



SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution



Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

19ITE310 - MOBILE APPLICATION DEVELOPMENT

III YEAR - VI SEM

UNIT 5 - TESTING MOBILE APPS AND TAKING APPS TO MARKET



White Box vs. Black Box Testing

White Box Testing

- Tests internal structures and code
- Code coverage analysis
- Example: JUnit for Android

Black Box Testing

- Tests functionality without internal knowledge
- Functional, UI, performance testing
- Example: Robotium, MonkeyTalk

White box focuses on internal code, while black box focuses on external functionality.



Unit for Android: Unit Testing Deep Dive



1

Setup

Use JUnit framework for Android.

2

Annotations

Use @Test, @Before, @After annotations.

3

Writing Tests

Test individual components/methods.

4

Mocking

Use Mockito for dependencies.

JUnit is a powerful framework for unit testing. Follow best practices like TDD and CI.





Automated Testing: Robotium and MonkeyTalk

Robotium

- Android UI testing framework
- Simulate user interactions
- Simple API
- Robust UI element identification

MonkeyTalk

- Cross-platform test automation
- Scripting tests
- Data-driven testing
- Reporting and analysis

Automated testing tools like Robotium and MonkeyTalk can significantly improve efficiency.



Mobile App Debugging Techniques



Android Studio Debugger

Use breakpoints and stepping.



Logging and Monitoring

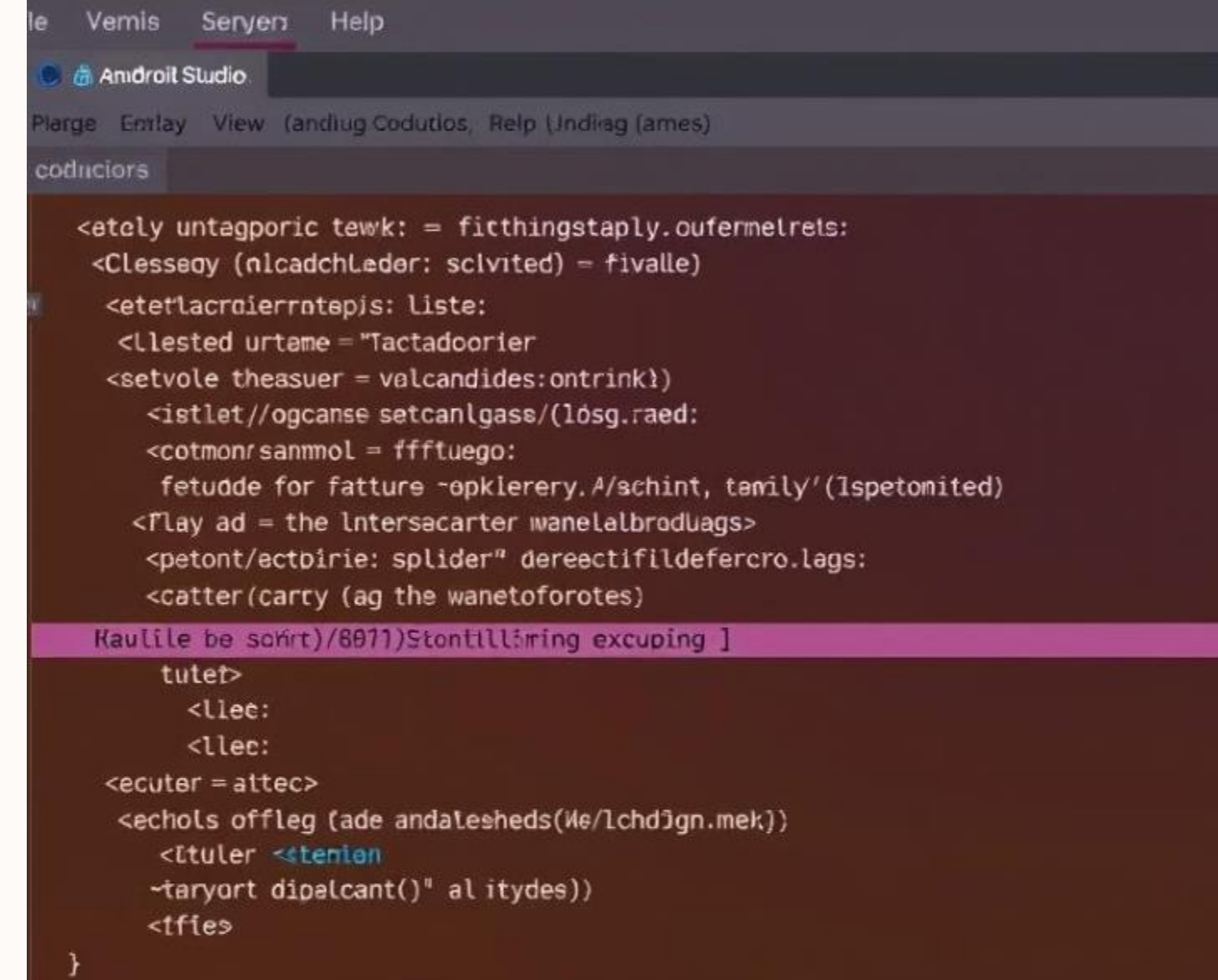
Use Logcat, Firebase Crashlytics.



Remote Debugging

Debug apps on physical devices.

Effective debugging is essential for identifying and resolving issues quickly.





Versioning Mobile Apps: Best Practices

Semantic Versioning

Use MAJOR.MINOR.PATCH.

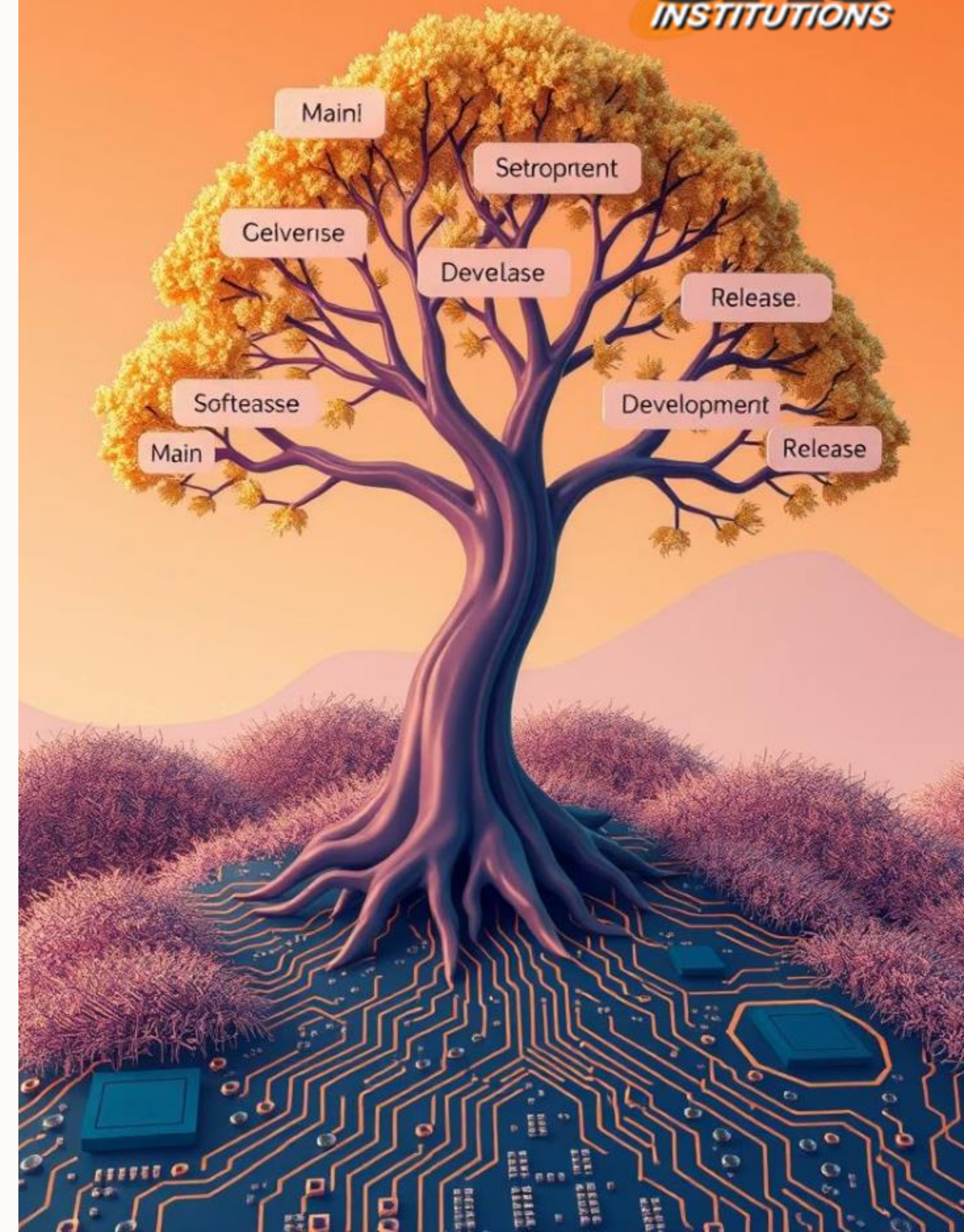
Build Variants

Debug, release, staging.

Git Branching

Gitflow, GitHub Flow.

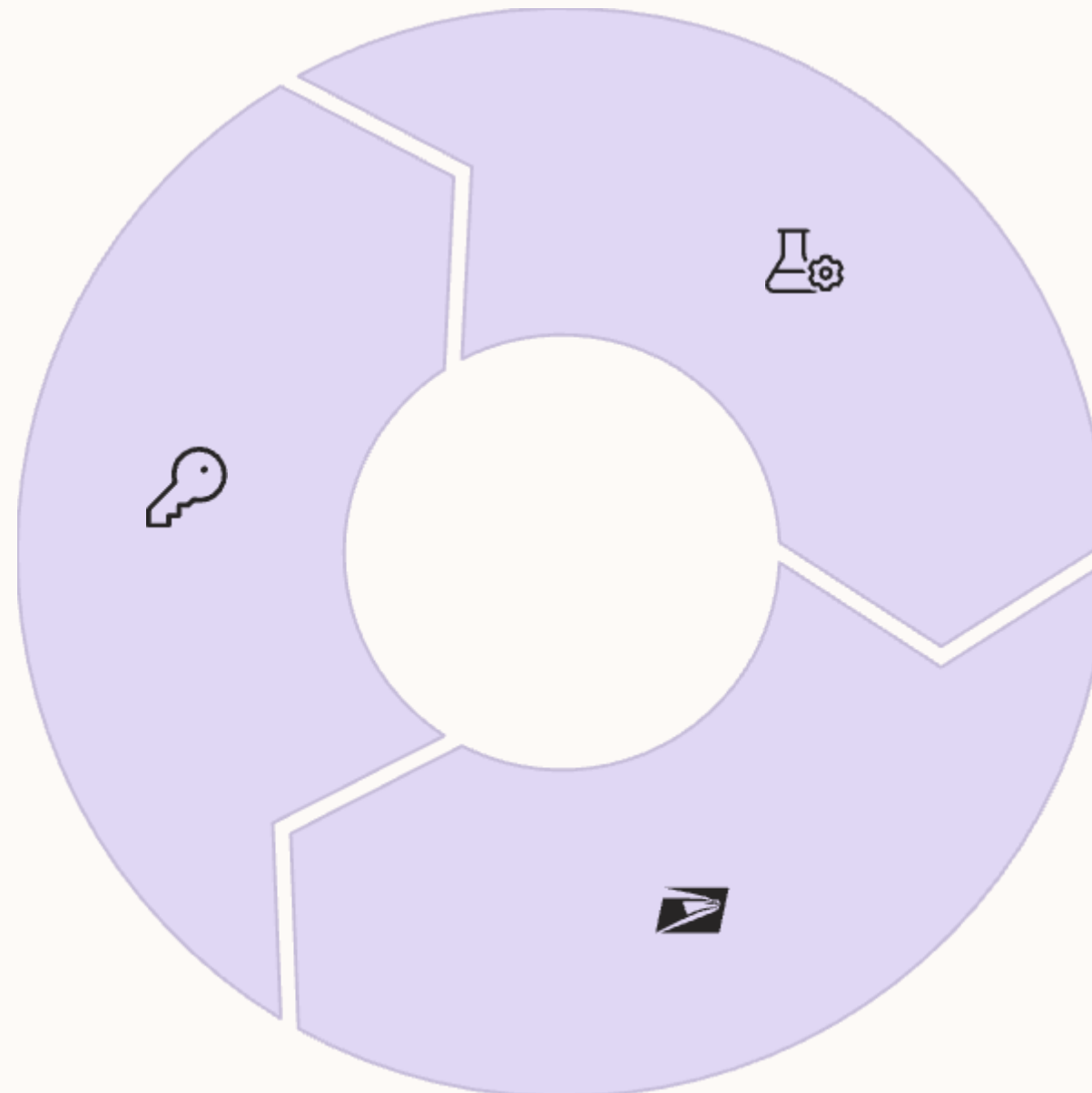
Proper versioning helps manage updates and maintain compatibility. Use Gradle for versioning.





Signing and Packaging Mobile Apps

Keystores
Create private keys.



Signing Configurations

Release vs. debug builds.

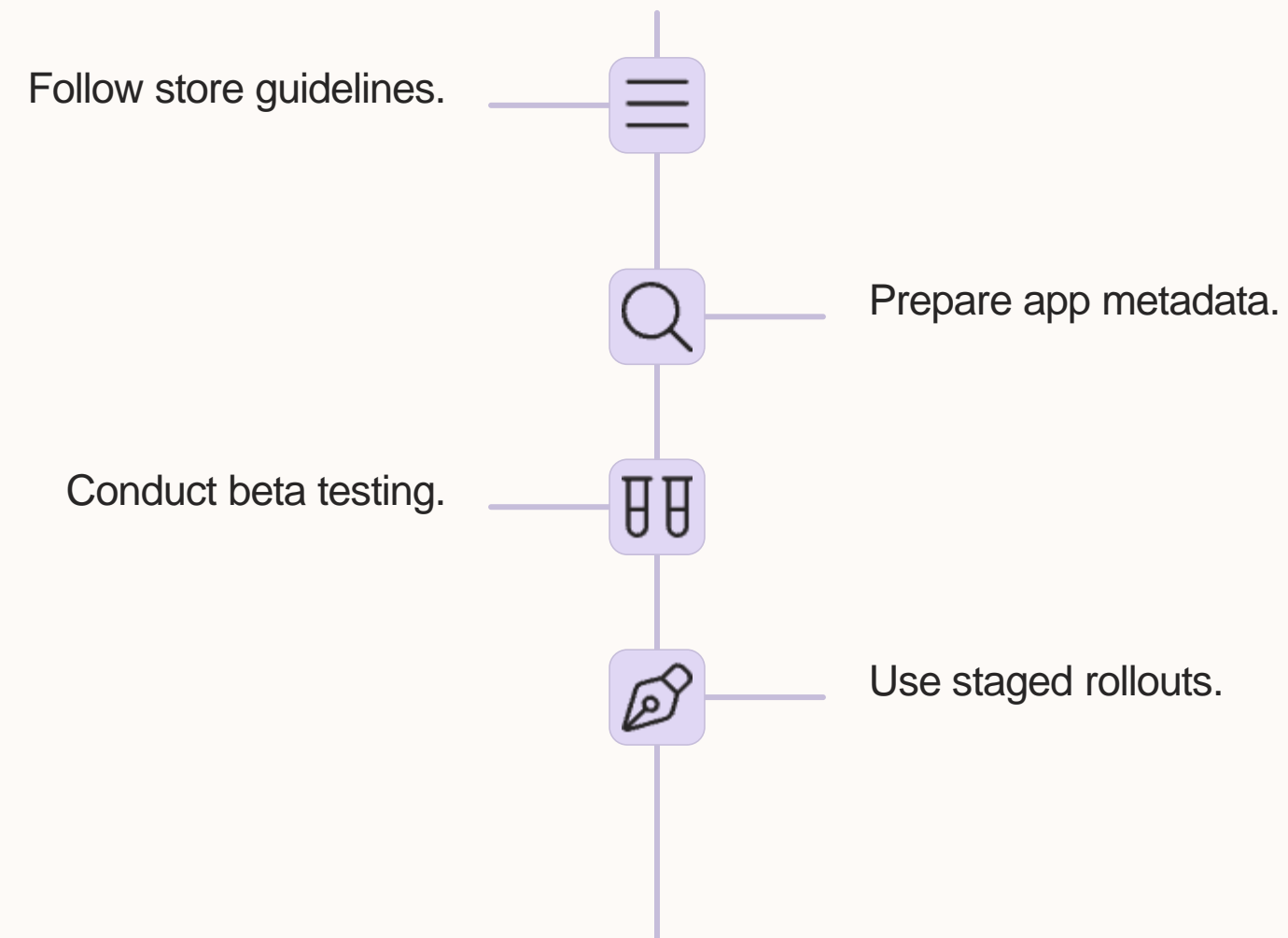
APK/AAB Formats

Use correct formats.

Signing and packaging are critical for app security and distribution. Consider code obfuscation.



Distributing Apps on Marketplaces



Careful distribution ensures a smooth launch. Monitor app performance and user feedback.