - *The three buckets represent three different stages of the plan and are usually called 1-year, 6-month and 3-month buckets.*
  - *1-year bucket is dedicated for long-term goals.*
  - *6-month bucket is where the main requirements of this plan are crystallized.*
  - *When a company is ready to start implementing the plan, the requirements are moved into the 3-month bucket and divided into tasks to be completed by the project team.*
  - *It is from this bucket that the team draws tasks during their on-demand planning meeting and starts working on the tasks.*



Scrumban Pros/Cons

- **Pros**
  - ***Saves time by planning when there's a demand***
  - ***Ideal for larger projects as tasks can be prioritized over time***
  - ***Reduced bottlenecks as visualized workflow identifies them fast***
  - ***Transparency allows everyone to see what's going on***
  - ***Easy to adopt for teams who don't have to learn new techniques***
  - ***Less stress for teams as everyone as equal footing in the project***
- **Cons**
  - ***No best practices as the method is still new***
  - ***Difficult to track because of self-directed teams***
  - ***Less control for project managers than traditional PM***





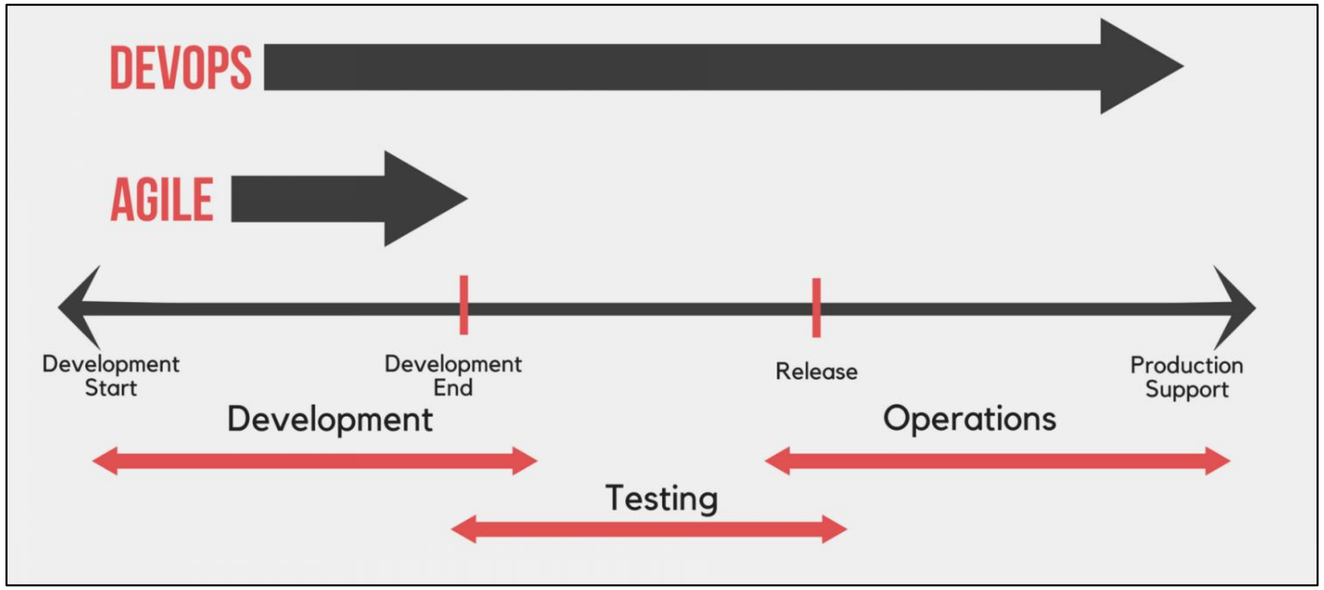
Scrum vs Kanban vs Scrumban

	Scrum	Kanban	Scrumban
Time base	1-4 weeks sprints	No time base - Kanban is event-driven	1-year, 6-months and 3-months buckets
Rules	Complete constrained process	few constraints mostly flexible process	Slightly restricted process
Roles	Product owner, Scrum master, scrum team and stakeholders	No specific roles required	No specific roles required
Event-Based	No - Once started sprints cannot be modified	Yes - On going work can react to the workflow	Yes - On going work can react to the workflow and cause On-Demand Planning
Board	Defined/resets each sprint	Persistent - the Kanban board	Persistent - the Scrumban board
Prioritization	Through backlog	Optional	Recommended on each planning
Work routines	The product owner manages tasks and assigns them to team members	Team members choose and pull tasks	The project manager push tasks in the To-Do column and team members choose and pull from there
Scope limits	Sprint limits the work amount	Work in progress limits current on going work amount	Work in progress limits and optional To-Do limit
Task size	What can be delivered in a single sprint	Any size	Any size
New items in an iteration	Not allowed	Allowed whenever the queue allows it (WIP limits)	Allowed whenever the queue allows it (To-Do & WIP limits)
Meetings	Sprint planning, daily stand-ups, sprint reviews and retrospectives	Avoidable	On-Demand Planning
Estimation	Has to be done before sprint has started	Optional	Optional
Planning routines	Sprint planning	Release/iteration planning, demand planning	Planning on demand for new tasks
Performance metrics	Burndown	Cumulative flow diagram, lead time cycle time	Average cycle time
Performance feedback	Sprint retrospective	Optional	Improvement events are an option

Source: https://medium.com/agileinsider/scrumban-the-unwanted-child-272b18482c0b

DevOps Methodology

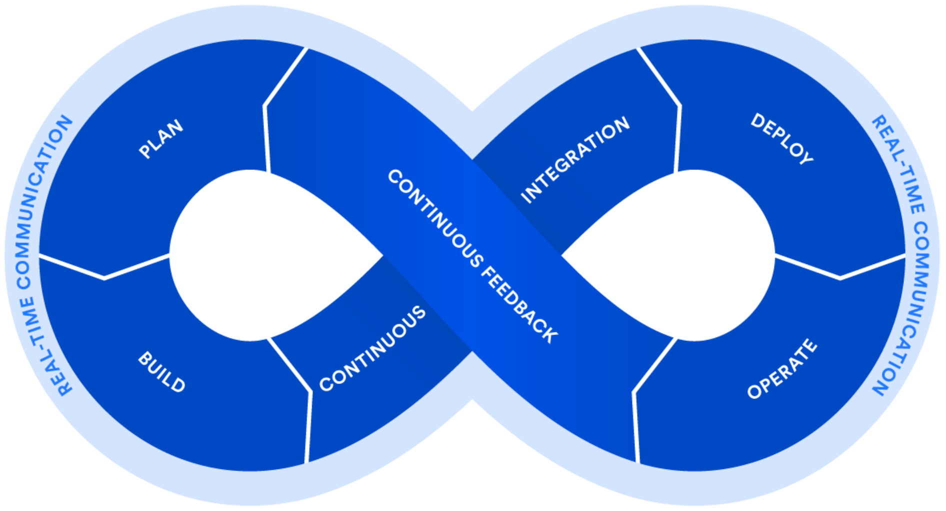
- Although both DevOps and Agile result in the development of software, they have different approaches, involve different groups and departments, and structure production differently.
- DevOps and Agile are not mutually exclusive.
  - ***DevOps** is a culture, fostering collaboration amongst all participants involved in the development and maintenance of software.*
  - ***Agile** can be described as a development methodology designed to maintain productivity and drive releases with the common reality of changing needs.*



Source: http://bridge-global.com/blog/devops-beyond-what-you-think-it-is/

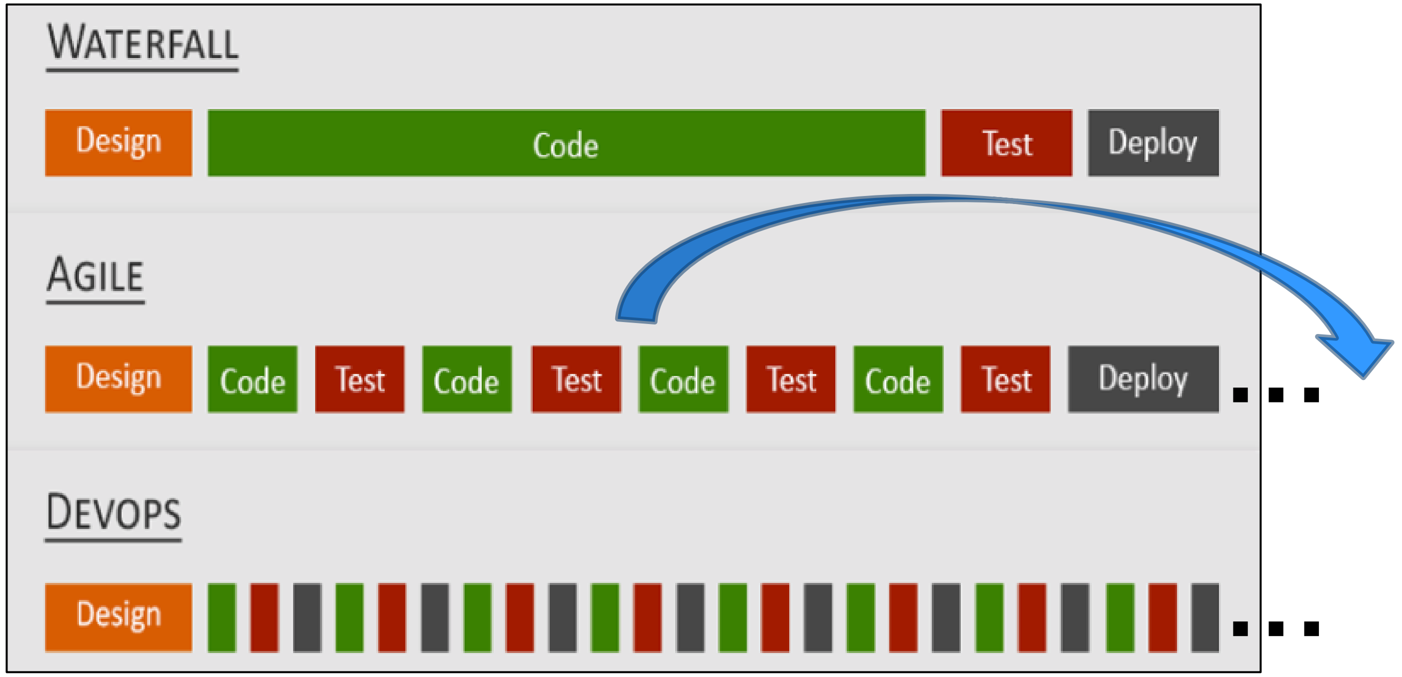
DevOps Details

- Continuous development Software delivered in multiple sprints.
- Continuous testing
  - *Automated testing tools used prior to integration.*
- Continuous integration
  - *Code pieces with new functionality added to existing code running code.*
- Continuous deployment
  - *Integrated code is deployed to the production environment.*
- Continuous monitoring
  - *Team operations staff members proactively monitor software performance in the production environment.*



Summary

- Although DevOps and Agile are different, when used together these two methodologies lead to greater efficiency and more reliable results.
- DevOps practices can be joined in part with Waterfall development
  - ***For example, the development team can use tools to automate the build.***
  - ***The siloed, staged nature of Waterfall however, means most DevOps practices are not applicable.***



Source: https://www.linkedin.com/pulse/transition-devops-jasmine-scott/

Activity – Team Project Revisited

- If you were to do the class project all over again, what Agile methodology would you choose (Scrum/Kanban/Scrumban)?
  - ***Come up with 2-3 reasons explaining why***
  - ***Any specific Agile activities you would change?***
- Identify some of the the DevOps practices you would use and how would they help improve your project?
- Describe what pros/cons your team could have had if this project had used the waterfall methodology
- Teams will discuss and share their answers with the class