Workshop 3, Development & QA

TIEA3290 Mobiilisovellus, 3 op, Mobiilisovelluksen toteuttaminen ideasta sovelluskauppaan

Introductions

Toni Hintikka
Project Manager

Jani Lirkki
Lead SW Developer

Jani Kerttula
Senior SW Developer

Prafull Sanas
Senior SW Developer

Timo Tarvainen
Senior Test Engineer

Homework after the design workshop

- Github repository
  - Repository ready
  - Readme.md included in the repository with basic feature list
  - Prepare to present during the Development workshop
  - Deadline 13.2.

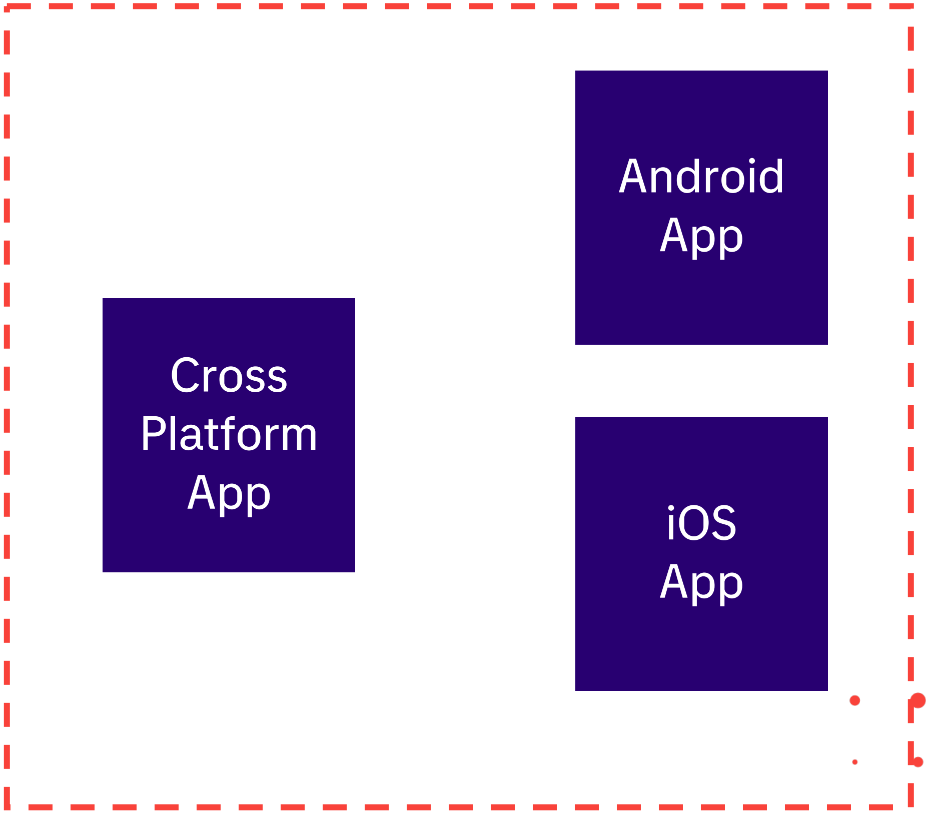
Agenda, Development & QA workshop

- Selecting the mobile platform and selecting the tech scope
- Version control and distributing apps for testing
- Quality Assurance
- Initial development plan
  - What features are in the scope?
  - What can be left out?
- Q&A and discussion with Tietoevry experts
- Homework after this workshop
  - Teams create very basic app (can be simple skeleton) that can be distributed through TestFlight or Google Play
  - Homework deadline 20.2.2023

- Goals of the workshop
  - Teams have Github repositories in place
  - Teams have selected the technology they plan to use
  - Teams have very basic QA plan
  - Teams have very basic development plan (list of features)

Simplify, Focus

Do you really need this on MVP?



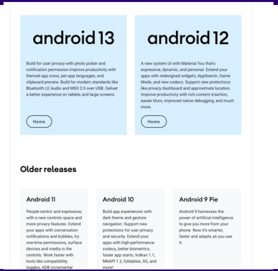
Lifespan of the mobile app

4/23

React native versions

From github seems new version every 2 weeks

Android technology



android 13

Build for user privacy with photo picker and notification permission.Improve productivity with themed app icons, per-app languages, and clipboard preview. Build for modern standards like Bluetooth LE Audio and MIDI 2.0 over USB. Deliver a better experience on tablets and large screens.

Home

android 12

A new system UI with Material You that's expressive, dynamic, and personal. Extend your apps with redesigned widgets, AppSearch, Game Mode, and new codecs. Support new protections like privacy dashboard and approximate location. Improve productivity with rich content insertion, easier blurs, improved native debugging, and much more.

Home

Older releases

Android 11

People-centric and expressive, with a new controls space and more privacy features. Extend your apps with conversation notifications and bubbles, try one-time permissions, surface devices and media in the controls. Work faster with tools like compatibility toggles, ADB incremental

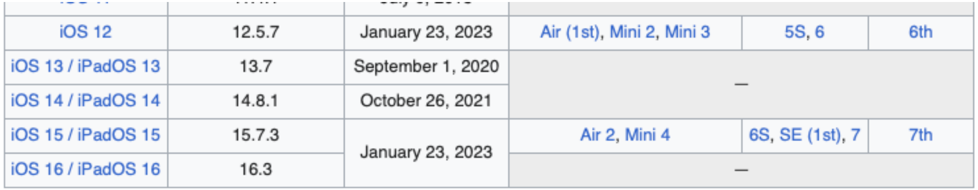
Android 10

Build app experiences with dark theme and gesture navigation. Support new protections for user privacy and security. Extend your apps with high-performance codecs, better biometrics, faster app starts, Vulkan 1.1, NNAPI 1.2, foldables, 5G, and more!

Android 9 Pie

Android 9 harnesses the power of artificial intelligence to give you more from your phone. Now it's smarter, faster and adapts as you use it.

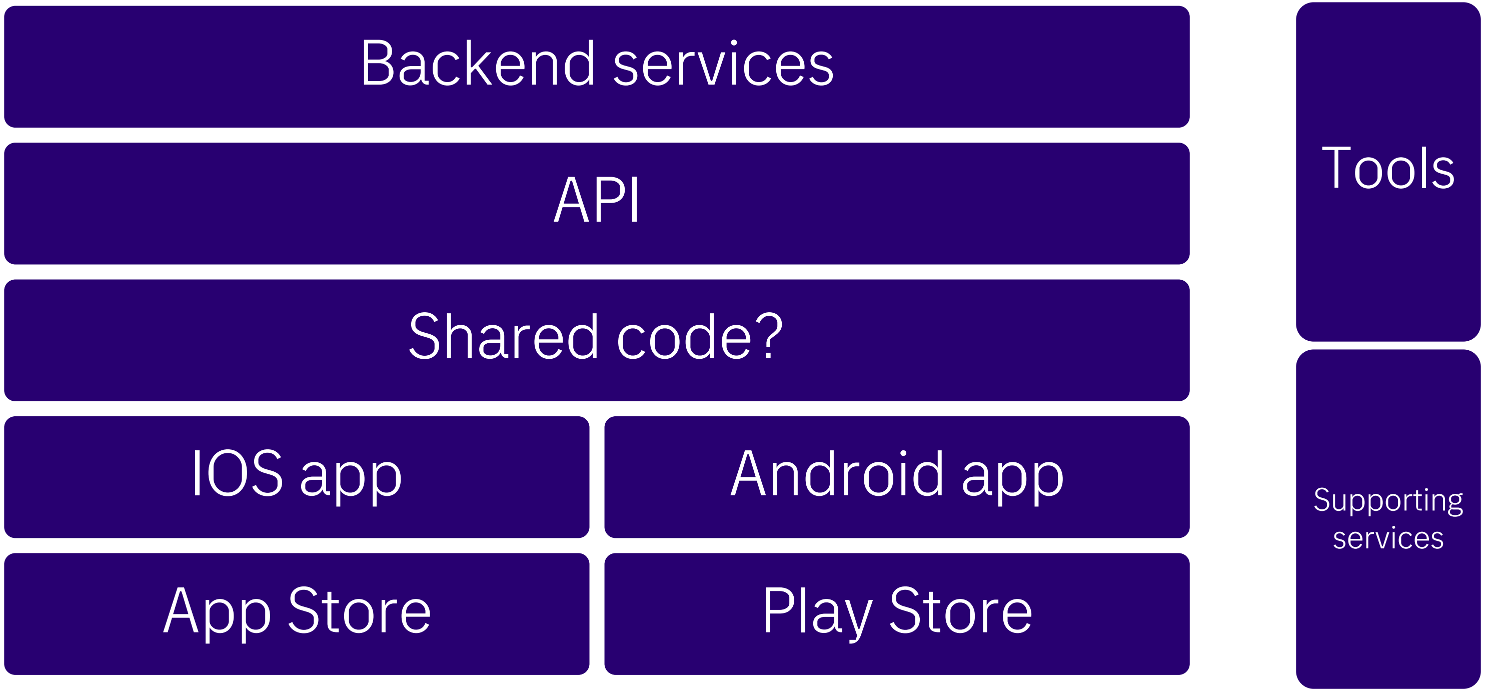
Apple technology



iOS 12	12.5.7	January 23, 2023	Air (1st), Mini 2, Mini 3	5S, 6	6th
iOS 13 / iPadOS 13	13.7	September 1, 2020	—		
iOS 14 / iPadOS 14	14.8.1	October 26, 2021			
iOS 15 / iPadOS 15	15.7.3	January 23, 2023	Air 2, Mini 4	6S, SE (1st), 7	7th
iOS 16 / iPadOS 16	16.3		—		

mvp

Tech scope high level



Backend services

API

Shared code?

IOS app

Android app

App Store

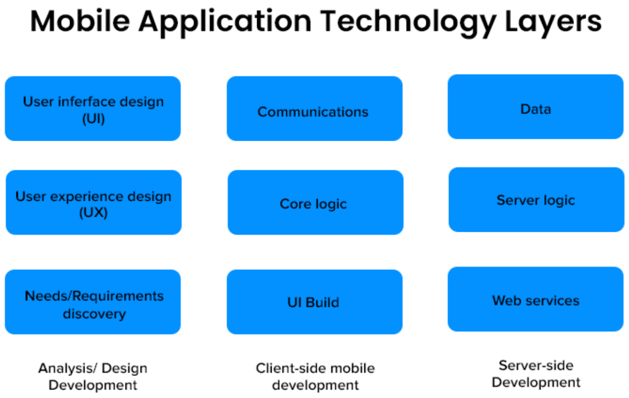
Play Store

Tools

Supporting services

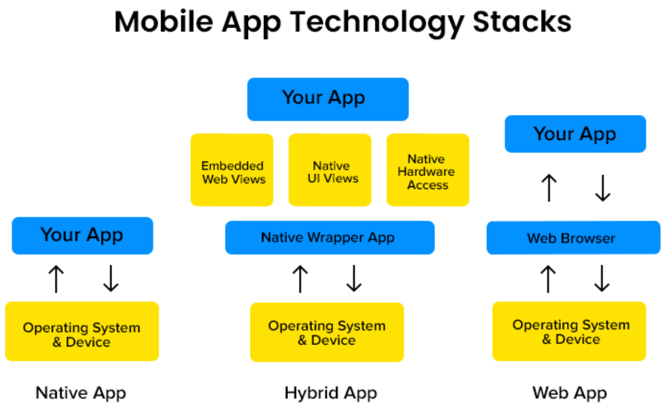
Tech Stacks to Launch Your App

- https://appinventiv.com/blog/technology-for-mobile-app-development/



Mobile Application Technology Layers

Analysis/ Design Development	Client-side mobile development	Server-side Development
User inferface design (UI)	Communications	Data
User experience design (UX)	Core logic	Server logic
Needs/Requirements discovery	UI Build	Web services



Mobile App Technology Stacks

Native app options

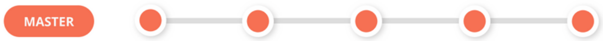
- Native iOS
- Native Android
- Cross platform application
  - Flutter
  - React Native (link to comparisation article)
  - .NET Multi-platform App UI (https://dotnet.microsoft.com/en-us/apps/maui)

Choosing your platform

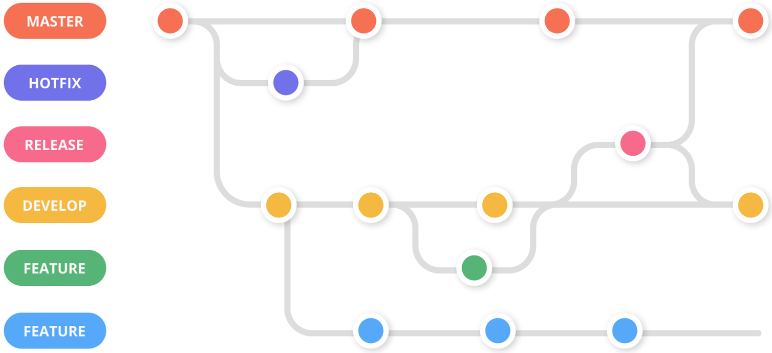
- What devices do end users use?
  - Do you need iOS and/or Android applications?
  - Do you need Cross Platform technology, or can you use Native technologies
- Platform maturity
  - Does the platform support all needed features you have planned
  - Does platform have active support and continuous releases
  - Can you find answers to possible questions that will rise in development
    - Stack Overflow etc…

Version Control (Git)

From this...



To this



Version Control

- There are many different practices on how to use version control efficiently
  - Release branches, feature branches etc.
- From the ground up it's important to set how do YOU handle version control
  - Do you have a stable "Main" branch, separate build branches, how to handle hotfixes and features
- A good version control foundation separates good code from broken
  - When you manage version control clearly, you can avoid unnecessary regressions that could have been prevented
  - Smaller == Better
- Code Review, Pull Request, "Ninja Commit", Dev-branch

- Discussion: What do you think good version control looks like? 10 min

CI/CD

- Automation of delivery
  - Microsoft App Center, Github Actions+Firebase,…
- When should you start build process, should the build go to deployment or just automate testing
  - Cost and resource management (build minutes)
  - Release-branch OR Main-branch build process, automated PR-test pipeline
- Signing process of application in build

Delivery

- How do you distribute builds to testing/production
  - Possible deployments:
    - **Google Play Store**
      - Internal, Closed, Production releases
    - **Testflight** (iOS specific)
    - Firebase distribution
    - App Center distribution
    - "Manual distribution"

Quality Assurance

1. QA vs QC
2. End user perspective
3. Scripted testing vs exploratory testing
4. Manual testing vs automation testing

WHAT IS QUALITY? (2min)

- What is quality assurance? (2min)
- Why is this important?

QA vs QC

- Quality Assurance
  - Focuses on processes and procedures that improve quality before and while development is taking place
    - Testing strategy
    - Test planning (test cases)
    - Documentation
    - Monitoring
    - Requirements review
    - User story grooming
  - When is QA ready?

- Quality Control
  - Focuses on testing activities that remain *after* development
    - Goal to make sure the app/functionality works as it should be through various methods of testing
    - Test execution and reporting
    - Beta testing with a group of end users
  - Giving feedback of the developed app/functionality
  - Goals
    - To find bugs/defects/glitches/anomalies (creating **new** information)
    - To assure that the software works as it should (creating trust)

End user perspective

- What is the end user's problem and how does my application solve it?
- How will the end user use my application?
- How well does my app solve the end user's problem? Is it easy to use and intuitive?
- What are the most important functionalities of my app?

- Know your customer/end user
  - Understanding end user's behaviour patterns is guiding the decisions you make while developing
  - Familiarity in your app (same kind of behaviour as in other mobile applications)
  - How many ways are there to use your app?

Scripted testing VS Exploratory testing

- Very different methods of testing

- Scripted testing
  - Creates trust that the app/functionality is working as it is planned to work
  - Testing against requirements for every functionality
    - Test cases, user stories

- Exploratory testing
  - Unscripted or very loosely scripted testing
  - Creates new information about the state of the app/functionality
  - Leans heavily on tester's own capabilities, experience, responsibility to assess, plan and execute testing
  - Might result in new bugs / anomalies. If there aren't any new findings, that is also a result!
    - Well, what is a bug? **If it bugs you, it's a bug!**

Manual testing vs automated testing

- Manual testing
  - Basic method of testing
  - Slower than automated testing
  - Prone to human error (it gets repetitive quite quickly)
  - Very good for exploratory testing, usability testing and ad-hoc testing
  - Handles complex scenarios better

- Automated testing
  - Takes time to setup, but it will save a lot of time in the long run
  - Cost-effective
  - Good for regression testing
  - Unable to do exploratory testing
  - Can be run for example in cloud environment during night time

Exercise time!

Write ten (10) bullet points / ideas on **how to test the mobile application** you are developing. (20min)

Presentations

- Team presentation and feature comments

Homework

- Do you have developer account?
- Teams create very basic app (can be simple skeleton)
- Distributed app through TestFlight and/or Google Play
  - You can create open test link.
- Send a link to the slack to #releases channel.
- Homework deadline 20.2.2023

Thank you

Learn more at tietoevry.com

tietoevry