



Road Runner
Sjálfvirkar gæðaprófanir á
fjarskiptaneti Nova

Vor 2013

Guðmundur Sveinsson

Jens Gísli Heiðarsson

BSc Tölvunarfræði

Leiðbeinandi: Birgir Kaldal Kristmannsson
Prófdómari: Stefán Freyr Stefánsson

T-404-LOKA
Tölvunarfræðideild

Inngangur	2
Lýsing verkefnis	2
Skipulag	3
Aðferðafræði	3
Hlutverkaskipting.....	4
Greining	4
Tól.....	5
Forritun	6
Android App	6
Vefþjónusta	6
Vefviðmót.....	7
Prófanir	8
Kerfisprófanir.....	8
Notendaprófanir.....	9
Framvinda	9
Sprettur 0	9
Sprettur 1	10
Sprettur 2	10
Sprettur 3	11
Sprettur 4	12
Sprettur 5	12
Sprettur 6	14
Samantekt	15
Tímar.....	15
Umsögn tengiliðs.....	16

Inngangur

Fjarskiptakerfi Nova er stórt og flókið fyrirbæri. Því er stýrt með mörgum færíbreytum (e. parameters) sem hafa áhrif á hegðun og gæði kerfisins. Í kerfinu eru gerðar um 1000 breytingar (stillingar) á ári til að ná fram sem bestri þjónustu við viðskiptavini. Hjá Nova eru keyrslupróf (e.drive-test) mikið notuð til þess að mæla gæði fjarskiptakerfisins en það eru prófanir sem gerðar eru af starfsmönnum og aðstoðarmönnum þeirra. Keyrt er um svæði sem dreifikerfi Nova nær til og hin ýmsu gildi mæld til að meta gæði kerfisins á hverjum tíma. Hingað til hafa prófarar þurft að nota sérhæfð símtæki sem stillt eru sérstaklega fyrir hverja prófun. Eftir að keyrslu lýkur hefur þurft að framkvæma innflutning (e.import) inn í greiningarkerfi til þess að sjá niðurstöður prófana.

Markmið þessa verkefnis var að smíða hugbúnað (e. app) í Android síma til að framkvæma þessar prófanir á einfaldan og sjálfvirkan máta. Road Runner gerir prófum kleift að fylgjast með gæðum kerfisins, sambandsstyrk á hverjum mældum stað og hvort slit verði á símtölum. Sérfæðingar geta skoðað niðurstöður prófana í vefviðmóti sem sendar eru sjálfvirkt inn með Android Appinu.

Lýsing verkefnis

Verkefnið stendur saman af fjórum hlutum: Android „appi“, vefþjónustu, gagnagrunn og vefviðmóti.

Í vefviðmótinu geta prófarar skráð símana sem verða notaðir í prófanir ásamt forskriftum fyrir þá. Forskrifta er plan fyrir símann, þ.e. hvenær hann á að hringja yfir daginn og hvað símtölin eiga að vera löng. Þessar forskriftur eru síðan skráðar í grunn og bíða eftir að síminn sæki þær. Í byrjun hvers sólarhrings, klukkan 00:00, nær appið í nýjustu forskriftuna sem skráð er á viðeigandi síma. Þegar appið fær forskriftuna senda setur það niður tímasetningar á símtölum yfir daginn. Þegar hvert símtal byrjar setur appið niður lengd símtalsins og á 5 sekúndna fresti tekur það „test“ sem inniheldur staðsetningu símans og helstu mæligildi sem til hans ná. Þegar appið skellir svo á sendir það niðurstöðurnar úr símtalinu til vefþjónustu sem skráir þær í grunn.

Þegar þær eru komnar í grunninn getur prófarinn skoðað allar niðurstöður sem eru í boði inni á vefviðmótinu undir „niðurstöðu“ flípanum. Þar getur hann líka takmarkað niðurstöðurnar og jafnvel séð aðeins úr prófunum sem hann framkvæmdi sjálfur.

Þegar prófarinn er búinn að finna þær niðurstöður sem hann vill skoða betur, hakar hann í þær og smellir á „sýna á korti“. Þar fær hann að sjá þær niðurstöður sem hann valdi á korti. Hver punktur á kortinu er ein prófun og þegar ýtt er á ákveðna prófun birtist gluggi sem sýnir mæligildin á þeim punkti. Auk þess sér hann línu í símasendinn sem síminn var að tala við þegar mælingin fór fram.

Skipulag

Aðferðafræði

Verkefnið var unnið eftir Scrum aðferðafræðinni og vefsíðan Trello notuð til að halda utan um verkefnalista. Trello var einnig notað fyrir Scrum-dashboardið, þar sem sögurnar voru færðar á milli „To Do“, „Doing“ og „Done“. Trello reyndist okkur mjög vel til hliðar við Scrum og ekkert vandamál kom upp tengt því. Við völdum að nota Trello í staðinn fyrir hefðbundnu post-it miðana vegna þess að þá gátum við skoðað sögurnar og verkefnalistann hvenær sem er og hvar sem er en ekki aðeins í vinnuáðstöðunni.

Í upphafi skiptum við tímanum sem við höfðum til að vinna verkefnið niður í spretti, þessir sprettir voru 7 talsins. Sprettur 0 var aðeins ætlaður í greiningu á verkefninu, engin forritun átti að byrja fyrr en í spretti 1. Sprettur 1-5 eru forritunarsprettir fyrir verkefnið. Verkefnið var því brotið niður í 5 „útgáfur“ sem við unnum eftir. Útgáfurnar voru tölusettar „Útgáfa 0.1“ að „Útgáfu 0.5“, sem er lokaafurð verkefnisins. Eftir hvern sprett, átti viðkomandi útgáfa að vera keyrandi og tilbúin til notkunar. Svona settum við upp útgáfurnar:

- Útgáfa 0.1:
 - o Appið getur hringt í símanúmer á fyrirfram ákveðnum tíma og haldið símtalinu gangandi í ákveðinn tíma áður en skellt er á.
- Útgáfa 0.2:
 - o Appið getur skráð niður öll mælanleg gildi sem síminn býður upp á meðan á símtali stendur og vistað í símann.
- Útgáfa 0.3:
 - o Appið getur haft samband við vefþjónustu og sent niðurstöður mælinga yfir 3G.

- Útgáfa 0.4

- o Prófarar geta búið til forskrift með vefviðmóti, sem geymt er í gagnagrunni og verður sent til símans í gegnum vefþjónustu.

- Útgáfa 0.5

- o Appið getur beðið vefþjónustuna um að fá senda forskriftu og fær þá senda rétta forskriftu eftir símanúmeri.

Hverri útgáfu skiptum við niður í „tasks“ og taldist útgáfan kláruð ef öll tösk undir útgáfunni hafði verið lokið. „Task“ taldist vera lokið þegar báðir aðilar í hópnum voru sammála um það. Seinasti spretturinn, sprettur 6, er aðeins ætlaður til lagfæringar og skýrsluvinnslu. Öll forritun á þá að vera búin og aðeins litlar lagfæringar leyfðar.

Við ákváðum í byrjun að hafa sprett 0 og 1 viku lengur en hina sprettina. Við reiknuðum með að það tæki okkur tíma að kynna okkur Android umhverfið, sem var svo raunin. Það var því ákveðið að hafa þá spretti 3 vikur en aðra spretti 2 vikur. Hópurinn hittist alla fimmtudaga og sunnudaga frá 8-17. Við reyndum einnig að hittast sem flest kvöld yfir vikuna þegar tækifæri gafst. Við ákváðum að vinna allt í sameiningu þar sem við erum aðeins tveir í verkefninu. Við vildum ekki vera á sitthvorum staðnum og/eða sitthvorum tímanum.

Til að halda utan um tímaskráningar notuðum við Excel. Allar áætlanir stóðust að mestu leyti.

Hlutverkaskipting

Hlutverkaskiptingin var einföld þar sem um tveggja manna hóp er að ræða.

Scrum master: Guðmundur Sveinsson

Project Owner: Gunnar A. Ólafsson

Forritarar: Guðmundur Sveinsson & Jens Gísli Heiðarsson

Prófarar: Guðmundur Sveinsson, Jens Gísli Heiðarsson & Benedikt Ragnarsson, yfirmaður fjarskiptasviðs Nova.

Greining

Við bjuggum til spretta plan með afmörkuðum sögum sem voru svo brotnar niður í „tösk.“ Markmiðið var að vera með keyrandi útgáfu af kerfinu að hverjum spretti loknum. Talsverður tími fór í að greina hvaða upplýsingar var hægt að fá úr testsímanum.

Tól

Tól	Í hvað voru þau notuð
Eclipse	Java þróunarumhverfi
Visual Studio 2012	Þróunarumhverfi fyrir vefþjónustur og vefviðmót
Azure	Hýsingarþjónusta frá Microsoft
Tortoise Svn	Subversion client (útgáfustýringarbiðlari)
Trello	Verkefnastjórnun
Google Drive	Til að vinna saman í skjölum á rauntíma. Geymir skjöl
Dropbox	Geyma gögn
Google Calander	Fundir og aðrir viðburðir
MS SQL Server Management Studio	Gagnagrunnsvinnsla
Advanced Rest Client	Prófanir á vefþjónustum
JSONLint	Sannprófa JSON strengi

Framework	Notkun
ASP.NET MVC 4	Vefforritun
Linq to sql	Tengingar vefviðmóts og vefþjónustu við gagnagrunn.
jQuery	Vefforritun
Javascript	Vefforritun
Twitter bootstrap	CSS
Google Maps API	Google kort í vefviðmóti
ASP.NET MVC 4 WebAPI	Vefþjónusta
Android SDK	Android kóðasafn, sýndarvélar og önnur þróunartól
ITelePhony	Yfirskrifar TelePhone manager í Android umhverfinu
Razor	Vefforritun

Forritun

Android App

Forritun á Android appinu byrjaði í sprett 1 og stóð yfir alveg til enda verkefnisins. Forritunin byrjaði hægt þar sem uppsetning á Android forritum er öðruvísi en við vorum vanir, en við vorum fljótir að átta okkur og gátum þá byrjað að forrita. Við byrjuðum á að nota IntelliJ sem þróunarumhverfi en skiptum fljótt yfir í Eclipse. Eclipse er aðal þróunarumhverfið fyrir Android og reyndist töluvert þægilegra í notkun.

Við notum Alarm Manager sem er innbyggður eiginleiki í Android SDK-inu til að stilla hvenær síminn ætti að hringja, skella á og taka test. Þá stilltum við „alarm“ sem fer í gang á ákveðnum tíma og framkvæmir ákveðna skipun. Þetta fannst okkur besta lausnin því ef síminn er ekki búinn að vera í notkun lengi og er sofandi vekur alarmið símann á meðan aðrar aðferðir hefðu ekki virkað þá.

Notast er við Location Manager úr Android SDK til að fá staðsetningar á símunum. Við notum GPS staðsetninguna ef það er möguleiki, ef ekki reynum við að fá sem nákvæmstu staðsetninguna eftir internet tengingu. PhoneStateListener úr Android SDK fylgist með stöðu símans í hvert sinn. Hann segir okkur hvort síminn sé í símtali eða ekki og gerir okkur kleift að fylgjast með því hvort símtal slitni eða ekki. Einnig mælir hann GSM sambandsstyrk og hvaða símsendi síminn talar við hverju sinni.

Location Manager og PhoneStateListener taka mikla orku og rafhlaða símans vildi gjarnan tæmast fljótt. Það vandamál var leyst með því að slökkva á þessum aðgerðum milli símtala og kveikja aftur á þeim rétt áður en síminn hringir.

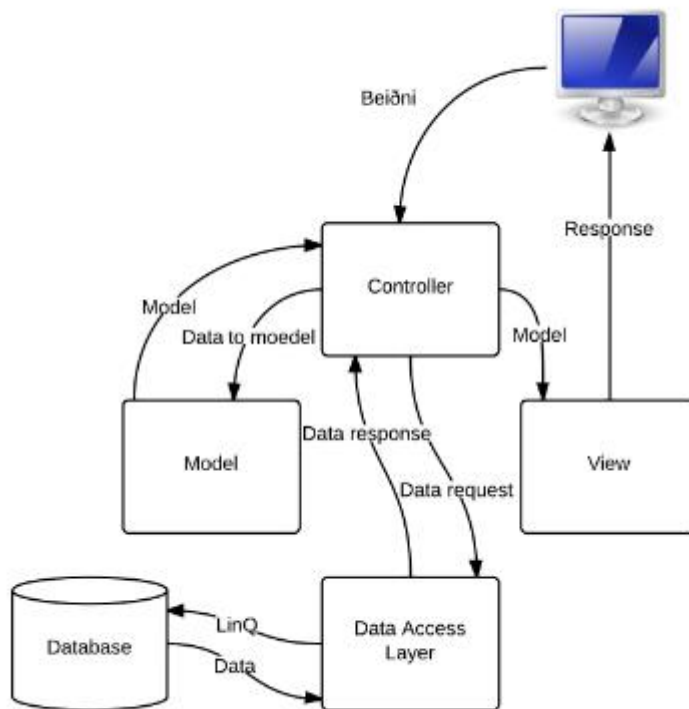
Vefþjónusta

Vefþjónustan hefur tvö hlutverk, annars vegar að skila síma forskriftu eftir því símanúmeri sem sent er og hins vegar að taka við niðurstöðum prófana og skrifa í gagnagrunn. Í fyrstu skrifuðum við WCF þjónustu en þrátt fyrir ítrekaðar tilraunir náðum við ekki sambandi við hana. Eftir að miklum tíma hafði verið varið í að fikta í web.config, endpointum og bindings ákváðum við að byrja uppá nýtt á vefþjónustunni og skrifa hana í ASP.NET MVC 4 WebAPI. Þar sem við vorum búnir að skrifa hana áður í WCF þjónustunni vorum við þokkalega snöggir að skrifa hana og hún virkaði mun betur en fyrri tilraunin.

Vefviðmót

Forritun á vefviðmótinu fór fyrst og fremst fram í sprettum 4, 5 og 6. Kröfurnar til vefviðmótsins í upphafi voru þær að prófari gæti skráð forskriftu og vistað hana í gagnagrunn. Þegar því var lokið var talsvert eftir af sprettinum svo við ákváðum að taka inn B og C kröfur sem fólust í því að gera prófara kleift að greina niðurstöður prófana í vefviðmótinu og tókst það vel til. Eins og áður kom fram er vefviðmótið skrifað í ASP.NET MVC 4 og við notuðum Razor fyrir HTML síður. Allar grafískar niðurstöður prófana eru teiknaðar niður á kort frá Google. Google býður upp á opinn API sem við notuðumst við og skrifuðum á mótí í javascript og jQuery.

Uppbygging kerfisins:



Kerfið er byggt upp eins og myndin sýnir. Beiðni um gögn koma frá client, controllerinn kallar í fall í Data Access Layernum og kallar það áfram niður í gagnagrunn með LINQ to SQL. Gögnunum er skilað aftur upp í controllerinn þar sem þeim er breytt í viðeigandi model og skilað inn í view-ið sem skilar response til client.

Viðmótið er sett upp sem ein síða og öll köll í controller frá client eru gerð með ajax köllum sem gerir okkur kleift að þurfa aldrei að endurhlaða síðuna. Flæðið í síðunni er skrifað í javascript og jQuery.

Prófanir

Kerfisprófanir

Prófanir fóru fram jafnt og þétt yfir allt verkefnið. Alltaf þegar við vorum komnir með keyrandi útgáfu gerðum við á því bæði handvirkar og sjálfvirkar prófanir. Þar sem kerfið mælir gæði símkerfis þurftum við að fara í nokkra bíltúra til að safna niðurstöðum. Við náðum í 2 - 3 þúsund niðurstöður í hverjum rúnti sem stóð í um klukkstund. Þær fóru þannig fram að við vorum með 1 - 2 síma, bjuggum til forskriftur fyrir þá báða og létum þá hringja í númer sem svaraði sjálfvirk, á meðan við keyrðum og borðuðum ís.



Kerfisprófanir

Þessar prófanir skiluðu okkur mikilvægum upplýsingum sem nýttust okkur til að fínþússa appið, til dæmis að slökkva á GPS á milli símtala til að spara rafhlöðu, geyma niðurstöður

prófana ef ske kynni að ekki næðist samband við vefþjónustu og senda þær síðar svo að niðurstöður tapist ekki.

Nova er með nákvæmar mælingar á því hvenær símtöl slitna og geta þeir flett upp í gögnum hjá sér á hvaða tíma hvaða símanúmer slítur sambandi. Við bárum saman tímunn á þeim punktum sem slit urðu í okkar kerfi við slitin í þeirra kerfi og stefndi það upp á sekúndu. Þessi samanburður staðfestir gæði Road Runner og geta þeir því treyst því að ef slit koma fram í kerfinu okkar þá hafa í raun og veru orðið slit. Og í fyrsta skipti geta þeir nú séð hvar slitin eiga sér stað en ekki bara hvenær.

Notendaprófanir

Við fengum dreifikerfis sérfræðinga Nova til að prófa kerfið og þeim fannst það frábært. Þeir töluðu mikið um einfalda og þægilega framsetning gagna í appinu og hversu auðvelt væri að setja í gang prófanir þar sem að þeir þurftu bara að fara inn í vefviðmótið og búa til forskriftu fyrir síma sem gæti verið á ferðinni hvar sem er. Þegar því var lokið gátu þeir annaðhvort fylgst með kortinu uppfærast eftir hvert símtal eða skoðað allar niðurstöðurnar í lok dags.

Framvinda

Eins og áður hefur komið fram voru 7 sprettir. Allir sprettirnir gengu vel, þótt hafi komið upp nokkrar hraðahindranir á leiðinni. Eini spretturinn sem fór ekki eftir áætlun var sprettur 2, við lengdum hann um eina viku því ekki náðist að klára allar kröfur í sprettinum á tilsettum tíma. Hér að neðan munum við taka það helsta úr hverjum spretti og burndown rit þess.

Sprettur 0

Lengd:

3 vikur

Sprettur 0 fór í að koma okkur fyrir í Nova fá leyfi á allan hugbúnað og þess háttar. Verkefninu var skipt niður eins og talað er um hér að ofan og sett niður skipulag fyrir önnina og hvernig við ætluðum okkur að klára verkefnið. Hvorugur okkar hafði skrifað kóða áður fyrir Android svo það fór talsverður tími í að kynna sér það umhverfi.

Sprettur 1

Lengd:

3 vikur

Kröfur:

Sem sími get ég hringt og skellt á eftir forskrift.

Verkefni:

Hanna uppsetningu forskriftar

Parsa forskriftu

Hlaða forskrift inn í appið

Láta símann hringja eftir forskriftu

Láta síma skella á eftir forskriftu

Spretturinn fór frekar rólega af stað þar sem við vorum ennþá að læra inn á Android SDK-inn. Eftir að forritun fór af stað, gekk allt mjög vel þar til við lentum í vandræðum með að ljúka símtölum þar sem Google hafði lokað á þann sjálfvirka möguleika eftir Android útgáfu 3.2. Við vorum ekki tilbúnir að fara niður í svo gamla útgáfu svo það fór smá tími í að finna lausn á því vandamáli. Ein hugmyndin var að ljúka símtalinu á hinum endanum þar sem móttakandinn verður „virtual“ númer sem hýst er hjá Nova og væri hægt að stilla þannig að það skellist á eftir fyrirfram ákveðinn tíma. Okkur líkaði ekki sú lausn þar sem það gengur þvert á kröfur sem gerðar voru til kerfisins í upphafi. Við eyddum talsverðum tíma í að finna lausn á þessu vandamáli og fundum að lokum library sem yfirskrifar Google SDK-inn og með því er bæði hægt að hringja og skella á. Að öðru leyti gekk spretturinn eins og í sögu og allar kröfur kláraðar að fullu.

Eftir sprettinn gat appið hringt í ákveðið símanúmer á ákveðnum tíma og haldið símtalinu gangandi í einhvern tíma áður en skellt var á. Alveg eins og markmið sprettsins var.

Sprettur 2

Lengd:

2 vikur (lengdur í 3 vikur)

Kröfur:

Sem sími get ég skráð helstu mæligildi stýrikerfis á x fresti í símtali til að safna upplýsingum um símkerfið

Verkefni:

Kanna hvaða mæligildi við fáum úr símanum
Setja alarms á x sekúndna fresti í símtölum
Öll testin úr hverju símtali skulu vera vistuð í lista
Setja upp staðsetningarkerfi
Setja upp PhoneStateListener

Spretturinn fór hægt af stað á meðan við vorum að kynna okkur hvaða mæligildi síminn gæti gefið okkur. Mikill tími fór í að finna út hvernig við gátum fengið sem mestar upplýsingar úr Android kerfinu. Við komumst að því að ekki er boðið uppá öll sömu gildin fyrir öll drefikerfi (3g/2g). Bestu niðurstöðurnar fengust úr 2g kerfinu, en þar sem NOVA notar aðeins 3g notuðum við að sjálfsgöðu aðeins þær niðurstöður.

Eftir að við vorum búnir að setja upp alla þessa test virkni, fannst okkur appið sjálft of hægt og þungt í keyrslu. Ákváðum við því að endurskipuleggja uppsetninguna á appinu. Eftir það vorum við lengi að fá nýju útgáfuna til þess að keyra eðlilega og þurfum við því að lengja sprettinn um 1 viku til að geta klárað það. Við tókum út fullt af klösum og „activities” og náðum að minnka appið um meira en helming sem varð til þess að það var léttara í keyrslu og stöðugra.

Eftir sprettin getur appið hringt, skellt á og tekið test eftir forskriftu, alveg eins og markmiðið var.

Sprettur 3

Lengd:

2 vikur

Kröfur:

Eftir hvert símtal skal síminn senda niðurstöður prófana í vefþjónustu og skrifa í grunn.

Verkefni:

Skrifa vefþjónustu sem tekur á móti niðurstöðum í JSON streng.

Parsa JSON strenginn frá appinu í klasa og skrifa í gagnagrunn.

Hanna, skrifa og setja upp gagnagrunn fyrir niðurstöður.

Parsa niðurstöður í appinu í JSON streng áður en þær eru sendar.

Hafa samband við vefþjónustu og senda niðurstöðurnar úr appinu.

Spretturinn gekk mjög vel, við skrifuðum gagnagrunninn í MS-SQL og vefþjónustuna í ASP.NET MVC 4 API í Visual Studio 2012. Illa gekk reyndar að ná sambandi við vefþjónustuna og erfitt var að prófa virknina til að byrja með. Það endaði með að við keyptum okkur

hýsingarþjónustu hjá Microsoft Azure. Þar hýstum við gagnagrunninn og vefþjónustuna. Seinna útvegaði Nova okkur server og færðum við þjónusturnar yfir í enda sprettsins.

Við bættum við debug mode í appið svo hægt væri að búa til forskriftu í appinu sjálfu. Það sparaði okkur mikinn tíma við að prófa virknina í appinu því það var mjög tímafrekt að vera í sífellu að búa til nýja forskriftu og hlaða í símann, aðeins til að prófa einn lítinn hlut.

Eftir sprettin getur appið hringt, skellt á og tekið test í símtalinu eftir forskriftu. Eftir að skellt var á símtalið var niðurstöðulistinn sendur í vefþjónustu og skrifað í grunn. Alveg eins og markmið sprettsins var.

Sprettur 4

Lengd:

2 vikur

Kröfur:

Prófarar geta búið til forskrift með vefviðmóti, sem geymist í gagnagrunni og verður sent til símans

Verkefni:

Skrifa vefviðmót fyrir prófara til að búa til forskrift.

JSON Parsa formið í klasa og vista í grunn.

Skrifa bakenda sem hefur samband við gagnagrunn.

Við byrjuðum á að skrifa viðmótið, það gekk vel og við vorum komnir með viðmót sem skrifði forskriftu í grunn í JSON streng eftir fyrstu vikuna. Ekkert var farið í útlitið á viðmótinu þar sem það var áætlað í sprett 5 og 6.

Eftir sprettin getur appið hringt, skellt á og tekið test í símtalinu eftir forskriftu. Eftir að skellt var á var niðurstöðulistinn sendur í vefþjónustu og skrifaður í grunn. Einnig gat prófari búið til forskriftu í vefviðmóti eins og markmið sprettsins var.

Sprettur 5

Lengd:

2 vikur

Kröfur:

Appið getur beðið vefþjónustu um að fá senda forskriftu og vefþjónustan sendir rétta forskriftu eftir símanúmeri símans.

Verkefni:

Sími biður um forskrift eftir símanúmeri.

Skrifa vefþjónustu sem skilar JSON streng úr gagnagrunni út frá símanúmeri.

Koma sambandi milli vefþjónustu og apps.

Láta appið vinna eftir forskriftu frá vefþjónustu, líkt og það hefur gert með harðkóðaðar forskriftur.

Í sprett 5 lentum við í smá vandræðum með að fá appið og vefþjónustuna til að vinna saman, við vorum lengi að fá JSON strenginn sem sendur er til appsins til að virka eins og hann átti að virka, það hafðist þó á endanum og eftir það gekk spretturinn mjög vel. Við komumst að því að ekki var hægt í öllum sínum að fá upp símanúmerið sem er í símanum, svo við þurftum að finna leið framhjá því. Við komumst að því að alltaf er hægt að fá „sim serial number“ eða ICCID sem er auðkenni á símakortum. Við tókum þetta upp á fundi með tengiliðnum okkar í Nova og komumst við að þeirri niðurstöðu að best væri að fá appið til að senda ICCID númerið á vefþjónustuna þegar það væri að biðja um forskriftu. Við gátum ekki búist við því að prófaranir kynnu ICCID númerið á öllum símanúmerunum sínum, því það númer eru 19 tölustafir, og einnig er krafa að kerfið vinni eftir símanúmerum. Þá bættum við nýrri töflu í grunninn okkar, töflu sem heldur utan um alla prófunarsíma í kerfinu. Prófarar þurfa þá að skrá símanúmer símana sem þeir ætla að nota og sim serial númerið á þeim. Þegar síminn biður svo um forskriftuna sendir hann sim serial númerið og vefþjónustann leitar að símanúmeri eftir serial númerinu og skilar forskriftu eftir því.

Útlit á vefviðmóttinu var búið til í sprettinum og að mestu leyti klárað.

Eftir sprettinn gat prófari skráð forskriftu í vefviðmóti sem svo skráðist í grunn. Sími gat náð í forskriftuna úr grunninum í gegnum vefþjónustu og hringt, gert prófanir og skellt á eftir henni. Þegar símtali var lokið sendi appið niðurstöður til vefþjónustu sem skráði þær í grunn.

Þegar allar kröfur í sprettinum voru búnar, áttum við smá tíma eftir svo við ákváðum að taka inn B kröfuna Google Maps. Það var búið að vera markmið hjá okkur allan veturinn að ná inn þessari kröfu þar sem hún myndi gera vinnslu niðurstaða mikið þægilegri fyrir alla og gera okkur kleift að sýna niðurstöðurnar miklu betur heldur en í töflum, eins og áætlað var í byrjun verkefnisins.

Tveir seinustu dagarnir í sprettinum tileinkuðum við Google Maps og vorum komnir með góða virkni á þeim tíma.

Sprettur 6

Lengd:

2 vikur

Kröfur:

Fínstilla alla hluta kerfisins og gera viðmót flott. Vinna úr niðurstöðum.

Lokaskýrsla.

Verkefni:

Lokaskýrslan og önnur pappírsvinna.

Fínstilla vefviðmót og alla virkni þar.

Fínstilla App viðmótið og alla virkni þar.

Google maps niðurstöðuvinnsla.

Þar sem við vorum byrjaðir á Google Maps hlutanum í sprett 5 héldum við áfram í því í byrjun sprett 6. Fljótlega fórum við í að yfirfara allan kóða og klára útlit á viðmótum. Í lok fyrri vikunnar í spretti 6 vorum við búnir að klára alla forritun og vorum komnir í „codefreeze”. Seinasta vikan frá mánudegi til skila á föstudeginum fór í skýrslugerð og yfirllestur.

Samantekt

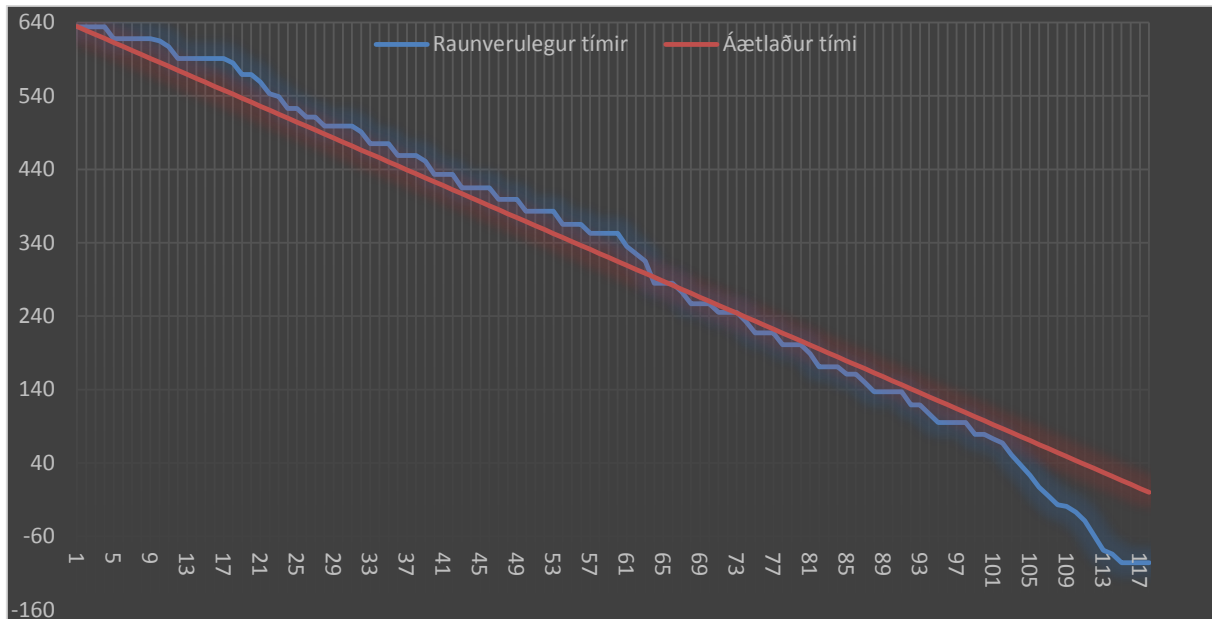
Verkefnið gekk að flestu leyti vel og stöndum við uppi með kerfi sem gerir ótrúlega hluti miðað við þann stutta tíma sem tók að skrifa það. Þetta verkefni hefur fengið okkur til að kynna okkur nýja hluti sem við hefðum eflaust ekki kynnt okkur fyrr en við stæðum frammi fyrir því að þurfa að nota í framtíðinni. Áður en verkefnið hófst höfðum við enga reynslu í að skrifa öpp fyrir android en höfum þrátt fyrir það skrifað app sem er mjög sérstakt að því leyti að það hefur engan notanda. Þetta er app sem á að liggja í bíl og á bara að virka. Það var mikil áskorun að forrita kerfi sem ekki fær neitt inntak frá notandanum á meðan það er í keyrslu. Við kynntum okkur vefþjónustur, bæði Soap og Rest, við skrifuðum Rest WCF þjónustu sem svo virkaði ekki. Við nýttum þá reynslu til að byrja upp á nýtt og finna aðrar leiðir til að leysa þetta vefþjónustu mál og enduðum við eins og áður kom fram á því að nota WebAPI til að skrifa þjónustuna.

Google maps er með góðan API sem við kunnum ekkert á og ákváðum við því þegar við áttum lausan tíma í lokin að lesa okkur til um API-inn og læra á hann. Við sjáum ekki eftir því þar sem möguleikarnir eru endalausir.

Heilt á litið erum við að öllu leyti betri forritarar, betri í að skipuleggja tíma okkar og skipuleggja verkefni. Ekki kemur að sök að vera orðnir samræðuhæfir við sérfræðinga á fjarskiptasviði Nova um hin ýmsu gildi í símkerfinu og hvernig sendar í símkerfum geta bæði unnið saman og á móti hver öðrum.

Tímar

Áætlaður tími verkefnisins voru 640 tímar. Við settum upp tíma- og verk áætlun út frá því. Við höldum okkur við þessa áætlun allan tímann, en undir lokin bættum við á okkur vinnu og unnum stundum frammá nótt. Ástæða þess að við bættum á okkur þessari vinnu var að við vildum koma inn Google Maps í vefviðmótið til að auðvelda greiningu niðurstaðna. Við sjáum ekki eftir þessari auka vinnu við að koma Google Maps inni kerfið þar sem okkur finnst það hafa lyft verkefninu uppá annan stall.



Umsögn tengiliðs

Þetta er í fyrsta skipti sem Nova ákveður að bjóða uppá lokaverkefni á tæknisviði fyrir útskriftarnema á háskólastigi. Fyrir valinu varð verkefni sem fjallar um: "Að gera sjálfvirkar gæðaprófanir á fjarskiptakerfi Nova við raunverulegar aðstæður".

Lagt var af stað með ágæta áætlun sem lögð var niður í nokkra spretti og gerðu nemendur verkáætlun sem vel hefur gengið eftir. Mjög skemmtilegt hefur verið að fylgjast með því hvernig frjóar hugmyndir nemenda hafa fengið líf. Notkun á kortagrunni Google kemur mjög vel út í lausninni en sú hugmynd er sprottinn frá hópnum sjálfum. Þar er með mjög góðum hætti settar fram lykilupplýsingar um "drive test" sem strax gefa góðar vísbendingar um hvar vinna þurfi í gæðamálum fjarskiptakerfis. Einnig var það fljótt ljóst að lausnina má auðveldlega nýta við prófanir vegna reikningagerðar þ.e. til að framkvæma endurtekin símtöl sem oft þarf að gera þegar þróaðar eru nýjar þjónustuleiðir. Nemendur voru að öllu leiti sjálfbjarga með uppsetningu á öllu þróunar- og keyrsluumhverfi sem var til fyrirmyndar. Starfsmenn Nova komu með ábendingar um hvaða mæligildi í sínum skipta mestu máli. Nemendur lögðu samt sem áður á sig mikla rannsóknarvinnu til að skilja þau gildi enn frekar og komu einnig með ábendingar um hvað mætti nýta að auki við það sem tæknimenn Nova höfðu lagt til. Samvinnan var því mjög góð fyrir báða aðila.

Gaman verður að sjá hvernig notkun kerfisins mun ganga. Ef vel gengur gæti verið grundvöllur til að taka málið enn frekar áfram út í hinn stóra heim. Kerfi líkt og þetta sem hér er komið vel á veg eru seld fyrir mjög háar upphæðir á alþjóða markaði. Með þessu verkefni getur því vel verið að lagður hafi verið grunnur af sprotafyrirtæki sem gæti átt framtíðina fyrir sér. Við þökkum nemendum sem tóku þátt í verkefnun kærlega fyrir sína vinnu og höfum í hyggju að ráða þá báða til frekari starfa hjá Nova.

Gunnar A. Ólafsson. Yfirmaður tæknisviðs Nova.