

Lappeenrannan teknillinen yliopisto
Teknicaloudellinen tiedekunta
Tietotekniikan osasto

KANDIDAATINTYÖ

Kuormitustestausvälineen valinta

Microsoft Office SharePoint Server 2007 – toteutuksessa

Kandidaatintyön aihe on hyväksytty 1.6.2009

Lupa loppuraportin esittämiseen 24.8.2009

Työn tarkastajana ja ohjaajana toimii prof. Jari Porras

Helsingissä 3.9.2009

Matti Kosonen (0219065)

Munkkisaarenkatu 8 B 42

00150 Helsinki

040-7617257

makosone@lut.fi

TIIVISTELMÄ

Tekijä: Matti Kosonen
Nimi: **Kuormitustestausvälineen valinta Microsoft Office SharePoint Server 2007 – toteutuksessa**
Osasto: Tietotekniikan osasto
Vuosi: 2009
Paikka: Helsinki

Kandidaatintyö. Lappeenrannan teknillinen yliopisto, Teknillistaloudellinen tiedekunta, 41 sivua, 2 liitettä

Ohjaaja ja tarkastaja: Prof. Jari Porras

Hakusanat: kuormitustestaus, web-sovellukset, SilkPerformer 2008, Visual Studio Team System 2008 Test Edition, Microsoft Office SharePoint Server 2007

Kuormitustestaus on osa web-sovellusten kehitystä ja käyttöönottoa. Sillä varmistetaan sovellusten toimivuus ennalta määrätyn kuorman alla. Microsoft Office SharePoint Server 2007 (MOSS) on palvelintuote nykyaikaisten web-sovellusten luontiin ja ylläpitoon. Työssä vertaillaan kahta eri kuormitustestaustyökalua: SilkPerformer 2008 ja Visual Studio Team System 2008 Test Edition ja valitaan MOSS – web-sovelluksille paremmin sopiva työkalu.

Työssä vertaillaan työkaluja niiden ominaisuuksien perusteella sekä suorittamalla kuormitustestausta testausta varten luodulle MOSS – web-sovellukselle. Vaikuttavien tekijöiden perusteella työkaluja arvioidaan ja tämän perusteella saadaan tulos vertailulle.

Työn tuloksena Visual Studio Team System 2008 Test Edition sopii paremmin MOSS – web-sovelluksen kuormitustestausvälineeksi. Vertailussa kuitenkin havaittiin, että työkalut ovat melko tasavertaisia, ja käytännön tilanteesta riippuu, kumpi sopii paremmin. Tämä työ auttaa valinnan teossa.

ABSTRACT

Author: Matti Kosonen

Name: **Choosing Load Test Tool for Microsoft Office SharePoint Server
2007 – Web Application**

Department: Information Technology

Year: 2009

Place: Helsinki

Bachelor's thesis. Lappeenranta University of Technology, Faculty of Technology Management, 41 pages, 2 appendixes

Supervisor and inspector: Prof. Jari Porras

Keywords: load testing, web applications, SilkPerformer 2008, Visual Studio Team System 2008 Test Edition, Microsoft Office SharePoint Server 2007

Load testing is a part of the development and deployment of web applications. Load testing confirms the functionality of the applications in pre-defined load situations. Microsoft Office SharePoint Server 2007 (MOSS) is a server product for the creation and maintenance of modern web applications. This work compares two different load testing tools. Also a tool, which is better suitable to the MOSS – web applications, is chosen.

In the work tools are compared on the basis of their properties and by load testing the MOSS – web application that was created for this purpose. On the basis of affecting factors the tools are evaluated. Based on the evaluation the result of the comparison is obtained.

As a result Visual Studio Team System 2008 Test Edition is better for load testing MOSS – web applications. In this comparison it was also apparent that the both tools are quite equal. In some MOSS cases SilkPerformer might be a better tool. The conclusion is that this report makes the difficult choice little easier.

SISÄLLYSLUETTELO

1	Johdanto.....	4
1.1	Tausta.....	4
1.2	Tavoitteet ja rajaukset.....	5
1.3	Työn rakenne.....	6
2	Teoriaosuus.....	6
2.1	Taustatietoa tekniikoista.....	7
2.1.1	Web-sovellusten kuormitustestaus.....	7
2.1.2	MOSS.....	8
2.1.3	SilkPerformer.....	9
2.1.4	Visual Studio Team System 2008.....	9
2.1.5	Kuormitustestausvälineiden lisensointi.....	10
2.2	Käsitteet ja termit.....	10
2.3	Tutkimusongelma.....	11
2.4	Tutkimusongelman ratkaisutapa.....	11
3	Käytännön osuus.....	12
3.1	Kuormitustestauksen suunnittelu.....	12
3.2	Kuormitustestausvälineiden arvioinnin suunnittelu.....	13
3.3	Kuormitustestausympäristön valmistelu.....	13
3.4	Kuormitustestauksen suoritus SilkPerformerilla.....	14
3.4.1	SilkPerformerin konfigurointi.....	14
3.4.2	Testiprojektin määrittely.....	15
3.4.3	Testien luominen.....	15
3.4.4	Työkuorman määrittely.....	18
3.4.5	Testien ajaminen ja seuranta.....	19
3.4.6	Tulosten tutkiminen.....	20

3.5	Kuormitustestauksen suoritus Visual Studiolla	20
3.5.1	Konfigurointi	21
3.5.2	Testiprojektin määrittely ja testien luominen	21
3.5.3	Testien ajaminen ja seuranta.....	24
3.5.4	Tulosten tutkiminen.....	24
3.6	Esille tulleet ongelmat.....	25
3.7	Tulosten yhteenveto	26
3.8	Tulosten tulkinta	26
4	Pohdinta.....	26
4.1	Tulosten merkitys.....	26
4.2	Tulevaisuus	27
5	Johtopäätökset	27
	Lähdeluettelo	29
	Liitteet.....	31

SYMBOLILUETTELO

CSV	Comma-Separated Values
DSL	Digital Subscriber Line
IIS	Internet Information Services
MSDN	Microsoft developer network
MVP	Microsoft Most Valuable Professional
MOSS	Microsoft Office SharePoint Server 2007
SA	Software Assurance
VSTS	Microsoft Visual Studio Team System
XML	eXtensible Markup Language

1 JOHDANTO

1.1 Tausta

Tämä työ käsittelee kuormitustestausta ja kuormitustestaustyökaluja Microsoft Office SharePoint Server 2007 (jatkossa MOSS) – toteutuksissa. Työn avulla voidaan arvioida työkalujen ominaisuuksia ja löytää oikea työkalu käyttökontekstiin.

Kuormitustestauksella mitataan testattavan sovelluksen kolmea laadullista ominaisuutta: suorituskykyä, skaalautuvuutta ja luotettavuutta [1]. Näiden laadullisten vaatimusten avulla voidaan esimerkiksi todeta järjestelmän kyky käsitellä tiettyä yhtäaikaista käyttäjämäärää [2] tai optimoida järjestelmän vaatima infrastruktuurin kapasiteetti [1]. Testattavasta kohteesta riippuen tarkasteltavat ominaisuudet voivat vaihdella ja yksityiskohtiin kiinnitetään eri tavoin huomiota. Tässä työssä kuormitustestauksen kohteena on MOSS – web-sovellustoteutus, jossa keskeisimpiä laadullisia vaatimuksia ovat suorituskyky ja saatavuus [3].

MOSS on Microsoftin valmistama ensisijaisesti yrityksille suunnattu yhteistoiminta- ja sisällönhallintapalveluja tarjoava palvelintuote. Microsoftin 27 tarjoaman tuotteen joukosta MOSS on viime vuosina nostanut tuntuvasti sijoitusta keskeisimpien Microsoft-tuotteiden joukossa. Ennusteiden mukaan tulevaisuudessa tullaan näkemään yhä enemmän MOSS -ratkaisuja myös web-sovelluksissa. [4, s.3] MOSS pitää sisällään sisällönhallinnan työkalut, kehittyneet etsintäominaisuudet, sekä tarjoaa tarvittaessa julkaisuominaisuudet yrityksen selaimella käytettäviä intranet-, extranet- ja web-palveluja varten [5]. MOSS on räätälöitävissä ja siihen usein integroidaan muita tuotteita ja järjestelmiä. Tämän takia MOSS:n kuormitustestaus ja riittävän suorituskyvyn optimointi on haastava tehtävä.

Kuormitustestausvälineitä on tarjolla suuri määrä. Kuormitustestausta voi tehdä omatekoisillakin ohjelmilla, mutta se ei ole laajemmissa toteutuksissa suositeltavaa. Lisäksi on tarjolla avoimen lähdekoodin sovelluksia, joissa asiakastuki usein puuttuu ja toiminnallisuudet ovat rajalliset. Palveluna tarjottavat kuormitustestausvälineet

rajoittavat sisäisissä verkoissa toimivia ja kehitysvaiheessa olevien toteutusten kuormitustestausta. [1] Tässä tutkimustyössä on valittu kaksi kuormitustestausvälinettä testiympäristöön rakennettavan MOSS -toteutuksen testaamiseen. Ensimmäinen työkaluista on Microsoftin Visual Studio Team System 2008 Test Edition (jatkossa VSTS), joka on osa Visual Studio Team Suite 2008:aa. Toinen työkalu on Borlandin SilkPerformer 2008 (jatkossa SilkPerformer).

1.2 Tavoitteet ja rajaukset

Tutkimustyön tavoitteena on saada kattava kuva edellä mainitusta kahdesta kuormitustestausvälineestä. Kuormitustestausvälineiden vertailun lopputuloksena saadaan arvio, kumpi testausvälineistä soveltuu paremmin ensisijaisesti web-käyttöön tarkoitetuissa MOSS -toteutuksissa. Kirjallisuuskatsauksessa perehdytään Web-sovellusten kuormitustestaukseen, MOSS – web-ratkaisujen ominaispiirteisiin, sekä mahdollisimman syvällisesti VSTS:n ja SilkPerformerin ominaisuuksiin. Lisäksi kuormitustestauksen suorittamista varten käytetään erilaisia, kirjallisuudesta löytyviä, ohjeistuksia ja oppaita.

Tutkimustyö tehdään tuotteiden (MOSS, VSTS, SilkPerformer) koekäyttölisenssiversioilla. Esiselvitystyön perusteella ainoat rajoitukset koskevat tuotteiden käyttöaikaan, joka vaihtelee 30 päivästä 180 päivään. Tämä ei kuitenkaan aiheuta ongelmia, koska empiirisen tutkimuksen vaihe suoritetaan alle 30 päivässä. MOSS -toteutus tulee olemaan mahdollisimman yksinkertaistettu. Tarkoituksena on toteuttaa siihen mahdollisimman paljon erilaisia toiminnallisuuksia, joita kuormitustestausta suorittaessa hyödynnetään. Sisällölliset ja muut ominaisuudet, jotka eivät vaikuta vertailuun jätetään huomiotta. Esimerkiksi esteettisyys ja sisällön järkevyyys ei erota työkaluja toisistaan, jolloin niitä ei myöskään tarvitse toteuttaa testitoteutukseen.

Todellisissa MOSS -toteutusprojekteissa lopullinen kokonaisuuden kuormitustestaus tulee suorittaa tuotantoympäristössä tai tuotantoympäristöä kokoonpanoltaan vastaavassa ympäristössä, koska kuormitustestauksella ensisijaisesti mitataan

toteutuksen toimintaa tietyllä laitteistokokoonpanolla. Tässä tutkimustyössä toteutettavaa MOSS -toteutusta ei asenneta suorituskyvyltään ja laajuudeltaan todellisuutta vastaavaan palvelinympäristöön, vaan testiympäristönä käytetään virtualisoitua 32-bittistä palvelinympäristöä.

1.3 Työn rakenne

Työ koostuu viidestä luvusta. Ensimmäisenä Johdanto-luku esittelee työn taustaa, siinä käsitellään työn sisältöä, syitä työn toteutukselle, sekä tapoja kuinka haluttuun lopputulokseen aiotaan päästä. Lisäksi kerrotaan työn tavoitteet, eli mitä työltä odotetaan ja vedetään selkeät rajat työn laajuudelle. Teoriaosuus-luvussa esitellään kirjallisuudesta löytyvää tutkimuskohteeseen liittyvää materiaalia. Siinä käsitellään kuormitustestausta, MOSS -tuotetta ja kuormitustestausvälineitä.

Kolmantena lukuna kandidaatintyössä kuvataan käytännön osuus. Ongelmaa lähdetään ratkaisemaan kahdesta näkökulmasta. Ensimmäisenä kirjallisuuskatsauksen pohjalta voidaan tehdä hyvin vertailua. Toisena verrataan kuormitustestausvälineitä empiirisen tutkimuksen keinoin, jolloin hyödynnetään testiympäristöstä saatuja havaintoja ja tuloksia. Tulosten merkitystä ja MOSS – toteutusten kuormitustestauksen tulevaisuutta pohditaan neljännessä luvussa. Viimeisenä lukuna on johtopäätökset.

2 TEORIAOSUUS

Työn teoriaosuudessa esitellään kirjallisuuskatsauksen pohjalta saatuja tietoja työssä käsiteltävistä aiheista. Keskeisimmät aiheet, joista esitellään taustatietoa tässä työssä, ovat kuormitustestaus, MOSS, SilkPerformer ja VSTS. Lisäksi alkupuolella esitellään termit ja käsitteet. Teoriaosuudessa kuvataan tutkimusongelma tarkasti, mikä on kuormitustestausvälineen valinta MOSS – web-sovellukselle. Myös tutkimusongelman ratkaisutapa kuvataan tarkemmin. Tiivistettynä tutkimusongelma ratkaistaan kirjallisuuskatsauksen ja empiirisen tutkimuksen keinoin, käyttäen apuna kuormitustestausvälineiden vertailuun luotua arvostelutaulukkoa.

2.1 Taustatietoa tekniikoista

Tässä osiossa esitellään työssä esiintyviä käsitteitä ja niihin työn kannalta oleellisesti liittyviä asioita. Tällöin nämä asiat ja niiden merkitys voidaan ymmärtää. Ensin esitellään web-sovellusten kuormitustestausta yleisellä tasolla. Työkalujen kuvauksen lisäksi esitellään verrattavien kuormitustestaustyökalujen lisensointia ja hintoja.

2.1.1 Web-sovellusten kuormitustestaus

Kuormitustestaus on yksi ohjelmistojen ja web-sovellusten testauksen muoto, joilla mitataan kolme laadullista ominaisuutta: suorituskykyä, skaalautuvuutta ja luotettavuutta [1]. Tässä työssä kuormitustestauksen kohteena on MOSS – web-sovellustoteutus. Tällaisen toteutuksen kaksi keskeisintä vaatimusta ovat suorituskyky ja saatavuus [3]. Kuormitustestauksen avulla voidaan tutkia infrastruktuurin kapasiteettia [1] tai järjestelmän suoriutumisen halutunsuuruisesta yhtäaikaisesta käyttäjämäärästä [2]. Testattavasta kohteesta riippuen tarkasteltavat ominaisuudet voivat vaihdella ja yksityiskohtiin kiinnitetään eri tavoin huomiota. Kuormitustestausta voidaan myös suorittaa eri laajuuksissa kuten komponentti-, infrastruktuuri- tai arkkitehtuuritasolla. [1]

Kuormitustestaukseen kuuluu erilaisia työvaiheita, joiden suoritusjärjestys ja sisältö voivat vaihdella käytettävän työkalun mukaan. Seuraavaksi esitellään näitä työvaiheita ja niiden suoritusta. Testit perustuvat yksinkertaisiin testitapauksiin, joiden vaikutusta pyritään muuttamaan parametrisoimalla ja muokkaamalla testejä manuaalisesti. Testien parametrisointi perustuu muun muassa sovelluksissa oleviin lomakkeisiin ja niissä oleviin kenttiin. Parametrisointiin voidaan luoda esimerkiksi logiikkaa, mikä huomioi jo tehtyjä valintoja. Sovellusten parametrien sekä työaseman ja palvelimen välisten transaktioiden lisäksi testeihin voidaan luoda piirteitä todellisista käyttäjistä, kuten satunnaisia odotusaikoja ja käyttäjäsyötteitä. Tämän perustana voidaan käyttää todellisten käyttäjien luomaa dataa. [6]

Testien luonnin lisäksi testauksessa tulee määrittää työkuorma, jolla järjestelmää kuormitetaan. Kuormitus tapahtuu kuormitustestaustyökalun luomilla virtuaalikäyttäjillä, jotka suorittavat määriteltyjä testejä web-sovelluksessa. Kuormaan vaikuttaa muun muassa käyttäjämäärän käyttäytyminen. Erilaiset kuormamallit kuvaavat erilaisia käyttäytymismalleja, jotka vaikuttavat web-sovelluksen kuormittumiseen ja tätä kautta tutkittaviin mittaristoihin. [6] Kuormitukseen voi vaikuttaa edellä mainittujen lisäksi erinäiset muut tekijät kuten virtuaalikäyttäjien käyttämät yhteydet ja selaintyytit. [7]

Kuormitustestauksen aikana ja sen jälkeen voidaan tehdä erilaisia havaintoja sovelluksen käyttäytymisestä kuorman alla. Havaintoja tehdään haluttujen mittaristojen perusteella esimerkiksi vasteajoista, saatavuudesta tai aloitetuista käyttäjäistunnoista. [6,8] Rasitustestit etsivät lisäksi järjestelmän pullonkauloja äärimmäisen kuorman alaisuudessa [6]. Testauksen laajuus, tarkkuus, luotavat testit, tutkittavat mittaristot ja kuormitusmallit tulee valita sen perusteella mitä testeillä halutaan selvittää [1].

2.1.2 MOSS

MOSS on Microsoftin valmistama palvelintuote, joka sisältää useita erilaisia yhteistoiminta- ja sisällönhallintapalveluja. MOSS on yleistynyt nopeasti yrityksissä ja jatkossa MOSS –ratkaisut todennäköisesti yleistyvät web-sovelluksissa. [4, s.3]

MOSS pitää sisällään kuusi erilaista ominaisuutta. Ensinnäkin MOSS -sovellus toimii portaalina, mikä mahdollistaa web-sovelluksen julkaisemisen Internetissä tai käyttämisen intra- tai ekstranettinä. Sovellus mahdollistaa myös käyttäjäprofiilien luomisen, autentikoinnin, ja sisällön personoinnin. Toisaalta MOSS luo mahdollisuuksia yhteistyöhön virtuaalisena työtilana sekä mahdollistaa virtuaalisen sosiaalisen verkostoitumisen ja kanssakäymisen. Kolmanneksi se mahdollistaa sisällön koko elinkaaren hallinnan sisältäen esimerkiksi arvioinnin, julkaisun, arkistoinnin ja hävittämisen. [4 s. 4-12]

Neljäntenä ominaisuutena MOSS -sovellus sisältää kehittyneet etsintätoiminnot web-sovelluksesta ja eri tietovarastoista. MOSS etsintätoiminto pystyy indeksoimaan lähes minkä tyyppistä sisältöä tahansa. Toiminto myös arvioi osumien relevanssia. Viidenneksi MOSS mahdollistaa tiedon keräämisen elektronisesti lomakkeiden avulla. Datan kerääminen ja validointi voidaan kohdistaa kaikille, joilla on jonkinlainen selain. Myös lomakkeiden luonti onnistuu helposti. Viimeisenä ominaisuutena MOSS tarjoaa liiketoimintatiedon hallintaa, jonka avulla päästään käsiksi tärkeään tietoon ja voidaan seurata tarvittavia mittaristoja. Lisäksi dataa pystytään helposti analysoimaan ja raportoimaan. [4 s. 7-11]

2.1.3 SilkPerformer

SilkPerformer on yksi johtavista kuormitustestausvälineistä tänä päivänä. Sitä käytetään arvioimaan mm. Internet-palvelimien, tietokantapalvelimien ja jaettujen sovellusten suorituskykyä niiden kehitystyön aikana ja ennen käyttöönottoa. SilkPerformer mainostaa itseään nopeana ja kustannustehokkaana kuormitustestausvälineenä. SilkPerformer mahdollistaa sadoilla tai jopa tuhansilla virtuaalikäyttäjillä kuormitustestauksen mahdollisimman pienillä resursseilla. Tuotteessa on myös tehokkaat raportointivälineet, joilla analysoidaan palvelimien suorituskykyä ja voidaan löytää ympäristön pullonkaulat, jotta ne voidaan korjata. [7]

2.1.4 Visual Studio Team System 2008

Visual Studio Team System 2008 Team Suite on järjestelmä, mikä sisältää sovelluskehitykseen tarvittavat työkalut koko kehitystiimille. VSTS sisältää omat työkalut arkkitehdeille, kehittäjille ja testaajille. Työkaluilla on yhtenäinen käyttöliittymä ja toimintalogiikka. Lisäksi järjestelmä sisältää yhteistyö- ja säilytyskomponentit. Näiden avulla yhteistyö eri toimintojen välillä helpottuu. Visual Studio Team Suite tarkoittaa ohjelmistopakettia, joka sisältää kaikki VSTS:n työkalut. [9, s. 881-886]

VSTS Team Edition on VSTS:n testaajille tarkoitettu työkalu. Työkalu pitää sisällään toiminnot yksikkö- ja kuormitustestaukseen sekä web-sovellusten testaukseen. Näiden lisäksi työkalu sisältää erilaisia toimintoja testien mittaamiseen ja analysointiin. [9, s. 890]

2.1.5 Kuormitustestausvälineiden lisensointi

Ohjelmistoilla on erilaisia hankintavaihtoehtoja, mikä tekee niiden vertailusta hankalampaa. Microsoft tarjoaa Visual Studio Team System 2008 Team Suite – ohjelmistopakettia yhdellä lisensointivaihtoehdolla, jossa jokainen käyttäjä tarvitsee oman lisenssin. Pelkän lisenssin hinta on noin 5600 €. Lisäksi on mahdollista hankkia Software Assurance (SA) palvelu, mikä sisältää muun muassa MSDN-tilauksen. SA premium -palvelulla saa myös oikeudet uusiin ohjelmistoversioihin palvelun voimassaolon aikana. Eri vaihtoehtojen valinta voi nostaa tuotteen hinnan 10 600 €:n. [10,11]

Toinen vaihtoehto on hankkia vain Visual Studio Team Systemiin kuuluva Visual Studio Team System 2008 Test Edition. Tällöin ohjelmiston mukana ei tule Team Suiten muita kokonaisuuksia. Tämän yksittäisen osan lisenssin hinta on noin 2 800 € ja lisäpalveluiden arvo nostaa tuotteen arvoa kuten edellä. [10,11]

Borland SilkPerformer ohjelmiston hankintahinta 1000 virtuaalikäyttäjän kapasiteetilla on noin 100 000 €. Microsoftin SA palveluun verrattavissa oleva tuki- ja päivityssopimus vuoden ajalle maksaa 18 000 €. Toinen vaihtoehto on vuokralisenssi, jolla ohjelmiston saa käyttöön esimerkiksi kuukaudeksi. Ohjelmiston kuukausivuokra on noin 19 000 €.

2.2 Käsitteet ja termit

- Kuormitustestaus on testityyppi, jolla testataan testauksen kohteen käyttäytymistä kuormituksessa. [12]

- Rasiustesti on kuormitustestauksen yksi muoto, jolla arvioidaan testin kohteen kykyä suoriutua yli sille määriteltyjen rajojen. [12]
- MOSS eli Microsoft Office SharePoint Server on palvelintuote, mikä sisältää yhteistoiminta- ja sisällönhallintapalveluita. [4]
- Web-sovellus on ohjelmistojärjestelmä, jota käytetään Internetin yli. [13]
- .NET on ohjelmistokomponenttikirjasto, mikä sisältää komponenttien luokkakirjastot ja ajoympäristön. [14]

2.3 Tutkimusongelma

Web-sovellusten kehittämisen yksi tärkeä osa on kuormitustestaus, jolla varmistetaan järjestelmän toimivuus todellisessa käyttöympäristössä. Kuormitustestausta varten on olemassa työkaluja, joiden avulla testaus voidaan suorittaa ja sen tuloksia analysoida. Käytännön ongelmana on se, miten käytettävä työkalu tulisi valita haluttuun käyttötarkoitukseen. Tässä työssä toteutetaan MOSS -toteutuksissa käytettävien VSTS- ja SikPerformer-kuormitustestausvälineiden vertailua. Työn tutkimusongelmana on valita työkaluista MOSS -toteutuksiin paremmin soveltuva vaihtoehto.

2.4 Tutkimusongelman ratkaisutapa

Ongelmaa lähdetään ratkaisemaan kahdella tavalla: kirjallisuutta ja olemassa olevaa dokumentaatiota tutkimalla sekä empiirisellä tutkimuksella. Työkalujen vertailua varten luodaan arviointijärjestelmä, jonka perusteella tuotteiden ominaisuuksia arvioidaan. Tämän avulla tuotteiden ominaisuuksia voidaan verrata keskenään ja luoda kokonaiskäsitys tuotteiden soveltuvuudesta. Ominaisuuksien painotus perustuu kirjallisuudesta saatuihin tietoihin ja arvioihin.

Kirjallisuuskatsauksen aluksi perehdytään Web-sovellusten kuormitustestaukseen yleisesti, jolloin löydetään työkaluilta vaadittavia ominaisuuksia. Kirjallisuuskatsauksen perusteella pyritään lisäksi saamaan tietoa kuormitustestausvälineiden ominaisuuksista muun muassa mihin kaikkeen tuotetta

voidaan käyttää, sekä minkälaista tietoa se tuottaa. Lisensointiin liittyvät asiat määrittelevät millaisia liiketoiminnallisia vaatimuksia työkalujen käyttö asettaa. Kirjallisuuskatsauksen perusteella voidaan rakentaa empiirinen tutkimus, jolla täydennetään kirjallisuudesta puuttuvaa tietoa aiheesta. Kirjallisuuskatsaus auttaa myös fokusoimaan empiirisen tutkimuksen relevantteihin tekijöihin.

3 KÄYTÄNNÖN OSUUS

Empiirisen tutkimuksen aluksi valmistellaan suunnitelmat, joiden mukaan listataan selvitettävät asiat, luodaan testiympäristö ja testitapaukset. Selvitettäviä asioita ovat mittaritot työkalujen vertailuun ja erilaiset käyttökohteet. Testiympäristö asennetaan virtualisoituun palvelinympäristöön. Testiympäristöön asennettu MOSS -asennus muokataan vastaamaan suunnitelmassa asetettuja vaatimuksia. Testiympäristöön asennetaan ja otetaan käyttöön myös VSTS ja SilkPerformer. Testitapauksista kuvataan yksityiskohtainen toimintaohje kunkin testitapauksen suorittamiseksi.

Kuormitustestaussuunnitelman perusteella toteutetaan kuormitustestausta molemmilla työkaluilla vuorotellen, siten etteivät työkalut vaikuta toistensa toimintaan. Työkalujen toimintaa arvioidaan myös huonosti mitattavien ominaisuuksien kuten käytettävyyden suhteen. Mahdollisuuksien mukaan testauksista kerätään kvantitatiivista dataa työkalujen analysointia varten. Lopuksi käsitellään esille tulleita ongelmia, tehdään yhteenveto tuloksista ja tulkitaan tuloksia.

3.1 Kuormitustestauksen suunnittelu

Käytännön osuus alkaa kuormitustestaussuunnitelman valmistelulla. Kuormitustestaussuunnitelmassa pyritään keräämään mahdollisimman paljon MOSS -Web-sovelluksille ominaisia piirteitä, joiden testaus kuormitustestausvälineellä halutaan suorittaa. Tällaisia ominaispiirteitä ovat muun muassa sisältönostot, haku, lomakkeiden täyttö, kirjautumistoiminnot ja niin edelleen. Lisäksi kuormitustestaussuunnitelmassa kuvataan minkälaisia käyttäjiä, selaimia ja

verkkoyhteyksiä kuormitustestauksessa halutaan simuloida. Kuormitustestaussuunnitelmassa ei oteta kantaa kuormitustestausvälineen ominaisuuksiin, vaan se tehdään täysin MOSS – web-sovelluksen ehdoilla. Kuormitustestaussuunnitelma on kokonaisuudessaan liitteessä 1.

3.2 Kuormitustestausvälineiden arvioinnin suunnittelu

Ennen varsinaisen käytännön työn aloittamista suunnitellaan kuormitustestausvälineiden arvioinnissa käytettävä mittaristo, johon työn empiirinen osuus perustuu. Kirjallisuuskatsauksen perusteella arviointityökaluun, joka on taulukkomuotoinen, listataan kuormitustestausvälineeltä vaadittavia ja toivottuja ominaisuuksia ja toimintoja. Ominaisuudet ja toiminnot arvotetaan pisteillä yhdestä viiteen sen perusteella, kuinka relevantti kukin ominaisuus on MOSS – web-sovelluksen kuormitustestauksessa. Pistemäärä yksi tarkoittaa sitä, että ominaisuudella on vähäinen merkitys ja viisi, että ominaisuus on erittäin tärkeä. Nolla-tason ominaisuuksia ei arvostelutaulukkoon ole järkeä listata, koska niillä ei ole merkitystä valintaa tehdessä.

Lisäksi taulukkoon tulee sarakkeet molemmalle kuormitustestausvälineelle, johon täytetään työkalukohtaiset arviot ominaisuudesta tai toiminnosta kuormitustestausta tehdessä. Arviot tarkistetaan vielä lopuksi, kun kuormitustestaus on suoritettu molemmilla työkaluilla, jotta arvioinnista tulisi mahdollisimman objektiivinen. Lopuksi lasketaan ominaisuuksille tai toiminnoille työkalukohtaiset pisteet kertomalla arvio MOSS -relevanssiarvolla. Lopuksi lasketaan työkalukohtaiset pisteet yhteen ja eniten pisteitä saanut työkalu valitaan. Kuormitustestausvälineiden arviointitaulukko on liitteenä 2.

3.3 Kuormitustestausympäristön valmistelu

Kuormitustestausympäristö asennetaan virtualisoituun palvelinympäristöön, jota ajetaan työasemalla Microsoft Virtual PC -ohjelmistolla. Käyttöjärjestelmänä virtuaalipalvelimella toimii 32-bittinen Windows Server 2003.

Kuormitustestauspalvelimelle asennetaan ja konfiguroidaan käyttövalmiiksi Microsoft Office SharePoint Server 2007. MOSS toimii tällöin niin sanotussa Single Server – asennuksena, jossa kaikki komponentit (edustapalvelin, sovelluspalvelin ja tietokantapalvelin) on asennettu samalle palvelimelle.

Seuraavana MOSS Web-sovelluksen valmisteluissa on web-sovelluksen luonti, joka luodaan MOSS:n keskitetyllä hallintatyökalulla oletusasetuksin. Luotava web-sovellus tulee ajettavaksi palvelimella toimivan IIS 6.0:n (Internet Information Services) päälle. Web-sovelluksen päälle luodaan MOSS:n keskitetyllä hallintatyökalulla sivustokokoelma käyttäen Julkaisemisportaali-sivustomallia. Luotava sivustokokoelma toimii kuormitustestauksen MOSS – web-sovelluksena.

Juuri luotu MOSS – web-sovellus muokataan vastaamaan suunnitelmassa asetettuja vaatimuksia, luomalla sivustolle muun muassa sisältönostoja, lomakkeita, sisältöä hakua varten ja käyttäjiä. Lisäksi esimerkiksi hakua varten on MOSS:n keskitetyllä hallintatyökalulla tehtävä sisällön täysi indeksointi. MOSS – web-sovelluksella ei ole muita vaatimuksia, kuin että kuormitustestausuunnitelmassa mainitut asiat on toteutettu ja että ne ovat käytettävissä kuormitustestauksessa.

3.4 Kuormitustestauksen suoritus SilkPerformerilla

Viimeisenä kuormitustestausympäristöön asennetaan kuormitustestausvälineet. VSTS ja SilkPerformer asennetaan oletusasetuksin ja asetukset konfiguroidaan vastaamaan virtuaalipalvelimen ominaisuuksia.

3.4.1 SilkPerformerin konfigurointi

Ohjelmiston asennuksen jälkeen kuormitustestausvälineeseen tulee asettaa globaalit asetukset, jotka eivät liity yksittäiseen projektiin. Asetukset koostuvat kolmesta kokonaisuudesta: Workbench, recorder ja remote agents. [7, 15]

1. Workbench sisältää kuormitustestauksen perusasetukset, testauksessa käytettävät työasemat, hakemistot, joita käytetään testeissä ja joita testi generoi sekä yleiset ulkoasuasetukset muun muassa skriptien ulkoasuun ja tulosten näyttämiseen.
2. Recorder sisältää perusasetukset, kuinka nauhoittaa ja muodostaa skriptejä. Siinä myös määritellään asetukset mitä verkkoliikenteestä tallennetaan ja kuinka niistä muodostetaan skriptejä. Recorder -osiossa määritellään testauskohteiden profiilit ja hallinnoidaan niitä. Profiilit sisältävät muun muassa tietoa testauskohteiden ohjelmistoista, jolloin se vaikuttaa oleellisesti testausdatan keräykseen. Recorder osiossa määritellään myös muita, esimerkiksi proxyihin ja portteihin, liittyviä asetuksia.
3. Remote agents sisältää tiedot testauksessa mahdollisesti käytettävistä muista etätyöasemista. Ne voivat olla lähiverkossa, Internetissä tai palomuurien takana, jolloin kontrollerin ja etäkoneiden välinen tiedonsiirto tulee määritellä. [7, 15]

3.4.2 Testiprojektin määrittely

Projekti sisältää perustiedot ja kaikki resurssit, mitä tarvitaan kuormitustestauksen suorittamiseksi. Näitä resursseja ovat mm. työkuorma, profiilit, testiskriptit, datatiedostot, agenttitietokoneet ja palvelinpään seuranta. [7, 15]

3.4.3 Testien luominen

Testiskripteillä kuvataan toimintoja, joita virtuaaliset käyttäjät simuloivat testauksen aikana. Testiskriptit luodaan SilkPerformerin BDL-skriptauskielellä (Benchmark description language), mikä muistuttaa Pascal-ohjelmointikieltä. Testiskriptejä voidaan luoda kolmella eri tavalla: nauhoittamalla, kirjoittamalla tai muokkaamalla. Helpoin tapa on luoda skriptit nauhoittamalla SilkPerformer Recorderin avulla. Recorder nauhoittaa verkkoliikennettä siten, että testi voidaan simuloida uudelleen. Toinen tapa on skriptien luonti manuaalisesti alusta lähtien. Kolmas tapa on muokata olemassa olevia esimerkkiskriptejä vastaamaan haluttua. [7, 15]

3.4.3.1 Testiskriptin nauhoitus

Tässä työssä skriptit luodaan nauhoittamalla. Skriptin nauhoituksessa valitaan käytettäväksi ennalta määriteltyä ohjelmistoprofiilia (Internet Explorer). Myös testauksen kohdesivusto määritellään nauhoituksen määrittelyn yhteydessä. Nauhoitus toteutetaan kuormitustestaussuunnitelman mukaisesti. Nauhoittaminen tapahtuu toimimalla kohdesivustolla, kuten virtuaalikäyttäjän halutaan toimivan kuormitustestaustilanteessa. Yhteen skriptiin nauhoitetaan yksittäinen toiminto tai lyhyt toimintaketju ja se tallennetaan skriptikantaan. Tallennuksen jälkeen skripti on manuaalisesti muokattavissa. Skriptejä luodaan niin monta, että kaikki suunnitelmassa kuvatut testit voidaan toteuttaa tuotettujen skriptien avulla. [7, 15]

3.4.3.2 Testiskriptin testaus ja analysointi

Skriptin toimivuus testataan TryScript-toiminnon avulla, joka toistaa nauhoitetut toiminnot, näyttää kulkevan datan sekä kirjoittaa lokit ja raporttitiedostot. Skriptin testauksen avulla voidaan tarkistaa, toimiiko skripti sellaisenaan, vai vaatiiko se virheiden korjausta. TryScriptistä saatuja tuloksia analysoidaan käyttämällä TrueLog Explorer -työkalua. Työkalussa on kolme eri toimintoa analysointiin ja näiden lisäksi skriptillä ajettu virtuaalikäyttäjän toiminta voidaan toistaa animoituna näkymänä. Seuraavassa käydään läpi TrueLog Explorer -työkalun analysointitoiminnot. [7, 15]

1. Virtuaalikäyttäjän yhteenvetoraportin katselmoinnissa analysoidaan käyttäjän toimintaa. Siinä kuvataan yksityiskohtainen yhteenveto yksittäisen virtuaalikäyttäjän toiminnasta ja ajallisista mittareista TryScript -testiajosta.
2. Virheiden etsintä -toiminnolla voidaan käydä kaikki virheet läpi.
3. Testiajoa ja nauhoitusajoa voidaan vertailla keskenään sivutilastoja, jotka sisältävät muun muassa erilaisia tilastotietoja sivun lataukseen liittyen. [7, 15]

3.4.3.3 Testiskriptien kustomointi

TrueLog Explorer:n toiminnolla voidaan korvata staattiset istuntotunnisteet dynaamisilla istuntotunnisteilla. TrueLog Explorer tunnistaa automaattisesti mahdolliset istuntokohtaiset tunnisteet ja tämän tiedon avulla ne voidaan muuttaa

dynaamisiksi aputyökalun avulla. Työkalulla voidaan tarvittaessa validoida muutoksen toimivuus. [7, 15]

Käyttäjien syötetietojen parametrisoinnilla voidaan nauhoituksessa luodut html-lomakkeiden syötteet parametrisoida. Työkalu tunnistaa html-lomakkeet ja toteuttaa parametrisoinnin skriptin arvoilla tai uusilla esim. ulkoisesta tietolähteestä haettavilla arvoilla. Samalla määritellään myös parametrien käyttötapa esimerkiksi arvojen satunnainen sijoitus. [7, 15]

Skriptiin voidaan lisätä erilaisia tarkistuksia, joilla varmistetaan siitä, että käyttäjä vastaanottaa kaiken datan palvelimelta kuormitustestauksen aikana. Tarkistus voi koskea esimerkiksi sivun jotakin tekstiosiota. Kuormitustestaus ei muuten paljasta datan katoamista, jolloin tarkistuksen avulla myös virtuaalikäyttäjän kokemat virheet voidaan havaita kuormitustestauksessa. [7, 15]

3.4.3.4 Käyttäjäprofiilien määrittely

Käyttäjäprofiili tarkoittaa virtuaalisen käyttäjän teknisiä ominaisuuksia: käytettävää selainta ja yhteyden nopeutta. Käyttäjäprofiilien määrittelyllä voidaan luoda erilaisia profiileita erilaisille käyttäjille. Käyttäjäryhmä tarkoittaa virtuaalisen käyttäjän käyttäytymiseen liittyviä ominaisuuksia. Käyttäjäryhmään kuuluvat virtuaaliset käyttäjät suorittavat samankaltaisia toimenpiteitä keskenään, mutta eivät kuitenkaan toimi identtisesti. Käyttäjäryhmiä voivat olla esimerkiksi ylläpitäjät ja peruskäyttäjät. Käyttäjäryhmän määrittelyssä määritellään käyttäjien toteuttavan tiettyntyyppisiä transaktioita käyttökerran aikana. Käyttäjätyyppi tarkoittaa käyttäjää, jolla on tietty käyttäjäprofiili ja käyttäjäryhmä, mutta myös oma yksilöllinen skripti, minkä käyttäjä toteuttaa. [7, 15]

3.4.4 Työkuorman määrittäminen

3.4.4.1 Vertailukohtien määrittäminen

Vertailukohtien (baseline) määrittämisessä etsitään järjestelmän kuormituksen ideaalitaso. Vertailukohta syntyy yhden käyttäjän tai käyttäjätyyppien kuormittamisesta järjestelmää. Kuormituksesta saadaan tulokset käytettävistä mittaristoista, jotka edustavat vertailukohtaa ja joihin kuormitustestauksen tuloksia verrataan. [7, 15]

Vertailukohtien määrittämiseen SilkPerformerissa on oma toimintonsa, jossa määritellään testattavat käyttäjätypit ja ajetaan määrittämissarja. Määrittämissarjan tulosten perusteella arvioidaan, onko vertailukohta hyväksyttävällä tasolla, jotta sitä voidaan käyttää vertailutasona kuormitustestauksessa. Arviointiin käytetään vertailukohtareporttia, joka sisältää yleistä informaatiota ja tietoa käyttäjätyypeistä. Yleinen informaatio sisältää tiedot projektista ja kyseisestä vertailukohtien määrittämissarjasta. Käyttäjätypit sisältävät tiedon käyttäjätyyppien vasteajoista tyypeittäin sekä muita mitattuja määreitä. [7, 15]

Vertailukohtien hyväksynnän yhteydessä voidaan myös määrittää halutut raja-arvot vasteajoille. Näitä käytetään kuormitustestauksen yhteydessä luokittelemaan vasteaikoja hyväksi, hyväksyttäväksi tai ei hyväksyttäväksi. Vertailukohtien hyväksynnän yhteydessä voidaan myös asettaa tarkkailuikkuna kuormitustestausta varten. Tarkkailuikkunalla voidaan tarkkailla haluttuja mittaristoja joko palvelimella tai työasemalla. Monitorointia voidaan käyttää kuormitustestauksen tukena reaaliaikaisesti sekä mittauksen tuloksia testauksen tuloksien vertailussa. [7, 15]

3.4.4.2 Kuormitustestauksen työkuorman määrittäminen

Työkuorman määrittäminen kuormitustestausta varten sisältää kuormitustestauksen toiminnan määrittämisen. SilkPerformerissa on sisäänrakennetut mallit erityyppisille kuormitustestauksille. Mallit ovat erilaisia simulaatioita erityyppiselle käyttäytymiselle. Eri mallit on esitelty seuraavassa: [7, 15]

1. Kasvava työkuorma aloittaa kuormituksen tietystä tasosta kasvattaen kuorman maksimiin.
2. Tasainen työkuorma pitää käyttäjämäärän koko ajan samana toistaen yksittäisten käyttäjien transaktioita jatkuvasti.
3. Dynaaminen työkuorma mahdollistaa käyttäjämäärän muuttamisen tietyissä rajoissa manuaalisesti.
4. Koko päivän työkuorma mahdollistaa käyttäjätyyppikohtaisten päiväjakaumien käyttämisen, jolloin erilaisia skenaarioita voidaan mallintaa.
5. Jonotyökuorma mallintaa tietyn määrän tapahtumia tietylle aikavälille, mutta tapahtumien ajoitus on satunnainen. Tällöin testi on suoritettu, kun kaikki tapahtumat on saatu suoritettua.
6. Verifiointityökuormalla verifioidaan kohde esimerkiksi regressiotestauksessa.
[7, 15]

Työkuorman asettaminen sisältää tarvittavien parametrien määrittämisen. Kokonaiskäyttäjämäärä voidaan asettaa tavoitteiden mukaisesti. Vertailukohdan perusteella SilkPerformer laskee yhtäaikaisten käyttäjien määrän oletusarvon. Myös vaadittu kaistanleveys tulee määrittää ja oletusarvo lasketaan automaattisesti vertailukohdan perusteella. Projektiin voidaan määrittellä useita erilaisia työkuormamalleja, mutta vain yhtä kerrallaan voidaan käyttää kuormitustestauksessa.
[7, 15]

3.4.5 Testien ajaminen ja seuranta

Testi ajetaan luodun kuormitusmalliin perustuen. Testin monitorointi testin aikana sisältää tietoja yksittäisistä virtuaalisista käyttäjistä, käyttäjäryhmistä ja palvelimesta. Käyttäjät ja käyttäjäryhmät sisältävät tietoa niiden tilasta, vasteajoista, suoritetuista tehtävistä, virhetilanteista ja koneen resurssien käytöstä. [7, 15]

Palvelimen tilaa seurataan Performance Explorer – työkalun avulla. Työkalu näyttää testin tuloksia reaaliaikaisesti sekä testin jälkeen. Seurattavia mittaristoja voi olla useita ja usealta eri palvelimelta. [7, 15]

3.4.6 Tulosten tutkiminen

SilkPerformerissa on kaksi työkalua tulosten tutkimiseen: TrueLog Explorer ja Performance Explorer. [7, 15]

1. TrueLog Explorerin avulla voidaan selata kuormitustestauksen aikana suoritettuja testejä, jotka ovat päättyneet virheeseen testin aikana. Sivukohtaisten tilastojen avulla voidaan löytää syyt virhetilanteisiin esimerkiksi vasteajoista.
1. Performance Explorerissa on luokiteltuna erilaista tulosdataa testistä.
 - a. Yleiset tiedot kuvaa perusmittarein mitattuna kokonaissuorituskykyä. Mittareita ovat esim. aktiivisten käyttäjien määrä, vasteajat ja virheiden määrät. Mittarit ovat myös esitetty aikasarjoina testin ajalta.
 - b. Yhteenvetotaulukot sisältävät mittareiden arvoja keskimääräisinä arvoina taulukkomuodossa.
 - c. Luokittelussa eri sivut luokitellaan niiden latausajan perusteella.
 - d. Käyttäjätyypeistä on esitetty yksityiskohtaisia tietoja niiden käyttäytymisestä ja erityyppisten tilanteiden määrästä.
 - e. Omissa kuvaajissa voi muokata kuvaajia haluamistaan mittaristoista.
 - f. Omissa taulukoissa voi muokata numeerisia taulukoita haluamistaan mittaristoista.
 - g. Yksityiskohtaiset kuvaajat sisältävät tarkat kuvaajat kuvaajista, jotka ovat mukana raportissa. Tällöin esim. yleisissä tiedoissa esitettyjä kuvaajia voidaan tutkia tarkasti. Kuvaajia voi myös tarkastella useaa kuvaajaa kerralla sekä kohdistetusti tiettyihin hetkiin. [7, 15]

3.5 Kuormitustestauksen suoritus Visual Studiolla

Kun kuormitustestaus SilkPerformerilla on suoritettu, tehdään kuormitustestaussuunnitelman mukainen testaus VSTS:llä.

3.5.1 Konfigurointi

VSTS:n asennuksen jälkeen tuote on lähes käyttövalmis. Ainoastaan käytettävä tietokanta on asetettava käyttöön. Tietokannan luontia varten VSTS:n mukana tulee tietokantaproseduuri, joka luo kannan. Viittaus tietokantaan tehdään VSTS:n asetuksista. [16]

3.5.2 Testiprojektin määrittely ja testien luominen

Testiprojekti luodaan kuten mikä tahansa Visual Studio –projekti. Projektimallina käytetään Testaus-mallia. Testiprojektiin voi lisä erilaisia testejä. Kuormitustestauksessa käytetään web-testejä ja kuormitustestejä. Web-testeillä nauhoitetaan toiminta web-sovelluksessa ja kuormitustestillä ajetaan web-testejä eri tavoin. [16]

3.5.2.1 Testin nauhoitus

Testi voidaan luoda nauhoittamalla, manuaalisesti kirjoittamalla tai tuomalla osia jo tehdyistä testeistä. Nauhoitettu testi voidaan generoida koodiksi, joka on .NET-kieltä. Nauhoittaminen onnistuu myös javascript, AJAX-kutsuista ja popup-ikkunoista. Dynaamiset parametrit, kuten sessiotunnisteet tunnistetaan automaattisesti. Testi nauhoitetaan web-testin nauhoitus-työkalulla. Työkalu integroituu selaimen ja nauhoittaa sivuston selailun aiheuttamat kyselyt verkkoliikenteen sijaan. Tällöin sivustosta riippuvat ominaisuudet kuten kuvat eivät vaikuta itse testiin. [16]

Testeihin voidaan lisätä odotusaikoja, jotka kuvaavat käyttäjän reagointia sisältöön. Sivuston latautumisaikaa ei lueta odotusajaksi. Kullekin testille voidaan määritellä tavoiteaika, johon sivuston latautuminen kokonaisuudessaan tulisi kyetä. Kuormitustestauksen tuloksena saadaan latausajan keskiarvo ja kertojen määrä, jotka ovat päässeet tavoiteaikaan. [16]

3.5.2.2 Testien testaus ja analysointi

Testien testauksella ja analysoinnilla tarkistetaan testien toimivuus. Testin ajon tuloksia voidaan tarkastella web-testien katseluohjelman avulla, jossa on kaksi eri osiota testin tutkimiseen. Näiden avulla epäonnistuneet kutsut voidaan löytää ja havaita virheeseen johtanut syy. [16]

1. Yleiskuvauksessa kuvataan HTTP-kysely, vasteajat sekä pakettien koko.
2. Tarkassa kuvauksessa kuvataan renderöity sivu, tehty kysely graafisesti ja raakadatana, vastaus, sivun konteksti ja yksityiskohdat. [16]

Testejä voidaan ajaa erilaisilla asetuksilla puuttumatta kuitenkaan koko projektin testiasetuksiin. Yhden testin ajossa voi vaikuttaa iteraatioiden määrään, selaimen, yhteyden nopeuteen ja odotusaikojen simulointiin. [16]

3.5.2.3 Testin muokkaaminen

Nauhoitettua kyselyä voidaan muokata web-test -editorin avulla. Itse testi on .NET-luokka, joka sisältää järjestyksessä kaikki testissä tehtävät HTTP-kyselyt. Testi kuvataan kyselypuuna, josta jokaista kyselyä voidaan erikseen muokata. Myös testin parametrit voidaan esittää puussa. Kyselyitä voidaan muokata joko C# tai Visual Basic kielillä. [16]

Testille voidaan asettaa ominaisuuksia, kuten virheentarkistuksia tai reagointia virheisiin. Sivustojen latautumista sisällöllisesti oikein voidaan myös testata, jolloin sivustojen toiminta kuormitustestauksessa voidaan validoida myös sisällöllisesti. Validointia voidaan tehdä myös tulostamalla testin kyselyihin tulevat vastaukset. Validointisääntöjä voidaan myös tehdä itse, jolloin voidaan luoda monimutkaisia tarkastuksia palvelimen vastauksille. [16]

Testiin voidaan lisätä kutsuja toisiin testeihin, jolloin testin aikana suoritetaan tarvittaessa myös jokin toinen testi. Testin syötedata voidaan tarvittaessa

parametrisoida ja lukea ulkopuolisesta tietolähteestä, joka voi olla tietokanta, csv- tai xml-tiedosto. [16]

3.5.2.4 Vertailukohtien määrittäminen

Vertailukohtien määrittämiseen ei Visual Studiosissa ole erityistä työkalua. Vertailukohta voidaan määrittää luomalla testi, joka asetetaan itse vastaamaan vertailukohtaa. Tulosten analysoinnissa tätä testiä voidaan käyttää vertailussa muihin toteutettuihin testeihin.

3.5.2.5 Kuormitustestauksen työkuorman määrittäminen

Työkuormaan vaikuttavat virtuaalikäyttäjien määrä ja testattavien testien määrä. Lisäksi testin työkuormaan vaikuttavat käytettävät selaimet ja virtuaalikäyttäjien käyttämät yhteydet. Virtuaalikäyttäjien määrä määritellään kuormamallina. Erilaisia kuormamalleja on kolme: [16]

1. Pysyvä kuorma pitää virtuaalikäyttäjien määrän jatkuvasti samana.
2. Asteittainen kuorma kasvattaa kuormaa alkukuormasta maksimikuormaan tietyin aikaväleihin.
3. Tavoiteperusteinen malli toimii kuten asteittainen kuorma, mutta maksimimäärä määritetään tietyn suorituskykymittarin perusteella. Tällöin esimerkiksi prosessorin käyttö voi määrittää testin maksimitason. [16]

Test Mix tarkoittaa sitä, miten paljon kutakin testiä halutaan ajaa. Tähän on kolme mahdollista vaihtoehtoa

1. Aloitetun testin perusteella testien määrä jakautuu prosentuaalisesti tasan.
2. Yhtäaikaisten käyttäjien perusteella yhtä aikaa suoritettavana on prosentuaalisesti tietyn verran kutakin testiä.
3. Testaustahdin perusteella kutakin testiä suoritetaan tietty määrä tietyssä ajassa virtuaalikäyttäjää kohden. [16]

Browser mix määrittelee, mitä eri selaimia käytetään testissä ja missä suhteessa. Network mix määrittelee, millaisia yhteyksiä virtuaalisilla käyttäjillä on ja missä suhteessa. Odotusaikaprofiilit kuvaavat sitä, miten kauan käyttäjiltä menee suorittaessa tiettyjä toimintoja. Profiileja voidaan muokata esim. normaalijakautuneeksi tai käyttää nauhoitettuja odotusaikoja. [16]

3.5.3 Testien ajaminen ja seuranta

Testeihin määritellään testien kesto, datan keruun taso sekä mitattavien mittareiden määrä ja mittausväli. Yhteen testiin voidaan linkittää useita eri asetuksia, jolloin sama testi voidaan ajaa eritasoisina testeinä. Testin ajon yhteydessä testiin valittuja mittareita voidaan seurata omasta näkymästä. [16]

3.5.4 Tulosten tutkiminen

Tuloksia voidaan analysoida yksityiskohtaisesti tai yleisellä tasolla. Yleisellä tasolla kuormitustestauksen yhteenvedossa esitetään viisi asiaa: [16]

1. Testiajon tiedot sisältävät yleiset tiedot testiajosta.
2. Yleistulokset sisältävät mittarit: kyselyitä sekunnissa, epäonnistuneiden kyselyiden määrä, keskimääräinen vasteaika ja keskimääräinen sivun latausaika.
3. Viisi hitainta sivua.
4. Viisi hitainta testiä.
5. Viisi hitainta SQL-operaatiota. [16]

Yksityiskohtaisissa tiedoissa on listattuna kaikki tärkeät tiedot kuudesta eri osa-alueesta. Kaikkien osa-alueiden yksilöt ja niihin liittyvät mittarit ja tapahtumat on listattu yksityiskohtaisesti. [16]

1. Testitulokset
2. Sivujen tulokset

3. Transaktioiden tulokset
4. Testattavan systeemin resurssit
5. Kontrollerin ja agentin resurssit
6. Virheet [16]

3.6 Esille tulleet ongelmat

Käytännön osuudessa ei tullut suuria ongelmia vastaan. Kaikki meni lähes suunnitelmien mukaan. Pahin ongelma oli todennäköisesti ajankäyttöön liittyvä. Käytännön osuudelle varattu aika ei riittänyt hyvin siihen, että kuormitustestausvälineisiin ja toimintaan käytännössä olisi ehtinyt perehtymään tarpeeksi syvällisesti. Tällöin jokin asia tai toiminto on voitu ymmärtää väärin. Kaikki toiminnot eivät myöskään olleet täysin selviä. Jotkin toiminnot eivät toimineet täysin. Syy tähän voi olla ohjelmien virheellinen käyttö tai asennusvirhe.

Toinen pieni ongelma oli testiympäristön tehon vähyys. Etenkin SilkPerformerin osalta ohjelman toiminta oli hitaampaa kuin sen ehkä tulisi olla. Ympäristön hitaus aiheutti myös sen, että työkaluihin perehtyminen jäi pienemmäksi kuin oli suunniteltu. Kolmas ongelma oli SilkPerformerin kokeiluversion rajoite, jossa virtuaalikäyttäjien määrä on rajoitettu kymmeneen. Tästä ei tosin seurannut ongelmia, koska laitteisto ei olisi pystynyt pyörittämään kovin paljoa suurempia virtuaalikäyttäjämääriä.

Viimeinen ongelma oli luoda kuormitustestaussuunnitelman mukainen kokoonpano kahteen eri työkaluun siten, että se toimisi samalla tavalla molemmissa. Tämän takia testitulosten vertailu tehtiin huomattavasti yksinkertaisimmilla kokoonpanoilla ja vähemmällä testitapausmäärällä kuin oli suunniteltu. Tästä huolimatta raportit olivat niin erilaisia, että niiden vertailu oli vaikeaa. Joitain yhtäläisyyksiä tuloksista pystyi huomaamaan, mutta tarkempi tuloksien vertailu jäi ajanpuutteen takia tekemättä. Muistutuksena vielä, että työn rajauksissa mainitaan, ettei kuormitustestauksen tuloksiin kiinnitetä paljoa huomiota.

3.7 Tulosten yhteenveto

Kirjallisuuskatsauksesta saa hyvän kuvan kuormitustestaustyökaluista ja niiden toiminnoista. Sen perusteella kuormitustestauksen suoritus kuormitustestaussuunnitelman mukaan onnistuu helposti. Kuormitustestauksen suoritus puolestaan osoittaa sen, että molemmat työkalut soveltuvat MOSS – web-sovellusten kuormitustestaukseen. Analysoitavaksi jää vain se kumpi työkaluista sopii paremmin.

Valintaa tehdessä käytetään apuna kuormitustestausvälineen vertailutaulukkoa (Liite 2) ja siihen syötetään arviot ja perustelut MOSS:n liittyen kuormitustestausta tehdessä. Vertailutaulukon mukaan parempi työkalu kuormitustestauksen suorittamiseksi MOSS – web-sovelluksella on VSTS.

3.8 Tulosten tulkinta

Tuloksesta voidaan päätellä, että valinta ei ole kovin kriittinen, vaan valinta on loppujen lopuksi tehty melko pienin perustein, koska työkalut eivät eroa toisistaan kovin merkittävästi MOSS – web-sovellusten testauksessa. Koska merkittäviä eroja tuotteiden välillä ei ole, keskeisimmiksi erottaviksi tekijöiksi paljastuvat VSTS:n eduksi seuraavat asiat: MOSS -kehittäjälle VSTS:n käyttöliittymä on ennestään tuttu ja esimerkiksi versionhallinta voidaan hoitaa kuormitustestauksen osalta samoin kuin kehityksen osalta, koska Visual Studiolla kehitetään MOSS -ratkaisuja. Skriptien ohjelmointikieli on myös sama kuin MOSS:ia ohjelmoitaessa. Kaiken lisäksi VSTS:n kustannukset tulevat halvemmaksi.

4 POHDINTA

4.1 Tulosten merkitys

Jokseenkin epävarma tulos tarkoittaa sitä, että valinta ei ole ehdoton, vaan aina tapauskohtaisesti on punnittava eri ominaisuuksia ja tehtävä päätös sen mukaan mikä tai mitkä ominaisuudet painavat eniten. Esimerkiksi jos halutaan tarkkaa raportointia

ja hankkia kuormitustestausvälineen vuokralisenssillä, tulee silloin todennäköisesti valita SilkPerformer.

Tämän työn ja sen tuotosten tehtävänä on auttaa kuormitustestausvälineen valinnan teossa, eikä antaa ehdotonta vastausta. Lisäksi tulee muistaa, että markkinoilla on paljon muitakin kuormitustestausvälineitä, jotka saattavat joissain tilanteissa olla parempia. Tässä työssä oli työmäärän rajaamiseksi valittu vain nämä kaksi, oletetusti parhaiten MOSS -ratkaisuihin sopiviksi.

4.2 Tulevaisuus

Ohjelmistoversiot päivittyvät nykyisin muutaman vuoden välein. Nyt on tiedossa uudet versiot työssä käytetyistä ohjelmistoista. MOSS:sta on tulossa uusi versio 2010, samoin Visual Studiosta. SilkPerformerin seuraava versio on jo 2009.

MOSS – web-sovellusten kuormitustestauksen osalta ei ole uutta tiedossa. Todennäköisesti näillä työkaluilla on pärjättävä ainakin muutama vuosi eteenpäin ennen kuin uskaltaa toivoa MOSS:n vaatimukset paremmin täyttävää kuormitustestausvälinettä.

5 JOHTOPÄÄTÖKSET

Työn tarkoituksena oli verrata kahta eri kuormitustestaustyökalua MOSS – web-sovellukselle : SilkPerformer 2008 ja Visual Studio Team System 2008 Test Edition. Työssä rakennettiin virtualisoituun palvelinympäristöön testausympäristö, jossa oli MOSS – web-sovellus ja kuormitustestausvälineet. Näiden lisäksi suunniteltiin kuormitustestaus ja arviointijärjestelmä kuormitustestausvälineen valinta varten. Kuormitustestaussuunnitelmassa keskityttiin testaamaan MOSS:lle ominaisia piirteitä. Arviointijärjestelmässä myös painopisteenä oli MOSS:n vaatimukset ja ominaisuudet.

Testiympäristössä suoritettiin kuormitustestaus suunnitelman mukaan ja sen perusteella tehtiin työkalujen arviointi. Arvioinnin tuloksena MOSS – web-sovelluksen kuormitustestaukseen soveltuu paremmin VSTS. Kuitenkin aina tapauskohtaisesti tulisi arvioida tilanne uudestaan, koska tilanteet ovat usein erilaisia, vaikka samoilla välineillä ja tekniikoilla tehtäisiinkin web-sovelluksia.

LÄHDELUETTELO

- [1] Borland. "Load, Stress and Performance Testing - Choosing a Load Testing Strategy". Internet dokumentti, URL:
http://myinfo.borland.com/forms/8915_borl_wp?docID=701300000008Lac
(31.5.2009).
- [2] G.A. Di Lucca, A.R. Fasolino, F. Faralli, U. De Carlini: Testing Web applications, Proceedings of International Conference on Software Maintenance, IEEE Computer Society Press, 2002, s. 310–319.
- [3] Microsoft TechNet. Plan for performance and capacity (Office SharePoint Server). Internet WWW-sivu, URL: <http://technet.microsoft.com/en-us/library/cc262971.aspx> (31.5.2009).
- [4] B. English: Microsoft Office SharePoint Server 2007 Administrator's companion. Microsoft Press, Redmond, 2007.
- [5] DirAction. "Microsoft Office SharePoint Server 2007 product overview". Internet dokumentti, URL: <http://www.diraction.ch/shopdocs/files/osps.pdf> (31.5.2009).
- [6] D. Draheim et al.: Realistic Load Testing of Web Applications, Proceedings of the Conference on Software Maintenance and Reengineering, IEEE Computer Society, 2006, s. 60-70
- [7] Borland. SilkPerformer 2008 R2 Help. Internet dokumentti, URL:
http://techpubs.borland.com/silk_gauntlet/SilkPerformer/2008R2/EN/SilkPerformer.pdf
(31.5.2009).
- [8] D. A. Menascé: Load Testing of Web Sites. IEEE Internet Computing, 6(4):70-74, 2002
- [9] L. Powers ja M. Snell: Microsoft Visual Studio 2008 Unleashed. SAMS, Indianapolis, 2008.
- [10] Microsoft Volume Licensing. Product configuration. Internet WWW-sivu, URL:<http://www.microsoft.com/licensing/mla/product.aspx> (21.8.2009)
- [11] Microsoft. "Microsoft® Visual Studio® Team System 2008 Licensing". Internet dokumentti, URL:
<http://www.microsoft.com/downloads/details.aspx?FamilyId=CE194742-A6E8-4126-AA30-5C4E969AF2A3&displaylang=en> (21.8.2009)

- [12] E. Veenendaal: Standard glossary of terms used in Software Testing V 1.3, ISTQB, 2007
- [13] W.G.J. Halfond, S. Anand ja A. Orso: Precise interface identification to improve testing and analysis of web applications, Proceedings of the eighteenth international symposium on Software testing and analysis, ACM, 2009, s. 285-296
- [14] .NET Framework Developer Center, .NET Framework Conceptual Overview, Internet WWW-sivu, URL:[http://msdn.microsoft.com/fi-fi/netframework/default\(en-us\).aspx](http://msdn.microsoft.com/fi-fi/netframework/default(en-us).aspx) (21.8.2009)
- [15] Borland. Web Load Testing Tutorial. Internet dokumentti, URL: http://techpubs.borland.com/silk_gauntlet/SilkPerformer/2007/WebLoadTestingTutorial.pdf
- [16] MSDN Library, Visual Studio Team System, Internet WWW-sivu, URL: <http://msdn.microsoft.com/en-us/library/fda2bad5.aspx> (21.8.2009)

LIITTEET

Liite 1 Kuormitustestaussuunnitelma

Liite 2 Kuormitustestausvälineiden arviointitaulukko

LIITE 1 KUORMITUSTESTAUSUUNNITELMA

Testi MOSS – web-sovelluksen kuormitustestaussuunnitelma

Testitapaukset

Kuormitustestauksen tavoitteena on testata MOSS – web-sovellukselle yleisimpiä ominaisuuksia. Nämä piirteet on koottu käymällä läpi perus MOSS -asennuksessa mukana tulevia komponentteja ja ominaisuuksia. Alla on kuvattu testitapaukset tarkasti yksittäisinä toimintoina. Testitapaukset 1–11 ovat peruskäyttäjän toimintoja ja 12–20 ovat ylläpitäjän toimintoja.

	Testitapauksen nimi	Testitapauksen kuvaus vaiheittain
1.	Sivustolle saapuminen	1. Anna sivuston etusivun latautua kokonaan. (Tässä testissä ei varsinaisesti tehdä mitään.)
2.	Navigointi sivustolla	1. Valitse päänavigaatiosta ”Alasivusto”-linkki. 2. Anna uuden sivun latautua kokonaan. 3. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
3.	Raskaan sivun avaus	1. Valitse päänavigaatiosta ”Raskas sivusto”-linkki. 2. Odota, että kaikki suuret kuvatiedostot ovat latautuneet ja näkyvät kokonaan sivulla. 3. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
4.	Asiakirjakirjaston selaus	1. Valitse etusivulta ”Asiakirjat”-kirjaston linkki. 2. Mene Tiedosto.txt kohdalle, valitse ”Näytä ominaisuudet”-alavetovaihtimesta. 3. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
5.	Suuren tiedoston lataus	1. Valitse etusivulta Asiakirjat-kirjaston alta isokuva-linkki hiiren kakkosnäppäimellä ”Tallenna kohde”. 2. Valitse tallennuskohteeksi Työpöytä. 3. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
6.	Listan avaus	1. Valitse etusivulta Lista-linkki, joka vie Listasivulle. 2. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
7.	Listan näkymän vaihto	1. Valitse etusivulta Lista-linkki, joka vie Lista-

		<p>sivulle.</p> <ol style="list-style-type: none"> 2. Vaihda listan näkymä vasemman laidan pudotusvalikosta "Lisätiedot"-näkyväksi. 3. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
8.	Haun suorittaminen	<ol style="list-style-type: none"> 1. Sivuston etusivulla on "Etsi"-kenttä. 2. Kirjoita kenttään "kuormitus". 3. Suorita haku painamalla Enter-näppäintä. 4. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
9.	Hakutulosten järjestyksen vaihto	<ol style="list-style-type: none"> 1. Sivuston etusivulla on "Etsi"-kenttä. 2. Kirjoita kenttään "kuormitus". 3. Hakutulosten yläpuolelta voi valita lajitteluperusteen, valitse "Muokkauspäivämäärä". 4. Suorita haku painamalla Enter-näppäintä. 5. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
10.	Sisältönostollisen sivun avaus	<ol style="list-style-type: none"> 1. Valitse päänavigaatiosta "Sisältönosto"-linkki. 2. Anna uuden sivun latautua kokonaan. 3. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
11.	Uudelleenkäytettävää sisältöä sisältävän sivun avaus	<ol style="list-style-type: none"> 1. Valitse päänavigaatiosta "Uudelleenkäytettävää sisältöä"-linkki. 2. Anna uuden sivun latautua kokonaan. 3. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
12.	Kirjautumisen vaihtaminen	<ol style="list-style-type: none"> 1. Valitse etusivulta "Tervetuloa, Järjestelmätili"-alavetovalikosta "Kirjaudu järjestelmään toisena käyttäjänä" -linkki. 2. Anna uuden sivun latautua kokonaan.
13.	Sivuston luonti	<ol style="list-style-type: none"> 1. Valitse etusivulta "Sivuston toiminnot" – alavetovalikosta "Luo sivusto". 2. Syötä Otsikko "Testisivusto1" ja URL-nimi "Testisivusto1". 3. Valitse "Luo". 4. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
14.	Tekstin syöttö sivulle	<ol style="list-style-type: none"> 1. Valitse etusivulta "Sivuston toiminnot" – alavetovalikosta "Muokkaa sivua". 2. Valitse "Muokkaa sisältöä"-linkki. 3. Kirjoita "Sisältöä1" ja tee rivinvaihto painamalla "Enter". 4. Valitse sivun ylälaidasta "Kuittaa sisään vedoksen jakamista varten".
15.	Kuvan syöttö sivulle	<ol style="list-style-type: none"> 1. Valitse etusivulta "Sivuston toiminnot" – alavetovalikosta "Muokkaa sivua". 2. Valitse "Muokkaa kuvaa"-linkki. 3. Kirjoita "Valittu kuva"-kenttään kuvan sijainti. 4. Valitse "Ok". 5. Valitse sivun ylälaidasta "Kuittaa sisään".

		vedoksen jakamista varten”
16.	Web partin lisäys sivulle	<ol style="list-style-type: none"> 1. Valitse etusivulta ”Sivuston toiminnot” – alavetovalikosta ”Muokkaa sivua”. 2. Valitse Yläosa –Web osa -alueen alapuolella oleva ”Lisää Web-osa” -linkki” 3. Valitse ”Asiakirjat” ja paina ”Lisää”. 4. Valitse sivun ylälaidasta ”Kuittaa sisään vedoksen jakamista varten”
17.	Navigaation muokkaus	<ol style="list-style-type: none"> 1. Valitse etusivulta ”Sivuston toiminnot” – alavetovalikosta ”Sivuston asetukset” ja sen alta ”Muokkaa siirtymisnäkyä”. 2. Valitse ”Yleinen siirtymisnäky” alta Testisivusto1 ja paina ”Piilota” 3. Valitse ”OK”.
18.	Listaelementin luonti	<ol style="list-style-type: none"> 1. Valitse etusivulta Lista-linkki, joka vie Listasivulle. 2. Valitse ”Uusi”. 3. Syötä uudelle listaelementille Otsikko ”Elementti1”. 4. Valitse ”OK”. 5. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
19.	Asiakirjakirjaston luonti	<ol style="list-style-type: none"> 1. Valitse etusivulta ”Sivuston toiminnot” – alavetovalikosta ”Näytä sivuston kaikki sisältö” 2. Valitse ”Luo”. 3. Valitse ”Asiakirjakirjasto”. 4. Syötä Nimi ”Asiakirjakirjasto1”. 5. Valitse ”Luo”. 6. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.
20.	Tiedoston vienti asiakirjakirjastoon	<ol style="list-style-type: none"> 1. Valitse etusivulta ”Asiakirjat”-kirjaston linkki. 2. Valitse ”Lataa palvelimeen”. 3. Kirjoita Nimi-kenttään tiedoston sijainti: ”C:/tiedostot/Tiedosto1.txt” 4. Valitse ”OK”. 5. Palaa takaisin etusivulle valitsemalla pikalinkki sivun vasemmasta yläreunasta.

Profiilit

Lisäksi kuormitustestausta tehdään erilaisilla käyttäjä-, selain- ja yhteyskombinaatioilla. Lisäksi joitain yllä mainittuja testitapauksia parametrisoidaan siten, että jos esim. 10 eri käyttäjää vievät tiedoston asiakirjakirjastoon, kasvaa jokaisen vietävän tiedoston nimen perässä oleva numero yhdellä (tiedosto1.txt,

tiedosto2.txt, tiedosto3.txt...). Tämä edellyttää myös valmisteluja sen osalta, että hakemistossa on käytettävissä tarvittavat tiedostot.

Alla on lueteltu käytettävissä olevat resurssit.

Käyttäjät: peruskäyttäjä, ylläpitäjä

Selaimet: Internet Explorer 7.0, FireFox 3.0

Yhteydet: DSL, Mobiili

Alla on kuvattu eri kombinaatiot ja niiden lukumäärät. Esimerkiksi MOSS-ylläpito on periaatteessa tuettua vain Internet Explorer – selaimella.

Peruskäyttäjä, Internet Explorer, DSL	36
Peruskäyttäjä, Internet Explorer, Mobiili	9
Peruskäyttäjä, FireFox, DSL	36
Peruskäyttäjä, FireFox, Mobiili	9
Ylläpitäjä, Internet Explorer, DSL	9
Ylläpitäjä, Internet Explorer, Mobiili	1

LIITE 2 KUORMITUSTESTAUSVÄLINEIDEN ARVIOINTITAU LUKKO

Ominaisuus	Ominaisuuden kuvaus	Tärkeys MOSS:n liittyen	Perustelu MOSS:n liittyen	Arvio	Visual Studio Team System 2008 Test Edition	Pisteet	Arvio	SilkPerformer	Pisteet
Virtuaalikäyttäjäprofiilit	Käyttäjien toimintojen simulointi	4	MOSS:n kannalta on tärkeää voida luoda erilaisia käyttäjärooleja	4	Tietyle virtuaalikäyttäjälle voi hyvin rajata vain tiettyjä toimintoja	16	4	Tietyle virtuaalikäyttäjälle voi hyvin rajata vain tiettyjä toimintoja	16
Selainprofiilit	Käytettävissä olevien selainten valikoima ja uusien liitettävyy	2	MOSS:ia käytetään ensisijaisesti Internet Explorerilla	3	Yleisimmät selaimet tarjolla	6	4	Monipuolinen valikoima	8
Yhteysprofiilit	Käytettävissä olevien yhteysmäärittysten valikoima ja uusien liitettävyy	2	MOSS:ia käytetään yleensä nopeilla yhteyksillä	3	Muutamia ennalta määrättyjä eri nopeusluokista	6	5	Monipuoliset ja kustomoitavissa	10
Nauhoitustoiminto	Nauhoitus-toiminnon toimivuus ja ominaisuudet	5	Kuormitustestausten toiminnot yleensä nauhoitetaan ja MOSS:ssa toimitaan yleensä pelkästään käyttöliittymän kautta	4	Integroituu selaimen, selkeä ja hyvin toimiva. Ei lisätoimintoja.	20	3	Hieman epäselvä ja epäluotettavan oloinen ja hidas. Mahdollisuus tehdä eri toimintoja ajon aikana mm lisätä ajastimia	15
Skriptien muokattavuus	Skriptien koodauskieli	3	MOSS:n kuormitustestausta varten skriptejä ei ole välttämätöntä muokata kovin paljoa	5	.NET, tuttua MOSS-kehittäjille	15	2	BDL (Pascalin tyylinen), saattaa olla vierasta	6
Skriptien parametrisointi	Parametrisointi ja tietolähteet	5	Parametrisointi on tärkeää MOSS:ssa, koska erilaiset käyttäjien toimet ovat usein kertaluontoisia esim. tiedoston vienti kirjastoon, että sellaisia toimintoja ei voi tehdä useita peräkkäin ilman parametrisointia	4	Erlaisia lähteitä, mm. tietokanta	20	4	Monipuoliset keinot ja tiedostotietolähteet. Osittain automatisoitua	20
Skriptien validointi	Validointitoiminnon toimivuus	3	Ei suurta merkitystä MOSS:n suhteen, mutta hyvä olla olemassa	5	Monipuoliset validointitoiminnot	15	3	Perus validointitoiminnot	9
Skriptien toimivuuden tarkistus	Skripti "testataan"	4	Skriptien toimivuus on tärkeää testata MOSS-sivustoilla, koska usein tarvitaan mm. parametrisointia	4	Helppo ajaa tarkistus	16	5	Helppo ajaa tarkistus. Monipuoliset tiedot saatavilla ajosta.	20
Resurssien seurattavuus	Eri toimintojen seurattavuus ajon aikana ja jälkikäteen	4	MOSS vaatii melko paljon resursseja, joten niiden monitorointi on tärkeää.	4	Palvelinpään tarkka seuranta, kuormitustestin aikainen seuranta sujuvaa	16	2	Hieman epäselvää joidenkin toimintojen seuraaminen ajon aikana.	8
Kuorman mallinnustavat	Tarjottavat kuomamallit	4	MOSS:ssa on usein paljon erilaisia käyttäjä ja web-sovellusten käyttö yleensä vaihtelee esim. vuorokauden aikojen mukaan. Siksi kuormaa on tärkeää päästä mallintamaan eri tavoin.	3	Perusmallit saatavilla	12	5	Useita malleja tarjolla	20
Raportit	Raporttien sisältämät tiedot ja luettavuus	3	Ei suurta merkitystä MOSS:n suhteen, mutta täytyy olla olemassa	2	Selkeät, mutta melko suppeat	6	4	Erittäin laajat, mutta vaikeaselkoiset	12
Käyttöliittymä	Käyttöliittymän intuitiivisuus ja selkeys	3	Ei suurta merkitystä MOSS:n suhteen.	5	Selkeä ja siisti, tuttu MOSS-kehittäjille, jotka usein käyttävät Visual Studiota MOSS:n kehittämisessä.	15	3	Melko monimutkainen ja sekava, vanhahtava ulkoasu	9
Toimivuus	Käytön sujuvuus	3	Ei suurta merkitystä MOSS:n suhteen.	5	Toimi sujuvasti pienelläkin laitteistokokoonpanolla.	15	3	Ei kovin intuitiivinen.	9
Dokumentaatio	Käyttöohjeiden selkeys ja laajuus	2	Ei suurta merkitystä MOSS:n suhteen.	3	MSDN on valkea lukuinen (vain selaimella luettavissa)	6	4	Hyvä ja tarkka käyttöohje, mutta toistaa liikaa itseään.	8
Lisensointi	Lisensointivaihtoehtojen joustavuus	4	Pienissä ja keskiuurissa MOSS-kehitysprojekteissa joustava lisensointi on tärkeää.	2	Hankintamallissa sitouduttava tuotteeseen pitkäksi aikaa	8	4	Monipuolinen lisensointitarjonta, mm. vuokraus hyvä vaihtoehto	16
Hinnoittelu	Aiheutuvat kustannukset	3	Ei suurta merkitystä MOSS:n suhteen.	4	Kallis, mutta MOSS-kehitykseen liittyen usein hyviä alennuksia	12	1	Erittäin kallis	3
Laitteistovaatimukset	Tarvittava laitteistot kuormitustestausten suorittamiseen	3	Ei suurta merkitystä MOSS:n suhteen.	4	Vaatii tietokannan toimiakseen	12	2	Suosittelavaa käyttää työasemia virtuaalikäyttäjien simulointiin kontrollikoneen lisäksi. Vaatii tehoa kontrollikoneelta.	6
Skaalautuvuus	Laajuus, jolla kuormitustestausta voidaan suorittaa.	4	MOSS-toteutukset vaihtelevat pienistä suurin, siksi myös kuormitustestausvälineen skaalaututtava.	5	Mahdollista suorittaa kuormitus agentti-koneilta	20	5	Mahdollista laajentaa Agent Poolia	20
Yhteydet muihin järjestelmiin	Liitettävyy muihin ohjelmistokehityksen työkaluihin	4	MOSS-kehittäjän kannalta olisi hyvä jos kuormitustestausta voisi tehdä samoilta työkaluilta kuin kehitystä tai, että ainakin välitietoisivat yhteensopivia.	5	Toimii osana Visual Studiota ja siten helppo integraatio esim. versionhallintaan	20	2	Yhdistettävissä Borlandin tuotteisiin	8
				74		256	65		223