

Android Mobile Application Development Project

Choosing a project and team

You'll be working on one mobile app project throughout the semester, so think of an idea that you're interested in and excites you! The result should be a production-level app that is well designed, easy to use and solves a problem for your target user.

Requirements:

- Multiple activities
- A well organized and structured app
- A well thought out and implemented organization and flow
- A clear user interaction model
- A comprehensive data management strategy
- Adapt to all supported device configurations
- Use of Material Components
- Consistent theme throughout the app including custom launcher icons

This project can be done individually or in a team. The ideal size for a team is 2-4 class members. This should be a collaborative effort. How each team delegates the work is up to the discretion of the team members and each student will be held responsible for their work.

Project Management Process

<https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/>

The waterfall development method has been around since 1970, initially from the manufacturing and engineering industry where a linear process makes sense. When used for software development it treats the process as a single path through different phases where each phase is completed before the next starts.

Analysis – design – development – testing – deployment – maintenance

The advantage of the waterfall process is that each phase is well defined and forces the client and development team to spend time thinking through the requirements and design up front.

The disadvantage is it creates a long development process where all requirements and design must be done up front and there's no opportunity to modify, improve, or fine-tune along the way.

Many products, especially software, require exploration and experimentation.

The agile development method takes an iterative approach where you break up projects into smaller chunks of work called iterations, or sprints, so you can quickly define, create, and iterate. Each sprint receives feedback and then priorities and tasks are set for the next sprint. This allows for client feedback, early user testing, time to react to business changes, and the ability to modify and fine-tune along the way if needed. It provides the agility to both create and respond to change. Some form of the agile development process is most commonly used today for software development.

Incremental delivery planning methods <https://www.linkedin.com/learning/agile-at-work-planning-with-agile-user-stories/incremental-delivery-planning-methods>

Agile Development Process

Agile is a development process where you break up projects into smaller chunks of work called iterations, or sprints, so you can quickly define, create, and iterate. This lifecycle makes it easier to adapt to user feedback and react to business changes. It provides the agility to both create and respond to change. As we only have 7 weeks we'll be following an abbreviated version of this process that will include two iterations/sprints.

Why Agile exists <https://www.linkedin.com/learning/agile-at-work-building-your-agile-team/why-agile-exists-2>

Agile Manifesto Principles <https://agilemanifesto.org/principles.html>

What is Agile? <https://www.linkedin.com/learning/devops-foundations-lean-and-agile/what-is-agile-2?u=42275329>

Phases

- Envision (vision)
 - Product vision - *why* are you doing this project?
 - Idea, not technology
 - Who is your target customer?
 - The vision should focus on the customers
 - What is the customer's product vision?
 - Product mission
 - How will you accomplish the vision?
 - Includes technology
 - Scope
 - Constraints
 - Success criteria
 - How will you know if the project accomplished its vision?
 - Meeting all the technical requirements doesn't ensure it accomplishes the project's vision
 - Project community
 - Who are the right participants to include in the project community?
 - Project team organization
 - How will the team deliver the product?
 - Outcome
 - **Project vision** (charter)
 - Vision
 - Objective
 - Target customer
 - Key benefits
 - Purpose
 - Mission
 - Product description/summary statement
 - Minimum Viable Product (MVP)
 - Scope/boundaries
 - Target users
 - Competition
 - Differentiation

- Success criteria
 - Criteria for meeting your vision
 - How will you measure success?
- Speculate (plan)
 - Refine information gathered in the Envision phase
 - Expand the product vision into more detail
 - Transform requirements into detailed features
 - A feature is a piece of functionality that focuses on a specific business need
 - Prioritize feature list and features from previous iteration
 - can use categories such as high, medium, low priority
 - Estimate the workload for each feature
 - In the first iteration you should estimate the work effort for all features
 - In future iterations estimates might need to be adjusted
 - Relative estimates are fine
 - Categorize using buckets
 - Identify features for upcoming iteration
 - Outcome
 - **Iteration plan**
 - Feature list for the current iteration
 - Once the scope of an iteration is agreed upon it should not change
- Explore (create)
 - Specifications and design are implemented
 - Daily stand-up meetings
 - Report progress
 - Identify issues
 - Outcome
 - **Implemented plan/feature list**
 - The explore phase ends either when the planned features for the sprint are completed or when the time allocated has been reached.
- Adapt (review)
 - Feedback
 - Feedback areas
 - Technical
 - Technical choices
 - Architecture
 - Materials
 - Proof of concept/tutorials
 - Technical research
 - Development
 - Aesthetics
 - Moodboard
 - Prototype
 - Asset creation
 - UI/UX
 - Low fidelity
 - Sketches
 - Wireframe
 - Paper prototype

- High fidelity
 - Digital prototype
 - Project interaction
 - Storyboard
- Stakeholders
 - Target users/audience
 - Sponsors
 - Mentors – content/industry, and technical
- Approaches
 - Stakeholder interviews
 - One-to-one interviews in which the user is asked to provide feedback
 - Perform a series of tasks in a prototype or a product
 - Collect feedback and validate flow, design and features
 - Develop a script with detailed questions
 - Ask questions based on a script
 - Capture and keep all answers and feedback
 - Summarize
 - Use cases and scenarios
 - A comprehensive list of scenarios that describe how a user interacts with the product
 - Ensure all possible actions are considered
 - Different scenarios needed for each type of user
 - Personas
 - Create a fictional character that highlights the behavioral patterns, needs, and motivations of your target audience.
 - Mental model
 - A model, or explanation, of how users see the world
 - Task analysis
 - Break down the required information and actions needed to achieve a task
 - Helps to understand the current system and its information flows
 - User Flow/ Experience map
 - a visual representation that illustrates the user's flow within a product that illustrates their needs, wants, expectations, and the overall experience on the way to a particular goal
 - Surveys
 - Collect data from a targeted group of respondents to gain information and insights related to your project
 - Capture and keep all answers and feedback
 - Summarize
 - Testing
 - Usability testing
 - Users are asked to perform a series of tasks in a product or prototype
 - Validates and collects feedback on flow, design, and features
 - A/B testing
 - Offer alternative approaches or versions of your prototype or product to users and compare the results and feedback

- Reference: [UX Design Methods & Deliverables](#)

Outcome

- **Documented feedback**
 - Document the feedback areas, stakeholder, approach, and methods used
 - Include the full set of interview questions, survey, user scenarios, etc..
 - Document the resulting feedback in detail
 - Include the complete set of feedback data
 - Document the feedback in whatever way you feel is most appropriate
 - Photos, sketches, renderings, screenshots, videos, text, etc
 - Summary of feedback
 - Summarize the findings from your feedback
- **Lessons learned**
 - Based on the feedback, what did you learn?
 - What was unexpected / what lessons did you learn?
 - What worked, what didn't work?
 - What did you accomplish? Reflect on your time management.
 - Any changes to be made?
- Adaptive actions are incorporated into the next iteration
- Capture, share and apply lessons learned
- Proceed to speculate phase for next iteration or close phase to end the project
- **Close**
 - Deliverables are completed
 - Outcome
 - Did you meet your success criteria from your project charter in the Envision phase?
 - Lessons learned
 - Future plans