

Pekka Rajala

**ANDROID TEST STATIONIN KÄYTTÖNOTTOKELPOISUUDEN
SELVITYS**

ANDROID TEST STATIONIN KÄYTTÖÖNOTTOKELPOISUUDEN SELVITYS

Pekka Rajala
Opinnäytetyö
Kevät 2021
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, laite- ja tuotesuunnittelu

Tekijä: Pekka Rajala
Opinnäytetyön nimi suomeksi: Android Test Stationin käyttöönottokelpoisuuden selvitys
Opinnäytetyön nimi englanniksi: Feasibility study of Android Test Station for Bittium
Työn ohjaajat: Kari Laitinen, Tapani Laakko
Työn valmistumislukukausi ja -vuosi: Kevät 2021
Sivumäärä: 37 + 1 liite

Työn aiheena oli tutustua Googlen kehittämään Android Test Station -testiautomaatiotyökaluun ja sen käyttöönottokelpoisuuden selvitys Bittiumin Tough Mobile 2 (TM2) -älypuhelimien kehitys- ja ylläpitoprojektille. Tavoitteena oli tuottaa perustellut syyt ohjelman käyttöönoton puolesta ja vastaan nykyisen järjestelmän tilalle tai rinnalle.

Aluksi asennettiin ja tutustuttiin ATS:n toimintaan, ominaisuuksiin ja käyttöön yleisellä tasolla. Tämän jälkeen vertailtiin tarkemmin sen toiminnallisuutta ja helppokäyttöisyyttä nykyiseen järjestelmään.

Tutkinnan aikainen versio ohjelmasta oli vielä toiminnaltaan ja luotettavuudeltaan liian epävakaata ja siitä puuttui projektille tärkeitä ominaisuuksia. Tästä syystä ohjelmaa ei vielä tulla ottamaan käyttöön. Ohjelman kehitystä tullaan kuitenkin seuraamaan ja Googlelle lähetetään palautetta toivotuista ominaisuuksista ja havaituista ongelmista.

Asiasanat: Android, testaus, testiautomaatio, Tough Mobile 2, Bittium, käyttöönottoselvitys.

ABSTRACT

Oulu University of Applied Sciences
Degree program in Information Technology, Option of Device and Product Design

Author: Pekka Rajala
Title of thesis: Feasibility Study of Android Test Station for Bittium
Supervisors: Kari Laitinen, Tapani Laakko
Term and year when the thesis was submitted: Spring 2021
Pages: 37 + 1 appendix

Android Test Station is a test automation tool with a web-interface developed by Google for running their Test Suites more easily. The program was first released in December 18, 2019 and had caught the attention of supervisors of Bittium Tough Mobile 2 -development and maintenance project. They wanted to know whether the program was beneficial for the project.

The goal of this thesis was to have a feasibility study made on Android Test Station and provide justifiable reasons for and against its usability in tandem with or as a replacement for the projects current test automation system.

I had been working with Google Test Suites for over 6 months and had gotten a good handle on using them and this knowledge assisted in learning how to use and review ATS functionality. I first learned how to use it and its functions and then compared the usability and functionality to the current system.

After review it was found that ATS has many compelling features, but is still missing a few key features important to the project in addition being a little unstable at the time of review. Because of this it will not be taken to use in the project.

Project will continue to monitor the development of ATS and a new study shall be made after needed key features are implemented.

Keywords: Android, testing, test automation, Tough Mobile 2, Bittium, Feasibility Study.

ALKULAUSE

Haluan kiittää Bittiumia sekä työnantajana että opinnäytetyön toimeksiantajana. Kaikkia kollegoita ja esimiehiä haluan kiittää heiltä saamistani neuvoista ja palautteesta, kärsivällisyydestä jatkuvia kyselyitäni kohtaan sekä minulle suomasta luottamuksesta tehtävän omatoimiseen suoritukseen. Erityisesti heistä kiitän Lauri Fleuriot-Pajusta, Tapani Laakkoa, Tomi Koutosta sekä Tommi Härmää.

Kiitäisin myös ohjaavia opettajia, Kari Laitista sekä Tuula Hopeavuorta heidän antamistaan palautteista ja ohjeista raportin kirjoittamiseen.

Työhön sisältyi paljon salassapidon alaista tietoa, josta en tässä raportissa voi kertoa, joten näistä tarkemmin tietoa yhtiön sisäisessä dokumentaatiossa.

Käyttönottokelpoisuuden arviointi opinnäytetyön aikana keskittyi Android Test Stationin versioihin R7–9. Tämän raportin kirjoitukseen kuluneena aikana Google julkaisi Android Test Stationista uusia versioita, joiden mukana ohjelmaan sisällytettiin useita täysin uusia ominaisuuksia. Näiden tutkimista ja arviointia en tähän opinnäytetyöhön valitettavasti saanut ajan puutteen vuoksi sisällytettyä.

7.1.2021

Pekka Rajala

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	7
1 JOHDANTO	8
2 BITTIUM	9
2.1 Bittium Tough Mobile 2	9
2.2 Tough Mobile 2:n ominaisuudet	9
3 ANDROID JA GOOGLLEN TEST SUITET	11
3.1 Android	11
3.2 Test Suiten tarkoitus	11
3.3 Google Test Suiten käyttö	12
3.4 Tulosten tulkinta ja uudelleenajo	13
3.5 Testiautomaatio	14
3.6 Hyväksyntä- ja päivittäinen testiajoprosessi	16
4 ANDROID TEST STATION	17
4.1 Ominaisuudet	17
4.1.1 Käyttöliittymä	17
4.1.2 Laitetoiminnot eli Device actionit	21
4.2 Työn aloitus ja eteneminen	22
5 HAVAINNOT	24
5.1 Käyttöönnoton puolesta	24
5.1.1 Fyysinen testiympäristö	24
5.1.2 Testien ajo ja yleinen käyttö	25
5.1.3 Ylläpito	25
5.2 Käyttöönottoa vastaan	28
6 YHTEENVETO	33
LÄHTEET	35
LIITTEET	
Liite 1 CTS Testiajoskripti	

SANASTO

AOSP	Android Open Source Project. Avoimen lähdekoodin Androidin kehitysprojekti
ATS	Android Test Station. Googlen kehittämä testiautomaatio-ohjelma
CTS	Compatibility Test Suite. Yksi Googlen hyväksyntätesteihin kuuluvista testipaketeista
Flässa	Koontiversion tiedostojen kopiointi ja purku puhelimeen
GMS	Google Mobile Services eli Googlen tarjoamat mobiilipalvelut, kuten Google Maps, Youtube, Gmail jne.
Hyväksyntätestit	Kaikkien Test Suite -pakettien kaikki testit
Koontiversio	Ohjelmistopakettien versio
Test Suite	Testiohjelma tai testipaketti
Testikone	Tietokone, jolla testejä ajetaan
Testilaite	Testattava Android-älypuhelin, Tough Mobile 2
Testimoduuli	Kokoelma tiettyyn toiminnallisuuteen keskittyviä testejä. Test Suitet sisältävät useita testimoduuleja. Esim. CtsCameraTestModule
TM2	Tough Mobile 2. Bittiumin kehittämä ja tuottama tietoturvakovennettu älypuhelin

1 JOHDANTO

Opinnäytetyön tavoitteena oli saada selvitettyä Android Test Stationin käyttöön-
oton kannattavuus Tough Mobile 2:n ylläpito- ja kehitysprojektissa ja muodostaa
perustellut syyt päätökselle. Android Test Station on Googlen kehittämä testiau-
tomaatiotyökalu. (1.)

Aloitin työt Bittium Wireless Oy:ssä helmikuussa 2020 osana Bittium Tough
Mobile 2:n kehitys- ja ylläpitoa. Pääasiallisena työtoimenkuvani on
ollut huolehtia Googlen hyväksyntätestien ajosta Tough Mobile 2:n Android 9
-ylläpito-päivitysversiolla, raportoida ja dokumentoida testeissä ilmi tulleet vir-
heet, verifioida näiden korjaukset, sekä ylläpitää testikoneiden ohjelmistot ja
Test Suite -versiot.

Opinnäytetyön aihetta päätettäessä olin työskennellyt Googlen hyväksyntätesti-
en ja testiohjelmien kanssa päivittäin jo puolen vuoden ajan, joten olin saavut-
tanut vahvan ymmärryksen näiden toiminnasta sekä Bittiumin käyttämästä tes-
tiautomaatiosta. Näiden hyvä tuntemus helpotti ja nopeutti Android Test Statio-
niin tutustumista ja arviointia.

Tässä opinnäytetyön raportissa kerron aluksi taustatietoina hieman Bittiumista
ja sen tuotteista ja palveluista. Googlen tarjoamista Test Suiteista kerron, mitä
ne ovat ja kuinka niitä käytetään. Lisäksi kerron näiden testitulosten tulkinnasta
sekä alustavasti testiautomaatiosta. Android Test Station -luvussa kerron sen
ominaisuuksista mahdollisimman objektiivisesti sekä työn aloituksesta ja sen
etenemisestä. Havainnot-luvussa kerron enemmän mielipidepainotteisesti työn
aikana tehdyistä havainnoista. Vertailen ohjelman hyviä ja huonoja ominaisuuksia
nykyiseen järjestelmään nähden. Yhteenvedossa kerron työn lopputuloksis-
ta ja omia pohdintoja ohjelmasta sekä ATS:n jatkosta projektin kannalta.

ATS:n arvioinnissa ja raportissa keskityin projektin tarpeisiin eli sen soveltuvuu-
teen Tough Mobile 2:n ja Bittiumin koventaman Android 9 -käyttöjärjestelmän
testauksessa. Vertasin myös ohjelman ominaisuuksia projektin käyttämään tes-
tiautomaatiojärjestelmään selvittääkseni, mitä parannuksia se toisi ja mitä omi-
naisuuksia siitä mahdollisesti puuttuu.

2 BITTIUM

Bittium Oyj on maailmanlaajuisesti toimiva suomalainen yhtiö, joka on erikoistunut tietoturvallisten ja luotettavien viestintä- ja liitettävyyseratkaisujen kehittämiseen ja tuottamiseen yli 30 vuoden ajan. Bittiumin päätoimisto sijaitsee Oulussa.

Päätuotealueet ovat lääketieteelliset biosensorit ja palveluratkaisut, armeijan taktiset kommunikaatoratkaisut, turvalliset kommunikaatio- ja tietoyhteysratkaisut sekä tutkimus- ja kehitystyön palvelut. Bittium Tough Mobile-tuoteperhe on osa turvallisia kommunikaatio- ja tietoyhteysratkaisuja. (2.)

2.1 Bittium Tough Mobile 2

Tough Mobile 2 on vuonna 2019 julkistettu Bittiumin Suomessa kehittämä ja tuottama tietoturvakovennettu toisen sukupolven Android-älypuhelin. Laite on suunniteltu luottamuksellisen datan käsittelyyn ja on, TechRadarin arvostelun mukaan, todennäköisesti maailman tietoturvalisin älypuhelin tällä hetkellä. (3.)

Usein varsinkin teknologia-alan yritykset tarjoavat työntekijöilleen erillisen työpuhelimien, jonka käyttö on rajoitettu vain työtehtäviin, jolloin työntekijät joutuisivat kuljettamaan kahta puhelinta mukanaan kaiken aikaa. Hyvin usein tämä työpuhelin kuitenkin monilla jää hyvin vähäiselle käytölle ja voi unohtua työpöydän laatikkoon. Tough Mobile 2 on suunniteltu sopimaan käytettäväksi työn ja vapaa-ajan älypuhelimena. Tuotteen pääasialliset kohderyhmät ovat viranomaiset, puolustusvoimat sekä muut ammattilaisorganisaatiot, jotka tarvitsevat tietoturvallista mobiiliviestintäratkaisua. (4.) Kuka vain tietoturvatietoinen yksityinenkin henkilö voi laitteen ostaa.

2.2 Tough Mobile 2:n ominaisuudet

Yleisimpien älypuhelimissa olevien painikkeiden, kuten äänisäädön ja virtanapin, lisäksi laitteessa on Privacy mode-, Emergency- ja Push-to-talk-painikkeet. Laitetasolla toteutetun Privacy mode -painikkeen painalluksella voi sulkea laitteen mikrofonit, kamerat ja Bluetoothin ja heikentää sensoritarkkuutta.

Emergency- ja Push-to-talk-painikkeet ovat ohjelmitavissa olevia nappeja ja tällä hetkellä nimensä mukaiset toiminnot vaativat erillisen ohjelman. (4.)

Tough Mobile 2 on saatavilla kolmena eri käyttöjärjestelmäversiona, jotka toimivat samanlaisissa laitteissa (kuva 1). Kenen tahansa hankittavissa on kaksi varianttia, joista toinen sisältää Googlen mobiilipalvelut, kuten esimerkiksi Google Play, Gmail ja Google Maps esiasennettuna. Ja toinen on Android Open Source Project (AOSP) -pohjainen ilman Googlen mobiilipalveluita. (5.) Kolmannessa versiossa, joka on erotettu nimikkeelle Tough Mobile 2 C, on molemmat käyttöjärjestelmät toisistaan täysin erotettuna. Tämän tarkoituksena on täysin eristää luottamuksellisessa työkäytössä oleva puoli sekä vapaa-ajalla käytettävä puoli. Kaikissa varianteissa on Bittiumin tekemiä tietoturvakovennuksia ja ominaisuuksia. (4.)



KUVA 1. Tough Mobile 2:n kolme käyttöjärjestelmäversiota (4)

3 ANDROID JA GOOGLLEN TEST SUITET

3.1 Android

Android on avoimeen lähdekoodiin ja Linux-ytimeen pohjautuva mobiili- ja älylaitteille luotu käyttöjärjestelmä (6). Avoimen lähdekoodin ansiosta kuka tahansa voi luoda ja kokeilla erilaisia ominaisuuksia ja toiminnallisuuksia. Tästä syystä monet älypuhelimien ja tablettien valmistajat käyttävät laitteissaan Android-pohjaisia käyttöjärjestelmiä sen sijaan, että kehittäisivät itse alusta alkaen täysin erillisen oman käyttöjärjestelmänsä.

3.2 Test Suiten tarkoitus

Test Suitet ovat kokoelmia Googlen kehittämistä automatisoiduista testeistä, jotka on jaoteltu kunkin Test Suiten sisällä omiin testialuekohtaisiin moduuleihinsa. Näillä testeillä varmennetaan Android-käyttöjärjestelmän ja sitä käyttävien laitteiden yhteensopivuus ja toimivuus. (7.) Jokaiselle Android-versiolle on oma versionsa näistä paketeista ja julkisesti saatavilla olevia Test Suiteja on mm. Compatibility Test Suite (CTS). Android 9 -versiossa CTS sisältää hiukan alle miljoona automatisoitua testiä. CTS Verifier sisältää lähes 100 testaajan itse käsin ajettavaa testiä ja CTS Instant Apps on erillinen Test Suite ainoastaan Android 9 -versiolle, joka sisältää noin 13 000 automatisoitua testiä.

Julkisten Test Suite -pakettien lisäksi on myös ei-julkisia NDA:n alaisia Android GMS Partnereille (Google Mobile Services yhteistyökumppaneille) käytettävissä olevia Test Suiteja, joista tarkemmin Bittiumin sisäisessä dokumentaatioissa ja Googlen GMS Partner -sivulla, jonne tarvitsee Partner -tunnukset nähdäkseen sivun sisällön.

Test Suite -pakettien avulla Android GMS -yhteistyökumppanit ja Android-kehittäjät voivat testata tekemiään muutoksia ja lisäyksiä varmistaen näiden toimivuuden sekä tarkistaa, etteivät nämä muutokset ja lisäykset riko Android-yhteensopivuutta. (7; 8.)

Test Suitet on tarkoitettu ajettavaksi Linux-ympäristössä terminaalikomennoin ja eri Android-versioiden Test Suite -pakettien ajamiseen voi olla eri vaatimuksia. Esimerkiksi eri Android-versiot vaativat eri Open Java Development Kit -version ja jotkut Test Suitet voivat vaatia lisäohjelmia toimiakseen. (9).

3.3 Google Test Suiten käyttö

Test Suiteista julkaistaan uusi versio neljännesvuosittain ja julkaisuhyväksynnässä testit ajetaan aina tuoreimmalla versiolla, joten tuotekehittäjienkin on suositeltavaa käyttää varmennustesteissään aina tuoreinta Test Suite -versiota. Test Suite ladataan zip-pakettina Googlen sivuilta ja puretaan testikoneessa haluttuun polkuun. Test Suite on purkamisen jälkeen heti käyttövalmis olettaen, että ohjeistuksen mukaiset taustaohjelmat ja asetukset ovat kunnossa. (9; 10.)

Alustavasti, mikäli laajempaa testiautomaatioympäristöä ei ole vielä rakennettu tai sille ei ole itsellä tarvetta, ajetaan testejä lokaalisti terminaalinäköymästä. Testiohjelman käynnistys-skripti löytyy Test Suite -kansion tools-kansion alta. Terminaalikäynnistyskomentoesimerkki (11; 12.):

```
$ /android-cts/tools/.cts-tradefed
```

Test Suiten sisältämät kaikki testit ajetaan komennolla run ja Test Suiten lyhenne.

```
cts-tf > run cts
```

Test Suite -pakettien testit on myös jaoteltu 64-bittisiin arm64-v8a- ja 32-bittisiin armeabi-v7a-testeihin, ja varsinkin CTS:n testejä ajettaessa on suositeltavaa ajaa vain jompaakumpaa kerrallaan, sillä pelkästään 64-bittisten testien kertaalleen ajossa voi kestää lähes kaksi vuorokautta. (13.)

```
cts-tf > run cts --abi arm64-v8a
```

Toisinaan on tarpeen ajaa pelkästään yhden moduulin testit, jolloin komentoon lisätään -m ja moduulin nimi.

```
cts-tf > run cts --abi <armVersion>-m <TestModule>
```

Testiajon voi rajata myös yhteen tiettyyn testiin tietyissä moduulissa lisäämällä moduulirajauksen jälkeen -t ja testin nimi.

```
cts-tf > run cts --abi <armVersion> -m <TestModule> -t <TestCase>
```

Esimerkkikomento yhden 64-bittisen kameratestin ajoon:

```
cts-tf > run cts --abi arm64-v8a -m CtsCameraTestCases -t android.hardware.camera2.cts.RecordingTest#testVideoSnapshot
```

Ensimmäisellä kerralla kuitenkin suosittelisin tarkistamaan Test Suiten ja testiympäristön toimivuuden ajamalla jonkin yksittäisen testin tai lyhyen testimoduulin kuten CtsAbiOverrideHostTestCases-moduulin.

```
cts-tf > run cts --abi arm64-v8a-m CtsAbiOverrideHostTestCases
```

Antamalla lisäkomennon *--shard-count* ja testattavien laitteiden lukumäärän on mahdollista pirstaloida (Sharding) testiajon testit useammalle saman koontiver-sion omaavalle testilaitteelle samanaikaisesti ajettavaksi. Tällä tavoin saadaan lyhennettyä testiajona kuluva aikaa. Tehokkuuden maksimoimiseksi Google suosittelee käyttämään pirstalointia kuudella tai useammalla testilaitteella. (11.) Taulukossa 1 on vertailu, kuinka pirstalointi vaikuttaa testikierrosten kestoon yhdellä, kahdella, kolmella ja neljällä laitteella ajettuna.

TAULUKKO 1. Pirstaloitujen CTS-testiajokierrosten kestot muodossa hh:mm:ss

Laitteiden lkm.	Yksi	Kaksi	Kolme	Neljä
Kierros 1.	41:17:53	21:02:41	13:33:31	10:04:00
Kierros 2.	0:43:00	1:25:02	0:27:13	0:13:34
Kierros 3.	0:29:32	0:14:20	0:17:05	0:06:55
Kierros 4.	0:26:13	0:03:59	0:05:20	0:03:01
Kierros 5.	0:19:27	0:06:49	0:02:32	
Kierros 6.	0:25:36	0:01:30		
Yhteensä	43:41:41	22:54:21	14:25:41	10:27:30

3.4 Tulosten tulkinta ja uudelleenajo

Testisetin ajon jälkeen voi olla useitakin epäonnistuneita testejä, mikä on normaalia. Googlen hyväksyntätesteissä on sallittua ajaa uudelleen epäonnistuneita

ta testejä useaan otteeseen. Testiajojen välissä voi tehdä tarvittavia toimenpiteitä testattavalle laitteelle, kuten laitteen uudelleenkäynnistyksen tai tehdasasetusten palautuksen ja mediatiedostojen siirron. Testiajotuloslistauksen saa näkyviin komennolla

```
cts-tf > list results tai list r
```

Kuvassa 2 on esimerkki kesäkuussa ajettujen CTS-testien tulospöytäkirjasta. Ensimmäinen kierros on ajettu komennolla: `run cts --abi armeabi-v7a` ja seuraavat ovat tämän ajon epäonnistuneiden testien uudelleenajoja. Tulospöytäkirjään on listattu testisession numero alkaen nolasta, montako testiä on mennyt hyväksytyksi läpi, epäonnistuneiden testien määrä, läpikäytyjen testimoduulien määrä, tulospöytäkirjan nimi, joka muodostuu testin aloitusajasta, testisuunnitelma, testatun laitteen sarjanumero, testatun koontiversion tunnisteen sekä tuotevariantti (kuva 2). Tulospöytäkirjan testiajojen epäonnistuneet testit saa uudelleenajettua komennolla

```
cts-tf > run retry --retry <SessionId>
```

```
cts-tf > list r
Session Pass Fail Modules Complete Result Directory Test Plan Device serial(s) Build ID Product
0 469223 156 318 of 322 2020.06.01_09.57.36 cts MR4 slate_gms_EEA
1 475848 256 320 of 322 2020.06.02_19.25.28 cts-retry MR4 slate_gms_EEA
2 475851 253 320 of 322 2020.06.03_02.22.22 cts-retry MR4 slate_gms_EEA
3 475853 251 321 of 322 2020.06.03_03.04.35 cts-retry MR4 slate_gms_EEA
4 476089 15 322 of 322 2020.06.03_08.12.28 cts-retry MR4 slate_gms_EEA
5 476104 0 322 of 322 2020.06.03_11.45.41 cts-retry MR4 slate_gms_EEA
cts-tf >
```

KUVA 2. Testitulospöytäkirjä

Tarkemmat testitulokset ja suuntaa antavat testien epäonnistumisen syyt löytyvät results-kansiosta tiedostosta `test_result_failures.html`. Tiedostosta löytyvät myös kaikki Test Suiteen sisältyvät moduulit aakkosjärjestykseen listattuna taulukkona, josta löytyy myös jokaisen moduulin testien kokonaislukumäärä sekä läpi menneiden ja epäonnistuneiden testien määrät omilla sarakkeillaan. (14.)

3.5 Testiautomaatio

Compatibility Test Suite vaatii jokaiselle testiajolle paljon yleensä erikseen tehtäviä asetusmuutoksia, kuten esimerkiksi mediatiedostojen testattavalle laitteelle siirtämisen, WiFi-yhteyden asetuksen, näytönlukituksen poistamisen ja USB

debugging -tilan päälle laiton. Täyden asetuslistauksen ja ohjeen löytää Androidin testiasetusohjeista (15).

Kuten kuvasta 2 näkyy, on kaikkien testien hyväksytysti läpi saamiseen täytynyt epäonnistuneita testejä ajaa uudelleen useita kertoja. Manuaalisesti jokaisen uudelleenajon aktivoiminen on aikaa vievää ja työlästä varsinkin, kun testattavana on useita tuotevariantteja ja Test Suiteja useilla eri testikoneilla.

Automaation avulla ei testaajan tarvitse olla koko ajan seuraamassa testiajosten etenemistä ja testien uudelleenajot saadaan käynnistettyä myös työaikojen ulkopuolella. Työkuormaa saadaan helpotettua paljon jo suhteellisen yksinkertaisilla testiautomaatioskripteillä. Kuvan 2 neljä ensimmäistä testiajota on esimerkiksi ajettu liitteen 1 kaltaisella skriptillä, joka jokaisen testiajon jälkeen tekee testattavalle laitteelle uudelleenkäynnistyksen ja ajaa edellisen testisession epäonnistuneet testit uudelleen. Kaksi viimeistä uudelleenajota on ajettu manuaalisesti. Ennen session 4 aloitusta on puhelimelle tehty tehdasasetusten palautus ja ennen viimeistä ajota on puhelimeen vaihdettu jäljellä olevien testien vaatima SIM-kortti. Automaation tärkeys on havaittavissa kuvassa 2 näkyvien testien aloitusajoissa. Skriptin avulla käynnistetyt uudelleenajot ovat käynnistyneet reilusti normaalien työaikojen ulkopuolella ja testaus saatiin valmiiksi kolmen työpäivän sisällä. Mikäli tämä sama olisi tehty täysin manuaalisesti, olisi testaus saatu valmiiksi vasta viidentenä työpäivänä.

Tough Mobile 2:n kehitys- ja ylläpitoprojektissa tarvitaan edistyneempää testiautomaatiota ja projektissa käytetäänkin avoimen lähdekoodin Jenkins-automaatiotyökalua (16). Tällä ohjelmalla mahdollistetaan testien uudelleenajon lisäksi muun muassa aina uusimman päivittäiskoontiversion tiedostojen kopiointi ja purku eli flässäys testattavaan laitteeseen sekä testilokien ja tulosten verkkokansioon kopioinnin automatisointi. Järjestelmä myös lähettää testien valmistumisesta ilmoituksen sähköpostitse. Tärkeimpänä kuitenkin omasta mielestäni Jenkins tarjoaa selkeän graafisen web-käyttöliittymän ja mahdollistaa useiden testikoneiden hallinnan yhdestä paikasta sen sijaan, että olisi useita etäyhteyksiä eri testikoneisiin.

3.6 Hyväksyntä- ja päivittäinen testiajoprosessi

Tough Mobile 2:n kehitys- ja ylläpitoprojektissa toimitaan Agile-menetelmän mukaisesti ja TM2:n Android-versioihin tehdään jatkuvasti sekä tietoturvaan liittyviä muutoksia että uusia toiminnallisuuksia. Näitä ei kuitenkaan heti lähetetä asiakkaiden laitteisiin. Muutokset kootaan tietyin aika- ja tavoitevälein julkaistaviksi ylläpitopaketeiksi. Googlen palveluita sisältäville ylläpitopäivityspaketeille tarvitaan julkaisuhyväksyntä ja hyväksynnän saadakseen on ylläpitopäivityspaketit lähetettävä Googlen testilaboratorioon, jossa niiden on läpäistävä kaikki hyväksyntätestit.

Näiden testien läpäisyn varmistamiseksi luodaan jatkuvasti päivittäiskoontiversioita tehdyistä muutoksista. Näille ajetaan omien testien lisäksi kaikki Googlen hyväksyntätestit aikaisemmissa luvuissa kuvatulla tavalla. Mikäli jokin testi ei useista yrityksistä huolimatta mene läpi, tehdään tästä tehtävänhallintajärjestelmään raportti, joka ohjataan ohjelmistokehittäjälle. Ohjelmistokehittäjä tutkii vian mahdolliset syyt ja tekee tarvittavat korjaukset seuraavaan päivittäiskoontiversioon, jolle virheen raportoija ajaa testin uudelleen verifioidakseen korjauksen. Jos testi ei vielä kukaan mene hyväksytysti läpi, palautetaan tehtävä takaisin ohjelmistokehittäjälle. Nämä vaiheet toistetaan, kunnes raportoija on saanut verifioitua korjauksen. Verifiointi suoritetaan sekä yksittäistestinä että täyden Test Suite -ajon yhteydessä. Joissain tapauksissa ohjelmistokehittäjä voi teettää erillisen katselmointikoontiversion raportoijan testattavaksi ennen korjauksen liittämistä seuraavaan päivittäiskoontiversioon.

Päivittäiskoontiversio-pakettien testejä ajetaan Tough Mobile 2:n kehitysversioilla ja ennalta määritettyjen tavoitteiden täytyessä tehdään ylläpitopäivityspaketeista erilliset julkaisukoontiversiot, joille ajetaan kaikki Googlen hyväksyntätestit TM2:n kuluttajaversioilla. Kun julkaisukoontiversiot läpäisevät Bittiumin ajamat hyväksyntätestit, lähetämme nämä versiot Googlen laboratorioon verifioitavaksi.

4 ANDROID TEST STATION

Android Test Station on Googlen kehittämä ilmainen testiautomaatio-käyttöliittymä, jonka ensimmäinen versio julkaistiin joulukuussa 2019, joten kyseessä on suhteellisen uusi työkalu Android-kehittäjille ja -testaajille. Tästä syystä Bittiumilla haluttiin teettää tutkielma sen käyttöönoton kannattavuudesta nykyisen testiautomaatiojärjestelmän tilalle tai rinnalle.

Google suosittelee asentamaan Android Test Stationin ja käyttämään sitä tietokoneessa, jossa käyttöjärjestelmänä on Ubuntu 18.04 sekä vähintään 8 GB keskusmuistia ja 100 GB vapaata kovalevytilaa. Ohjelmaa ajetaan virtuaalisessa Docker-Containerissa, jota tutummin kutsutaan kontiksi. Tällöin siihen ei pysty suoraan vaikuttamaan selainpohjaisen käyttöliittymän ulkopuolelta. (1; 17.)

4.1 Ominaisuudet

Android Test Stationin erikseen huomioimisen arvoiset pääominaisuudet voidaan jakaa selainpohjaiseen käyttöliittymään ja Device Actioneihin eli laitetoimintoihin. ATS:n käyttöohjeet sekä muuta lisätietoa löytyy osoitteesta: <https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide>.

4.1.1 Käyttöliittymä

Android Test Suiten käyttöliittymänäkymät on jaettu viiteen välilehteen, jotka ovat Test Suites, Test Plans, Test Results, Devices ja Settings. Test Suites -välilehdellä on lueteltu käytettävissä olevat Test Suitet eri Android-versioille ja testiajot pääasiassa käynnistetään tätä kautta (kuva 3). Test Suiteja voi kopioida ja luoda myös itse. Oma Test Suite luotaessa sille voi muun muassa valita käytettävän Test Suite -version ja ajokomennon. Voi esimerkiksi luoda usein ajettaville yksittäisille testimoduulijajoille, kuten kameramoduulille, oman Test Suiten tähän näkymään. (18.)

Test Suites		Create a new test	
Name	Description		
CTS 10.0 (ARM)		Run test	Edit
CTS 6.0 (ARM)		Run test	Edit
CTS 7.0 (ARM)		Run test	Edit
CTS 7.1 (ARM)		Run test	Edit
CTS 8.0 (ARM)		Run test	Edit
CTS 8.1 (ARM)		Run test	Edit
CTS 9.0 (ARM)		Run test	Edit
CTS for Instant Apps 9.0 (ARM)		Run test	Edit
NOOP		Run test	Edit

KUVA 3. Test Suite -välilehden näkymä (18, Figure 7)

Run test -painikkeella aloitetaan halutun testiajon konfigurointi. Tästä avautuvasta näkymästä voidaan määrittää testiajon parametrit, kuten ajokomento, uudelleenajojen määrä sekä lisänimikkeet testiajolle (kuva 4). Kuvassa 4 näkyy itse luomani Test Suite, jossa testit rajattu 64-bittisiin testeihin.

Android Test Station
Version: prod_R9.202009.001
ADB: 29.0.6-6198805

Testing

- Test Suites
- Test Plans
- Test Results**

Devices

- Devices

Settings

- Settings

Individual Tests

← Schedule a Test Run

1 Configure Test Run — 2 Select Devices — 3 Add Actions — 4 Set Test Resources

Name *
Oma CTS 9.0 (64bit) R13

Command
cts --abi arm64-v8a

Find supported CTS parameters from [here](#).

Retry Command
retry --retry 0

Previous Test Run
Local

ID

Set to resume a previous local test run

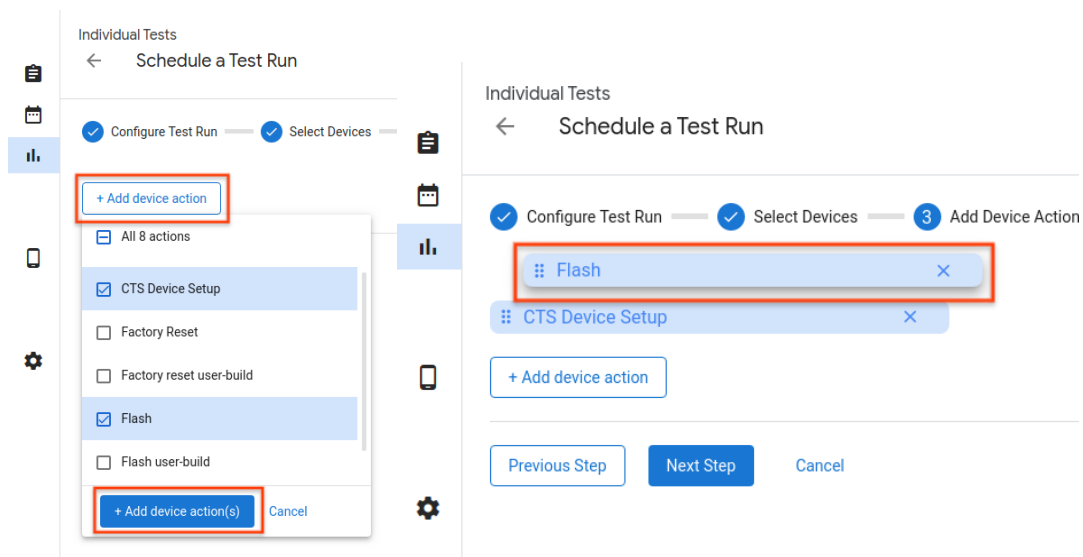
Advanced Settings ▾

Labels

Next Step Cancel

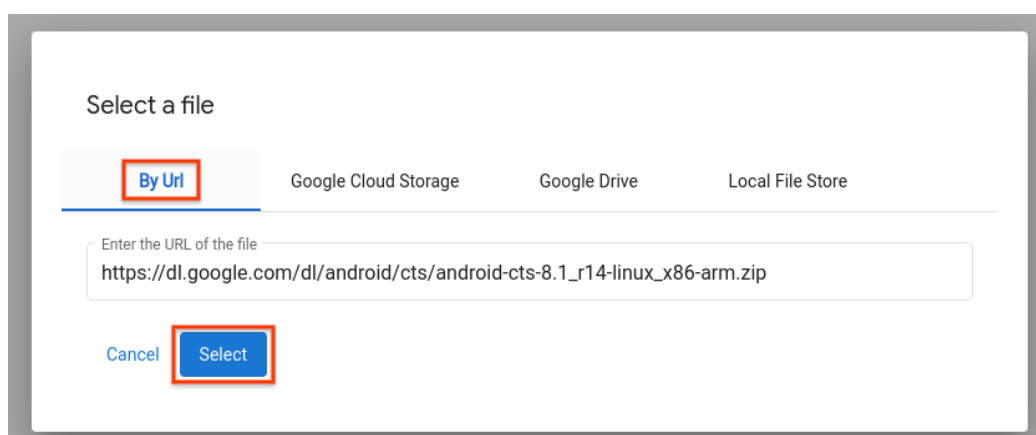
KUVA 4. Itse luodun testiajon konfigurointinäkymä lisäkomennoilla

Seuraavaksi valitaan testiajossa käytettävät laitteet, yksi tai useampia, sitten valitaan käytettävät laitetoiminnot ja niiden järjestys (kuva 5) (19).



KUVA 5. Laitetoimintojen valinta ja järjestys (20, Figure 12 & 13)

Lopuksi määritetään testissä käytettävien testiresurssien latauslinkit (21). Testiresursseja ovat esimerkiksi Test Suite -versio, mediatiedostot ja flässättävä koontiversio. Testiresursseille ATS oletusarvoisesti tarjoaa verkko-osoitetta heidän palvelimeltaan, mutta resurssit voi määrittää ladattavaksi myös Google Cloud- ja Google Drive -palveluista tai nämä voi ladata valmiiksi ATS-kontin sisäiseen muistiin (kuva 6).



KUVA 6. Testiresurssien latauspaikan valintanäkymä (21, Figure 15)

Sisäisten tietoturvamäärittysten vuoksi ei tarjottuja pilvipalveluratkaisuja tulisi projektissa käyttämään, joten näiden toimivuuden testausta ei nähty tarpeellisena.

Test Plans -välilehdellä voi luoda ajastettuja tai manuaalisesti käynnistettäviä testisuunnitelmia. Testisuunnitelmiin voi valmiiksi asettaa usein ajettavien testiajojen parametrit, jolloin testiajon saa käynnistettyä yhtä nappia painamalla sen sijaan, että asettaisi nämä erikseen jokaisen testiajon käynnistykseen yhteydessä. Tämän voi myös luoda Test Suites -välilehtikappaleessa mainittuja yksittäisten moduulien testiajosuunnitelmia. (22, Figure 20.)

Käynnissä olevia ja valmistuneita testiajoja voi seurata kuvassa 7 näkyvältä Test Results -välilehdeltä (23). Testiajot ovat listattuna luomisjärjestyksessä ja jokaisesta on nähtävillä testiajon nimi, käytetty testipakettiversio, testilaitteen sarjanumero, koontiversio, laitteen tuotekoodi, testiajolle mahdollisesti annetut lisänimikkeet (Labels), luomisajasta kulunut aika, testiajon status sekä epäonnistuneiden testien suhde onnistuneisiin. Näytettäviä testiajoja voi rajata testistatuksen sekä Filter-hakukentässä kuuden ensimmäisen sarakkeen tietojen mukaan. Myös itselle turhia sarakkeita voi rajata vähemmäksi.

Test	Test Package	Run Targets	Device Build	Product	Labels	Created	Status	Test Failures
A TS:n toimivuustarkistus CTS 9.0 (32bit) R13	CTS 9.0_r13		MR5	sdm710		3 months ago	Completed	0 / 1
Oma CTS 9.0 (32bit) R13	CTS 9.0_r13	+1	MR5	sdm710		3 months ago	Completed	20 / 476098
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13		MR5	sdm710	Sharding	3 months ago	Completed	2 / 471541
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13	+1	MR5	sdm710	Sharding	3 months ago	Completed	15 / 471541
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13	+1	MR5	sdm710	Sharding	3 months ago	Completed	20 / 471541
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13		MR5	sdm710		3 months ago	Canceled	50 / 476098
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13		MR5	sdm710		3 months ago	Completed	157 / 476098
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13		MR5	sdm710		3 months ago	Error	0 / 0
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13		MR5	sdm710		3 months ago	Canceled	16 / 471541
Oma CTS 9.0 (ARM-64) R13	CTS 9.0_r13		MR5	sdm710		3 months ago	Completed	126 / 471541

KUVA 7. Testiajonäkymä

View-painikkeella saa näkyviin kyseisen testiajon tarkemmat tiedot (24). Tässä näkymässä käynnistettäessä määritetyt testikierrokset näkyvät omina töinään ja

jokaisesta nähtävillä omat tarkemmat tiedot, kuten status, testitulokset ja lokit (kuva 8).

The screenshot shows the Android Test Station interface. At the top, it says 'Android Test Station' and 'Version: prod_R9.202009.001 ADB: 29.0.6-6198805'. The main content area is titled 'Test Results' and shows a specific test run: 'Test Run b1713259-da5a-4282-8d70-37728a98bf4e'. The status is 'Completed' with a red triangle icon and '22 / 471541'. Below this, there are buttons for 'Rerun', 'View Output Files', and 'Export Result'. The configuration section shows 'Test: Oma CTS 9.0 (ARM-64) - CTS 9.0_r13', 'Previous Run: 96f09991-1ae3-4129-b289-d614682e6249', and 'Labels:'. The 'Devices Tested' section shows a table with columns for Serial, Product, and Build ID. The table has one row with a redacted serial number, 'sdm710', and 'MR4.1'. Below this, there is a table with columns for Progress, Jobs, Logs, Test Results, Test Resources, and Config. The table shows several test attempts, all with a status of 'Completed' and a red triangle icon. The first attempt is 'Attempt 36a9ef36-6505-48d6-b2ee-9306b0f7b385' with '33 / 471456'. The second is 'Attempt efc4a806-e127-4d4f-bff7-f2f8855aa0bb' with '24 / 33'. The third is 'Attempt dd91293e-8d3a-452d-9892-f2af59c12f7b' with '22 / 24'. The fourth is 'Attempt f04bf213-979d-4603-aff3-aa4a763bb06d' with '22 / 22'. The final row shows '08/29/2020 7:28 AM | Test run reached final state'.

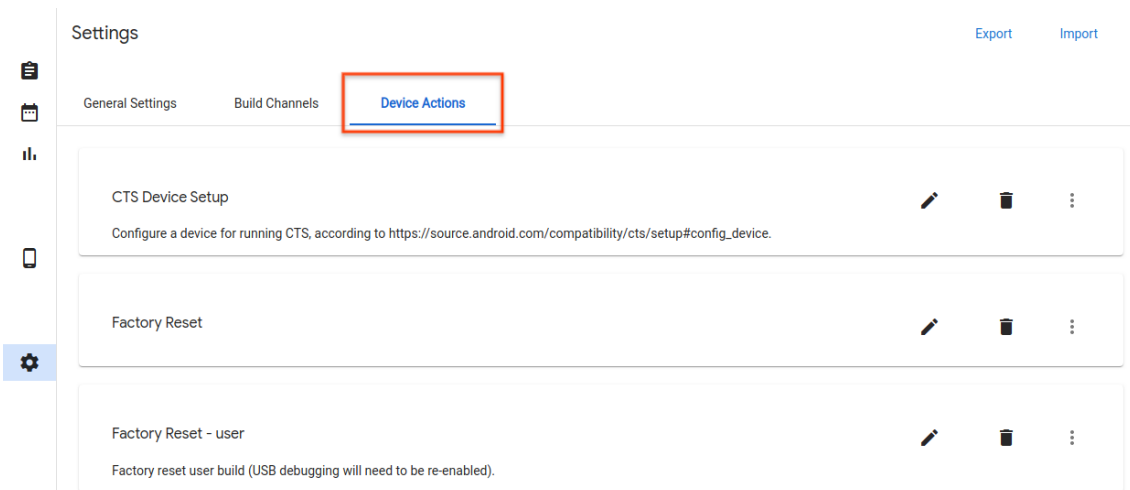
KUVA 8. Yksittäisen testiajon tarkastelunäkymä

Käynnissä olevan testiajon voi halutessaan keskeyttää sekä valmistuneen testiajon käynnistää uudelleen. Uudelleenkäynnistyksessä ohjelma tarjoaa samat vaiheet kuin testiajon ensimmäisessäkin käynnistyksessä ja käytettyjä parametreja voi muokata. Halutessaan voi lisätä tai poistaa käytettäviä laitetointoja. Tulokset ja muut testiajon tuottamat tiedostot voi kopioida ohjelmasta verkkolevylle tarkempaa tarkastelua ja dokumentaatiota varten. (24.)

Asetukset-välilehdeltä löytyvät Google Cloud-, Drive- ja Partner -palveluiden aktivointi sekä laitetointojen muokkaus ja luonti (kuva 9).

4.1.2 Laitetoiminnot eli Device actionit

Automaatiota helpottamaan Android Test Stationissa on Device actioneitä eli laitetointoja. Näillä korvataan testiajon vaatimia manuaalisesti tai skripteillä tehtäviä toimenpiteitä. Julkisesti ohjelmassa on saatavilla muutamia yleisiä laitetointoja, kuten Flash, Reboot, Factory Reset ja CTS Device Setup (kuva 9) (20).



KUVA 9. Device Actions -valikko (25, Figure 35)

CTS Device Setup muun muassa siirtää mediatiedostot laitteeseen, liittää testattavan laitteen ATS:n asetuksissa määriteltyyn WiFi-verkkoon sekä tekee muita muutoin manuaalisesti tehtäviä asetusmuutoksia. Laitetoiminnot ovat pohjimmiltaan skriptejä, jotka käyttävät Androidin lähdekoodista löytyviä luokkia ja funktioita. Laitetoimintoja voi myös luoda itse ja Google tarjoaa hyvät ohjeet tähän. (25.) Partnereille ATS tarjoaa myös muita laitetoimintoja, joista olen kertonut tarkemmin yhtiön sisäisessä dokumentaatiossa.

4.2 Työn aloitus ja eteneminen

Aloitin työn selvittämällä ja kartoittamalla, minkälaista laitteistoa ja ohjelmistoa ATS:n käyttö vaatii ja mitkä olisivat sen tärkeimmät ja välttämättömimmät ominaisuudet TM2-ylläpitoprojektin kannalta. Loin työn etenemissuunnitelman, johon sisällytin mahdolliseen käyttöönottoon vaaditut kriteerit sekä muita asioita, joita tulisi työn aikana selvittää, arvioida ja ottaa huomioon. Näitä olivat esimerkiksi käytön ja ylläpidon helppous, yhteensopivuus projektin muiden järjestelmien kanssa sekä kustannustehokkuus.

Arvioin tarvitsevani noin kaksi kuukautta ohjelman käytön opetteluun, tutustumiseen ja mahdollisimman perusteelliseen arviointiin. Sovimme esimieheni kanssa tämän suuruisen aikavälin elokuun puolesta välistä lokakuun puoleenväliin, jonka aikana jättäisin nykyiset muut työtehtäväni vähemmälle ja keskittyisin

ATS:n tutkimiseen. Tämän jälkeen jatkaisin normaaleja työtehtäviäni ja työstäisin opinnäytetyön raporttia muiden töiden ohella ja vapaa-aikana.

Tein yhtiön sisäisen Confluence-sivun, jonne dokumentoin työn eri vaiheet, suunnitelman, tarpeellisen laitteiston, työn aikana tekemäni havainnot, esiintyneet ongelmat ja näiden mahdolliset ratkaisut. Jätin esimiehelleni anomuksen laitetarpeista ja laitteet saatuani tein ATS:n, projektin ja TM2:n vaatimat asetusmuunnokset sekä taustaohjelmien asennukset.

Android Test Stationiin tutustumisen aloitin kahdella testattavalla TM2:lla ja yhdellä testikoneella, jossa oli Ubuntu 18.04 -käyttöjärjestelmä, 8 GB keskusmuistia ja 500 GB:n kovalevy, josta vähintään 250 GB vapaata tallennustilaa. Aikomuksena oli myöhemmässä vaiheessa tarpeen mukaan lisätä testattavia laitteita ja testikoneita. Ohjelman asensin Googlen ohjesivun manuaalitavan mukaisesti. Samalla tein tästä asennuksesta tiivistetyn ohjeen projektin käyttöön, sillä Googlen tarjoamat ohjeet oli jaoteltu kolmelle eri verkkosivulle. (1; 26; 27.) Myöhemmässä vaiheessa kokeilin myös Googlen tarjoamaa asennustyökalua toiseen tehokkaampaan testikoneeseen.

Työn aikana testasin kaikkien Test Suitejen toimintaa ajamalla sekä yksittäistestejä että täysiä testipaketteja sekä kehityksen alla olevilla päivittäiskoontiversioilla että jo julkaisuhyväksynnän saaneilla ylläpitokoontiversioilla useita kertoja. Laitetoimintojen toimivuudet tarkistin ensin yksittäin ilman Test Suitea. Toiminnan varmistuttua kokeilin useita laitetointoja yhtäaikaisesti, jonka jälkeen varmistin näiden toimivuuden vielä eri Test Suitejen kanssa yhteisajossa. Esiintyneet ongelmat, näille löytyneet ratkaisut sekä muut havainnot kirjasin Confluence-dokumentaationsivulle.

ATS:n käytön tutkimisen yhteydessä jatkoin pienessä määrin myös aikaisempia työtehtäviäni ajaen Googlen hyväksyntätestejä nykyisellä automaatiojärjestelmällä. Täten sain hyvin myös vertailtua mm. testiajojen kestoa ja stabiiliutta, käytön helppoutta sekä muita yleisen käytön eroavaisuuksia. Tutkinnalle varatun ajan jälkeen jatkoin normaaleja työtehtäviäni ja ryhdyin kirjoittamaan tätä raporttia vapaa-aikana sekä tilanteen salliessa muiden työtehtävien ohella.

5 HAVAINNOT

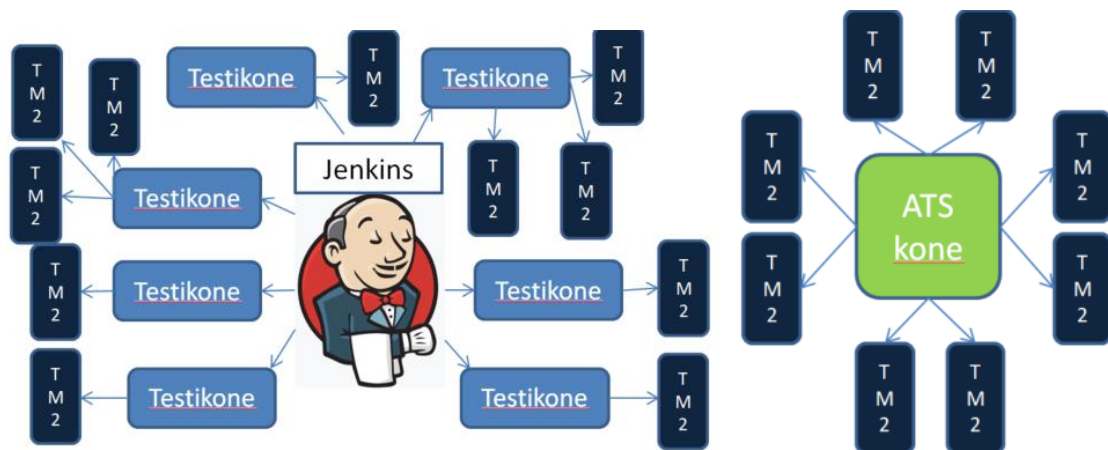
Tässä luvussa kerron tekemistäni havainnoista julkisesti saatavilla olevien Test Suitejen ja ominaisuuksien käytöstä ja toimivuudesta. Salassapidon alaisista havainnoista löytyy tietoa yhtiön sisäisestä dokumentaatiosta.

5.1 Käyttöönoton puolesta

Kerron tarkemmin Android Test Stationin hyvistä puolista ja eduista nykyiseen järjestelmään verrattuna.

5.1.1 Fyysinen testiympäristö

Kuvassa 10 on kuvailtu, kuinka projektin nykyinen Jenkinsiä hyödyntävä testiautomaatiojärjestelmä eroaa Android Test Stationin mahdollistavasta järjestelmästä. Nykyisessä järjestelmässä on käytössä useita Jenkinsiin liitettyjä testikoneita ja useimmissa niissä on vain yksi testattava laite, koska yhdellä testikoneella pystyy ajamaan vain yhtä Test Suitea kerrallaan. Testikoneissa, joissa useampia testilaitteita, ajetaan kaikille siinä kiinni oleville testilaitteilla samaa Test Suitea aina pirstaloituna.



KUVA 10. Fyysisen testiajoympäristön vertailu Jenkinsin ja ATS:n väliltä

Android Test Stationissa puolestaan on mahdollista ajaa useita eri Test Suiteja eri puhelimille samanaikaisesti. Tälläkin tosin on rajansa. Googlen suosittelemalla 8GB:n keskusmuistin omaavalla testikoneella jo kolmen Test Suiten yhtä-

aikainen ajo voi olla liikaa ja testikone voi kaatua, mutta kahden yhtäaikainen ajo onnistuu ongelmitta. Tehokkaammalla testikoneella on useammankin Test Suiten yhtäaikainen ajo mahdollista. Tämä voisi potentiaalisesti tuoda projektille säästömahdollisuuksia laitteistokustannuksissa. Jos ATS otettaisiin projektissa käyttöön, ei yksi kone siltikään riittäisi, vaan tarvittaisiin useampia ylläpitopäivysten ja mahdollisten vikatilojen esiintymisen varalta. Testauksen ja testikoneiden tarve voi myös kasvaa ajan myötä siinä määrin, että täytyisi lisätä testikoneita ja on harkittava, kuinka tehokkaita testikoneita tultaisiin käyttämään.

5.1.2 Testien ajo ja yleinen käyttö

Molemmissa järjestelmissä voi asettaa halutun määrän uudelleenajokierroksia. Näiden valmistuttua täytyy nykyisellä järjestelmällä testiajoja jatkaa manuaalisesti terminaalinäkömästä. ATS:llä uudelleenajon voi käynnistää nappia painamalla ja tällöinkin voi valita uudelleenajokierrosten määrän. Yksittäis- ja moduulitestejäkin voi valita uudelleenajon yhteydessä.

ATS:llä voi aina testin käynnistysvaiheessa valita yhden tai useamman testattavan laitteen ja pirstaloida testit näiden kesken automaattisesti. Nykyisellä järjestelmällä skriptit on rakennettu siten, että testiajo käynnistettävä aina joko pirstaloituna tai yksittäiselle testilaitteelle ja aina samoille testilaitteille.

Nykyisessä järjestelmässä jokaisella testikoneella on asetustiedostoon määritetty siinä käytettävien testilaitteiden tiedot, kuten sarjanumerot ja puhelinnumerot. Tämän vuoksi testilaitteiden ja näiden SIM-korttien vaihdon yhteydessä täytyy nämä muutokset tehdä myös testikoneiden asetustiedostoihin. ATS:llä voi testilaitteita ja SIM-kortteja lisätä, poistaa ja vaihtaa enemmän Plug and Play -tyylisesti.

5.1.3 Ylläpito

Uuden Test Suite -version päivitys

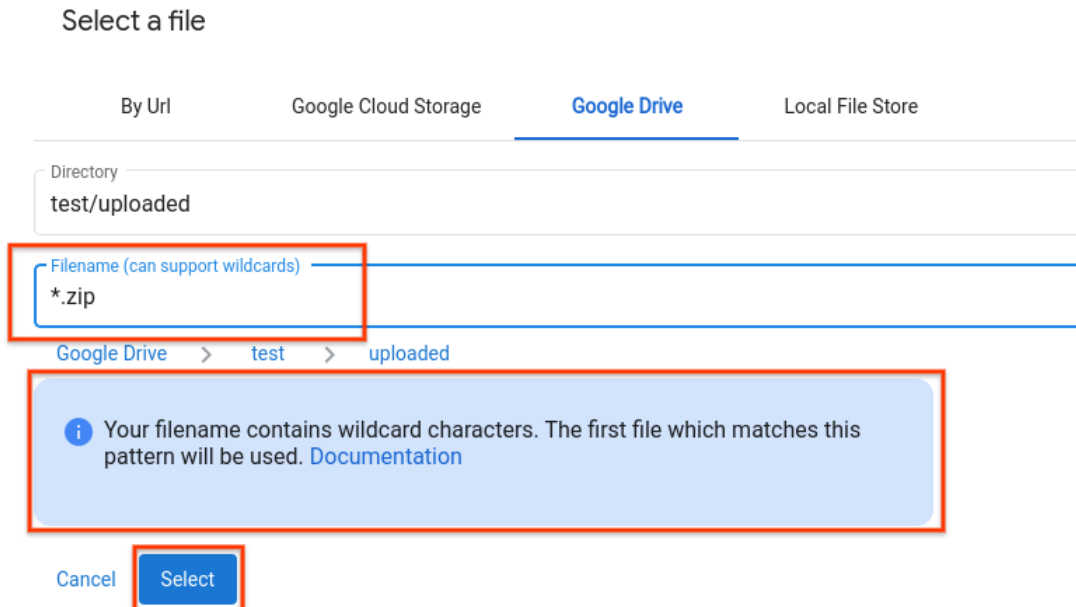
Test Suitet päivittyvät neljännesvuosittain ja nykyisellä järjestelmällä nämä käydään lataamassa zip-tiedostoina ennalta määritettyyn polkuun verkkolevylle. Tämän jälkeen jokaisella testikoneella ajetaan skripti, joka kopioi ja purkaa nä-

mä tiedostot ennalta määritettyyn testiajo-kansioon. Tämän jälkeen täytyy Jenkins-asetustiedostoihin käydä päivittämässä uusien Test Suite -versioiden binäärit ja uusien Test Suitejen pitäisi olla valmiit käytettäväksi. Ajetaan kuitenkin aina nopea testi toimivuuden tarkistamiseksi. ATS:llä Test Suite -version päivitystapa riippuu siitä, käytetäänkö lataustapana tarjottua verkko-osoitetta tai pilvipalveluita vai tallennetaanko se ATS:n paikallistiedostoksi.

Verkko-osoitteen käyttö on CTS- ja CTS Instant Apps -pakettien kanssa helpoin vaihtoehto, sillä tarjottu osoite on selkokielen ja riittää, että siihen päivittää uusimman Test Suite -versionumeron. Tällöin ohjelma lataa kyseisen Test Suite -version Googlen palvelimelta ensimmäisen testiajon käynnistyksessä. Jatkossa kyseinen versio on kontin muistissa, eikä sen uudelleenlataukseen kulu lisäaikaa. (21, Figure 15.)

Paikallistiedostoksi tallentamalla pitäisi Test Suite -versiot nykyiseen tapaan ladata verkkolevylle ja ladata se ohjelman käytettäväksi. Tämä oli itselleni mieluisin vaihtoehto, sillä Partner -pakettien verkko-osoitteet eivät ole julkisten tapaan selkeästi muokattavissa. Test Suite -versiot kopioitaisiin joka tapauksessa verkkolevylle talteen ohjelmistokehittäjien käytettäväksi. (21, Figure 18.)

Pilvipalveluratkaisuja ei tarkemmin testattu, aikaisemmin mainittujen sisäisten tietoturvamäärittysten vuoksi. Niiden ainoa mahdollinen etu kahteen edelliseen verrattuna olisi, ettei latauspolkua tarvitse erikseen muuttaa aina uuden version tullessa (kuva 11).



KUVA 11. Google Drive -latauspolun valinta (21, Figure 17)

Latauspolun voi asettaa aina valitsemaan uusin tai suurimman binäärin omaava versio sille asetetusta kansioista (21, Figure 16 & 17). Tällöinkin pitäisi kyllä kaikki Test Suite -versiot käydä erikseen lataamassa näihin pilvipalveluratkaisujen kansioihin. Mediatiedostojen sekä muiden mahdollisten testiresurssien lataus ja käyttö tapahtuu edellä mainittujen kanssa vastaavanlaisesti.

Testikoneen uudelleenasetus ja uuden testikoneen käyttöönotto

Toisinaan on tarpeen ottaa käyttöön uusi tai asentaa uudelleen vanha testikone. Nykyisessä järjestelmässä tätä prosessia varten on projektissa luotu automaattioskripti, joka tekee tarvittavat asetusmuutokset sekä asentaa ennalta määritetyt ohjelmat. Uuden testikoneen käyttöönottoon kuuluu lisäksi manuaalinen asetustiedoston luonti sekä muita mahdollisia manuaalisesti tehtäviä toimenpiteitä.

Android Test Stationin asennus on manuaalisestikin suhteellisen helppoa Dockerin ansiosta. Taustaohjelmia ei tarvitse juurikaan asentaa itse, vaan ne tulevat konttiin ATS:n mukana. Kertaalleen ympäristöön sopivaksi asetetusta testikoneesta voi ottaa asetustiedostokopion, jonka uuteen testikoneeseen ATS:ään lisäämällä saadaan samat laitetoiminnot ja testisuunnitelmat suoraan käyttöön. Halutessaan voi ottaa käytössä olevasta ATS:stä täyden varmuuskopion, jonka voi syöttää joko uuteen tai uudelleenasetettuun testikoneeseen.

Täydessä varmuuskopiossa tulevat laitetoimintojen, asetusten ja testisuunnitelmien lisäksi kaikkien testiajojen tulokset ja lokit sekä paikallistiedostoiksi tallennetut tiedostot.

5.2 Käyttöönottoa vastaan

Suurta osaa käyttöönottoa vastaan olevista havainnoista ei voi suoraan vertailla nykyiseen järjestelmään, vaan ne ovat lähinnä ominaisuuksia, jotka voisivat toimia paremminkin ja voivat vaatia hieman laajempaa selitystä. Selkeyttääkseni kuvauksia olen pyrkinyt jaottelemaan havainnot aihealueittain.

Testiajonäkymä

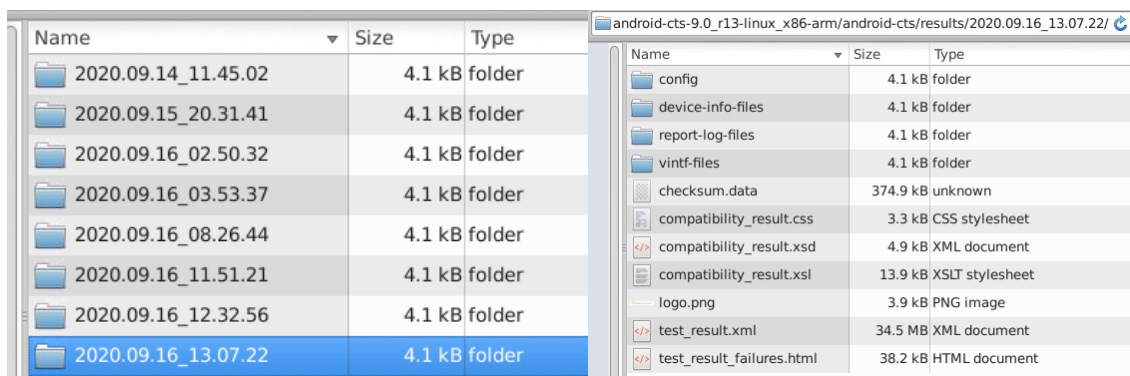
Koska testiajot listautuvat luomisajan mukaan niin, että tuorein on aina ylimmäisenä, eikä tätä järjestystä voi muuttaa kuin hakurajauksella, hautautuvat pitempään kestävätestiajot jatkuvan testauksen seurauksena näkymättömiin (kuva 7). Tästä syystä voi testaajalla unohtua kyseisen testiajon tarkistus, ellei hän ole tehnyt erillistä muistutusta itselleen kaikista testiajoista. Nykyisessä järjestelmässä tämä ratkaistu siten, että järjestelmä lähettää testaajalle sähköpostin testiajon valmistuttua. Tätä tai vastaavaa ominaisuutta ei tällä hetkellä ATS:stä löydy. Yksi tapa tämän parannukseen voisi olla esimerkiksi pitää käynnissä olevat testiajot aina listan ylimmäisimpinä ja järjestämällä listauksen testiajojen valmistumisajan mukaan.

Luomisaika on ilmoitettu siitä kuluneen ajan mukaan (kuva 7). Selkeämpi olisi, jos luomisaika ilmoitettaisiin päivämäärällä ja kellonajalla. Toisinaan voisi haluta järjestää testiajot esimerkiksi käytetyn Test Suiten mukaiseen aakkosjärjestykseen. Sen lisäksi olisi toivottavaa, että tässä näkymässä ilmoitettaisiin myös testiajojen kokonaiskestot.

Testiajoja ei ole mahdollista poistaa testiajonäkymästä. Toisinaan olisi tarpeen poistaa esimerkiksi keskeytettyjä ja useamman kuukauden vanhoja testiajoja näkymän selkeyttämiseksi. Näytettäviä testiajoja voi kyllä rajata, mutta nämä rajoitukset eivät tallennu, vaan poistuvat aina näkymästä pois käytäessä. Testiajojen tuottamat loki ja tulostiedostot on mahdollista poistaa, mutta testiajot silti jäävät testiajonäkymään tarkasteltavaksi.

Testitulosten sekä lokien taltiointi

Testitulosten ja lokien kopiointi Android Test Stationista on epäkäytännöllistä ja aikaa vievää. Manuaalisesti sekä Jenkinsillä ajettujen testikierrosten tulokset ja logit tallentuvat Test Suite -kansion logs- ja results-kansioihin, josta tulokset on helppo kopioida verkkolevylle talteen. Epäonnistuneiden testien tarkemmat tulokset voi tarkistaa useammastakin tiedostosta, mutta helpoin on käyttää test_result_failures.html-tiedostoa (kuva 12).



KUVA 12. Manuaalisesti ajettujen testikierrosten tulokansiot sekä sisältö

Kuvan 8 mukaisesta yksittäisen testiajon näkymästä Export Result -painike tarjoaa vain kyseisen testiajon viimeisimmän testikierroksen tulokansion, kun pitäisi saada taltiointia tulokset kaikilta kierroksilta. Tässä tulokansiossa tulosten tarkempaan selaukseen on tarjolla ainoastaan test_result.xml, joka on html-tiedostoa raskaampi käsitellä, eikä sisällä kaikkia tarvittavia tietoja.

View Output Files -painikkeesta avautuu kuvan 13 mukainen näkymä, jossa on yhden kansion alle listattuna kaikki sen testikierroksen tuottamat tiedostot, mukaan lukien edellä mainitusta tulokansioista puuttuva html-tulostiedosto. Tästä näkymästä voi tiedoston avata ja tarkastella tuloksia tarkemmin.

data / app_default_bucket / test_runs / c31f6766-0e85-477c-a021-047c2af8f538 / output / 5843477422145536 / fc5cc5a3-f893-49ad-8538-400b748a5c74

Name	↓	Mimetype	Modified	Size
tool-logs	↓	inode/directory	2020.12.10 09:07:23	
2020.12.10_08.33.04.zip	↓	application/zip	2020.12.10 09:07:23	551.89 KiB
compatibility_result.css	↓	text/css	2020.12.10 09:07:23	3.26 KiB
compatibility_result.xsl	↓	application/xslt+xml	2020.12.10 09:07:23	16.17 KiB
device_compatibility_matrix.xml_16617508782342409230.txt.gz	↓	text/plain	2020.12.10 09:07:23	382 B
device_logcat_setup_KM184500436_5611416642390279153.txt.gz	↓	text/plain	2020.12.10 09:07:23	111.99 KiB
device_logcat_tearardown_KM184500436_8195614554397740111.txt.gz	↓	text/plain	2020.12.10 09:07:23	80.51 KiB
device_logcat_test_KM184500436_16010642697473830953.txt.gz	↓	text/plain	2020.12.10 09:07:23	667.71 KiB
device_manifest.xml_14220162149517266753.txt.gz	↓	text/plain	2020.12.10 09:07:23	2.44 KiB
executeShellCommandLog_KM184500436_11109674620106103964.txt.gz	↓	text/plain	2020.12.10 09:07:23	8.02 KiB
FILES	↓	text/plain	2020.12.10 09:07:23	3.41 KiB
framework_compatibility_matrix.xml_10874138422751316576.txt.gz	↓	text/plain	2020.12.10 09:07:23	6.33 KiB
framework_manifest.xml_3448140782170039535.txt.gz	↓	text/plain	2020.12.10 09:07:23	636 B
host_adb_log_2476492349517961062.txt.gz	↓	text/plain	2020.12.10 09:07:23	556 B
host_log_16084537967669519772.txt.gz	↓	text/plain	2020.12.10 09:07:23	88.96 KiB
logo.png	↓	image/png	2020.12.10 09:07:23	3.77 KiB
suite-exclude-filters5736311400760723403.txt_7629754878306686627.txt.gz	↓	text/plain	2020.12.10 09:07:23	56.08 KiB
suite-include-filters119219492765113512.txt_3833250291051178559.txt.gz	↓	text/plain	2020.12.10 09:07:23	1.21 KiB
test_result.html	↓	text/html	2020.12.10 09:07:23	4.21 MiB
test_result.xml	↓	application/xml	2020.12.10 09:07:23	1.18 MiB
test_result_failures_suite.html	↓	text/html	2020.12.10 09:07:23	52.23 KiB

KUVA 13. ATS:llä ajetun yksittäisen testikierroksen tuottamat tiedostot

Menemällä kansiorakenteessa askeleen ylöspäin avautuu kuvan 14 mukainen näkymä, jolloin voi kopioida talteen koko testiajon testikierrokskansiot. Nämä kansiot tosin on nimetty hyvin sekavasti, eikä niistä nimen perusteella erota niiden järjestystä. Lisätyötä tuottaa niiden selkeä nimeäminen sekä näiden jälkikäteen tarkastelu, sillä ne ovat TGZ-tiedostomuotoon pakattuja.

data / app_default_bucket / test_runs / c31f6766-0e85-477c-a021-047c2af8f538 / output / 5843477422145536

Name	↓	Mimetype	Modified	Size
36c4de65-9cc6-4a22-a7f3-2d5f5df45634	↓	inode/directory	2020.12.10 08:27:22	
fc5cc5a3-f893-49ad-8538-400b748a5c74	↓	inode/directory	2020.12.10 09:07:23	

KUVA 14. Yhden ATS -testiajon kahden testikierroksen tuloksansiot

Laitetoiminnot

Valitettavasti kaikki tarjotut laitetoiminnot eivät olleet Tough Mobile 2:n kanssa yhteensopivia. Tästä johtuen suurin ATS:n käyttöönottoa vastaan toiminut ominaisuus oli, ettei ATS:n tarjoama koontiversion flässäysprosessi ole yhteensopiva Bittiumin Tough Mobile 2:lle luodun prosessin kanssa. Sain kuitenkin luotua laitetoiminnon, joka mahdollistaa testilaitteen manuaalisen flässäyksen samalla testikoneella. Tämän vuoksi manuaalisen työn ja testaukseen käytettävän ajan määrä lisääntyisi merkittävästi. Nykyisessä järjestelmässä Jenkins suorittaa halutun koontiversion flässäyksen testilaitteeseen testiajon käynnistyksen yhteydessä. Ainoat täysin toimivat laitetoiminnot olivat Reboot sekä CTS Device Setup.

Testien ajo ja yleinen käyttö

Yksittäisten testien ja lyhyiden testimoduulien ajo ATS:llä on manuaaliseen ajotapaan verrattuna hidasta. ATS luo jokaiselle testiajolle aina oman konttinsa ja lataa siihen tarvittavat testiresurssit. Manuaalisesti ajettuna yhden minuutin kestävän testin ajoon ilman laitetoimintoja ja etukäteen ladattua Test Suite -versiota käyttäen menee ATS:llä ajettuna noin neljä minuuttia. Mikäli Test Suite -versio täytyy ladata verkosta, kestää ATS:llä noin seitsemän minuuttia. Tätä kestoa ei edes huomaa isompia testipaketteja ajettaessa, mutta lyhytkestoisten testien ajo voi tästä syystä olla testaajalle turhauttavaa. Kuitenkaan ATS:n käynnissä ollessa ei samalla koneella ole mahdollista tehdä manuaalitestiajoja. Terminaali-ikkunasta Test Suiten käynnistys, testien ajo sekä muiden liitännäisten taustaohjelmien käyttö ATS:n käynnissä ollessa voi häiritä ATS:n toimintaa ja ohjelma täytyisi sammuttaa ja käynnistää uudelleen. Yksittäisten testien ajo on hyvin tärkeää korjausten verifiointissa sekä virhelokien talteenotossa.

Mielestäni sopivin vaihtoehto testiresurssien käytölle on paikallistiedostoiksi tallentaminen. Verkko-osoitteen käyttö julkisten Test Suite -pakettien ja mediatiedostojen kanssa on helppoa, mutta hidasta. Hyväksyntätestetit on tehtävä aina tuoreimmalla Test Suite -versiolla, mutta verkko-osoite oletusarvoisesti ei osoita tuoreimpaan julkaistuun Test Suite -versioon. Tarjottu versio voi olla jopa vuoden vanha. Ainoastaan julkisesti saatavilla olevien Test Suitejen verkko-osoite

on selkeä ja helposti päivitettävissä. Partner-pakettien osoite täytyisi itse etsiä ja kopioida. Tämä olisi Googlen puolelta helposti ratkaistavissa, sillä ATS:stä julkaistaan lähes kuukausittain uusi versio, jonka yhteydessä nämä osoitteet voisi päivittää.

Vajaan kahden kuukauden käytön jälkeen alkoi testikoneen massamuistitila käydä vähiin. Tilaa ei ollut varmuuskopion ottamiseen, joten jouduin poistamaan muuhun testaukseen käytettäviä tiedostoja. Varmuuskopion koko zip-tiedostoksi pakattuna oli yli 70 GB. Siirsin varmuuskopion verkkolevyille ja poistin yli viikon vanhat testiajojen tiedostot. Massamuistitilaa oli kuitenkin yli kaksi kertaa enemmän Googlen suositukseen nähden.

Yleisesti ottaen ATS toimii suhteellisen hitaasti ja vaikuttaa kohtalaisen epävaakalta. Kahden kuukauden aikana ajoin ATS:llä yli 150 testiajoa, joista jopa kaksi kolmasosaa keskeytyi tai muutoin epäonnistui testaajasta johtumattomista syistä. Testiajot saattoivat keskeytyä ilman selkeää syytä tai ohjelma kadotti yhteyden testattavaan laitteeseen. Työn aikana ATS sai useaan otteeseen testikoneen toimimaan niin hitaasti, että se katkaisi etäyhteyden eikä siihen saanut paikallisestikaan yhteyttä. Usean tunnin odotuksen jälkeenkään koneeseen ei saanut yhteyttä ja kone täytyi sammuttaa ja käynnistää uudelleen toiminnallisuuden palauttamiseksi. Toisinaan ATS väitti testiajon valmistuneen, vaikka olisi ollut tuhansia testejä ajamatta.

6 YHTEENVETO

Aiheena oli tutkia Android Test Stationin käyttöä ja ominaisuuksia. Tavoitteena oli tuottaa perusteltuja argumentteja sen käyttöönoton puolesta ja vastaan. Tarkoitus oli lopulta verrata sitä nykyiseen testiautomaatiojärjestelmään ja selvittää, onko sitä kannattavaa ottaa lainkaan käyttöön, voisiko sitä käyttää nykyisen järjestelmän rinnalla vai voisiko ATS kokonaan korvata nykyisen järjestelmän.

On myönnettävä, että itselläni oli jo ennen työn aloitusta muodostunut suhteellisen vahva mielipide, että Android Test Stationia ei kannata ottaa käyttöön. Tämä vahvisti hankaluuksien ilmentyessä negatiivista näkemystä ohjelmasta. Pyrin toiminnallisuuksia tutkiessani kuitenkin pysymään mahdollisimman objektiivisenä. Mielipide työn edetessä heittelikin yllättävän usein puolelta toiselle dokumentoimattomien ominaisuuksien ilmentyessä.

Android Test Stationissa on, käyttöönottoa vastaan kirjoittamistani havainnoista huolimatta, paljon lupaavia ja potentiaalisesti testaustyötä helpottavia ominaisuuksia. Google kehittää ATS:ää käyttäjiltä saamiensa palautteiden perusteella jatkuvasti. Ohjelman R1-versio julkaistiin joulukuussa 2019 ja R9 tuli käyttöön syyskuun loppupuolella. Valitettavasti ainakin R9-version aikana ohjelma oli kuitenkin vielä liian epävakaa oloinen ja siitä puuttui tiettyjä projektille tärkeitä ominaisuuksia, joten sitä ei ainakaan vielä oteta käyttöön. Tätä raporttia kirjoittaessani olen seurannut uusien versioiden julkaisukoosteita. Ohjelmaan on tulutkin näiden mukaan useita uusia ja mielenkiintoisia ominaisuuksia sekä korjauksia, joiden perusteella olen osan tekemistäni havainnoista jättänyt pois. (28.)

Itse työ, eli ATS:iin tutustuminen ja arviointi, onnistui sille varatussa ajassa, mutta selkeästi joko aliarvioin raporttiin tarvittavan ajan tai yliarvioin oman kyvykkyyteni sen kirjoittamiseen. Täten kirjoitusprosessi venyi mielestäni turhan pitkäksi. Tästä kuitenkin opin jatkossa huomioimaan raporttien kirjoittamiseen vaadittavan ajan aikatauluja suunniteltaessa. Kirjoitusprosessia viivästytti myös salassapidon alaisen osuuden karsiminen raportista sekä ilmentyneet työkiireet.

ATS:n kehityksen seurausta ja dokumentointia tullaan projektissa jatkamaan ja projektille tärkeiden ominaisuuksien ilmaantuessa tullaan ATS todennäköisimmin ottamaan ainakin osittaiseen käyttöön.

LÄHTEET

1. Android Test Station. 2020. Android. Saatavissa: <https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide>. Hakupäivä 3.8.2020.
2. About Bittium. 2020. Bittium. Saatavissa: <https://www.bittium.com/about-bittium/facts-figures/company-overview>. Hakupäivä 20.9.2020.
3. Athrow, Desire. 2020. This is probably the world's most secure smartphone right now. TechRadar. Saatavissa: <https://www.techradar.com/news/this-is-probably-the-worlds-most-secure-smartphone-right-now>. Hakupäivä 13.10.2020.
4. New Standard for Ultra Secure Mobile Communications. 2020. Bittium. Saatavissa: <https://toughmobile2.bittium.com/> Hakupäivä 17.8.2020.
5. Bittium Tough Mobile 2. 2020. Bittium. Saatavissa: <https://webshop.bittium.com/product/70/bittium-tough-mobile-2>. Hakupäivä 17.8.2020.
6. What is Android. 2020. Android. Saatavissa: <https://www.android.com/what-is-android/>. Hakupäivä 10.11.2020.
7. Android Compatibility Program Overview. 2020. Android. Saatavissa: <https://source.android.com/compatibility/overview>. Hakupäivä 10.11.2020.
8. Device compatibility overview. 2020. Android. Saatavissa: <https://developer.android.com/guide/practices/compatibility>. Hakupäivä 10.11.2020.
9. Java Development Kit for Ubuntu. 2020. Android. Saatavissa: <https://source.android.com/compatibility/cts/setup#JDK>. Hakupäivä 3.8.2020.
10. Compatibility Test Suite Downloads. 2020. Android. Saatavissa: <https://source.android.com/compatibility/cts/downloads>. Hakupäivä 3.8.2020.
11. Running CTS tests. 2020. Android. Saatavissa: <https://source.android.com/compatibility/cts/run>. Hakupäivä 3.8.2020.
12. CTS v2 Command Console. 2020. Android. Saatavissa: <https://source.android.com/compatibility/cts/command-console-v2>. Hakupäivä 3.8.2020.

13. Android ABIs. 2020. Android. Saatavissa:
<https://developer.android.com/ndk/guides/abis>. Hakupäivä 10.11.2020.
14. CTS v2 sample test summary. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/cts/interpret#cts-v2-sample-test-summary>. Hakupäivä 3.8.2020.
15. Android device configuration. 2020. Android. Saatavissa:
https://source.android.com/compatibility/cts/setup#config_device. Hakupäivä 3.8.2020.
16. Build great things at any scale. 2020. Jenkins. Saatavissa:
<https://www.jenkins.io/>. Hakupäivä 3.8.2020.
17. What is a Container? 2020. Docker. Saatavissa:
<https://www.docker.com/resources/what-container>. Hakupäivä 10.11.2020.
18. Selecting a test. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#select-a-test>. Hakupäivä 11.11.2020.
19. Configuring test run. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#configure-test-run>. Hakupäivä 11.11.2020.
20. Adding device actions. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#add-device-actions>. Hakupäivä 11.11.2020.
21. Setting test resources. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#set-test-resources>. Hakupäivä 11.11.2020.
22. Creating a test plan. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#create-a-test-plan>. Hakupäivä 11.11.2020.
23. Viewing test runs. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#test-run-list>. Hakupäivä 11.11.2020.
24. Test run details. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#test-run-details>. Hakupäivä 11.11.2020.

25. Creating a new device action. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-user-guide#create-a-new-device-action>. Hakupäivä 11.11.2020.
26. Install Docker Engine on Ubuntu. 2020. Docker. Saatavissa:
<https://docs.docker.com/engine/install/ubuntu/>. Hakupäivä 3.8.2020.
27. Post-installation steps for Linux. 2020. Docker. Saatavissa:
<https://docs.docker.com/engine/install/linux-postinstall/>. Hakupäivä 3.8.2020.
28. Android Test Station Release Notes. 2020. Android. Saatavissa:
<https://source.android.com/compatibility/tests/development/android-test-station/ats-release-notes>. Hakupäivä 3.8.2020.

Yksinkertainen Compatibility Test Suiten testiajoskriptiesimerkki

```
1  #!/bin/bash
2
3  CTS="android-cts-9.0_r13-linux_x86-arm" #Test suite version variable
4  CTSM="android-cts-media-1.5"           #CTS Media folder variable
5
6  #####
7  # Copy media files to DUT
8  #####
9
10 cd /Path/to/$CTSM/$CTSM/
11 ./copy_media.sh all
12 ./copy_images.sh
13
14 #####
15 # Delete old CTS logs and results
16 #####
17
18 cd /Path/to/$CTS/android-cts/logs/
19 rm -fr 2020*
20 cd /Path/to/$CTS/android-cts/results/
21 rm -fr 2020*
22
23 #####
24 # Start CTS testing
25 #####
26
27 cd /Path/to/$CTS/android-cts/tools/
28 ./cts-tradefed run commandAndExit cts --abi arm64-v8a      #Launches cts-tradefed and full 64bit testrun
29 #./cts-tradefed run commandAndExit cts --abi armeabi-v7a  #Launches cts-tradefed and full 32bit testrun
30 adb reboot
31 sleep 120
32 ./cts-tradefed run commandAndExit retry --retry 0
33 adb reboot
34 sleep 120
35 ./cts-tradefed run commandAndExit retry --retry 1
36 adb reboot
37 sleep 120
38 ./cts-tradefed run commandAndExit retry --retry 2
```