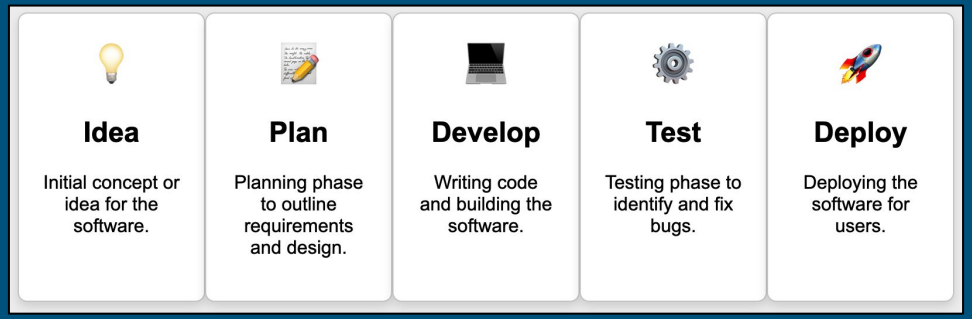
Software Development Process

COM/CSC 271
Week 2: Process & Design

Software Development Life Cycle (SDLC)

- Every piece of software goes through a similar path from idea to launch day.
- The **software development life cycle (SDLC)** describes a process for planning, creating, testing, and deploying software.
    - Each step creates an **output**–whether an idea, document, diagram, or piece of working software–which is then used as the **input** for the next step and so on until you hit your goal.



MacKay (2019), Software Development Process: How to Pick The Process That's Right For You

Why Use SDLC?

Benefits of Using an SDLC:

- Gives a clear process and guides your development steps.
- Makes building new tools cheaper, more efficient, and less stressful.
- Defines communication channels and expectations between developers and project stakeholders.
- Sets clear roles and responsibilities for your entire team (designers, developers, project managers).
- Provides an agreed-upon “definition of done” for each step to avoid scope creep and keep the project moving.
- Formalizes how to handle bugs, feature requests, and updates.

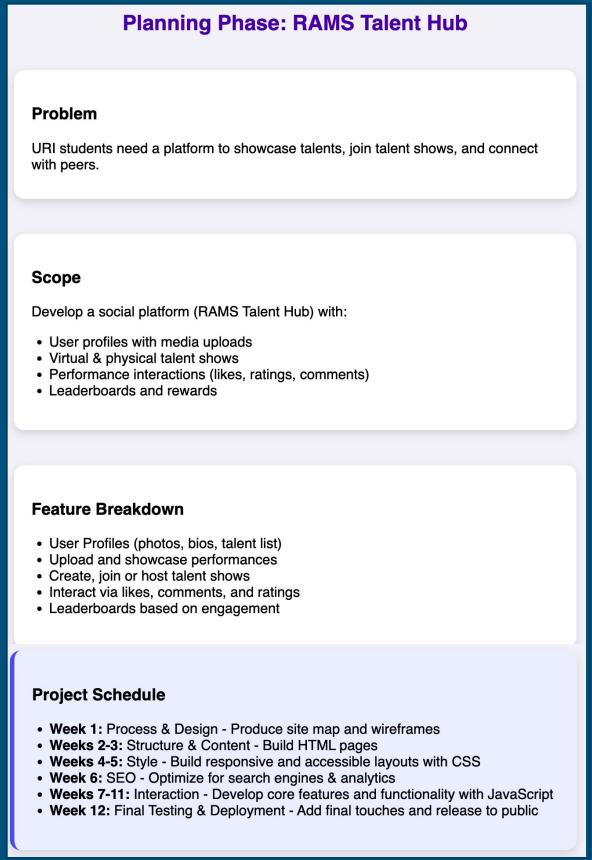
Without SDLC, easy to fall into the following traps:

- Poor collaboration and communication among team members.
- Poor or no estimation of time to complete project.
- Missing or poorly planned features or functionality.
- Poor or no prioritization of features.
- Inability to add or remove members from the project.
- Lead to changes in the project’s scope at any point after the project begins (scope creep).
- Software doesn't actually meet the needs of users.

MacKay (2019), Software Development Process: How to Pick The Process That’s Right For You

Planning Phase

- A plan is put together, breaking down the project into smaller tasks, usually with dates when each task needs to be completed.
- This phase consists of:
  - Defining the problem and scope of the project.
  - Developing an effective outline for the upcoming development cycle.
  - Catching problems before they affect development.
  - Securing the funding and resources needed to make the plan happen.
  - Setting the project schedule.



Planning Phase: RAMS Talent Hub

Problem

URI students need a platform to showcase talents, join talent shows, and connect with peers.

Scope

Develop a social platform (RAMS Talent Hub) with:

- User profiles with media uploads
- Virtual & physical talent shows
- Performance interactions (likes, ratings, comments)
- Leaderboards and rewards

Feature Breakdown

- User Profiles (photos, bios, talent list)
- Upload and showcase performances
- Create, join or host talent shows
- Interact via likes, comments, and ratings
- Leaderboards based on engagement

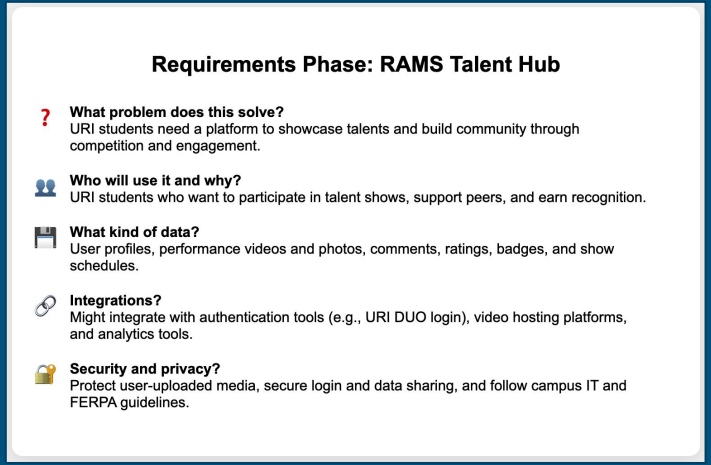
Project Schedule

- **Week 1:** Process & Design - Produce site map and wireframes
- **Weeks 2-3:** Structure & Content - Build HTML pages
- **Weeks 4-5:** Style - Build responsive and accessible layouts with CSS
- **Week 6:** SEO - Optimize for search engines & analytics
- **Weeks 7-11:** Interaction - Develop core features and functionality with JavaScript
- **Week 12:** Final Testing & Deployment - Add final touches and release to public

MacKay (2019), Software Development Process: How to Pick The Process That's Right For You

Requirements Phase

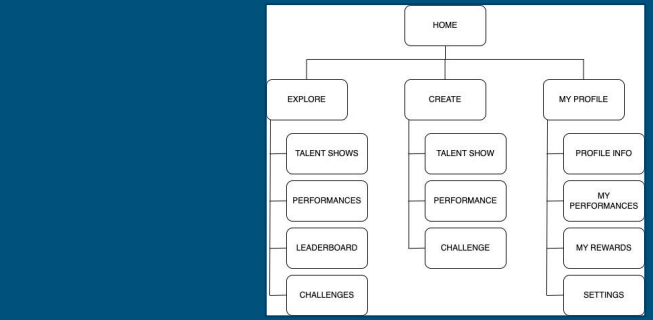
- Gathering all the specific details required for the software as well as determining the first ideas for prototypes.
- This phase consists of asking questions about the specifics around this project:
  - What problem does this solve?
  - Who is going to use it and why?
  - What sort of data input/output is needed?
  - Will you need to integrate with other tools?
  - How will you handle security and privacy?

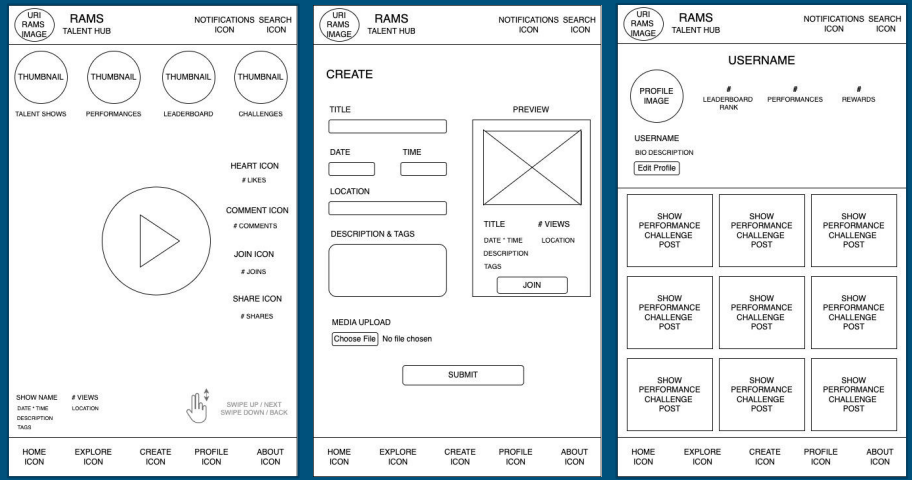


MacKay (2019), Software Development Process: How to Pick The Process That's Right For You & Preston (2020), System Development Life Cycle Guide

Design Phase

- Designers or software engineers put a design together to be reviewed to make sure it is what the user wants and that it will work.
- During the design phase you document the ideas with words and sketches of how a program is supposed to work, what are the parts, what are the inputs and outputs, and how it flows.
- This stage helps your team and your client validate ideas and get valuable feedback before you commit your ideas to code.
    - Much easier and cheaper to change than written code! If changes need to be made, you simply change the sketches and documentation.

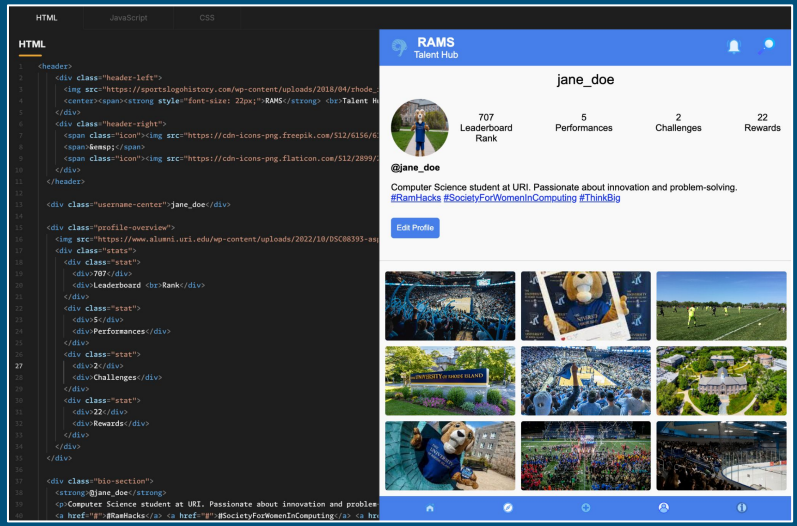




MacKay (2019), Software Development Process: How to Pick The Process That's Right For You & Preston (2020), System Development Life Cycle Guide

Development Phase

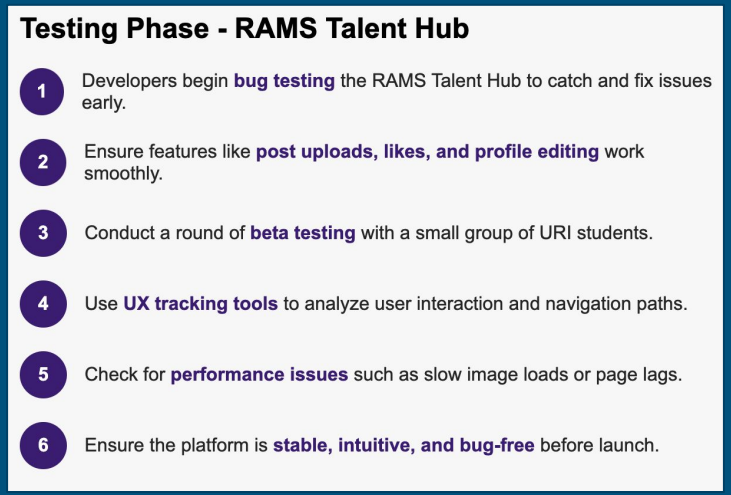
- This is the code writing phase.
- The development stage is the part where developers actually write code and build the application according to the earlier design documents and outlined specifications.
- Developers will choose the right programming code to use based on the project specifications and requirements.
- The goal of this phase is to stick to the plan, and build clean and efficient software.



MacKay (2019), Software Development Process: How to Pick The Process That's Right For You & Preston (2020), System Development Life Cycle Guide

Testing Phase

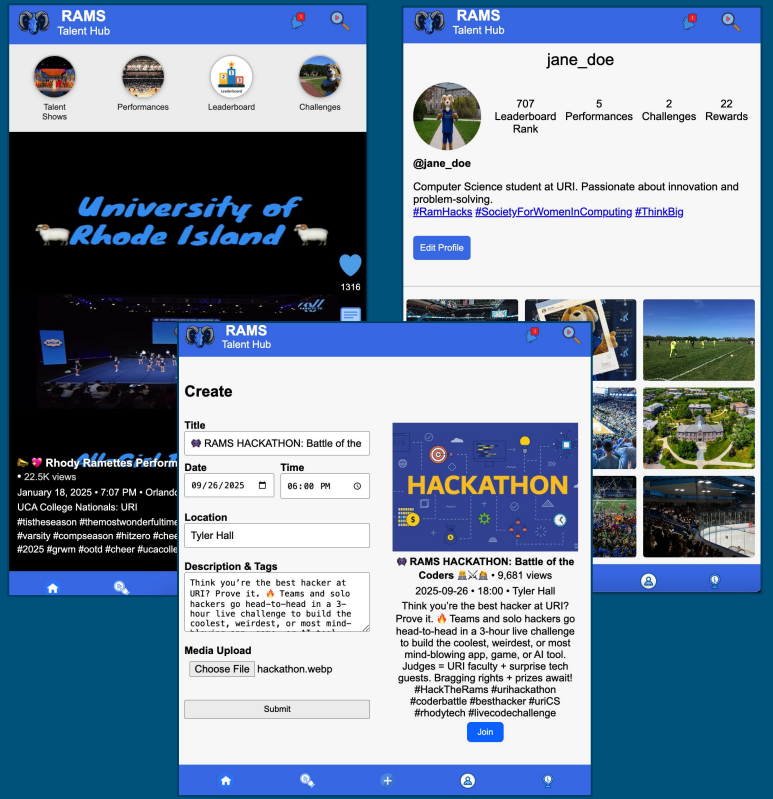
- During the testing stage, developers will go over their software with a fine-tooth comb, noting any bugs or defects that need to be tracked, fixed, and later retested.
- Testing varies in length—it could be quick for minor updates or take weeks for major ones!
- Once all planned features are in place, deeper testing begins:
    - Releasing the product to a small group of beta testers.
    - Using UX tools to track how users interact with it.
- It is important to make sure you are not shipping buggy software to real customers!



MacKay (2019), Software Development Process: How to Pick The Process That's Right For You & Preston (2020), System Development Life Cycle Guide

Deployment & Maintenance Phase

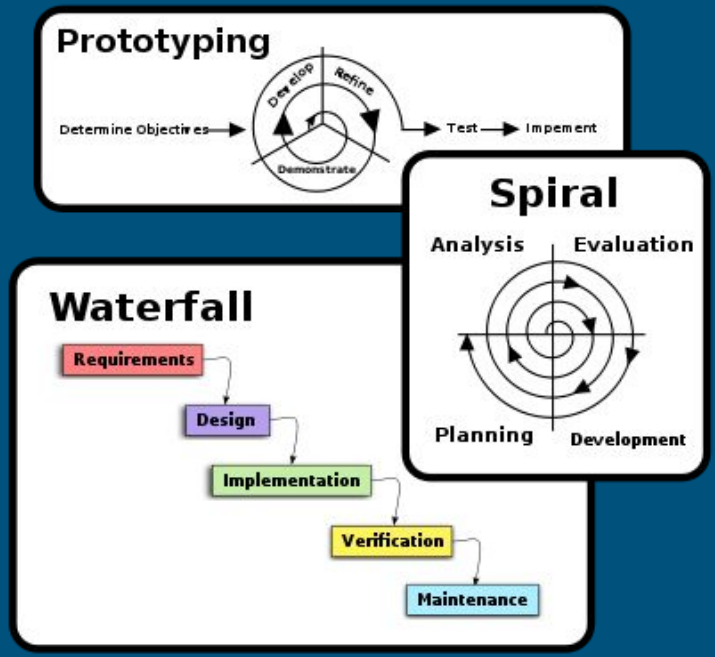
- The deployment phase is launching your software to all your users and releasing it into the appropriate market.
- The software product will need updates from time to time because a user found a bug or something changed (like the operating system) so the program needs to be updated.
- Developers must now move into a maintenance mode and begin practicing any activities required to handle issues reported by end-users.
    - Basic upkeep and maintenance of your software to ensure uptime and customer satisfaction.
    - As people begin to use your software, they will undoubtedly find bugs, request new features, and ask for more or different functionality.



MacKay (2019), Software Development Process: How to Pick The Process That's Right For You & Preston (2020), System Development Life Cycle Guide

Software Development Processes

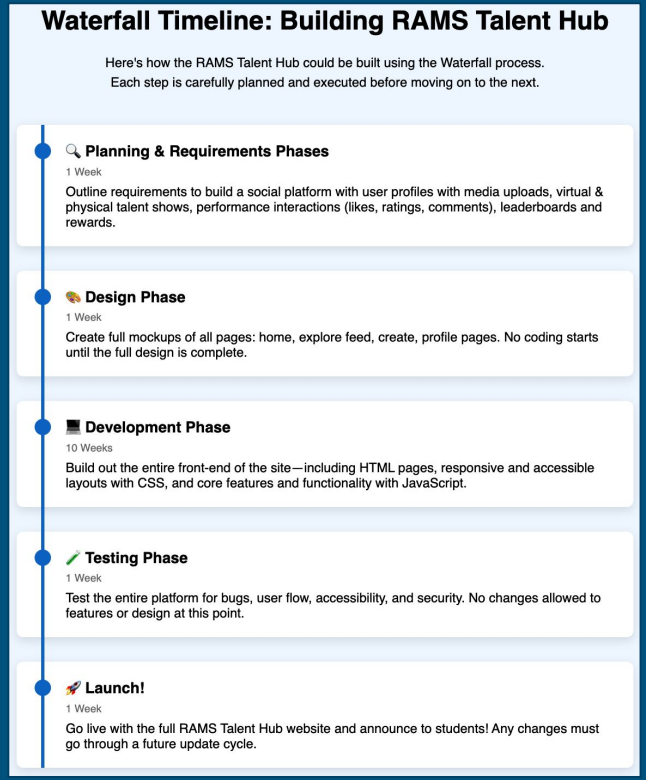
- The software development life cycle is more of a guideline for building software.
  - How you check off each phase, when and in what order is up to you.
- Over the years, a number of different **software development processes** have been formalized to tackle more and more complex projects.
  - Which process you use depends on your goals, the size of the project and your team, and other factors.



MacKay (2019), Software Development Process: How to Pick The Process That's Right For You

Waterfall

- **Waterfall** is one of the oldest and most traditional models for building software.
- Follows each step of the software development lifecycle **in a strict, linear sequence**.
  - You must finish one phase completely—like planning—before moving to the next—like design.
  - Each step waits for the one before it to be fully complete.
- In order to complete a project, you **first** need to **know everything** that **needs to be done** and **in what order**.
  - Design and develop all features of software in one fell swoop!



MacKay (2019), Software Development Process: How to Pick The Process That's Right For You

Waterfall

✅ **Who It's For:**

- Teams with rigid structures and documentation needs.
- Works best when your goals, requirements, and technology stack are unlikely to radically change during the development process.
- Best suited for larger organizations (like government agencies) that require sign-offs and documentation on all requirements and scope before a project starts.

🚫 **Who It's NOT For:**

- If you are testing a new product, need user feedback mid-stream, or want to be more dynamic in your development process.
- Slow and does not adapt well to change.
- Won't be creating and testing prototypes and changing your mind along the way.
- Might end up committing to the wrong path without knowing it until launch day.

MacKay (2019), Software Development Process: How to Pick The Process That's Right For You

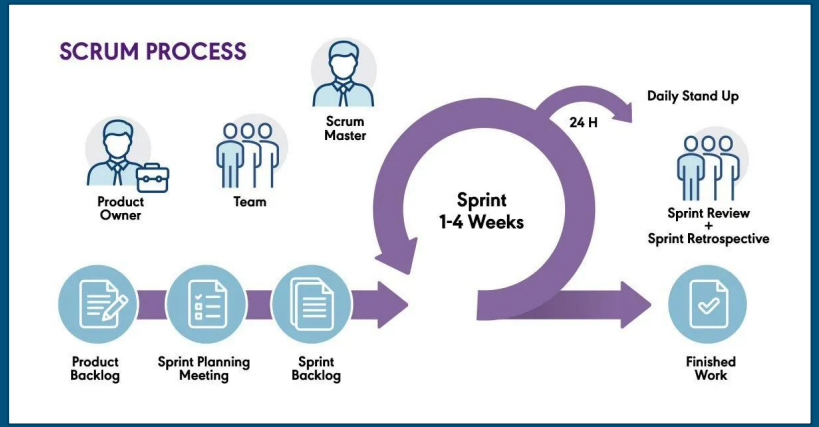
Agile & Scrum

- **Agile** and its most popular methodology—**Scrum**—take a dynamic, iterative approach to development.
- Teams work in **“Sprints”** of 2 weeks to 2 months to build and release usable software to customers for feedback.
- **Agile** is all about moving fast, releasing often, and responding to the real needs of your users, even if it goes against what is in your initial plan.
    - Moving in one direction with the understanding that you will change course along the way.

MacKay (2019), Software Development Process: How to Pick The Process That’s Right For You

Scrum

- **Scrum** is a methodology of Agile.
- Scrum recommends breaking things into small pieces of work and timeboxing our development cycles.
- Scrum sets roles like:
  - **Product Owner:** Defines product vision, priorities, accepts & rejects work performed.
  - **Team:** Builds the product and its features.
  - **Scrum Master:** Keeps the process running smoothly and the team on track, sets sprint duration and removes impediments.



MacKay (2019), Software Development Process: How to Pick The Process That's Right For You

Agile & Scrum

✅ **Who It's For:**

- Dynamic teams doing continuous updates to products.
- Allows tighter feedback loops throughout the software development process so you can adapt and react to real customer needs.
- Favored by most startups and technology companies testing new products or doing continuous updates to long-standing ones.
- As testing takes place after each small iteration, it is easier to track bugs or roll back to a previous product version if something more serious is broken.

🚫 **Who It's NOT For:**

- Team's with extremely tight budgets and timelines.
- Projects can easily go over their initial timeframe or budget, create conflicts with existing architecture, or get derailed by mismanagement.
- Agile and Scrum takes dedication and a solid understanding of the underlying process to pull off properly.
    - Important to have at least one dedicated Scrum master on your team to make sure sprints and milestones are being hit and the project does not stall out.

MacKay (2019), Software Development Process: How to Pick The Process That's Right For You

Processes & Plans Are Just Guesses

- Every software development process and method comes down to five basic principles:
  - Know what you're building and why.
  - Choose the process that feels right for you and your team's goals.
  - Design and build working software.
  - Put it in users' hands and listen to their feedback.
  - Use that feedback to make it better.

- Remember, it's a lifecycle. If you don't get it right the first time around, understand why it didn't work, try a different process and start again.

MacKay (2019), Software Development Process: How to Pick The Process That's Right For You