# Support App Center test on Test Site

- Goal
- Background
- Open Questions
- Test-executor difference between App Center and Test Site
- Make plan for app center
- Areas that need to be adjusted:
  - Choose conditions before make plan
    - test-device-service adjustments:
  - make plan async create test run
    - test-case-service adjustments:
  - make plan After Testing
    - test-executor adjustments

| Version | ChangeLog | Date |
| --- | --- | --- |
| 1.0.0 | 1. Additions: Build the Executor Project with Maven, Add Parent and Sub-Project Level `pom.xml` Configurations | 2023.10.07 |
| 1.0.0 | Design : <br><br> 1. Update: Differences Between Running `test-executor` on App Center and Test Site <br><br> 2. Update API Definition: Pull All `executor_case` by `test_platform` & `device_model` | 2023.09.28 |

## Goal

1. Create a app center test plan on test-site and then start a test runner on app center.  ( Starting a test run - Visual Studio App Center )

## Background

1. App center test is available only on UAT
2. Test cases type: upgrade and regular test case
3. Executor is build using maven
4. Run test on app center : Starting a test run - Visual Studio App Center

## Open Questions

1. How to know which app to test in the code
   a. App center directly installs the app to the device, and the script locates and operates the app through bundleId
2. How to know which device to use for testing in the code?
   a. Assigned by app center, device information can be obtained directly from system environment variables.
3. If there is a newly released app during the test, do you need to use the latest app to install and test the remaining cases?
   a. Platform & app center ignore app updates. We do not need to retest
4. If the test script changes during the test, do we need to stop the app center test and initiate a new app center test for the remaining cases?
   a. Platform & app center ignore executor updates. We do not need to retest
5. Final report generation

a. Change to manual or job to trigger the final test (currently manually controlled)

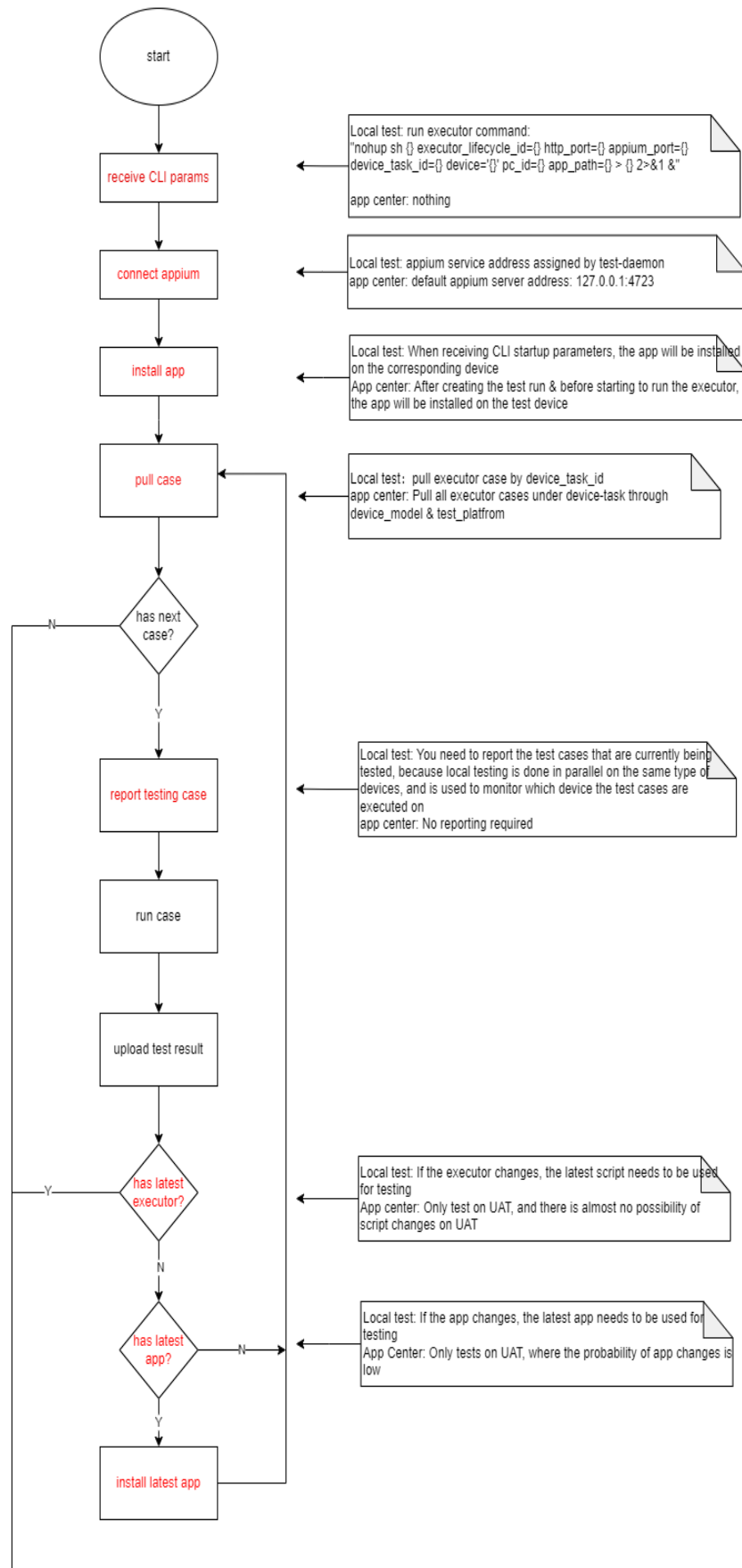6. Do you need to record which version and revision number of the app used?

a. Yes

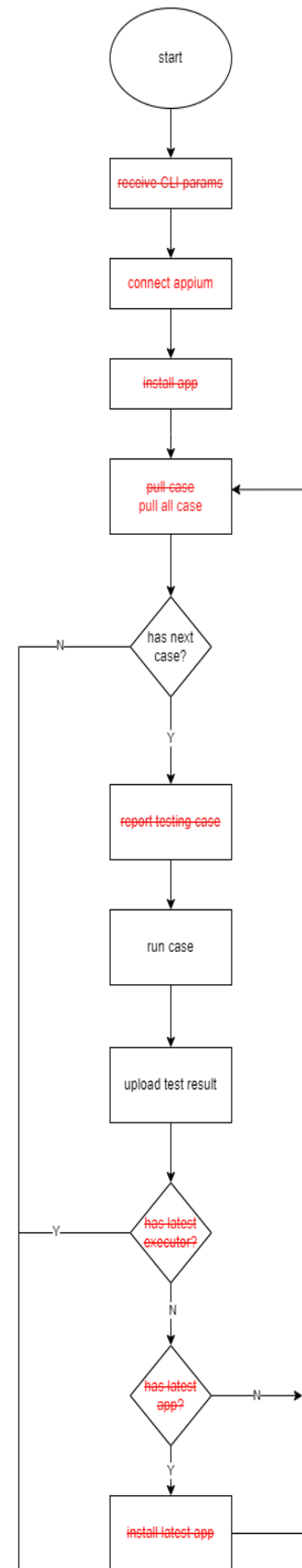## Test-executor difference between App Center and Test Site

**Difference:**

1. The objects that operate the executor are inconsistent:

   a. Local test: operated by the locally installed test-daemon service. Test-executor needs to receive the CLI parameters passed by test-daemon to know the test tasks to be executed.

   b. App center: operated by the app center. After running, the test tasks to be executed are known by calling the find device task API.

2. The methods of obtaining test cases are inconsistent:

   a. Local test: call the pull case API `ExecutorCaseAJAXWebService.nextForTest()` through polling to obtain the test cases to be executed

   b. App center: obtain device_task through device_model&test_platform type, including all test cases to be run and the installation path of the app (required when doing upgrade test)

3. The methods of obtaining the app installation path are inconsistent:

   a. Local test: provided when running executor and completed by calling the `AppPackageWebService.getLatestPath()` API of test-daemon

   b. App center: completed by calling the find device task API of test-api.

4. The methods of installing apps are inconsistent:

   a. Local test: installation performed by executor code.

   b. App center: installed to the device by app center.

5. Inconsistent ways to obtain devices:

   a. Local test: obtained by parsing CLI parameters and passed to test-executor by test-daemon

   b. App center: obtained by reading the system environment variables of the app center server

6. Inconsistent ways to end the executor life cycle:

   a. Local test: executor calls the kill executor API on test-daemon to end the life cycle

   b. App center: After the task is executed, the executor needs to actively exit the program.

7. Inconsistent addresses for connecting to appium server:

   a. Local test: 127.0.01:xxxx, port allocated by test-daemon
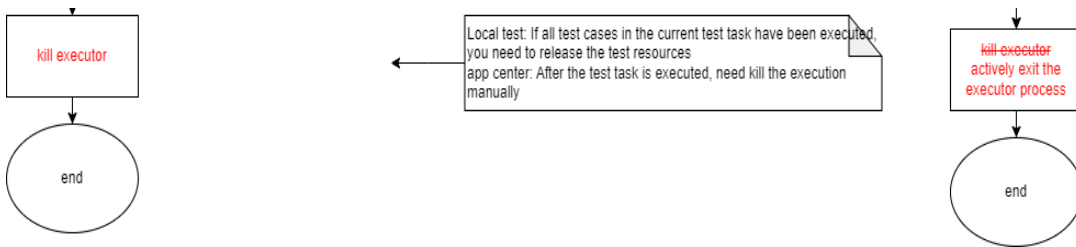
   b. App center: 127.0.0.1:4723, fixed

**Flow Difference：**

The running process of the executor on the local test platform
Red font: the difference when running the test in the app center

Running process of executor on app center

**Left flowchart:**

start

receive CLI params

Local test: run executor command:
"nohup sh {} executor_lifecycle_id={} http_port={} appium_port={} device_task_id={} device='{}' pc_id={} app_path={} > {} 2>&1 &"

app center: nothing

connect appium

Local test: appium service address assigned by test-daemon
app center: default appium server address: 127.0.0.1:4723

install app

Local test: When receiving CLI startup parameters, the app will be installed on the corresponding device
App center: After creating the test run & before starting to run the executor, the app will be installed on the test device

pull case

Local test: pull executor case by device_task_id
app center: Pull all executor cases under device-task through device_model & test_platfrom

has next case?

N

Y

report testing case

Local test: You need to report the test cases that are currently being tested, because local testing is done in parallel on the same type of devices, and is used to monitor which device the test cases are executed on
app center: No reporting required

run case

upload test result

has latest executor?

Y

N

Local test: If the executor changes, the latest script needs to be used for testing
App center: Only test on UAT, and there is almost no possibility of script changes on UAT

has latest app?

N

Y

Local test: If the app changes, the latest app needs to be used for testing
App Center: Only tests on UAT, where the probability of app changes is low

install latest app

**Right flowchart:**

start

receive CLI params

connect appium

install app

pull case
pull all case

has next case?

N

Y

report testing case

run case

upload test result

has latest executor?

Y

N

has latest app?

N

Y

install latest app

kill executor

end

Local test: If all test cases in the current test task have been executed, you need to release the test resources
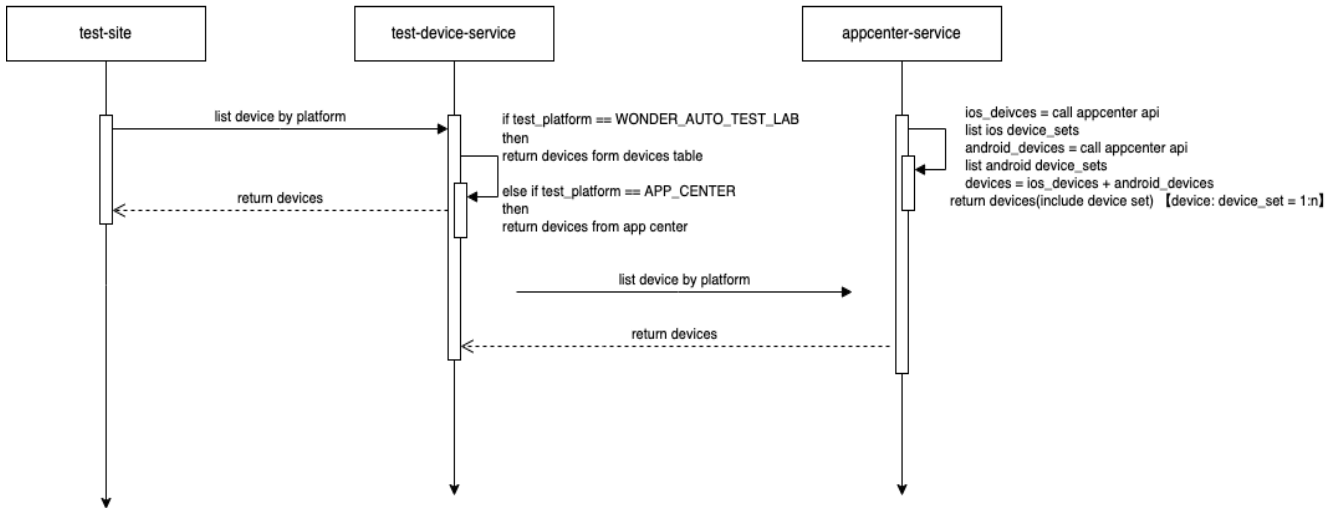app center: After the test task is executed, need kill the execution manually

kill executor
actively exit the executor process

end

# Make plan for app center

1. list device by platfrom:  call REST API of app center

list device from app center



| test-site | test-device-service | appcenter-service |

list device by platform

if test_platform == WONDER_AUTO_TEST_LAB then
return devices form devices table

else if test_platform == APP_CENTER then
return devices from app center

return devices

ios_deivces = call appcenter api
list ios device_sets
android_devices = call appcenter api
list android device_sets
devices = ios_devices + android_devices
return devices(include device set) 【device: device_set = 1:n】

list device by platform

return devices

2. make plan :

make plan for app center

```
test-site-frontend        test-site        test-case-service              appcenter-service              package-library              appcenter  official
```

PlanAJAXWebService.create()

PlanWebService.create()

if test_platform == APP_CENTER
then
    create device set for device task  [device_set : device_set = 1:1 ]
    create tasks  (pending)
    create plan  (pending)
    ...
    start test run on app center (async testing)
    return plan id

return id

return id

TestRunService.create()

(https://learn.microsoft.com/en-us/appcenter/test-cloud/starting-a-test-run)

return test_run

package executor with maven

return

create device set

return id

get download URL od latest app

return download_url

download app

create test run

return id

init hash file list

upload app-file & test-file

trigger test

validate test state

3. pull task



| test-executor | test-api | test-case-service |

ExecutorCaseAJAXWebService.nextAllForTest()    ExecutorCaseWebService.nextAllForTest()

return executor_cases    return executor_cases

# Areas that need to be adjusted:

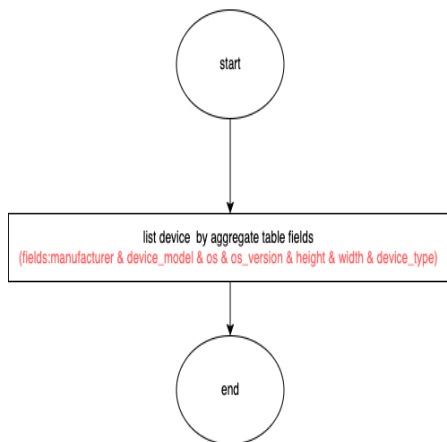## Choose conditions before make plan

### test-site adjustments:

1. The make plan for app center operation should only be available in UAT
2. List device: All devices in all stable device sets, temporary device sets should not be displayed
3. When searching on the interface, you can search based on the device set, for example: ios-first
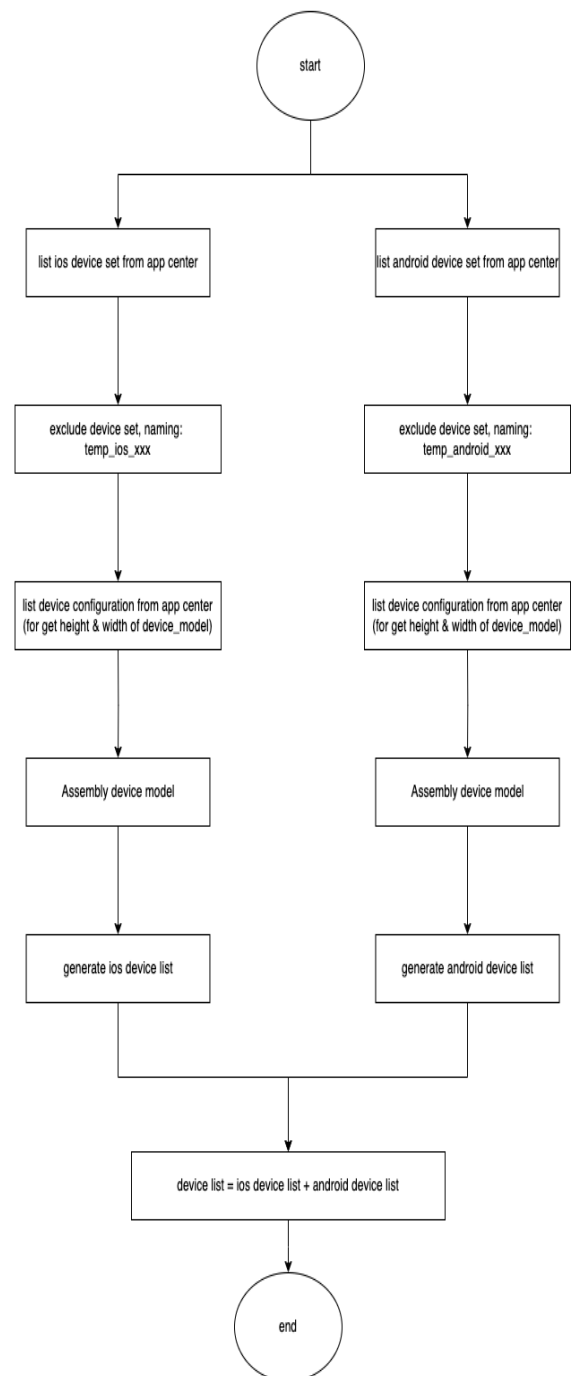
### test-device-service adjustments:

1. modify API for list device: Get devices by platform type
   a. Process:

```
1  1.Because the device set does not contain information such as device resolution, it is necessary to obtain it
   through the API for obtaining device details
2  2.Devices between device sets may overlap, so devices between different device sets will be merged
3
4
```

## list device for WONDER_AUTO_TEST_LAB

**start**

list device by aggregate table fields
(fields:manufacturer & device_model & os & os_version & height & width & device_type)

**end**

## list device for APP_CENTER

**start**

| list ios device set from app center | list android device set from app center |

exclude device set, naming:
temp_ios_xxx

exclude device set, naming:
temp_android_xxx

list device configuration from app center
(for get height & width of device_model)

list device configuration from app center
(for get height & width of device_model)

Assembly device model

Assembly device model

generate ios device list

generate android device list

device list = ios device list + android device list

**end**

b. API :

- Get Device Set: get {{app-center}}/v0.1/apps/:owner_name/:app_name/owner/device_sets
- Create Device Set: post {{app-center}}/v0.1/apps/:owner_name/:app_name/owner/device_sets
- Delete Device Set: delete {{app-center}}/v0.1/apps/:owner_name/:app_name/owner/device_sets/:id
- Get Device Details: get {{app-center}}/v0.1/apps/:owner_name/:app_name/device_configurations

# make plan async create test run

## test-case-service adjustments:

**Original Make plan Requirement：**

1. Only the latest version of the executor is allowed to perform upgrade test cases
2. For upgrade tests, the number of device_tasks is counted as follows:
   a. When only non-upgrade tests are included: number of device_tasks = number of devices
   b. When only upgrade tests are included: number of device_tasks = number of old_app_versions * number of devices
   c. When both upgrade and non-upgrade are included: number of device_tasks = number of old_app_versions * number of devices + number of devices

**After implementing make plan for app center：**

1. Reference: 🪟 Starting a test run - Visual Studio App Center
2. Make plan API, create a new device set based on the selected device
3. Download appcenter app
4. Pack the executor into a JAR package
5. Create a test run by calling the REST API and test it on appcenter.
6. If the test ends: plan.status==done or cancel, delete the device set
7. Configure the repo key in the server maven setting.xml: (Configure maven in dockerfile, customize Maven settings, and copy the settings.xml file to the specified location in the container)
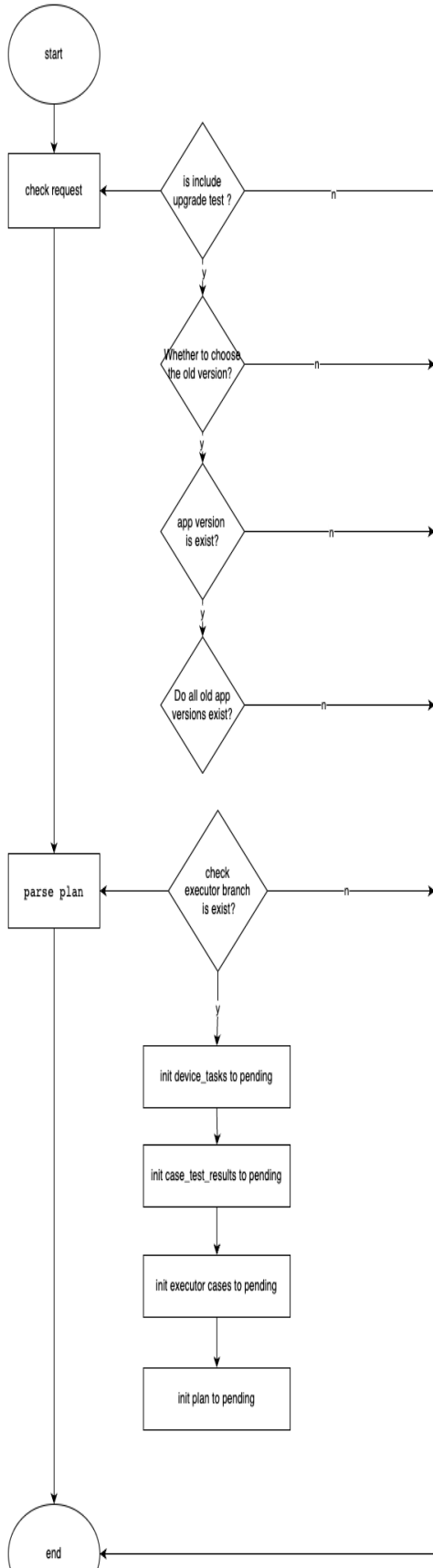
```
1    <servers>
2      <server>
3         <id>maven-ct</id>
4         <username>foodtruckinc</username>
5         <password>${env.AZURE_PACKAGES_ACCESSTOKEN}</password>
6      </server>
7    </servers>
```

dockerfile (settings.xml is placed in the docker directory of the project)
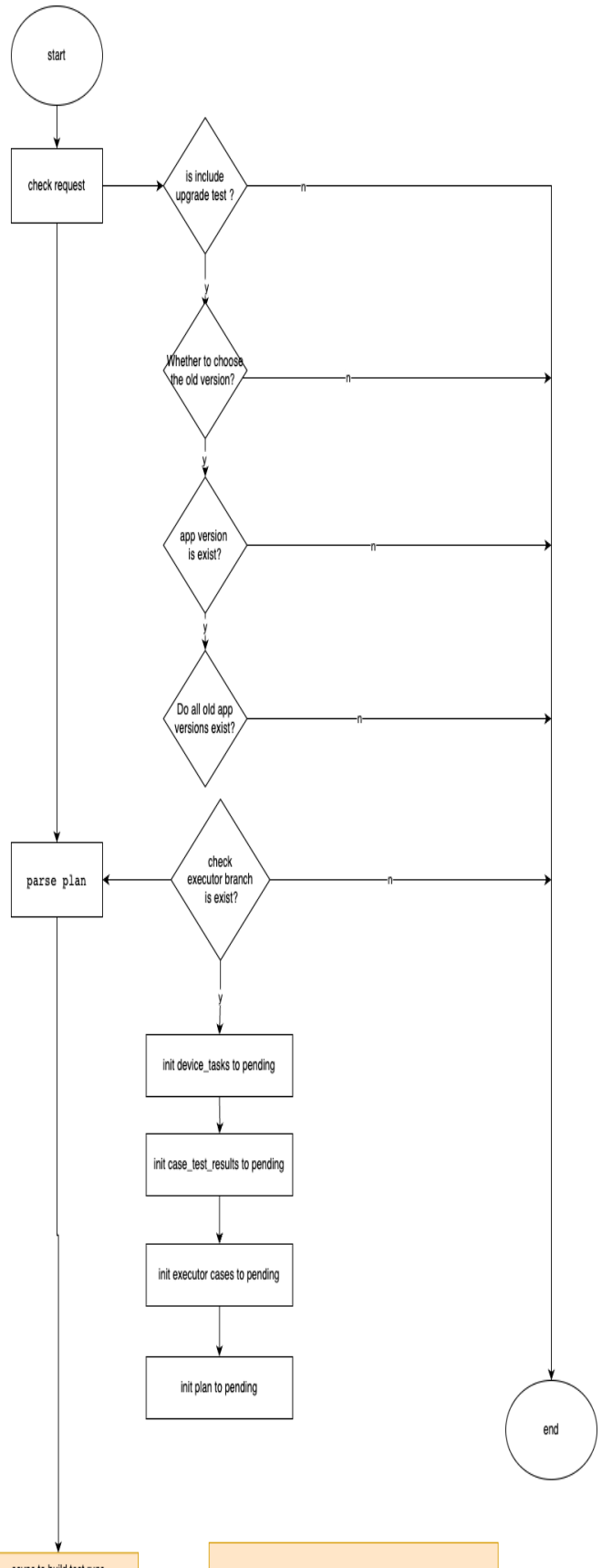
```
1   ...
2   RUN        apk update && apk add unzip
3   RUN        wget https://dlcdn.apache.org/maven/maven-3/3.9.4/binaries/apache-maven-3.9.4-bin.zip && unzip
    apache-maven-3.9.4-bin.zip && rm apache-maven-3.9.4-bin.zip
4   ENV        MAVEN_HOME /apache-maven-3.9.4
5   ENV        PATH $MAVEN_HOME/bin:$PATH
6   COPY       settings.xml $MAVEN_HOME/conf/
7   ....
```
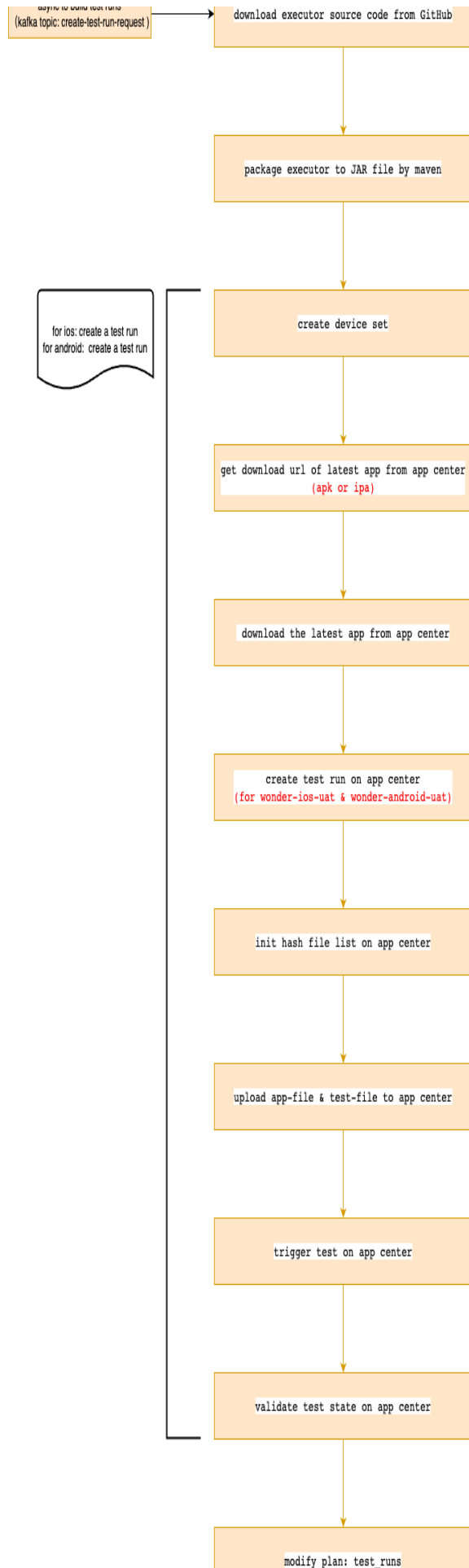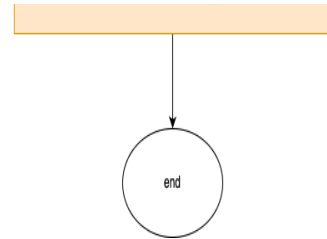
**Make plan Flow**

## Left flowchart

make plan for WONDER_AUTO_TEST_LAB

**start**

check request → is include upgrade test ? —n→

is include upgrade test ? —y→ Whether to choose the old version? —n→

Whether to choose the old version? —y→ app version is exist? —n→

app version is exist? —y→ Do all old app versions exist? —n→

Do all old app versions exist? —y→ check executor branch is exist? —n→

check executor branch is exist? —y→ init device_tasks to pending

check executor branch is exist? → parse plan

init device_tasks to pending → init case_test_results to pending → init executor cases to pending → init plan to pending

parse plan → end

**end**

## Right flowchart

make plan for APP_CENTER
ps: yellow->difference red->remove

**start**

check request → is include upgrade test ? —n→

is include upgrade test ? —y→ Whether to choose the old version? —n→

Whether to choose the old version? —y→ app version is exist? —n→

app version is exist? —y→ Do all old app versions exist? —n→

Do all old app versions exist? —y→ check executor branch is exist? —n→

check executor branch is exist? —y→ init device_tasks to pending

check executor branch is exist? → parse plan

init device_tasks to pending → init case_test_results to pending → init executor cases to pending → init plan to pending

**end**

async to build test runs

```
async to build test runs
(kafka topic: create-test-run-request )  ────►   download executor source code from GitHub

                                                  package executor to JAR file by maven

for ios: create a test run                        create device set
for android:  create a test run

                                                  get download url of latest app from app center
                                                  (apk or ipa)

                                                  download the latest app from app center

                                                  create test run on app center
                                                  (for wonder-ios-uat & wonder-android-uat)

                                                  init hash file list on app center

                                                  upload app-file & test-file to app center

                                                  trigger test on app center

                                                  validate test state on app center

                                                  modify plan: test_runs
```

**Pull task process**

Get all test cases of the earliest created task from the executor_cases table

## make plan After Testing

### test-executor adjustments

1. main entry:

   a. If it is an app center test

   b. Call API: Get device task (case & app) according to platform type & device_model

   c. No need to configure app download address in property file

2. No need to replace the latest app and executor version (same version with different revision number) during case test

3. **Maven configuration requirements:**

   a. Only keep one pom.xml file

   b. The test startup class entry configured in pom file is consistent with the Main entry of platform test

   c. Configure remote repo configuration in pom file:

```
1  <repository>
2      <id>maven-ct</id>
3      <name>maven-ct</name>
4      <url>https://pkgs.dev.azure.com/foodtruckinc/Wonder/_packaging/maven-local/maven/v1</url>
5  </repository>
```

   d. profiled: `prepare-for-upload` change to `uat` , when doing packing need to direct to specific configuration environment  ▦ Prepari
   ng Appium Tests for Upload - Visual Studio App Center

   e. Dependency Version Usage Method, Version+Change to Specifc Version, Num+method cannot successfully package in maven, eg:
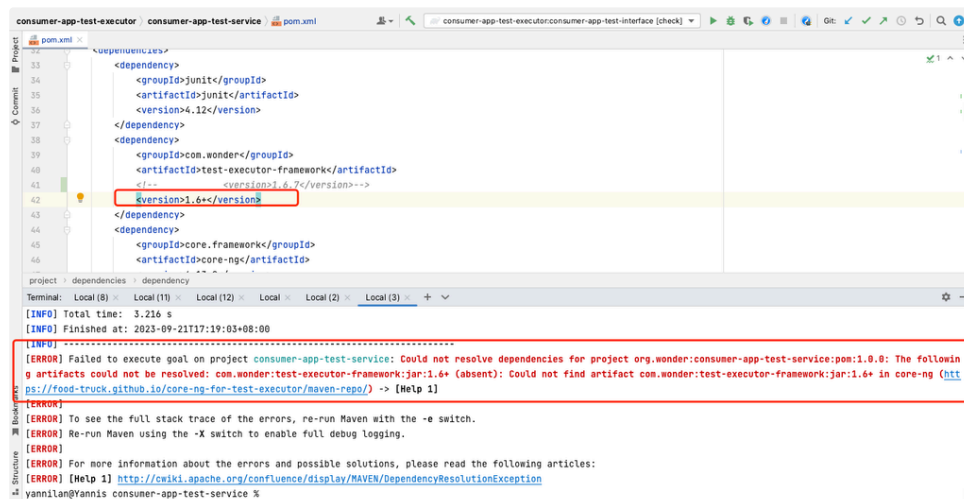


4. app center: After the task is completed, the executor needs to actively exit the program.

## 5. **Build Maven managed executor project**

```
1  Project Build
2  consumer-app-test-executor
3  - pom.xml
4  - consumer-app-test-interface
5  -- pom.xml
6  - consumer-app-test-service
7  -- pom.xml
8
```

Parent level pom.xml configuration: `consumer-app-test-executor/pom.xml`

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"
3           xmlns="http://maven.apache.org/POM/4.0.0"
4           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
5      <modelVersion>4.0.0</modelVersion>
6
7      <groupId>com.wonder</groupId>
8      <artifactId>consumer-app-test-executor</artifactId>
9      <packaging>pom</packaging>
10     <version>1.0.0</version>
11     <properties>
12         <consumer-app-test-interface.version>1.0.0</consumer-app-test-interface.version>
13         <consumer-app-test-service.version>1.0.0</consumer-app-test-service.version>
14         <java.version>11</java.version>
15     </properties>
16     <modules>
17         <module>consumer-app-test-interface</module>
18         <module>consumer-app-test-service</module>
19     </modules>
20     <repositories>
21         <repository>
22             <id>maven-ct</id>
23             <name>maven-ct</name>
24             <url>https://pkgs.dev.azure.com/foodtruckinc/Wonder/_packaging/maven-local/maven/v1</url>
25         </repository>
26         <repository>
27             <id>core-ng</id>
28             <name>core-ng-repo</name>
29             <url>https://food-truck.github.io/core-ng-for-test-executor/maven-repo/</url>
30         </repository>
31         <repository>
32             <id>maven-local</id>
33             <name>maven-local</name>
34             <url>file:/${user.home}/.m2/repository</url>
35         </repository>
36     </repositories>
37
38     <build>
39         <plugins>
40             <plugin>
41                 <groupId>org.apache.maven.plugins</groupId>
42                 <artifactId>maven-compiler-plugin</artifactId>
43                 <version>3.2</version>
44                 <configuration>
45                     <source>${java.version}</source>
46                     <target>${java.version}</target>
```

```xml
47                  <encoding>UTF-8</encoding>
48              </configuration>
49          </plugin>
50          <plugin>
51              <groupId>org.apache.maven.plugins</groupId>
52              <artifactId>maven-source-plugin</artifactId>
53              <version>2.4</version>
54              <configuration>
55                  <attach>true</attach>
56              </configuration>
57              <executions>
58                  <execution>
59                      <id>attach-sources</id>
60                      <phase>verify</phase>
61                      <goals>
62                          <goal>jar-no-fork</goal>
63                      </goals>
64                  </execution>
65              </executions>
66          </plugin>
67          <plugin>
68              <groupId>org.apache.maven.plugins</groupId>
69              <artifactId>maven-surefire-plugin</artifactId>
70              <version>2.5</version>
71              <configuration>
72                  <skip>true</skip>
73              </configuration>
74          </plugin>
75          <plugin>
76              <groupId>org.apache.maven.plugins</groupId>
77              <artifactId>maven-jar-plugin</artifactId>
78              <version>3.3.0</version>
79              <executions>
80                  <execution>
81                      <id>default-jar</id>
82                      <phase>package</phase>
83                      <goals>
84                          <goal>jar</goal>
85                      </goals>
86                  </execution>
87              </executions>
88          </plugin>
89      </plugins>
90    </build>
91 </project>
```

Child level pom.xml configuration: `consumer-app-test-executor/consumer-app-test-interface/pom.xml`

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"
3           xmlns="http://maven.apache.org/POM/4.0.0"
4           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
5      <modelVersion>4.0.0</modelVersion>
6      <artifactId>consumer-app-test-interface</artifactId>
7      <packaging>jar</packaging>
8      <version>1.0.0</version>
9      <parent>
10         <artifactId>consumer-app-test-executor</artifactId>
11         <groupId>com.wonder</groupId>
```

```xml
12          <version>1.0.0</version>
13          <relativePath>../pom.xml</relativePath>
14      </parent>
15      <dependencies>
16          <dependency>
17              <groupId>core.framework</groupId>
18              <artifactId>core-ng-api</artifactId>
19              <version>6.13.9</version>
20          </dependency>
21          <dependency>
22              <groupId>junit</groupId>
23              <artifactId>junit</artifactId>
24              <version>4.12</version>
25          </dependency>
26          <dependency>
27              <groupId>com.wonder</groupId>
28              <artifactId>test-executor-framework</artifactId>
29              <version>1.6.7</version>
30          </dependency>
31      </dependencies>
32  </project>
```

Child project level pom.xml configuration `consumer-app-test-executor/consumer-app-test-service/pom.xml`

```xml
1   <?xml version="1.0" encoding="UTF-8"?>
2   <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"
3            xmlns="http://maven.apache.org/POM/4.0.0"
4            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
5       <modelVersion>4.0.0</modelVersion>
6       <groupId>org.wonder</groupId>
7       <artifactId>consumer-app-test-service</artifactId>
8       <packaging>jar</packaging>
9       <version>1.0.0</version>
10
11      <parent>
12          <artifactId>consumer-app-test-executor</artifactId>
13          <groupId>com.wonder</groupId>
14          <version>1.0.0</version>
15          <relativePath>../pom.xml</relativePath>
16      </parent>
17
18      <dependencies>
19          <dependency>
20              <groupId>com.wonder</groupId>
21              <artifactId>consumer-app-test-interface</artifactId>
22              <version>${consumer-app-test-interface.version}</version>
23          </dependency>
24          <dependency>
25              <groupId>junit</groupId>
26              <artifactId>junit</artifactId>
27              <version>4.12</version>
28          </dependency>
29          <dependency>
30              <groupId>com.wonder</groupId>
31              <artifactId>test-executor-framework</artifactId>
32              <version>1.6.7</version>
33          </dependency>
34          <dependency>
35              <groupId>core.framework</groupId>
```

```xml
36                <artifactId>core-ng</artifactId>
37                <version>6.13.9</version>
38            </dependency>
39            <dependency>
40                <groupId>org.aspectj</groupId>
41                <artifactId>aspectjrt</artifactId>
42                <version>1.9.7</version>
43            </dependency>
44            <dependency>
45                <groupId>io.qameta.allure</groupId>
46                <artifactId>allure-junit4</artifactId>
47                <version>2.21.0</version>
48            </dependency>
49            <dependency>
50                <groupId>com.github.spotbugs</groupId>
51                <artifactId>spotbugs</artifactId>
52                <version>4.3.0</version>
53            </dependency>
54            <dependency>
55                <groupId>com.fasterxml.jackson.core</groupId>
56                <artifactId>jackson-databind</artifactId>
57                <version>2.14.2</version>
58            </dependency>
59        </dependencies>
60
61        <profiles>
62            <profile>
63                <id>uat</id>
64                <build>
65                    <plugins>
66                        <!--copy dependencies to upload/dependency-jars-->
67                        <plugin>
68                            <groupId>org.apache.maven.plugins</groupId>
69                            <artifactId>maven-dependency-plugin</artifactId>
70                            <version>2.10</version>
71                            <executions>
72                                <execution>
73                                    <id>copy-dependencies</id>
74                                    <phase>package</phase>
75                                    <goals>
76                                        <goal>copy-dependencies</goal>
77                                    </goals>
78                                    <configuration>
79                                        <outputDirectory>${project.build.directory}/upload/dependency-jars/
80                                        </outputDirectory>
81                                        <useRepositoryLayout>true</useRepositoryLayout>
82                                        <copyPom>true</copyPom>
83                                        <addParentPoms>true</addParentPoms>
84                                    </configuration>
85                                </execution>
86                            </executions>
87                        </plugin>
88                        <plugin>
89                            <groupId>org.apache.maven.plugins</groupId>
90                            <artifactId>maven-help-plugin</artifactId>
91                            <version>2.2</version>
92                            <executions>
93                                <execution>
```

```xml
                                <id>generate-pom</id>
                                <phase>package</phase>
                                <goals>
                                    <goal>effective-pom</goal>
                                </goals>
                                <configuration>
                                    <output>${project.build.directory}/upload/pom.xml</output>
                                </configuration>
                            </execution>
                        </executions>
                    </plugin>
                    <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-resources-plugin</artifactId>
                        <version>3.3.1</version>
                        <executions>
                            <execution>
                                <id>default-resources</id>
                                <phase>process-resources</phase>
                                <goals>
                                    <goal>resources</goal>
                                </goals>
                                <configuration>
                                    <resources>
                                        <resource>
                                            <directory>
                                                ${project.basedir}/conf/uat/resources
                                            </directory>
                                        </resource>
                                    </resources>
                                </configuration>
                            </execution>
                            <execution>
                                <id>default-testResources</id>
                                <phase>process-test-resources</phase>
                                <goals>
                                    <goal>testResources</goal>
                                </goals>
                            </execution>
                            <execution>
                                <id>copy-classes</id>
                                <phase>package</phase>
                                <goals>
                                    <goal>resources</goal>
                                </goals>
                                <configuration>
                                    <outputDirectory>${project.build.directory}/upload/test-classes</outputDirectory>
                                    <resources>
                                        <resource>
                                            <directory>
                                                ${project.build.outputDirectory}
                                            </directory>
                                        </resource>
                                    </resources>
                                </configuration>
                            </execution>
                            <execution>
```

```xml
                                <id>copy-testclasses</id>
                                <phase>package</phase>
                                <goals>
                                    <goal>testResources</goal>
                                </goals>
                                <configuration>
                                    <outputDirectory>${project.build.directory}/upload/test-
classes</outputDirectory>
                                    <resources>
                                        <resource>
                                            <directory>
                                                ${project.build.testOutputDirectory}
                                            </directory>
                                        </resource>
                                    </resources>
                                </configuration>
                            </execution>
                        </executions>
                    </plugin>
                </plugins>
            </build>
        </profile>
    </profiles>

    <build>
        <plugins>
            <plugin>
                <groupId>org.codehaus.mojo</groupId>
                <artifactId>aspectj-maven-plugin</artifactId>
                <version>1.14.0</version>
                <configuration>
                    <complianceLevel>${java.version}</complianceLevel>
                    <source>${java.version}</source>
                    <target>${java.version}</target>
                    <showWeaveInfo>true</showWeaveInfo>
                    <verbose>true</verbose>
                    <Xlint>ignore</Xlint>
                    <encoding>UTF-8</encoding>
                    <forceAjcCompile>true</forceAjcCompile>
                </configuration>
                <executions>
                    <execution>
                        <id>compile</id>
                        <goals>
                            <goal>compile</goal>
                        </goals>
                    </execution>
                    <execution>
                        <id>test-compile</id>
                        <goals>
                            <goal>test-compile</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
```

```xml
                <version>2.20</version>
                <executions>
                    <execution>
                        <id>default-test</id>
                        <phase>test</phase>
                        <goals>
                            <goal>test</goal>
                        </goals>
                    </execution>
                </executions>
                <configuration>
                    <testFailureIgnore>true</testFailureIgnore>
                    <excludes>
                        <exclude>**/*Test.java</exclude>
                        <exclude>**/Test*.java</exclude>
                    </excludes>
                    <includes>
                        <include>**/AppCenterExecutor.java</include>
                    </includes>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```