

## **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** 

Broadcast & APIs/ Mobile Application Development/Vikneshkumar.D/IT/SNSCT







# White Box vs. Black Box Testing

#### White Box Testing

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

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

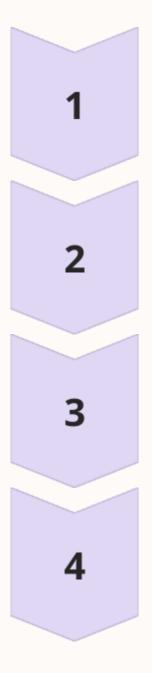
#### **Black Box Testing**

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









#### Setup

Use JUnit framework for Android.

#### Annotations

Use @Test, @Before, @After annotations.

#### Writing Tests

Test individual components/methods.

#### Mocking

Use Mockito for dependencies.

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





- Constitution of the contraction of the second second
- And States in the second states in the second states of the second states in the second state
- THE R. P. LEWIS CO., LANSING, MICH.
- the second se

The second se

- the state of the second state and the second state of the second s
- A STATE OF A STATE OF
- The second second
- and the state of t
- and an other states of the second states of the second states of the second states of the second states of the

- CONTRACTOR OF THE OWNER.
- the law and set of the set of the set of the set



# Automated Testing: Robotium and MonkeyTalk

#### Robotium

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

#### **MonkeyTalk**

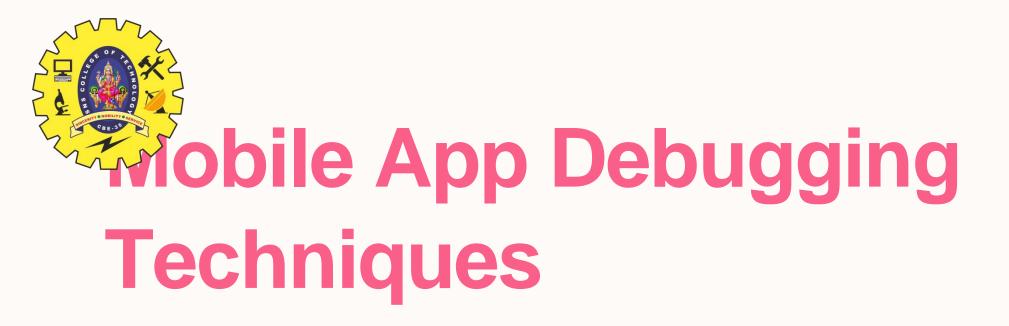
- Scripting tests
- Data-driven testing
- Reporting and analysis

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



Cross-platform test automation







#### **Android Studio** Debugger



#### **Remote Debugging**

Debug apps on physical devices.

Use breakpoints and stepping.



Use Logcat, Firebase Crashlytics.

Effective debugging is essential for identifying and resolving issues quickly.



le Vemis Seryen Help
🖲 🙆 Anidroit Studio
Plarge Enrlay View (andlug Codutios, Relp (Indiag (ames)
coduciors
<pre><etcly (nlcadchleder:="" <clesseqy="" sclvited)="fivalle)&lt;/pre" tewk:="fitthingstaply.oufermetrets:" untagporic=""></etcly></pre>
<pre><etetlacroierrntepis: liste:<="" pre=""></etetlacroierrntepis:></pre>
<pre><llested dereectifildefercro.lags:<="" pre="" urteme="Tactadoorier &lt;/pre&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;pre&gt;&lt;setvole theasuer = valcandides:ontrinkl) &lt;istlet//ogcanse setcanlgass/(losg.raed:&lt;/pre&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;&lt;pre&gt;&lt;cotmonr sanmol = ffftuego:&lt;/pre&gt;&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;fetudde for fatture -opklerery. A/schint, tawily'(lspetonited)&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;&lt;flay ad = the intersacarter wanelalbroduags&gt;&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;&lt;pre&gt;&lt;petont/ectpirie: splider"></llested></pre>
<catter(carry (ag="" th="" the="" wanetoforotes)<=""></catter(carry>
Kaulile be somrt)/6071)Stontllläming excuping ]
tute?>
<llee:< th=""></llee:<>
<li><li><li><li><li><li><li><li><li><li></li></li></li></li></li></li></li></li></li></li>
<ecuter =="" attec=""></ecuter>
<echols (ade="" andalesheds(we="" lchd)gn.mek))<="" offleg="" th=""></echols>
<ituler <tenion<br="">-taryort dipalcant()" al itydes))</ituler>
<tfies< th=""></tfies<>
}

# ersioning Mobile Apps: Best Practices

#### **Semantic Versioning**

Use MAJOR.MINOR.PATCH.

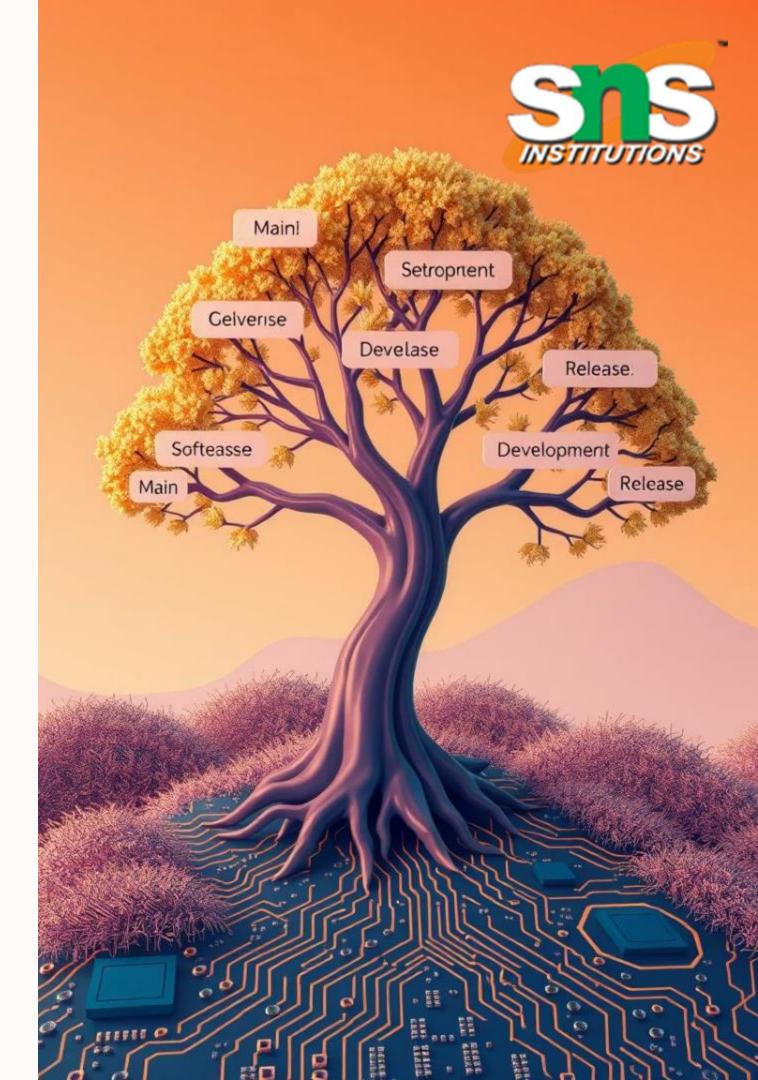
#### **Build Variants**

Debug, release, staging.

#### **Git Branching**

Gitflow, GitHub Flow.

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





 $\sum$ 



Create private keys.

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



#### **Signing Configurations**

Release vs. debug builds.

<u>L</u>ø

 $\geqslant$ 

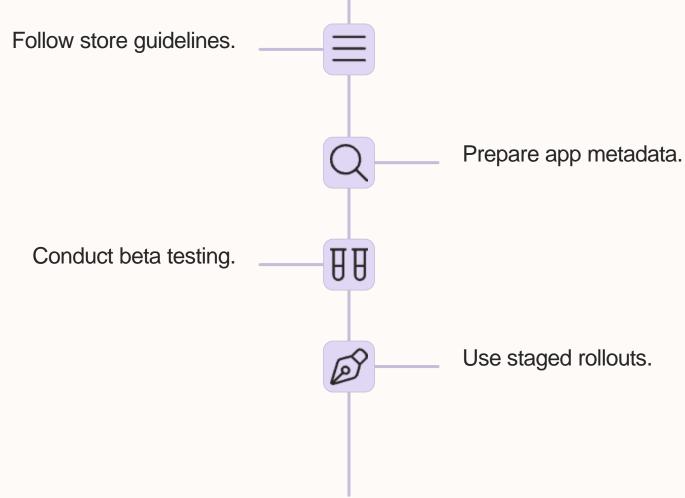
#### **APK/AAB Formats**

Use correct formats.





## **Distributing Apps on Marketplaces**



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

