Mobile App[n] Types

① → Access by installing, costly to develop.
Subject

Native ② → Separate codebase for each platform.

→ Access by app store, one code base, multiple platforms → android → Separate Builds are generated from single code base
→ ios
→ SB
→ WindPase → Access by URL, 1 codebase, 1 URL

	Native	Hybrid	Mobile Web
① Performance	Fast	sr	sr
② Camera Integ[n], calender, Contacts	③ Can Access device native features eg. Camera, GPS contacts.	Possible	Limited (or No)
③ Languages: objective ←→ ios Java → Android			HTML 5, CSS, Javascript
④ Developm[nt] updates/New Version	④ updates by installing new version	faster	fastest
⑤ Security	High	High	weaker

Native:

→ install[n] thro[g] App Store
→ can use device native ft[s] Camera, contacts, GPS etc.
→ costly to develop.
→ updates are not easy
↳ dev[t] for specific Plateform

→ accessed through URL
→ can not use few device features
→ less costly to develop.
→ for multiple platform-

Can not Buy All devices
↓↓ used to save cost
↓↓
emulator & Sim[r] are not the true repres[n] of devices.

Website → Web App[n]
Desk Top App[n] 4",5,7,9,10"
Mobile App[n] → more complex to test

① multiple screensizes & OS combin[n]
② Landscap & postrait orient[n] : Orient[n] Tg
③ N/W Tg: 2G, 3G, 4G, WIFI
   But In desktop only broadband or dial-up connection

④ Interrup[n] Tg: calls/SMS/push notific[n]
   lock/unlock, USB insert/Remove
   (Alarm, Reminder)

system Back key scenario: brings calling page
click to call feature for phone no.
Add to phone contact[r]
GPS/Direction features

⑤ Security & Privacy concern
   many apps can access device phonebook, pictures, videos, these are user personal data
   Any defect around the misuse of data
   → unintentional access

⑥ Mobile app has feature of accessing camera & phone book of device.

Android App vs iOS App

→ Toast Notific^n feature in android only → Build/ Install^n process → sharept
→ Camera disable for a part^r app in iOS phone. → HockyApp
→ .ipa/.apk

less Secure → Android vs iOS → more Secure

1. open source S/W developed by Google. | proprietry S/W (closed source S/w) owned by Apple | 5. Android has more tathy challenges than iOS
   | | ↳ device config^n (size, Manfat)
2. ↳ Android can be modified by Developer (Dev^r can try their own customised version of android) | ↳ can be modified & readressed only by Apple | ↳ OS various
   | | ↳ Security: info security, code security
   | ↳ iOS is designed to run only on Apple hardware (iphones)
3. Android support diff^t types of phones HTc, Samsung etc | ↳ Restricted Access: apps can be download only from Apple's app store.
4. android apps can be downloaded from Google Play store or from websites |

Testing that can be performed on Mobile are:—

1) Usability Tg

2) Compatibility Tg (across different ① devices ② Browsers ③ OS ④ Screen Size)

3) Install/Uninstall Tg — Install^n→ should be smooth, should not take much time.
   Uninstall^n→ should remove the app related file from device.
   if app stores picture & videos then these media file should not be removed.
   e.g. whatsapp, facebook.

4) Functionality Tg

5) Performance Tg (Load/Stress Tg)

6) Security Tg ⟳ session Timeout
   Fortress Authentic^n when login from a new device.
   No sensitive data is stored on the device by app.

7) Memory Leakage → Its OS default behavior that if an app takes excessive consumes
   memory OS terminates the app & user experience will not be smoother.

8) Interruption Testing

9) N/W based Tg (2G 3G 4G 5G WiFi)

10) Orient^n Tg

Responsive Website: → codebase is same for Mobile & Desktop
   ↳ app^n adjust/fit its content as per the screen size of System/device.

Mobile Web: → separate code base specific to mobile.

Non-responsive Website → separate code base specific to Desktop.

For mobile pt of view all UI related test cases should be checked in all diff^t screen size devices

Total TC = UI related TCs * 3
Note3A → HAT → Nexus5
5", 7" & 9/10

→ Highly Recommeded.

Usability Testing : ⇨ helps to improve end user satisfaction
⇨ saves a product from failure

Best Way to understand how real Users experience your app".

{ How easy to use
{ How easy to learn

Effectiveness
→ Easy to use & learn
→ Is content, color, Icons, Images
         Used are pleasing

S/W
Failure reasons :

→ where to click next ?
→ which page to go ?
→ which Icon represents what ?
✓→ Error msg are not consistant or
     effectively displayed.
✓→ Session time out not sufficient

Efficiency :
         Navigations: eg. drawer
                          Menu
✓→ Scroll bar should not be used
                          frequently
→ Uniformity in th format of
                     screen / webpag
✓→ Provision to search within
                     your app"

5 people uncover 80%.
         can
↳ usability Problems
(experts
+
Non exp)

Accuracy
→ No outdated data
→ No broken links   rates etc

Best Practices : Usability Tg.
→
     Start testing during early stage of design & devt
     you can conduct V.Tg on your comp-etitors product before you begin devt
     This will help you determine usability standards for your Target Audience
                                                              Customer

↳ It is costly, Need lot of expert resources.
                                        → How many errors
                                            user can make
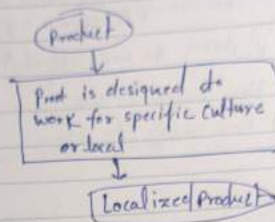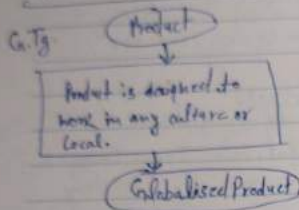   Errors: How much time
                                        app" is taking to
                                        perform a task
Adv
     Higher customer satisfaction & Retention
     Repeat Business

Globaliz^n vs Localiz^n Tg :

G.Tg

(Product)
↓
[Product is designed to work in any culture or local.]
↓
(Globalised Product)

(Product)
↓
[Prod is designed to work for specific culture or local]
↓
[Localized Product]

G.Tg is done to ensure that app^n can function in any culture or locale (language, territory & codepage)

checks proper functionality of the product.

L.Tg process of checking the localized version of product for that particular culture or locale settings

The area affected by L.Tg are UI & content

Load vs Stress Tg :-

If you are testing normal expected load, this is load testing. (You know system will be used by up to 100 users at a time)

But when you want to determine, How the system behaves under extreme load & when it breaks, this is stress testing.

Why RTM : ReqID vs TestCases → Business Rules, Error Messages

→ To ensure all the requirements are covered in our TC.

→ in case of any change in Req^t we can trace the impacted testcases

Deliverables :

1. Test Plan
2. TestCases
3. RTM, TestData
4. Status Report (Daily/weekly)
5. SIT Exit Report

Team Handling: Freshers, New Team-memb,
KT, explain the way/Process
you work.

Roles & Responsibilities

As a SPOC/Module Lead for Project → offshore co-ordinator

(Assign Task
  Review Task
  Attend variance call
  Send Status Daily/weekly
  Req. change Mangt

(1) → assigning task to team members (2) → Monitoring & Reviewing that work done by team mem

→ If they ~~are~~ are in design phase
   what Test Case/Test scenario they hv written?
   what test data they hv prepared, suggesting what cud be other possible scenario.

→ If in execut^n phase, All the team memb^rs assign variances
   to me then I review those ~~~~ whether all the
   details are proper or not then assign to Onsite Co-ordinator

(3) → Attending variance call with onsite Co-ord^n, Dev^r, BA & other
   Stakeholders (Env't Team, Test Data Team, Back end Team)
   Onsite Co-ordinator explains each defect one by one & if further clarif^n
   required I used to explain. Also used take status of some blocker defects
   After I assigned defect to OC^r   from Dev^r
   ⮑ Onsite co-ord used to assigne it to appropriate team

(4) sending Daily/weekly status Report   Dev Team
                                                       Env't Team
~~(5)~~ → Management of changes in requirement.   Back End Team
          ~~updatin~~
          ⮑ Analysis impact & ask team m^r to update TCs accordingly.

Being a Lead
ensure all the deliverables are delivered on time
To be aware of other projects running in || to your proj { for device plang
                                                          { Any related Issues
                                                            /dependency/
Ensure all team-member are given KT of app^n &
                        are well aware of process.
Motivate the team member by appreciating them, encouraging them learn
~~Keep the~~   for their effort
       new skills.
Create an emotion/friendly bondiy with them.
                                  go for outing
                                  go for lunch/snack/dinner together
                          Let them feel included & valued in the team

Mobile Testing options → on real devices → can not access camera,
→ Emulators → cost effective but with limitations → not have characteristic of
→ Real devices kept on cloud (device forms given on rent real device
(Remote Devices) by company like Amazon
↳ To get cost effective & better coverage sol^n sauce labs)
you can use [real + Emul^{rs}]
a combi^n of [Devices] → cost-effective, so take advantage
of it

→ get the account
created, Pay on
time basis usage.

→ Devices: The biggest Mobile Tg challeng
→ different version of OS.
→ diff^t screen sizes Read device: Most costly
→ Diff^t Manufacturers. Emulators : less costly.
Remote Devices: less costly.

N/W based Tg
Switching b/n N/W cond^{ns}

Frequency band of GSM : Multiple (850/900/1800 MHz)
CDMA: single 850 MHz

Wireless Technology Dominant worldwide. Global Market share
except US. 80% share globally.
GSM: ⇒ Global System for Mobile → 80%
CDMA → Code Division Multiple Access 20%
↓
Dominant in U.S.

GPRS |Edge |2G |3G |4G |LTE is A path followed
→ Long Term Evalution → to achieve
↓ 4G speed
A standard for high speed wireless
comm^n for mobile phones

Subject Test case Types in TGS - CLBC

Date

4 categories
- UI & content
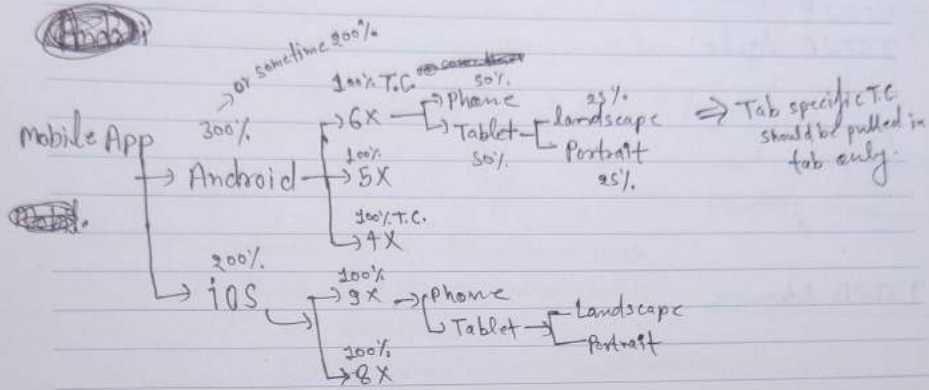- Error Messages
- Functional
- End to End Flows (Happyflow)

Testing Exec^n
Phases

1) DIT → (Design Integr^n Tg (Dev^t Independ Tg)
2) SIT → System Integrat^n Testg
   (SIT1 / SIT2 cycles)
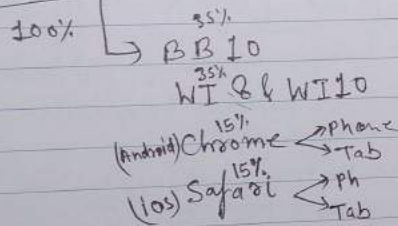   SIT - CM → alem Mangemt
3) UAT → Uses A/C Tg.

Execut" Plan & Device/Plateform Coverage

Say original T.C. = 100

Android

Mobile App

or sometime 200%

300% → Android ──┬──→ 6X ── 100% T.C. ─ Count → Phone 50%
        Tablet ─ Landscape 25%
        Portrait 25%
50% ⇒ Tab specific T.C. should be pulled in tab only.

100% → 5X

100% T.C. → 4X

200% → iOS ──┬──→ 9X → Phone
        Tablet → Landscape / Portrait

100% → 8X

Mobile Web

100% ──→ 35% BB 10
       35% WI 8 & WI 10
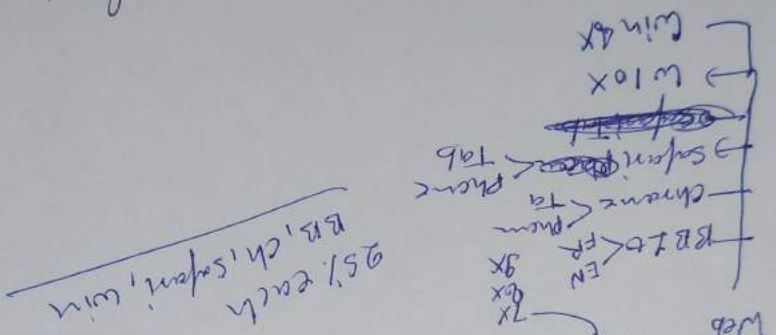   (Android) Chrome 15% < Phone / Tab
       (iOS) Safari 15% < Ph / Tab

⇒ T.C. Coverage on diff" plateform can be manipulated depending on length of time lines & No. of resources
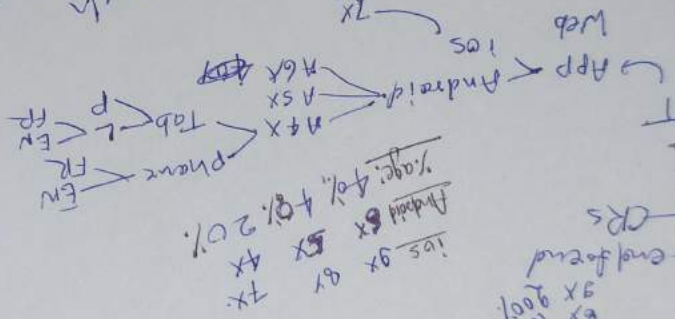
⇒ Try to get optimum coverage.

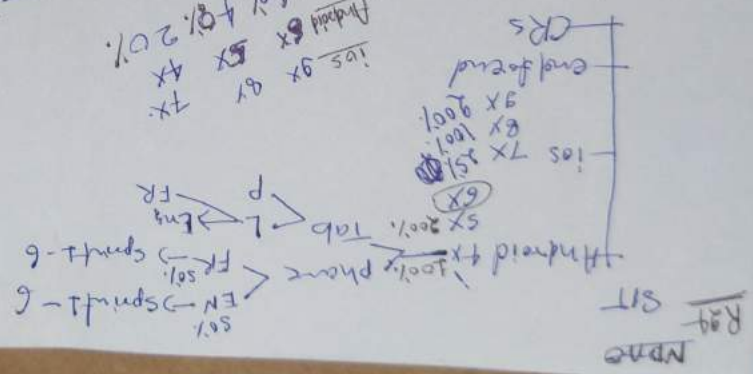SIT CM : → Generally we put failed T.C. & other critical Testcases.
← open plan story

w in 4x
w 10x
⑤ safari → Tab
chrome → Ta
RBIt < FR → Pheme
EN 9x
8x
7x
95% each   RB, Ch, safari, win

APP < Android → Tab   iOS
Web   ios
SIT   phone → FR → EN
FR
Cee Marvel   A 4x → Tab → L → EN
A 5x   P   FR
%age. 40%, 40%, 20%.   A 6x
Android 6x 5x 4x
ios 9x 8x 7x

CRs
employeed
ios 7x 85%
6x 100%.
9x 900%.
SITT

public class
SessionPersonal
NGR Features
whenever user
changes device or
Resolution, he
has to Sign
Again
} offline
} Omniture

NDno
Req   SIT
Android 4x 100% phone < EN → Spindl-6
95%.   FR → Spindl-6
5x 80%. Tab → L → Eng
P   FR