

90 ajatusta testitapauksista



Matti Vuori, Tampereen teknillinen yliopisto, Tietotekniikan laitos

Testitapaus on klassisin testien määrittelyn konstruktio ja niitä suunnitellaan ja suoritetaan joka hetki valtavia määriä. Jossain mielessä testitapaukset ovat joskus maailman tärkein asia ja toisaalta liiallinen keskittyminen niihin saattaa aiheuttaa monen muun aivan oleellisen asian ohittamisen. Testitapausten luonne on ehdottomasti miettimisen arvoinen. Luokaamme siis pieni katsaus niihin erilaisista näkökulmista.

Perusidea

1. Testitapauksessa etsitään jokin testattava asia, jonka toimintaa koetellaan jollakin täsmällisellä tavalla, usein "syötteellä". Syötteen antamisen jälkeen katsotaan, miten järjestelmä käyttäytyi ja oliko käyttäytyminen odotettua. Stereotyyppisesti vertaillaan todellista tulosta odotettuun.
2. Suoritus tehdään hallitussa tilanteessa ja siksi tapauksen määrittelyyn liittyvkin kohteen sopiva alustus ja koko ympäristön saattaminen otolliseen tilaan.
3. Samoin sen suorituksen jälkeen systeemi palautetaan usein johonkin tilaan – esim. siivotaan pois luodut tiedostot – jotta muille testitapauksille on sopivan "puhdas" alkutilanne.
4. Ja vaikka stereotyyppisesti katsellaan vain "tuloksia", käytännössä pitää katsella kaikkia kohteen reaktioita ja systeemin muutoksia ja havaintoja kokonaisuuden toiminnasta.
5. Jos testitapaus suoritetaan automaatiolla, havainnointi on vaikeampaa kuin silloin, jos testauksen tekee ihminen. Toisaalta automaatiikka näkee paremmin systeemin sisäisiä asioita, esim. tapahtumia tietokannassa.
6. Se tapa, jolla testitapaus suoritetaan, on käsitteellisesti usein erotettu "testiproseduuriksi", mutta käytännössä esitellään suoritustekniikkaan sidottuja askeleita, joihin kaikkiin voi liittää omia tarkistuksiaan.



7. Testitapauksen suoritustapa on siis "vapaasti" valittavissa. Ajatus on hieno, koska sama tapaus voidaan laittaa ihmisen, instrumentoidun automaation tai vaikka robotin suoritettavaksi täsmälleen samanlaisena.
8. Klassisesti ajatellaankin, että testitapauksia suoritetaan ensin manuaalisesti ja sitten, kun niiden toistotarve ilmaantuu, testaus voidaan automatisoida.
9. Tämä on järkevää erityisesti regressiotestausta varten, mutta samalla tietysti menetetään läsnä olevan ihmisen havainnointikyvyn käytön mahdollisuus.

Testitapauksella on monta roolia

10. Testitapaus on luonnollisesti testisuunnittelun arkifakta ja sellaisena osa testiarkkitehtuuria.
11. Se on testaajan ajattelun ja tahdon ilmentymä.
12. Se on testauksen työmäärän yksikkö ja kustannustekijä.

13. Se on testauslogistiikan seurattava elementti. Niitä ajetaan testauksen kohteelle virtana, erilaisina paketteina.

Standardin määritelmä

14. Uusi standardi ISO/IEC/IEEE 29119-1¹ määrittää testitapauksen näin: "set of test case preconditions, inputs (including actions, where applicable), and expected results, developed to drive the execution of a test item to meet test objectives, including correct implementation, error identification, checking quality, and other valued information".
15. ISTQB:n sanaston uusin versio² määrittää sen näin: "test case: A set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement. [After IEEE 610]."
16. ISTQB:n sanasto erottelee lisäksi kaksi testitapauksen abstraktiotasoa: a) high level test case: "A test case without concrete (implementation level) values for input data and expected results. Logical operators are used; instances of the actual values are not yet defined and/or available. See also low level test case." Synonyymejä: abstract test case, logical test case. b) low level test case: "A test case with concrete (implementation level) values for input data and expected results. Logical operators from high level test cases are replaced by actual values that correspond to the objectives of the logical operators." Synonyymi tälle on "concrete test case".

Historia

17. En tiedä, kuka keksi käsitteen, mutta sitä käyttivät ainakin jo Herbert Leeds ja Gerald Weinberg vuonna 1961 kirjassaan "Computer programming fundamentals".
18. He eivät määritelleet testitapaukselle rakennetta tai muutakaan formaalia – vain sen mihin niillä pyritään ja testataan (normaalitilanteet, äärirajat, poikkeukset).

¹ ISO/IEC/IEEE 29119-1 Software and systems engineering -- Software testing -- Part 1: Concepts and definitions

² Standard Glossary of Terms used in Software Testing. Version 2.4. International Software Testing Qualifications Board ISTQB. 50 p.

19. "Määrittelyjen" aikakausi alkoi myöhemmin, kun pyrittiin tiukasti standardoimaan testitapaustenkin rakenne – ja yhdenmukaistamaan kaikki muukin testauksen määrittely ja dokumentointi.
20. Tämä työ tuotti alan gurujen kirjoissaan esittämiä kuvauksia ja myöhemmin IEEE 829:n ensimmäisen version³ määrittelyt testitapaukselle ja testiprocedureille.
21. Standardoitu rakenne auttaa testitapausten kuvauksen harmonisointia organisaatioissa, niiden suunnittelun systematisointia, suunnittelun mahdollistamista kaikenlaisella työvoimalla, testitapausten auditoitavuutta ja potentiaalia niiden suorittamisen automatisointiin.
22. Koska testitapaus on ollut toiminnallisen testauksen ydintä ja toiminnallinen testaus on ollut dominoiva testityyppi, on selvää, että testitapauksilla on ollut erittäin vahva rooli testauskulttuurissa.
23. Kautta vuosikymmenten on tehty paksuja kirjoja tekniikoista, joilla testitapauksia suunnitellaan.
24. Testauksen opetuksessa ne ovat usein kurssien ensimmäisiä asioita.

Vahvoja merkityksiä

25. Testitapauksella on vahvoja merkityksiä ohjelmistokehityksen ja teollisuuden kulttuurissa.
26. Heti kun sana edes mainitaan, syntyy assosiaatio, että nyt otetaan laatu vakavasti ja on jo pohdittu tarkasti, miten systeemiä aiotaan koetella. Testitapaus on aivan eri asia kuin abstrakti "testaus".
27. Termiä käyttävä henkilö todistaa sillä oman ammattilaisuutensa.
28. Samoin termiin liittyy magiaa. Testitapauksen olemassaolo varmistaa laatua symbolisesti. Se on kuin Mustanaamion hyvä merkki ohjelmiston leuassa.
29. Ja koska se on hyvä merkki sinänsä ja hyvään liittyy kulttuurissamme se, että enemmän on parempi kuin vähemmän, mitä enemmän testitapauksia on, sitä paremmin asioiden koetaan olevan.

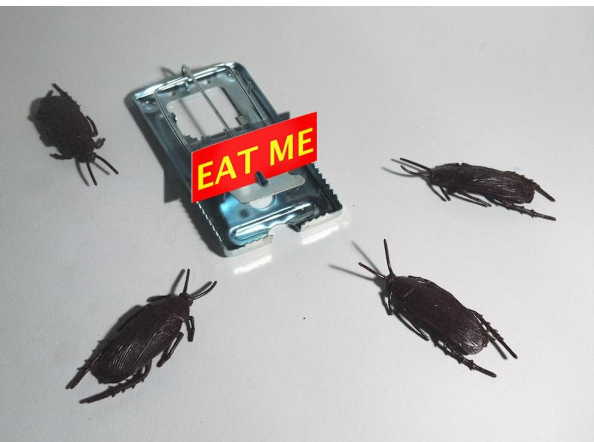
³ IEEE 829-1983 - IEEE Standard for Software Test Documentation



30. Testitapauksella on siis kulttuurinen rooli monella tasolla: se on osa retoriikkaa, osa uskomusjärjestelmää ja osa arkista toimintaa.
31. Jos laatukulttuuri ei organisaatiossa ole vahvaa, voi puheella hämätä itseään ja toisia.
32. Mutta puhe oikeilla termeillä tukee laatukulttuurin kehittymistä.

Yksin ja joukossa

33. Testitapauksella on yksilöllinen luonne. Jokainen niistä tekee aivan tiettyä asiaa, jolla on jokin selkeä tarkoitus.
34. Mutta samaan aikaan ne ovat kuin parvi, jonka voima tulee siitä, että pieniä asioita on kenties valtava määrä.
35. Testitapaukset ovat kuin ansoja, joihin odotetaan bugien astuvan.



36. Tämä on erilainen taktiikka kuin tutkivassa testauksessa, jossa testaaaja on kuin villipetoa aktiivisesti jäljittävävä metsästäjä.
37. Sellainenkin metsästäjä voi jättää regressioloukun sopivaan paikkaan, jos peto vaikka paikallisi aiempaan ruokapaikkaansa...
38. Ja tietenkin, jos metsää ansoitetaan, kannattaa ensin tehdä ansoja isoimmille saaliseläimille – pienempiä pyydetään sitten myöhemmin.

Kaikella on hintansa

39. Jokaisen testitapauksen suunnittelu, toteutus, ylläpito ja suorittaminen maksaa.
40. Siksi kannattaa laatia vain sellaisia testitapauksia, jotka tuottavat relevanttia tietoa tai muuta hyötyä.
41. Joskus se kustannus tulee ajasta, sillä aika on rahaa. Testitapausten suorittaminen ja palautteen saaminen on tärkeää olla nopeaa, että asiat etenevät.

Käytöstä mittarina

42. Testitapausten läpäisyn tai läpäisemättömyyden on arveltu olevan hyvä laadun mittari.
43. Tässä on perää silloin, kun jokainen testitapaus uniikisti testaa jotain relevanttia asiaa relevantilla tavalla, mikä ei ole missään nimessä varmaa.
44. Yksikkötestauksen tasolla testitapauksilla on historiallisesti enemmän relevanssia kuin ylempillä testaustasoilla, koska matalalla tasolla ei oikeasti vielä "tiedetä", millaisen todellisen virheen läpäisemätön testitapaus aiheuttaisi käyttäjän tai prosessien toiminnassa.
45. Mutta samalla tämä koettu merkitys on tuottanut patologioita, joissa testitapausten määrää on pidetty testauksen laadun mittarina, mitä on nykyisin opittu kritisoimaan.

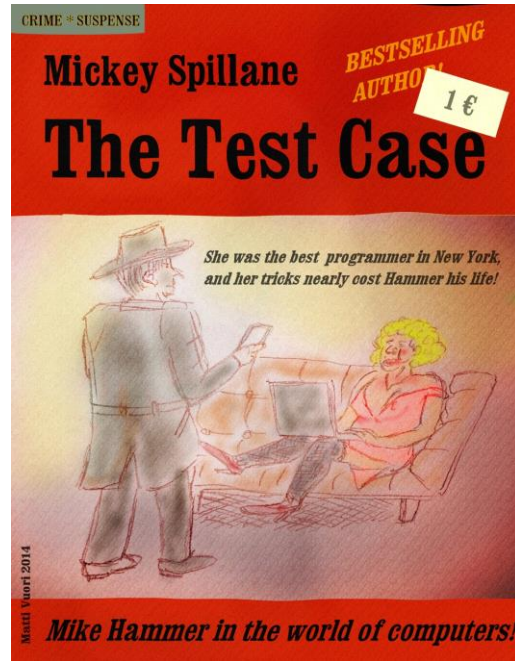
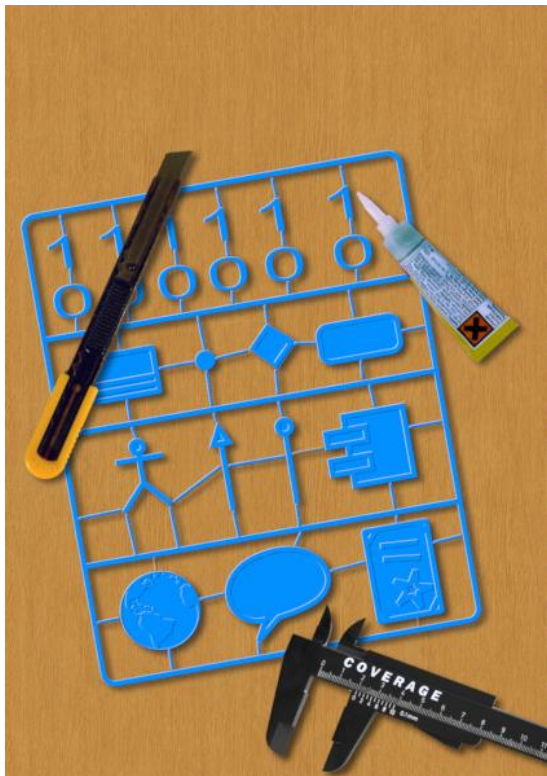
Luonteen kritiikkiä

46. Testitapaukset ovat kuin maalilaukaukset jääkiekossa ja niiden harjoittelu. Ollaan ikään kuin kontekstivapaasti sopivassa paikassa maalin edessä ja pyritään laukaisemaan kiekko maalivahdin ohi valitsemalla se "syöte" (eli mihin ammutaan) sopivasti. Mutta maailma ei aina ole sellainen, eikä maali ole niin selkeä. Ja jääkiekostakin on 1960-luvun jälkeen ymmärretty paljon muuta kuin tuo elementti.

47. Niinpä nykyään korostetaankin joukkuepeliä, pelitapaa jne. keskeisempänä. Niiden elementtien avulla päästään useimmin siihen tilanteeseen, että kiekko saadaan maaliin lähes luonnostaan. Kun luodaan hyvä pelitilanne (skenaario, käytötapaus, tarina – testiympäristössä) ja tarkka silmä avoimille paikoille (ymmärrys siitä mitä testataan, test conditions), oikea laukaus ("ratkaiseva syöte") löytyy lähes suunnittelematta, ikään kuin edellä kehitellyn väistämättömänä jatkumona. Kyse ei silloin ole laukaisutaidosta, vaan osaamisesta olla oikeaan aikaan, oikeassa valmiustilassa maalin edessä paikassa, johon kiekko tulee.

Maailmankuvan ikoni

48. Testitapausten maailmankuva on analyttinen. Siinä kaikki voidaan jakaa ja palastella aina vain pienempiin kohteisiin ja toiminta kohdistaa löydettyyn jakamattomaan asiaan. Tehdään systeemin jäsennyttä, jonka perusteella sen piirteet jaetaan. Niistä tunnistetaan kohteita, joita voisi testata ja relevantteja testattavia asioita. Funktionaalisille kohteille mietitään sopivia syötteitä ja tapoja suorittaa se syötteen syöttäminen. Alimmalla tasolla tehdään kaikki mahdollinen ja sitten luodaan katsaus ylempään kokonaisuuden tasoon.
49. Joskus tämä maailmankuva on niin vahva, että



ajatellaan, että mitään muuta ei tarvita laadun näkemiseen kuin testitapausten tarjoamia fragmentteja toiminnasta. Se on sama kuin kehittäisi tiimiä vain kehittämällä yksilöitä, kiinnittämättä huomiota tiimin toimintaan tiiminä.

50. Testitapausten ajattelutapa on myös idealistinen. Niissä näkyy teknologia puhtaana ja ilman ulkoisia vaikutuksia. Tapahtuu ajattelun puhdistumista mennessä alimmille testaus-tasolle, joissa vain puhdas suorituslogiikka on tarkastelun kohteena.
51. Puhtauteen liittyy myös se, että periaatteessa testiproseduuri on erotettu testitapauksesta.
52. Systeeminen ajattelu tulee toki esille testauksen kokonaisuudessa, jossa ylempillä testaus-tasolla otetaan koko järjestelmän toiminta huomioon ja realismia tuodaan mukaan joka kerroksella huipentuen hyväksymistestauksen ja järjestelmän validoinnin näkökulmaan, jossa olemassa oleva todellisuus on systeemin toiminnan referenssi.
53. Maailmankuvaan liittyy myös deterministisyyden ajatus. Kaikki on päätettyä ja harkittua. Sattumalla ei ole sijaa ja se pyritään karsimaan pois testauksesta.

Estetiikasta

54. Edellä mainitut puhtaus ja ideaalisuus ovat selkeitä testitapausten esteettisiä periaatteita.

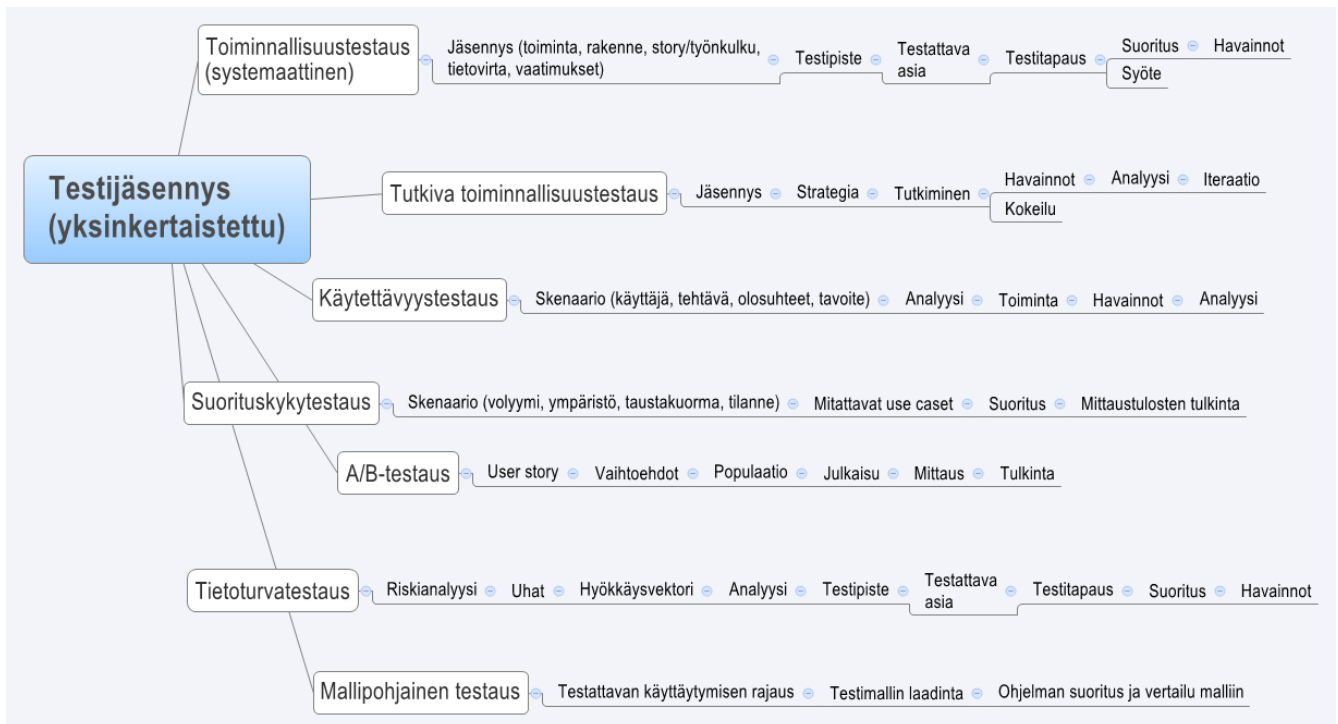
55. Yksinkertaisuus on myös tärkeää. Yksinkertainen testitapaus, jolla on selkeä tarkoitus on kuin hyvä työkalu, hyvä kirves tai puukko.
56. Toisaalta testitapauksiin liittyy myös rumuutta. Kun testitapauksessa ollaan erityisen kovakouraisia – esimerkiksi ”vedetään töpseli irti”, se ei ole kaunista, vaan melkein hirvittää testaajaakin. Toisaalta silloin on jo mukana väistämättömyyden ja kohtalon kauneutta.
57. Näistä esteettisistä periaatteista on paljon todellista hyötyä, sillä ne luovat testitapauksille tavoitteellisuutta, tehokkuutta, ylläpidettävyyttä ja auditoitavuutta.

The case of the case

58. Termissä on kaksi osaa: testi ja tapaus. Tapaus on maailmassamme monessa käytössä: oikeustapaus, sairaustapaus, tapaustutkimus, käyttötapaus.
59. ”Tapaus” on yhdentyyppisen asian yksilöllinen ilmentymä, joka muussa käytössä pakatoi kaikesta ilmentymään liittyvän: tilanteen, ongelman, tiedot, kohteen, ratkaisun, mukana olevat toimijat, historian.
60. Se on kuin pienoismaailma, jonka puitteissa asia otetaan esille, käsitellään ja ratkaistaan.
61. Yhden tapauksen puitteissa ei käsitellä muita tapauksia.
62. Käsitteen avulla saadaan siis asia hienosti hallittavaksi.
63. Toisaalta se antaa asioista liian hyvin pake-toidun kuvan: jo yksikin testitapaus luo assosiaation, että sen ratkaisu on jotain oleellista. Tämä näkyy mm. tilanteissa, joissa vaaditaan, että jokaista vaatimusta varten pitäisi olla ainakin yksi testitapaus. Yksikin niistä ”riittää”!
64. Voisiko miettiä muita käsitteitä?
65. Tilastollisesta ja diagnostisesta näkökulmasta jokainen testitapaus on vain näyte. Siis testi-näyte, laatu-näyte, näyte käyttäytymisestä.
66. Se käsite antaa ymmärtää, että yksi näyte on vain yksi monista, vaihtoehtoinen muille, eikä yksinään kerro vielä mitään. Sen avulla ei voida luoda minkäänlaista varmuutta systeemin toiminnasta tai laadusta.
67. Näytteenotto on kuitenkin ”observoivaa” – siinä vain tarkkaillaan asioita, eikä pyritä vaikuttamaan kohteeseen. Puuttuu hyvän testitapausten ”tappamisen meininki”...
68. Entä testitarina? Elämmehän aikaa, jolloin suositaan tarinoita niiden kertovuuden ja rikkauden ja vuoksi. Tarina olisi kuitenkin selvästi jotain testitapausta suurempaa – pieni skenaario jossain testipisteessä. Mutta sehän kuulostaisi hyvin mielekkäältä.
69. Vähän tätä lähellä ovat skenaariot ja käyttötapausten testaus, mutta ne ovat toiminnan maailmasta lähteviä termejä.

Fokusmekanismeja eri testityypeissä

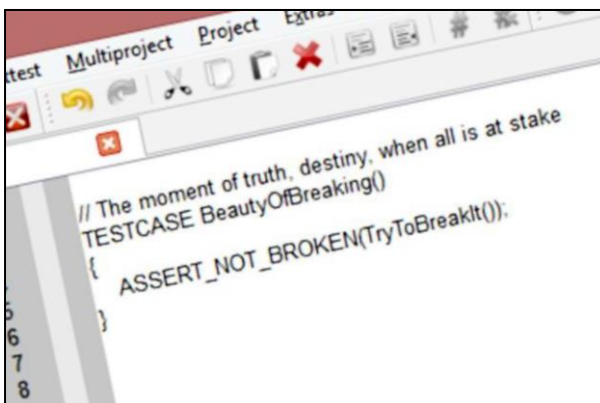
70. Testitapaukset ovat tyypillisiä systemaattisessa toiminnallisuustestauksessa. Siinä testaus huipentuu siihen, että ihminen suunnitelmalla löytää syötteen, jonka sitten syöttää testauksen kohteeseen ja tarkistaa mitä tapahtui.
71. Siihen päästään luomalla jäsenitys jostakin näkökulmasta (mm. toiminta, rakenne, story/työnkulku, tietovirta, vaatimukset) ja tunnistamalla niiden kautta sopiva ”testipiste”, josta päästään tarttumaan kohteeseen – vaikkapa tiedon syöttö lomakkeelle. Siitä on helppo tunnistaa testattavia asioita ja suunnitella testitapauksia ja niiden syötteitä.
72. Kohdetta havainnoidaan ennen ja jälkeen ja havaintojen ”erotus” kertoo siitä, mitä on tapahtunut. Erityisesti pyritään usein keksimään syötteitä vastaava ”odotettu tulos”, jonka vertailu todelliseen on hyvä mittari testin onnistumiselle.
73. Tutkivassa toiminnallisuustestauksessa tehdään samanlaista jäsenystä, mutta testausta ohjaa suoraviivaisten testipisteiden sijaan jokin strategia kohteen käyttäytymisen selvittämiseen. Sen puitteissa tehdään dynaamisesti ohjattua kokeilua, saadaan siitä havaintoja ja iteroidaan, kunnes käyttäytyminen on kylliksi ymmärretty.
74. Kokeilu on lähellä testitapausten ”just-in-time”-suunnittelua, mutta siitä puuttuu mentaalista painolasti odotettujen havaintojen ja tilanteen kontrollin suhteen.



75. Suorituskykytestaus ja kuormitustestaus ovat luonteeltaan tästä erilaisia. Testauksen fokuksena ei ole mikrotapahtuma, vaan laajempi skenaario – käytön volyyymi, ympäristö, taustakuorma, tilanne kokonaisuutena. Siihen kontekstiin suunnitellaan mielenkiinnon (tiedon tarpeen) kohteena olevia käyttötappauksia (tms.), jotka suoritetaan hallitusti ja samalla tehdään mittauksia. Mittaustuloksia ei hyväksytä suoraan, vaan ne vaativat analyysiä ja tulkintaa.
76. Mallipohjainen testaus lähtee testattavan käyttäytymisen rajauksesta ja testimallin laadinnasta. Usein tässä testauksessa ei ole testitappauksia, vaan ohjelmaa suoritetaan kaikki relevantteja suorituspolkuja pitkin ja suoritusta vertailaan mallin antamaan ”oikean” käyttäytymisen

referenssiin.

77. Mallipohjainen testaus on yksi esimerkki testauksesta, jossa myös sattumalla on sijaa – sekä yhden systeemin toiminnassa että systeemien vuorovaikutuksessa.
78. Toinen tapa tuoda testaukseen satunnaisuutta on ”monkey”-testaus, jossa ohjelmaa käytetään käyttöliittymästä täysin satunnaisella tavalla. Siihen maailmaan ei testitappauksen käsite istu luontevasti.
79. Fuzz-testaus soveltaa myös usein satunnaisuutta testidatan muuntelussa. Onko jokainen muunnos testitappaus? Ei ole aivan mielekkäintä ajatella niin. Muuntelutaktiikka on tärkein alimman tason määrittely.
80. Tietoturvatestaus alkaa riskianalyysillä ja sen osana uhkamallinnuksella. Konkreettista testausta tukee hyökkäysvektorien tunnistaminen. Niiden avulla voidaan tunnistaa testipisteitä ja tehdä testausta toiminnallisuustestauksen logiikalla.
81. Tietenkin tietoturvatestauksen osana – tai vahvasti synergisenä parina – on esimerkiksi käytettävyydestestaus, koska käyttäjän näkökulma ja virheet ovat kriittinen alue tietoturvassa.
82. Käytettävyydestestauksessa lähdetään yleensä laajemmilta vesiltä. Testattavana on ihmisen-tekniikka-järjestelmän toiminnan jokin tavoitteellinen skenaario, jonka annetaan kul-



kea ihmisen tahdon mukaisesti. Tärkein läsnä oleva ihminen on koehenkilö, joka tekee tehtäväänsä sen tavoitteisiin pyrkien yrittämättä tehdä mitään erityisen testaavaa. Testisuunnittelija miettii etukäteen skenaariot ja luo olosuhteet sellaisiksi, että mahdollisiin ongelmiin tullaan tekemisen kuluessa törmäämään. Skenaarioita suoritetaan suhteellisen kontrolloiduissa oloissa, tehdään havaintoja koehenkilön ”pärjäämisestä” ja niitä analysoidaan.

83. A/B-testauksen lähtökohtana on jokin käyttäjätarina, jonka puitteissa tutkitaan eri toteutusvaihtoehtojen paremmuutta. Keskeistä on siis muutaman vaihtoehtojen suunnittelu ja mielekkään testiasetelman laatiminen – missä tärkeää on miettiminen, millaisilla mittauksilla paremmuutta mitataan. Eri vaihtoehtojen tarjoaminen eri populaatioille ohjelmiston julkaisulla on testin valmistelun ydintä. Data hankitaan mittaamalla versioiden käyttöä ja sitä analysoidaan testin aikana ja jälkeen.
84. Näissä kaikissa näkyy kuitenkin samanlainen jatkumo: testattavan tilanteen skouppaus, miettiminen mitä arvoa systeemiltä odotetaan, fokuoitouminen keskeiseen, olosuhteiden luominen, jonkinlainen tilanteen toteuttaminen ja havaintojen tekeminen sekä havaintojen arviointi.

Johtopäätöksiä

85. Testitapaus on erinomainen käsitteellinen ja käytännöllinen väline varsinkin ”tekniseen” testaukseen, jossa kohdetta ”pommitetaan” erilaisilla syötteiden variaatioilla.
86. Sillä on myös keskeinen asema kulttuurisena konstruktiona, mihin liittyy riskejä organisaatioissa, joissa laatukulttuuri ei ole kypsää.
87. Liiallinen asioiden näkeminen testitapauksina on vaarallista – erilaiset asiat tarvitsevat erilaisia käsitteitä.
88. Testauksen maailmassa on paljon testityyppejä, joissa testitapauksen käsite ei ole mielekäs.
89. Ja aivan sama haaste on muussakin testauksen kehittämisessä... toiminnallisuustestauksen kulttuurihistoria ei riitä enää toiminnan ohjenuoraksi.
90. Kannattaa miettiä ja kehitellä käsitteistöä edelleen, jotta se muuttuvassa maailmassa pysyy mahdollisimman ilmaisuvoimaisen.

Matti Vuori on toiminut laatuun ja toiminnan kehittämiseen liittyvien asioiden parissa jo neljällä vuosikymmenellä tutkijana, konsulttina, kouluttajana ja opettajana ja ehtinyt nähdä ja miettiä monia erilaisia vastakkainasettelun lajeja. Nykyisin hän on töissä Tampereen teknillisellä yliopistolla.