

Week 9

Frameworks for Working in Teams

Chaufournier & Wood
CSCI 2541

This project is huge, how do we possibly get started working on it?

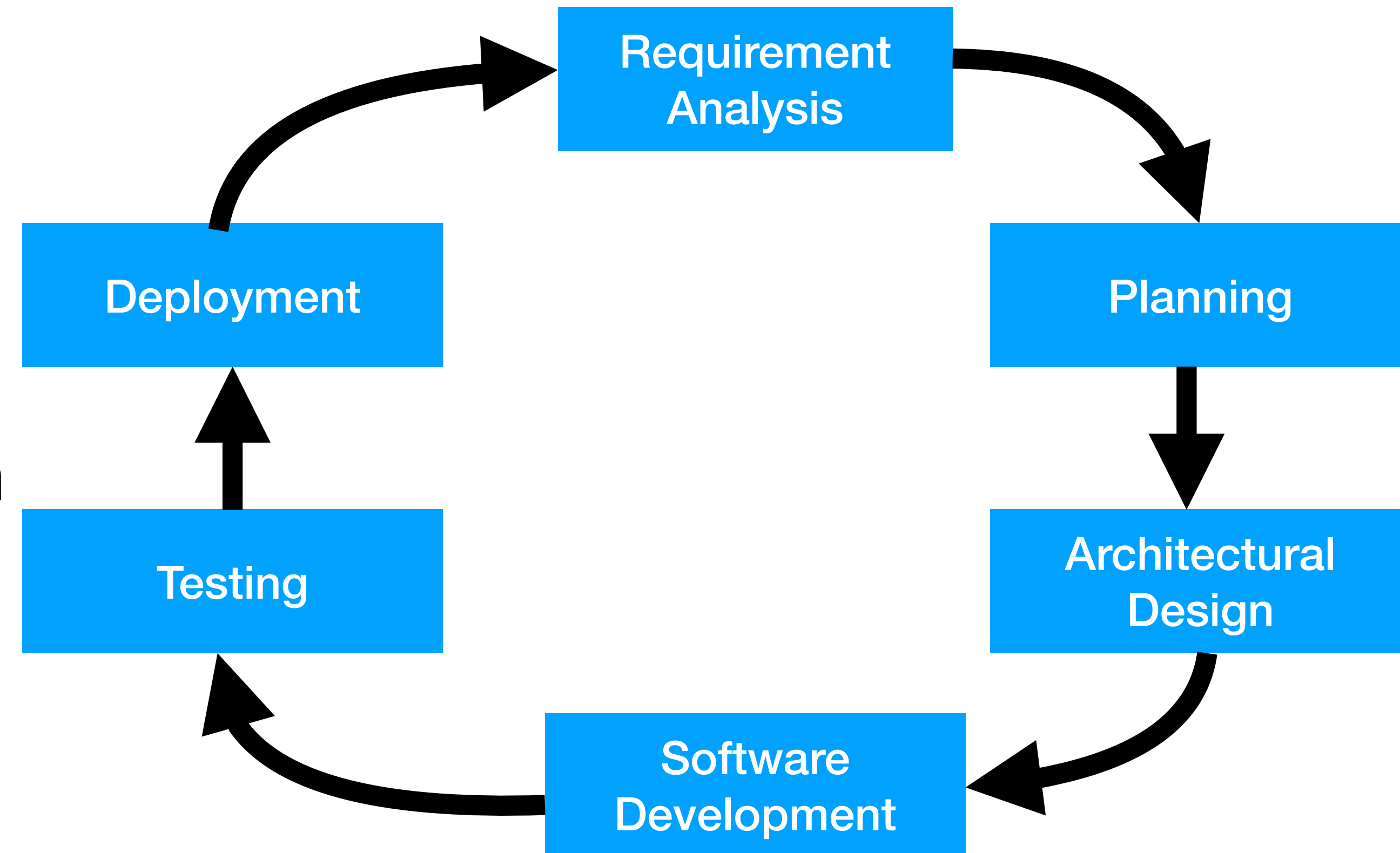
**How do we decide who should
work on what?**

How do we make sure we can
deliver on time?

**How do we make progress
without being in the same place?**

Software Development Lifecycle (SDLC)

- “A process that produces software with the highest quality and lowest cost in the shortest time possible.”
- Provides a set of phases “to quickly produce high-quality software which is well-tested and ready for production use.”
- A strong emphasis on testing, as you need to ensure code quality at every release.



Agile Software Development!

Agile

- A methodology for delivering software to customers faster and with limited headache.
- Instead of having large product launches, product features are released in smaller increments.
- Requirements and designs are continuously reevaluated to make teams flexible to change
- Key concepts are open communication, collaboration, adaptability, trust.
- A value system rather than a framework or defined set of steps.

If Agile is just a set of values how
does it help us?

This is where **Scrum** and **Kanban**
come in.

Scrum

Scrum

- Scrum is a framework for getting things done
- Scrum encourages teams to learn from their past, work together on problems, and reflect to always improve.
- Designed to allow teams to improve and be adaptive to change as they work on projects and reprioritize features as needed.
- Work is organized into increments that are completed in sprints.
- Releases generally happen at the end of a sprint or a series of sprints.

Scrum Concepts

- Scrum board - used to represent stories in progress.
- Sprint - A period of time (usually 2 weeks) of which work will be completed.
- A User Story - The smallest unit of work, usually written in human readable terms
 - i.e. When I click the login button, I am logged into the website.
- Product Backlog - The total list of all todo stories relating to the product. Includes new features, bug fixes, and enhancements
- Sprint Backlog - The list of all todo stories to be completed by the end of the sprint.
- Increment - The goal for the end of the sprint. What should be completed at the end of your sprint period.

The Scrum Ceremonies

- **Backlog Grooming:** Prioritize items and clean stories to keep the backlog up to date
- **Sprint Planning:** Deciding what stories will be undertaken for the next sprint.
- **Sprint:** The time period during which developers actually undertake the work for the increment and make progress. Usually around 2 weeks.
- **Daily Stand Up:** Allows for everyone to update their status and keep everyone on the same page and voice concerns.
- **Sprint Review:** Team gets together to demo what was completed during the sprint.
- **Sprint Retrospective:** Team meets to discuss things that went well and that need work.

Scrum in Review

- A incremental process that favors working in sprints and deploying features on a set release schedule.
- Key metric is burndown, the number of stories completed in a sprint.
- It's the entire teams responsibility to learn from past mistakes and work together to grow.
- Semi inflexible but well organized..

Kanban

Kanban

- Relies on Real-time communication of capacity and full transparency of work
- The amount of work in progress is matched to team capacity to make sure things are kept on schedule and no one is overwhelmed.
- Work items are represented visually on a kanban board so any team member can view the status of items.
- The kanban board is the single source of truth for progress. All impediments and blockers are clearly made visible.
- It's the entire team's responsibility to ensure items are moving efficiently through the process.
- Releases generally happen as features make its way through the whole process.

Kanban Concepts

- Kanban Board - A visual board with columns representing the state of work. i.e. Todo, In Progress, Code Review, Done
- A User Story - The smallest unit of work, usually written in human readable terms.
- Product Backlog - The total list of all todo stories relating to the product.
- WIP - The work in progress limit. The total number of stories a developer can take on at a given time.
- Cycle Time - The amount of time it takes for a unit to travel through the process from In Progress to Done.

A Sample Kanban Board

Github includes native support for kanban

🏠 thelimeburner / demo-aws-app

👁️ Unwatch 1 ⭐ Star

<> Code ⚠️ Issues 7 🔗 Pull requests 🎮 Actions 📁 Projects 1 📖 Wiki 🛡️ Security 📈 Insights ⚙️ Settings

Sample Kanban Board

Updated 6 minutes ago

🔍 Filter cards

+ Add cards

3 To do + ...

⚠️ Feature 2/ add login page ...
#4 opened by thelimeburner

⚠️ Feature 3/implement session storage of user cart ...
#5 opened by thelimeburner

⚠️ Feature 7/ Purchase Product Endpoint ...
#9 opened by thelimeburner

2 In Progress + ...

⚠️ Feature 1/ Add new home page ...
#3 opened by thelimeburner

⚠️ Feature 5/ Define database schema ...
#7 opened by thelimeburner

1 Code Review + ...

⚠️ Feature 6/ List Products Endpoint ...
#8 opened by thelimeburner

1 Done + ...

⚠️ Feature 4/ Configure database ...
#6 opened by thelimeburner

The Kanban Process

thelimeburner / demo-aws-app

Unwatch 1 Star

Code Issues 7 Pull requests Actions Projects 1 Wiki Security Insights Settings

Sample Kanban Board
Updated 6 minutes ago

Filter cards + Add cards

3 To do

- Feature 2/ add login page #4 opened by thelimeburner
- Feature 3/implement session storage of user cart #5 opened by thelimeburner
- Feature 7/ Purchase Product Endpoint #7

2 In Progress

- Feature 1/ Add new home page #3 opened by thelimeburner
- Feature 5/ Define database schema #7 opened by thelimeburner

1 Code Review

- Feature 6/ List Products Endpoint #8 opened by thelimeburner

1 Done

- Feature 4/ Configure database #6 opened by thelimeburner

1. Team plans features and places stories in todo

2. Developer picks a story from the top of todo and moves to in progress.

3. Developer finishes dev work and asks for code review. Moves Story here.

4. Developer completes code review and moves story to done once deployed.

5. Repeat

Kanban in Review

- A continuous process that favors limiting work in progress and deploying features as they finish.
- Key metric is minimizing the cycle time for a story
- It's the entire teams responsibility to keep stories moving through the process.
- Low overhead framework and very flexible to change.

So how can we use this in our projects?

Meet with your team and discuss your product. Come up with user stories for features.

Creating User Stories

Example: When I click the login button, I am logged into the website.

- *Assign a title:* feature/user login
- *Add a Description:* **When:** I click the login button, **What:** I am logged into the website **Why:** so I can do secure actions.
- *Assign Difficulty (Story Points):* 3
 - This allows you to estimate how long it will take and can be useful in your planning.
- *Assign Acceptance Criteria:*
 - A login form is presented
 - Logic for checking user implemented
 - Redirected to login page
 - Tested

**Decide whether to use scrum or
kanban to plan your work**

If **Scrum**, decide how long your sprint is and who will take on what for your first sprint.

If **Kanban**, arrange your backlog to priority and devs can start picking stories from the top.

Start Developing!

If **Scrum** check in with each other occasionally for updates.

If **Kanban** monitor the board and ask questions as needed.

Rinse, Repeat!

Activity 1

- Meet with your team and discuss what sort of framework you think would be helpful for your your project.
- Modify your team charter to include a new section on what approach you will try for your project.
 - You don't have to adhere to these frameworks 100% but we think it will help. You won't be graded on how well you adhere to these processes.
- Work with your team to divide up your phase 1 project into user stories
 - You can use GitHub issues or a simple markdown file in your repo root directory