Intro to Mobile App Dev

Written by Thapanapong Rukkanchanunt

Outline

- History of Mobile Development
- Qualities of Mobile Application

Early 2000s

- Mobile application development was primarily platform-specific, with developers creating apps separately for iOS, Android, and other mobile platforms.
- Native development involved using platform-specific languages and tools like Objective-C for iOS and Java for Android.
- With an increasing usage of iOS devices, the developers at that time needed to learn all platform-specific languages.

Mid 2000s

- To overcome the fragmentation and complexity of native app development, developers turned to web technologies.
- This was rather clever because all platforms have access to web browsers or a tool to generate web content.
- Web applications were developed using HTML, CSS, and JavaScript and were accessible through mobile web browsers and were platform-agnostic, but they lacked access to device hardware and native features.

Late 2000s

- Hybrid application frameworks like PhoneGap (later became Apache Cordova) emerged, allowing developers to build mobile applications using web technologies while gaining access to native device features through plugins.
- These applications could be packaged and distributed through app stores and were a compromise between web and native applications.
- This reduced some stress to mobile developers.
- In this period, Ruby has emerged as one of the languages for mobile development.

2010s

- Developments became more pleasing as cross-platform native development frameworks gained popularity.
- Tools like Xamarin and React Native allowed developers to write code once and deploy it on multiple platforms.
- Xamarin used C# and .NET for cross-platform development, while React Native used JavaScript and React to achieve the same goal.

2015

- Progressive Web Apps (PWAs) emerged.
- PWAs are web applications that leverage modern web technologies to provide a native-like experience. It’s like web but has its own render.
- They can be installed on the user's home screen and work offline. PWAs offer a way to develop cross-platform apps that don't require installation from app stores.
- Today’s example is Spotify.

2017

- The mobile application development changed drastically as Google introduced Flutter, an open-source UI framework, and Dart, a programming language, for building natively compiled applications.
- Flutter allows for building apps for multiple platforms with a single codebase.
- Flutter gained popularity for its expressive UI, hot-reloading, and performance.

2020s

- Two new frameworks were released for platform-specific development.
- Apple introduced SwiftUI, and Google introduced Jetpack Compose, both of which are modern UI frameworks that aim to simplify native app development for iOS and Android, respectively.
- These frameworks use declarative syntax and offer a more streamlined approach to UI development.

We will use Flutter

- Dart is short
- Fast development (Experience varies but generally yes)
- Cross-platform (iOS and Android simultaneously!)
- Support Material Design 3
- Cute mascot! :))

What makes mobile app great?

Simplicity

Speed

Visual

Flexibility

Security

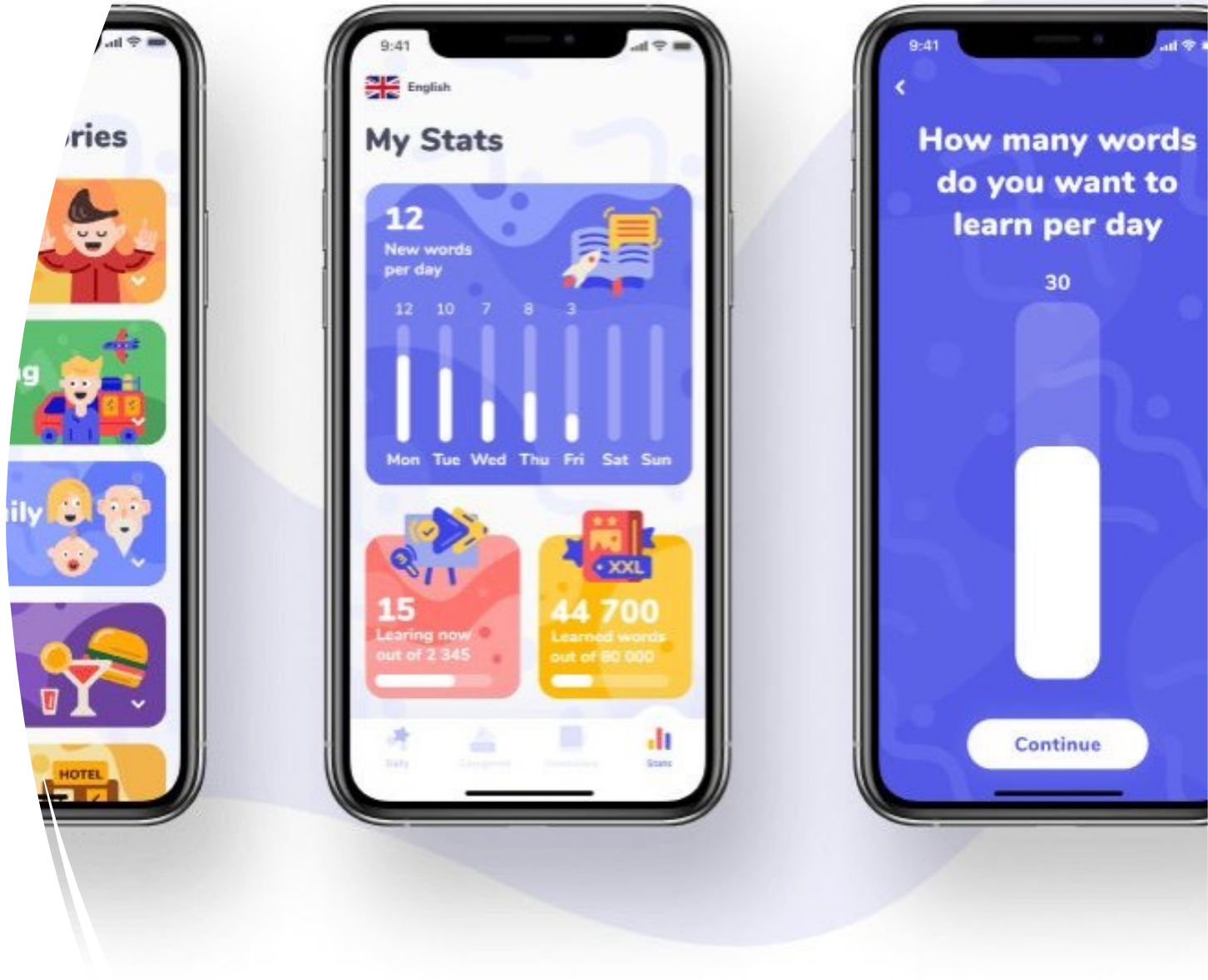
Search

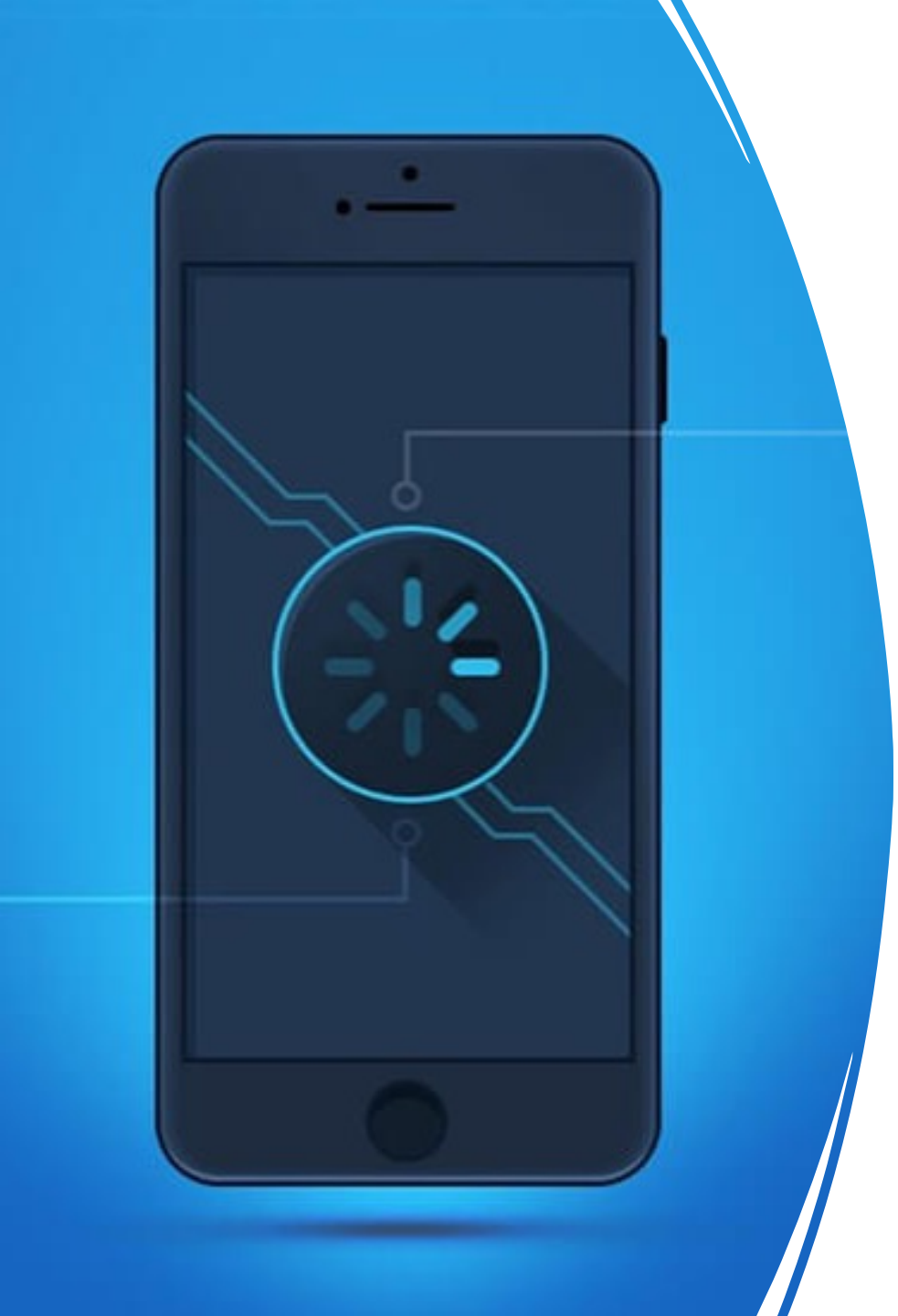
Push notification

Updates

Simplicity

- Easy nagivation – How to get to what you want as quickly as you can
- Intuitive control flow – People don’t read instruction
- No cluttered screen – Too many things can be distracting





Speed

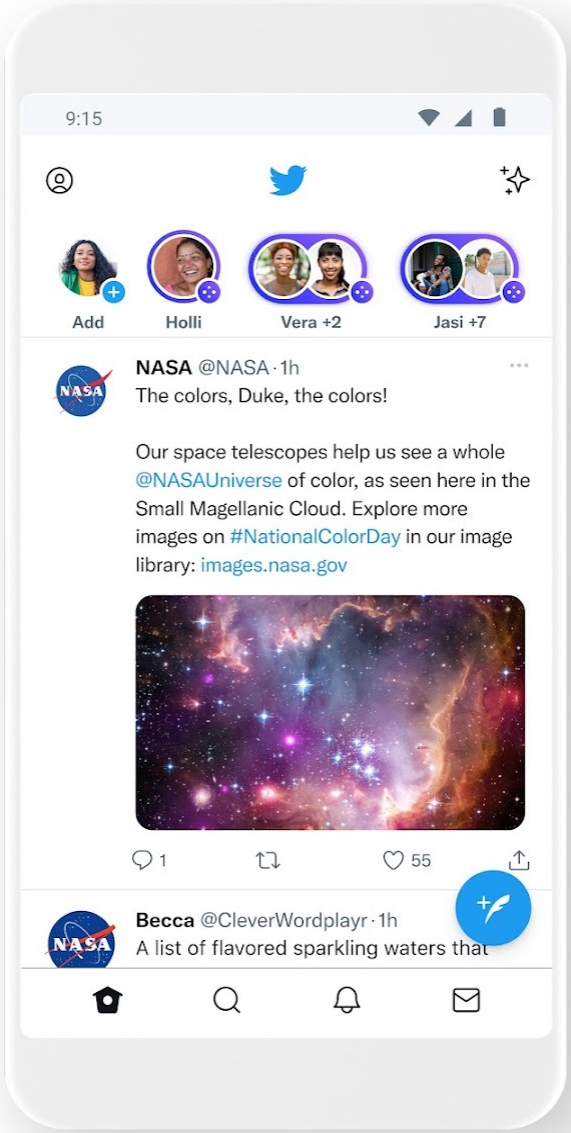
- Fast loading screen
- Loading images, assets
- People hate waiting!!
  - 0-2s is ideal
  - 2-4s is tolerable
  - 4s+ is bad

Visual

- Beautiful, eye-catching, and attractive design.
- Image resolution, image color depth, text size, and color contrast are suitable for users of all genders and ages.

Flexibility

- Availability in both iOS, Android and must have same functionality!
- Version exclusive will steer your customer away

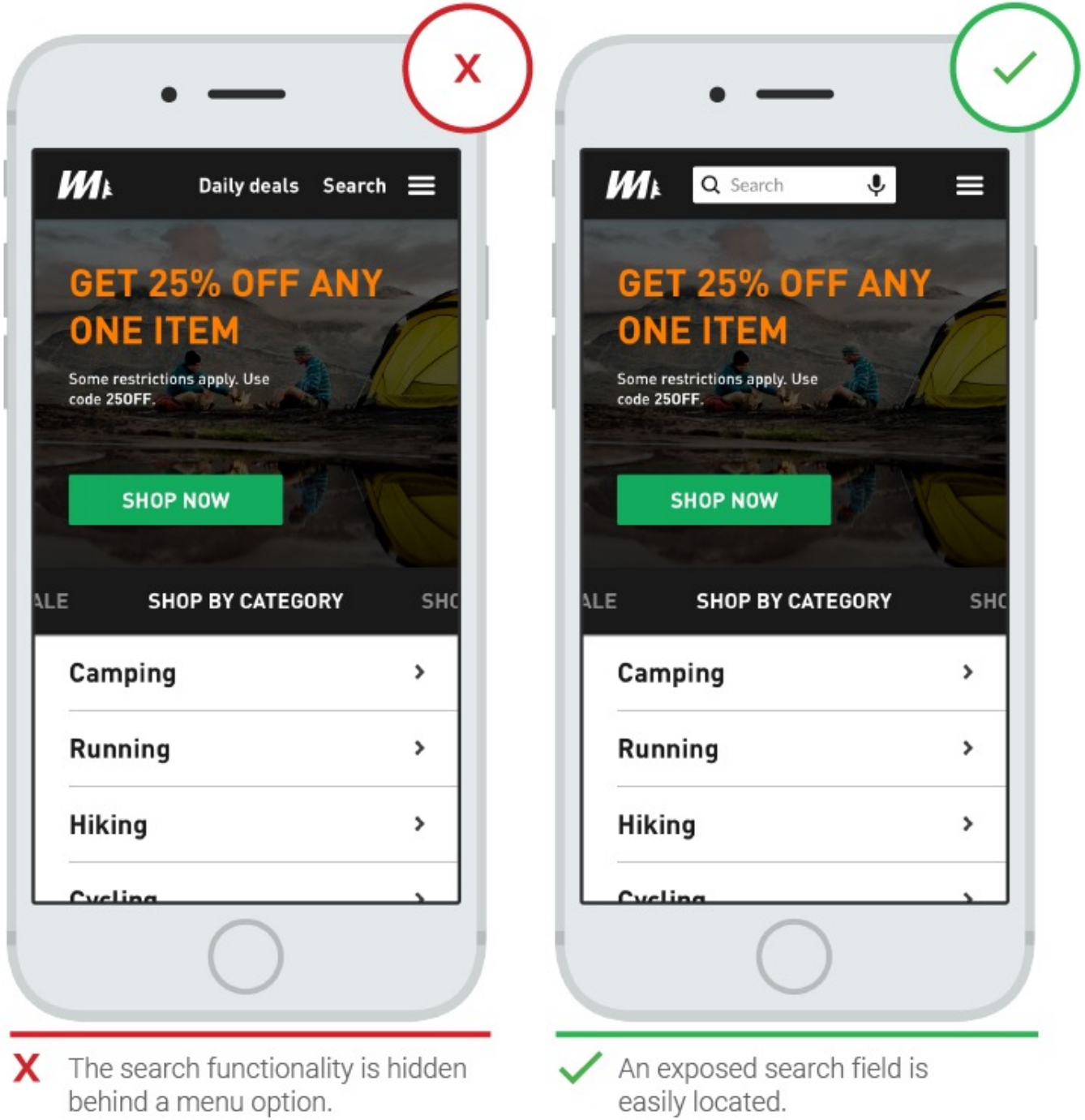


Security

- Handle sensitive information.
  - Credit card number
  - Identification number
- Handle data traffic.
  - Personal Data Protection Act (PDPA)
- Handle access control.
  - Who has permission on what.

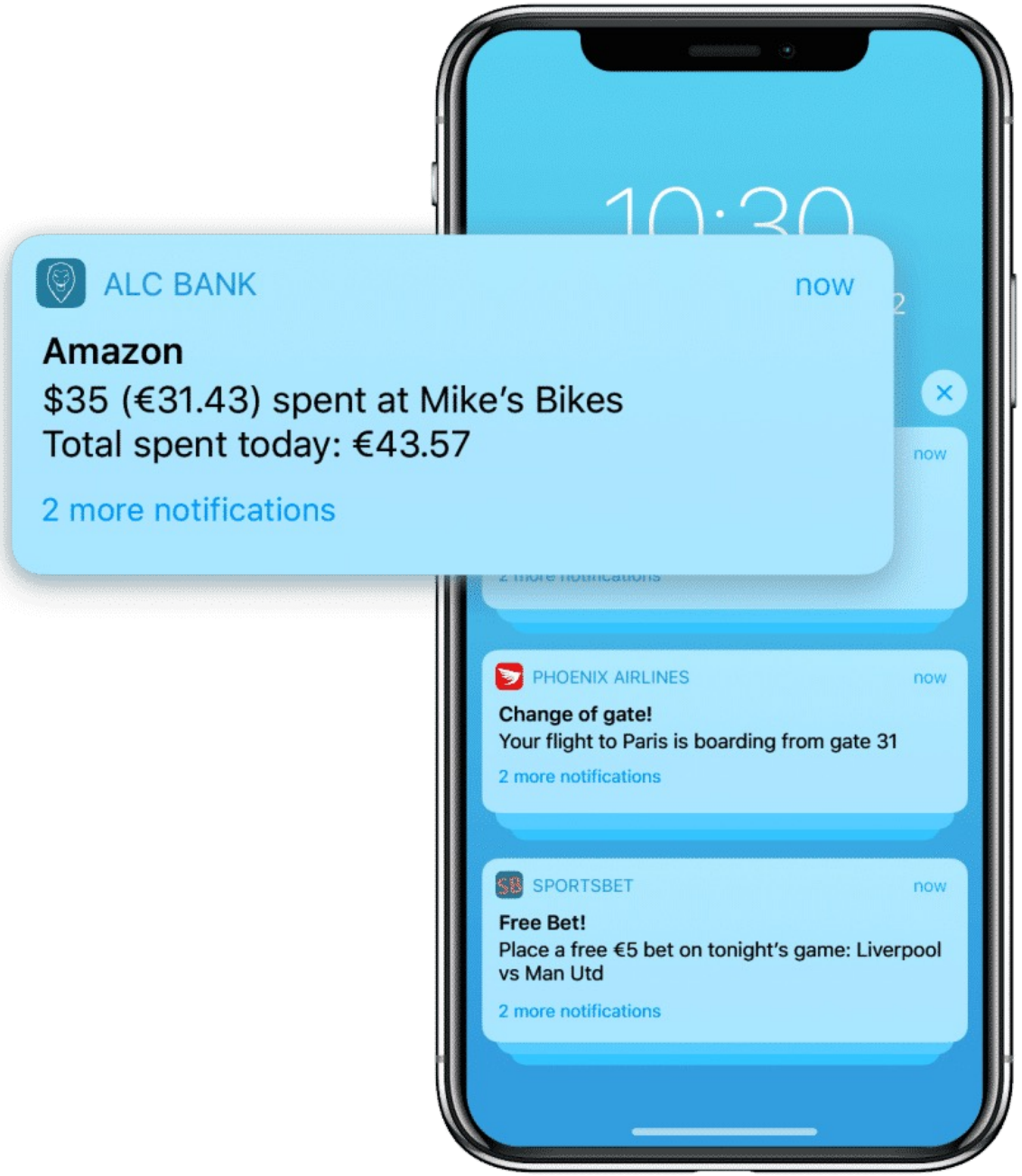
Search

- Should be able to search within App.
  - Autocompletion is very useful.
  - Reduce menu clutter.
- Important for business and social.
  - Search for products.
  - Search for friends.
- Not so much for games
  - Maybe search for heroes.



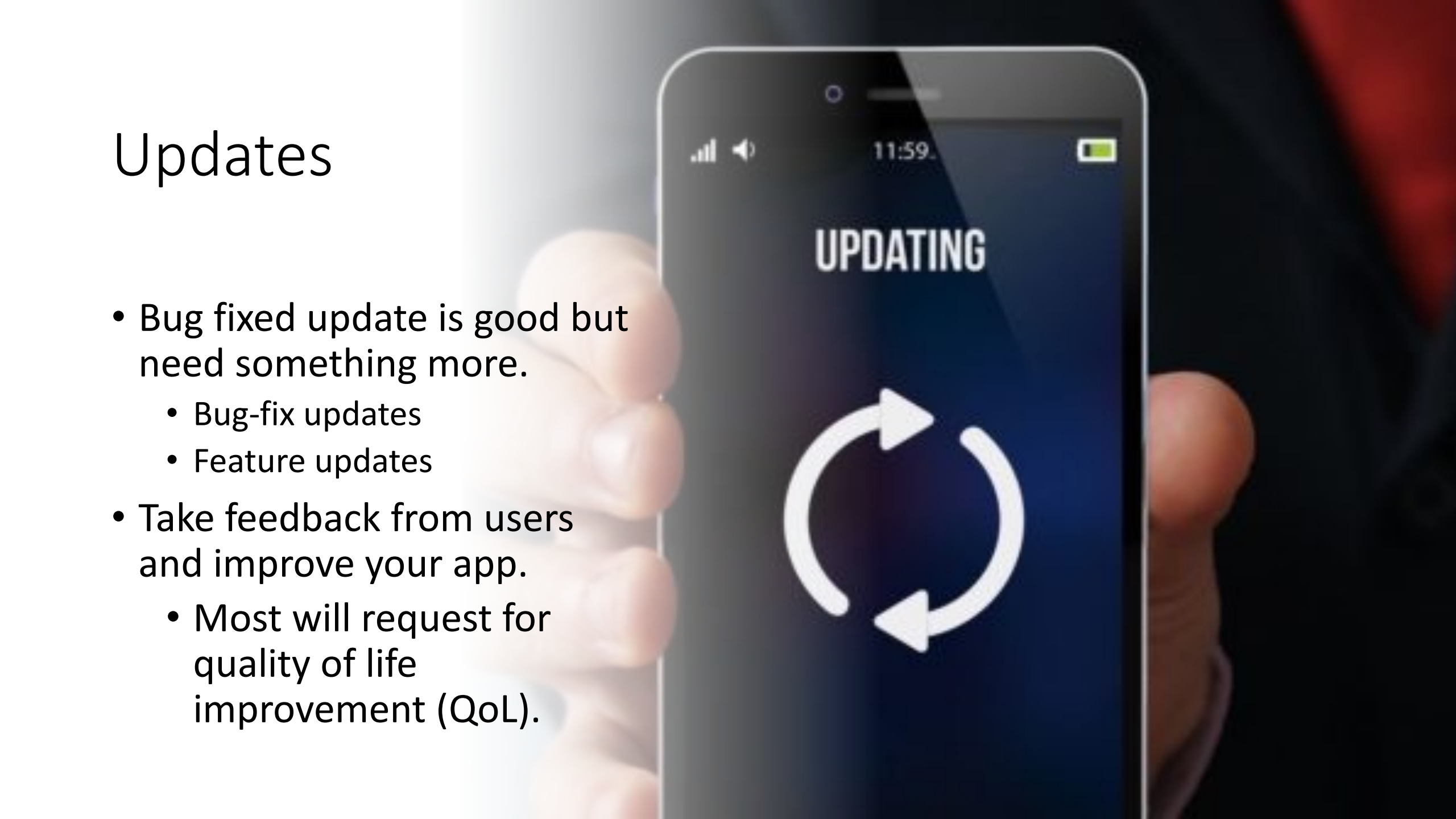
Push Notification

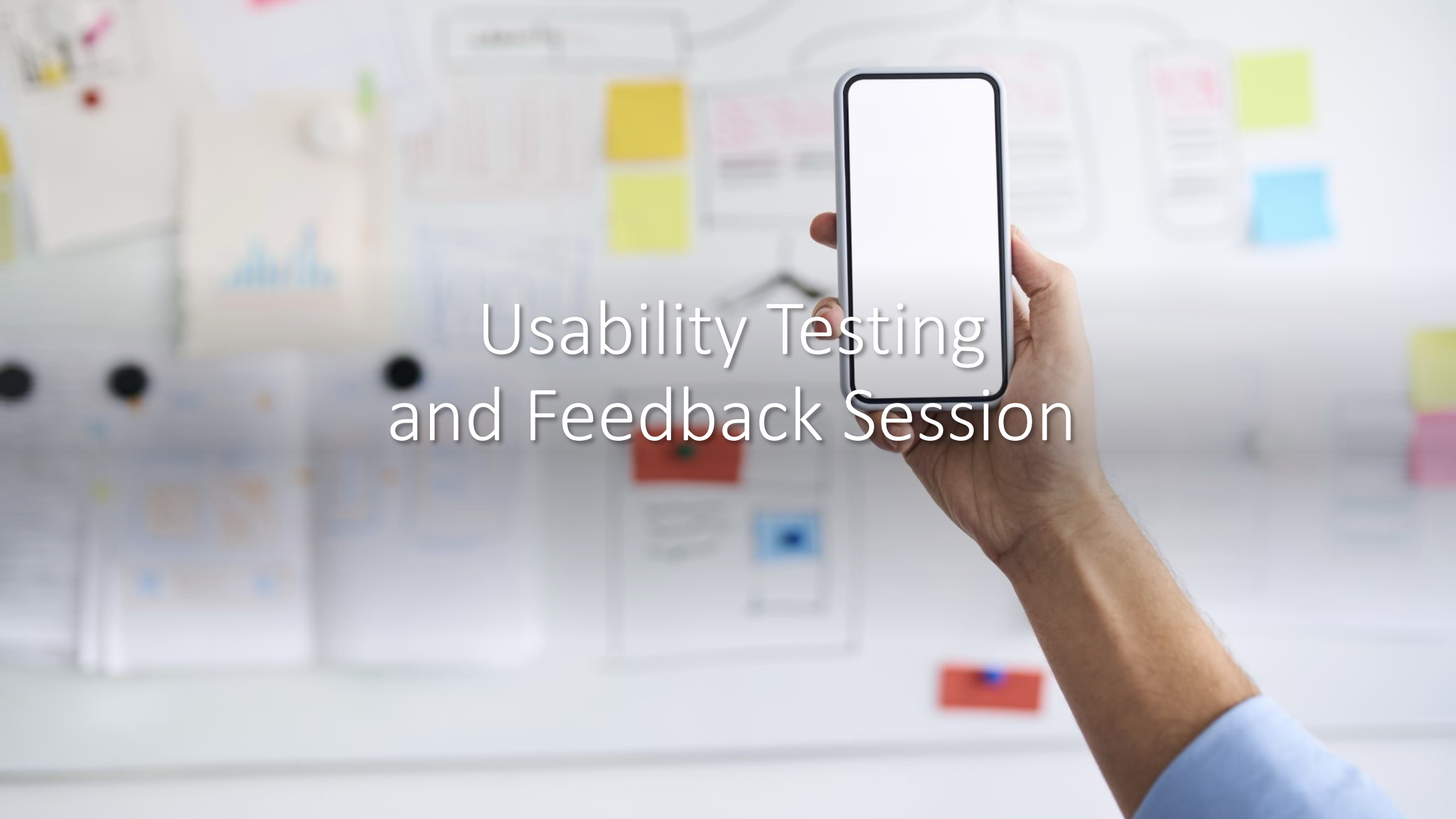
- Always assume that your clients don’t open your app all the time.
- Push notification lets your customer return to app.
  - Full stamina in games.
  - Receive important messages.
  - Need quick actions.
- Must be customizeable (in phone or app setting).



Updates

- Bug fixed update is good but need something more.
  - Bug-fix updates
  - Feature updates
- Take feedback from users and improve your app.
  - Most will request for quality of life improvement (QoL).





Usability Testing and Feedback Session

Active Activity

- Divide into 10 Groups
- One group will get one application to test
- 10 min to test the app. Focus on identifying any usability issues, such as navigation challenges, unclear instructions, or issues with layout and design.
- 5 min to list feedback (both positive and negative)
- 3 min per group to share feedback with the class.