

Valoa pilvestä –sensoriverkkoprojekti

Testaussuunnitelma

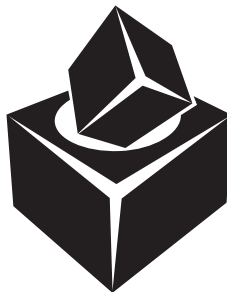
Mika Ahonen

Heljä Ekman

Jukka Friman

Merika Peltola

Janne Uusitupa



Versio: 2020–5–11

Julkinen

11. toukokuuta 2020

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2020		
Tilaaaja	__.__.2020		
Ohjaaja	__.__.2020		

Tietoa dokumentista

Tekijät:

- | | |
|-----------------------|------------------------------|
| • Mika Ahonen (MA) | mika.k.ahonen@student.jyu.fi |
| • Heljä Ekman (HE) | hememavu@student.jyu.fi |
| • Jukka Friman (JF) | jukkaf@gmail.com |
| • Merika Peltola (MP) | merika.m.peltola@jyu.fi |
| • Janne Uusitupa (JU) | janne.t.uusitupa@jyu.fi |

Dokumentin nimi: Valoa pilvestä -projekti, Testaussuunnitelma

Sivumäärä: 11

Tiedosto: testaussuunnitelma.tex

Tiivistelmä: Dokumentti on osa Kokkolan yliopistokeskuksessa toteutettavaa sensoriverkkoprojektia. Dokumentissa kuvataan testauksen kohteet ja testaustavat mahdollisimman yksityiskohtaisesti.

Avainsanat: Esineiden internet, IoT, LoRa, sensoriverkkoprojekti, NB-IoT, projekti-työ, projektihallinta, testaussuunnitelma.

Versiohistoria

Versio	Muutokset	Tekijät
2020–1–25	Dokumentin luonti, runko valmis.	HE
2020–5–7	Dokumentin muokkaus, tekstien tuonti	MA
2020–5–8	Ulkoasun hienosäätöä	HE
2020–5–10	Testitapauksien lisäys	MA
2020–5–11	Dokumentin viimeistely	HE

Tietoa projektista

Projektin tarkoitus on kehittää Laukaan jääkiekkokaukalolle automatisoitu valojen kytkentä. Projektissa hyödynnetään Microsoftin Azure-pilvipalvelua sekä LoRa- ja NB-IoT-teknologioita kytkeäkseen jääkiekkokaukaloon asennettavat anturit valoihin niin, että järjestelmä havaitsee kävijän ja kytkee valot käytön mukaisesti päälle ja pois.

Tekijät:

- | | |
|-----------------------|---|
| • Mika Ahonen (MA) | <code>mika.k.ahonen@student.jyu.fi</code> |
| • Heljä Ekman (HE) | <code>hememavu@student.jyu.fi</code> |
| • Jukka Friman (JF) | <code>jukkaf@gmail.com</code> |
| • Merika Peltola (MP) | <code>merika.m.peltola@jyu.fi</code> |
| • Janne Uusitupa (JU) | <code>janne.t.uusitupa@jyu.fi</code> |

Tilaaaja:

- | | |
|----------------|------------------------------------|
| • Riku Immonen | <code>riku.j.immonen@jyu.fi</code> |
|----------------|------------------------------------|

Ohjaajat:

- | | |
|-------------------------|---|
| • Tuomo Härmänmaa | <code>tuomo.harmanmaa@chydenius.fi</code> |
| • Veli-Matti Tornikoski | <code>veli.tornikoski@chydenius.fi</code> |

Sisältö

1	Johdanto	1
2	Termit	2
3	Testauksen kohteet ja tavoitteet	3
4	Testausympäristö ja testauksen apuvälineet	4
5	Toteutuksen organisointi ja raportointi	5
6	Testausstrategia	6
7	Testitapaukset	8

1 Johdanto

Tässä dokumentissa on esitelty testaussuunnitelma Kokkolan yliopistokeskus Chydeniuksen sensoriverkkoprojektikurssin Valoa pilvestä -projektille.

Projektin idea on lähtöisin Jyväskylän yliopistolla meneillään olevasta JKL-Open-projektista. Projektin tilaajana on Laukaan kaupunki, joka haluaa lisätä jääkiekkokaukaloihin käyttömukavuutta ja turvallisuutta asentamalla niihin automaattisesti syttyvät valot. Projektissa hyödynnetään langattomia radioteknologioita ja projektin resurssit tulevat JKL-Open-projektilta. Projektissa käytetään sensoriteknologioita sekä pilvipalveluita. Projektiryhmä koostuu viidestä tietotekniikan maisterivaiheen opiskelijasta.

Tässä testaussuunnitelmassa kuvataan Valoa pilvestä-sensoriverkkoprojektissa syntyvän järjestelmän testaus ja sen organisointi. Dokumentissa kuvataan testausympäristö ja miten toteutettavaa kokonaisuutta ja sen osia tullaan testaamaan. Testauksesta kirjoitetaan testausraportti.

2 Termit

Azure	Microsoftin pilvipalvelualusta. Sisältää kaikki alustassa käytettävät tuotteet.
IoT	<i>Internet of Things</i> , esineiden internet. Konsepti, jossa erilaiset esineet ja koneet vuorovaikuttavat keskenään.
LoRa	<i>Long Range</i> . Pitkän kantaman, matalan virrankulutuksen radioteknologia, kuten NB-IoT.
MQTT	<i>Message Queue Telemetry Transport</i> , julkaise-tilaa-malliin perustuva, kevyt kommunikaatioprotokolla, kehitetty laitteiden väliseen kommunikaatioon.
NB-IoT	<i>Narrowband IoT</i> . Kapeakaistainen, matalan virrankulutuksen radioteknologia, joka rakentuu LTE-tekniologian päälle.

3 Testauksen kohteet ja tavoitteet

Testauksen kohteena ei ole yksittäinen sovellus vaan kahdesta erilaisesta laitteesta ja pilvipalvelusta koostuva kokonaisuus. Testattava toteutus on tarkoitettu käytettäväksi jääkiekkokaukalossa, joka sijaitsee Laukaalla. Kehitystyön lopputuloksen on tarkoitus havaita liikettä kaukalosta ja sen perusteella sytyttää kaukalon yhteydessä olevat ulkovalot.

Testauksella tarkoitetaan prosessia, jonka avulla varmistetaan, että toteutettava tuote on vaatimusten mukainen ja että kaikki sen ominaisuudet toimivat halutulla tavalla. Esineiden internetin ja teollisen internetin testaukseen liittyy useita haasteita. Näitä ovat erityyppisten resurssien, kuten sensoreiden, aktuaattoreiden ja viestiyhteyksien määrä, sekä toisiinsa kytketyt eri järjestelmät. Pelkän lähdekoodin testaaminen ei riitä, vaan erinäisten laitteiden, verkkojen ja pilvipalveluiden yhteentoimivuuden varmistaminen ovat osa testauskokonaisuutta.

Näitä haasteita voi ratkaista suunnitellulla ja hierarkkisella testauksella. Kunkin laitteen kohdalla on ensin testattava radioyhteyden kuuluvuutta, ennen kuin voi testata viestin lähetystä. Lisäksi kutakin laitetta on syytä testata ensin erikseen ja sitten vasta yhdessä. Testauksen vianetsinnässä auttaa se, että kunkin laitteen voi kytkeä tietokoneeseen, ja lukea sen kirjoittamia viestejä. Lisäksi verkkoyhteyksien viestiliikenne on luettavissa käytettävien palveluiden lokeista.

Testauksen tavoitteena on tässä dokumentissa määriteltyjen testitapausten avulla löytää mahdollisimman suuri osa järjestelmän virheistä ja korjata nämä. Tavoitteena on toimiva järjestelmä. Testauksella voidaan havaita mahdolliset regressiot, joita saattaa tulla esimerkiksi kehitysprojektin päättymisen jälkeen, kun sensorit ovat kaukalossa kesän ajan. Näin ylläpitäjä voi varmistua siitä, että kukin osa toimii erikseen ja yhdessä.

4 Testausympäristö ja testauksen apuvälineet

Testausympäristö koostuu projektissa käytetyistä sensori- ja mikropiiriteknologioista (Dragino LoRa mini, ADXL362 kiihtyvyysanturi ja XBee), tarvittavista viestiyhteyksistä (Digitan LoRaWan ja Elisan NB-IoT-verkko) sekä Microsoft Azuren pilvipalvelun ohjelmointi- ja tiedontallennusympäristöstä.

Testauksen kohteina ovat laitteisto, tietokanta ja viestintä. Laitteisto kattaa sekä havainnoinnista vastaavan sensorin että valoja kontrolloivan aktuaattorin. Tietokanta koostuu Azuressa olevasta SQL-tietokannasta, johon tallennetaan sensorin lähettämä data. Viestintä kattaa sekä LoRa-viestinnän sensorilta Azureen asti että MQTT-viestinnän Azuresta XBeele.

Fyysisinä testausympäristönä toimivat projektiryhmän käytössä olevat työskentelytilat eri paikkakunnilla sekä järjestelmän todellinen kohdeympäristö Laukaan Valkealan koulun jalkapallokenttä ja jääkiekkokaukalo.

Testauksen apuvälineitä ovat mikropiirien kehitysalustat, eri ohjelmointiympäristöt, testityngät (eng. *test stub*), apuohjelmat ja tarvittaessa eri mittauslaitteet ja muut työkalut. Tavoitteena on mahdollisimman monipuolinen ympäristö.

5 Toteutuksen organisointi ja raportointi

Valoa pilvestä -projektin testaukseen osallistuu koko projektiryhmä. Testaus tullaan toteuttamaan useassa osassa, koska projektin jäsenet asuvat eri paikkakunnilla ja työ on jaettu eri osa-alueiden välillä ryhmänjäsenten kesken. Kaikilla ei myöskään ole jokaisen osan testaamiseen tarvittavia välineitä. Testaukset jaetaan siten, että parhaiten kykenevä henkilö toteuttaa testauksen.

Joustavana ohjenuorana on että jokainen kehittäjä yksikkötestaa oman koodinsa. Integrointitestaamiseen osallistuvat eri osioiden tekijät yhdessä, ja järjestelmä- ja hyväksymistestauksen tekee koko projektiryhmä. Testauksessa hyödynnetään mahdollisuuksien mukaan paritestausta.

Testauksen tuloksista raportoidaan testausraportissa ja raportin mahdollisissa liitteissä. Raportissa nimetään testaaja ja kuvataan selkeästi testausympäristö, testitapaukset sekä testauksen tulokset. Se sisältää kuvauksen testauksen kohteena olevasta laitteesta tai ohjelmistosta sekä kertoo, mitä ominaisuuksia testattiin. Testitapauksia ja tuloksia voidaan havainnollistaa erilaisten taulukoiden ja kuvaajien avulla. Tulosten pohjalta tehdään tarvittavat muutokset, jonka jälkeen testit suoritetaan uudelleen.

Raportteja kirjataan toteutuksen rinnalla. Yksittäisistä testeistä kerrotaan testaajan nimi, suorituspäivämäärä, testattava kohde, testausympäristö, kuvaus testistä, odotettu tulos ja havaitut virheet. Dokumentoinnin muoto voi vaihdella joustavasti riippuen testitapauksesta.

Testi on läpäisty, kun se on testaussuunnitelman mukaan hyväksytysti suoritettu ja siitä on tehty raportti. Janne Uusitupa valvoo tässä projektissa koodin laatua ja Mika Ahonen testauksen toteutumista.

6 Testausstrategia

Ketterä ohjelmistokehitys ja ketterä testaus perustuu priorisointiin. Etusijalla kehitystyössä ovat järjestelmän tärkeimmät osiot ja osiot testataan kriittisyysjärjestyksestä. Testausta pyritään tekemään vain sen verran, mitä kyseinen järjestelmä vaatii toimiakseen luotettavasti. Ketterissä menetelmissä tiimi on vastuussa testausmenetelmistä, ja menetelmille ei ole määritelty tiukkoja raameja. Testaus on jatkuva prosessi, jota pyritään suorittamaan toteutuksen rinnalla. Samalla kun kirjoitetaan koodia, kirjoitetaan testitapauksia. Testitapauksia muodostetaan käyttötapauksista, konkreettisia testitapauksia johdetaan myös koodaajien ja asiakkaan välisistä keskusteluista.

Ketterissä menetelmissä painotetaan kehittäjän itsensä kirjoittamia yksikkö- ja integraatiotestejä. Järjestelmän oleelliset toiminnallisuudet mm. integraatiot testataan huolella.

Seuraavia testauksen menetelmiä voidaan käyttää projektissa.

Sisäinen testaus (eng. *structural testing, white-box testing*) pyrkii testaamaan ohjelmistokomponenttien virheettömyyden. Sisäisessä testauksessa ohjelmiston toteutus tunnetaan, joten testitapaukset laaditaan toteutuksen perusteella.

Ulkoinen testaus (eng. *functional testing, black-box testing*) puolestaan tarkastelee ohjelmistokomponenttien tuottamien tulosten oikeellisuutta suhteessa syötettyihin arvoihin.

Tutkivassa testauksessa (eng. *explorative testing*) pyritään löytämään ohjelman mahdollisia vikatiloja. Testaaminen suoritetaan ilman tarkkaa testisuunnitelmaa, eikä se perustu etukäteen dokumentoituihin testitapauksiin. Tutkiva testaus ei vaadi suuria ennakkovalmisteluja ja sopii hyvin käytettäväksi ketterissä projekteissa. Tutkivassa testauksessa testaajat hyödyntävät kokemusta ja ymmärrystä testattavasta järjestelmästä. Menetelmä perustuu rinnakkaiseen testien suunnitteluun, suorittamiseen ja oppimiseen. Testejä suunnitellaan testauksen aikana, ja saatua informaatiota käytetään hyödyksi uusien testien luomiseksi. Menetelmä on kustannustehokas, kun tarkka etukäteissuunnittelu jätetään pois. Tuloksena syntyy raportti, joka kertoo mitä on testattu, mistä löytyi virheitä ja mitä tulee testata jatkossa. Projektissa voidaan hyödyntää myös osittain tutkivaa testausta.

Testausprosessi voidaan jakaa seuraaviin testaustasoihin.

Yksikkötestauksen tavoite on testata järjestelmän jokaisen yksittäisen komponentin oikea toiminta. Yksikkötestauksen suorittaa ohjelmoija toteutuksen yhteydessä. Yksikkötestin kirjoittamista voidaan helpottaa rakentamalla testikomponentteja tai testitynkiä. Löydetyt virheet korjataan välittömästi kehittäjän toimesta. Testit uusiin korjausten jälkeen.

Integraatiotestauksen kohteena ovat rajapinnat eli eri osien väliset yhteydet. Testaus tapahtuu selvittämällä moduulien keskinäistä ja niiden välistä toimintaa. Apuna tässä ovat ohjeet, määrittely- ja suunnitteludokumentit. Integrointitestauksessa keskitytään komponenttien välisiin rajapintoihin. Tarvittaessa voidaan kirjoittaa testaamiseen tynkiä.

Järjestelmätestauksessa järjestelmän kaikkia osia testataan yhtenä kokonaisuutena ja varmistetaan, että kokonaisen järjestelmän toiminnot vastaavat vaatimuksia. Tarkoituksena on testata, kuinka hyvin ohjelmisto toimii oikeassa olosuhteissa. Prosessin tai datavirran kulkua testataan sovelluksen alusta sovelluksen loppuun. Samalla huomioidaan liitännät ja rajapinnat muihin järjestelmiin.

Lisäksi voidaan testata suorituskykyä, rasituksen sietoa, turvallisuutta tai yleistä toiminnallisuutta verkossa. Testaus pyritään suorittamaan niin, että projektiryhmän jäsenet testaavat eri alueita kuin ovat itse toteuttaneet.

Hyväksymistestaus on viimeinen testausvaihe ennen järjestelmän käyttöönottoa. Tässä testataan, täyttääkö sovellus asiakkaan asettamat vaatimukset kokonaisuudessaan. Hyväksymistestaus tehdään yhdessä loppukäyttäjän kanssa betatestauksena. Jos järjestelmästä löytyy hyväksymistestauksen aikana vikoja, ne korjataan ja korjaukset todennetaan ennen julkaisua. Kehitystä ja testaamista jatketaan kunnes asiakas antaa hyväksyntänsä.

Integraatio-, järjestelmä- ja hyväksymistestaus tehdään ulkoisena testauksena. Käyttöönottovaiheessa tehdään lopullinen järjestelmä- ja hyväksymistestaus. Vaiheen on tarkoitus olla nopea, niin että siinä vain todetaan, että järjestelmä toimii oikein. Tämä vaatii, että aiemmat testitapaukset on saatu menemään läpi.

7 Testitapaukset

Tässä kuvataan järjestelmälle suoritettavat testitapaukset. Testitapausten sisältö määräytyy järjestelmän vaatimusten ja käyttötapausten mukaan.

Testitapaus kuvaa mitä ja miten testataan. Testitapaukset nimetään suoritettavien toimintojen mukaan. Testien tuloksista raportoidaan tässä dokumentissa kuvatulla tavalla. Testitapauksissa määritellään testien kohteet, suoritusvaiheet, syötteet, tulosteet, vaatimukset ympäristölle, erityiset vaatimukset ja riippuvuus muista testeistä.

Testitapauksen suoritus alkaa käymällä läpi testitapauksen suoritusvaiheet ja ympäristövaatimukset testitapauksen kuvauksesta. Jos suoritusvaiheita ei ole kerrottu, tulee niiden käydä ilmi testitapauksen yleiskuvauksesta.

Eri tyyppiset testit voivat vaatia eri tyyppistä dokumentointia, siksi dokumentoinnin muoto voi vaihdella joustavasti riippuen testitapauksesta.

ID	Testitapaus, kuvaus (alkutila, esiehto, ympäristö, toiminto, syötteet, askeleet, vaatimukset, ohjelmistoversio)	Odotettu tulos, lopputila
S1	Anturilaitteeseen kytketään virta ja se käynnistyy virheettömästi. Alkutila: Laite on pois päältä	Anturi käynnistyy, testaa itsensä ja tulostaa monitoriin määrätyn viestin.
S2	ADXL362-kiihtyvyyssanturin self-test	Laite suorittaa self-testin onnistuneesti. Sensori tulostaa positiivisen testituloksen monitoriin aina käynnistyksen yhteydessä.
S3	Liikesensori havaitsee liikettä. Sensori lukee akseliensa x, y sekä z arvon. Alkutila: S1 ja S2 testit ok.	Arvot tulostuvat monitoriin. Sensori nukahtaa liikkeen puuttuessa asetusten mukaisesti ja herää jälleen, kun sitä liikutetaan.
S4	Liikesensori lähettää viestin Digitan LoRa reitittimille.	Sensorilta saadaan tyhjä tai muu merkityksetön viesti kulkemaan LoRa käyttäen Digitan palvelimelle.
S5	Sensori lähettää viestin, jonka jälkeen se nukkuu määrätyn ajan säästääkseen virtaa.	Havaitaan, että anturikokonaisuuden toiminta loppuu, ja laitteisto nukkuu.

S6	Sensori toimii toivotulla tavalla lähettäen viestin Digitan palvelimelle, nukahtaen sitten määrätyn kes- toiseen uneen ja herää jälleen mittaakseen tärinää.	Sensori lähettää onnistuneesti vies- tin ja vaipuu sitten pakotettuun uneen, jonka aikana se ei reagoi suu- rempaankaan värinään. Sensori he- rää määrätyn ajan jälkeen ja lähettää uuden viestin vain, jos se havaitsee värinää. Jos liikettä ei havaita, vai- puu sensori uudelleen uneen, josta se herää värinää havaittuaan. Senso- ri ei näytä vikatoimintaa edes pitkän päälläoloajan jälkeen.
S7	Sensori toimii toivotulla tavalla oikeissa olosuhteis- sa. Sijaintipaikkana sensorien lopullinen käyttöym- päristö.	Kun sensori on kiinnitetty valittuun paikkaan kohdeympäristössä, lähet- tää se viestin havaittuaan liikettä. Se ei lähetä viestiä tuulen tai muun ul- koisen häiriön aiheuttaman liikkeen vuoksi.
S8	Sensorin toimintavarmuus ja viiveen tarkastelu Anturi lähettää säännöllisin väliajoin viestin riippu- matta havaitusta värinästä.	Laitteen toiminta ei pääty virheti- laan. Laitteen heräämisessä ja lähet- tämässä havaitut viiveet kirjataan.
D1	LoRa-gateway viestin vastaanotto. Vastaanotetaan sensorinoodilta viesti. Alkutila: Sensorinoodi on konfiguroitu päätelait- teeksi Digitan LoRaWan-verkkoon.	Digitan palvelussa nähdään vastaa- notettu viesti.
D2	LoRa-gateway viestin välittäminen. Viesti välite- tään HTTP Post -protokollalla Azureen. Alkutila: Gateway on saanut viestin sensorilta. Sen- sorin viestit on konfiguroitu välitettäväksi Azuren pilvipalveluun.	Viesti otetaan vastaan Azuressa.
A1a	Azure-funktion kutsuminen ja laitteen käyttöoikeu- den tarkastaminen virheellisellä laiteavaimella.	Lokille merkintä 403. Data ei tallen- nu tietokantaan.
A1b	Azure-funktion kutsuminen ja laitteen käyttöoikeu- den tarkastaminen oikealla DevEUI:lla	Lokille merkintä 200. Data kirjautuu oikein tietokantaan.
A2a	Azure funktion kutsuminen kutsulla, jossa JSON ra- kenne on väärä.	Azure funktion suoritus onnistuu, paluukoodi 400. Lokille kirjautuu paluukoodi 400. Kantaan ei tallennu mitään. XBee:lle ei lähde viestiä.
A2b	Azure funktion kutsuminen kutsulla, jonka JSON rakenne on oikea, mutta siitä puuttuu joku tarvit- tavista arvoista.	

A3a	MQTT-lähetys Azuresta XBeele, kun XBee on verkossa ja kuittaa saamansa viestin	Funktion suoritus päättyy odotetusti. Lokille kirjataan XBeen kuittaus. Tietokantaan kirjataan XBeen ID.
A3b	MQTT-lähetys Azuresta XBeele, kun XBee on verkossa, mutta ei lähetä kuittausta.	Funktion suoritus päättyy odotetusti. Lokille kirjataan XBeen kuittauksen puuttuminen. Tietokannan XBee-kenttä jää tyhjäksi.
A3c	MQTT-lähetys Azuresta XBeele, kun XBee ei ole verkossa.	Funktion suoritus päättyy odotetusti. Lokille kirjataan, että XBee ei kuittanut. Tietokannan XBee-kenttä jää tyhjäksi.
A4	Suorituskykytestaus useammalla yhtäaikaaisella lähetyksellä. Lähetetään viestejä sekä samoilla, että eri laitetunnuksilla.	Kaikkien lähetykset otetaan onnistuneesti vastaan, MQTT-viestit lähtee XBeele ja tiedot tallentuvat kantaan.
X1	XBee kytkeytyy NB-IoT-verkkoon. Alkutila: Laitteessa on NB-IoT-SIM-kortti, ja siihen on kytketty tietokone ja XCTU.	Verkkoyhteyden muodostus NB-IoT-verkkoon nähdään XCTU:n monitorissa.
X2	XBee yhdistää Azuren IoT Hubiin. Alkutila: Testi X1 on läpäisty.	Laite yhdistää Azuren IoT Hubiin. Laitteen tila Azure IoT Explorerissa on Connected.
X3	Xbee vastaanottaa Azurelta Direct Method viestin Azure IoT Explorerilta. Alkutila: Testit X1 ja X2 on läpäisty.	Viestin vastaanotto todetaan PyCharmin konsolista.
X4	XBeen kytketään releellä verkkovirtaan.	KytKentä toimii ja valo syttyy.
IJ1	Järjestelmä- ja integraatiotestaus: Laukaan kokonaisuuden testaus. Testaus toteutetaan Laukaan Valkolan koulun kentällä. Testauspaikka on sama kuin varsinainen toimintaympäristö, joten testauksen avulla voidaan selvittää koko järjestelmän toiminta lopullisessa paikassa.	Järjestelmä toimii odotetusti: sensori havaitsee liikettä, lähettää viestin ja aktuaattori vastaanottaa viestin, valot syttyvät.
IJ2	Järjestelmä- ja integraatiotestaus: JSON viestin välittäminen Azuren funktiolla Xbee-laitteelle.	Viesti kulkee perille asti, valo syttyy ja viesti tallennetaan tietokantaan.
DO1	Ohjeistuksen ja dokumentaation testaus.	Laaditut ohjeistukset ja dokumentaatio vastaavat todellisuutta

Ei-toiminnallisten vaatimuksien testaus

Suorituskykytestauksen avulla varmistetaan, että järjestelmä täyttää sille asetetut tehokkuusvaatimukset.

Testaamatta jäävät ominaisuudet ja toiminnallisuus

Esim. skaalautuvuuden vaikutuksia ei tulla erikseen testaamaan. Sovelluksen siirrettävyyttä muihin fyysisiin ympäristöihin ei testata.