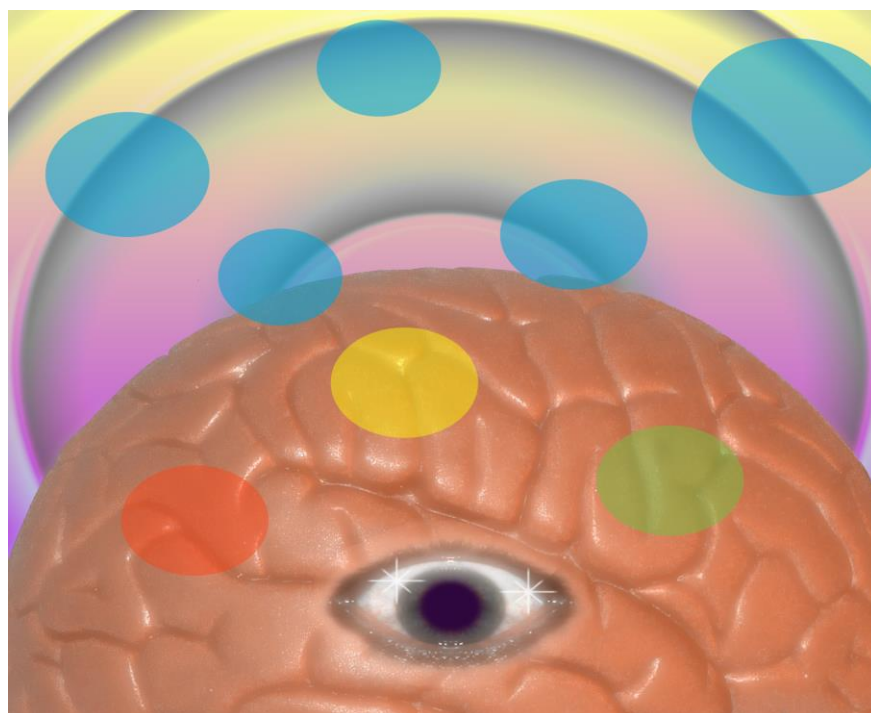


Laatu ja testaus

Arvioinnista, tutkijoista ja
testaajista, ja datasta
testausta tehostamaan

2/2019



TestausOSY
Finnish Association of Software Testing

Julkaisija Testauksen osaamisyhteisö
TestausOSY

Julkaisupaikka Tampere

Lehden arkisto:

<https://www.sytyke.org/testausosy.fi/magazine>

Lehti on laadittu vapaaehtoisvoimin.

Artikkelien oikeudet kuuluvat kirjoittajille.

Tekninen taittotyö: Matti Vuori

Kansikuva: Matti Vuori

TestausOSY on Sytykkeen osaamisyhteisö,
mutta avoin kaikille



Sisällysluettelo

3. Lukijalle. *Matti Vuori*
4. Muutama ajatus asioiden arvioinnista.
Matti Vuori
12. Mitä yhteistä on tutkijoilla ja testaajilla?
Tuula Pääkkönen
15. Dataa testausta tehostamaan – vai
astutaanko kehittäjien varpaille? *Ville
Laitila*
18. Muistojen tiellä... ensimmäinen
kohtaamiseni testiautomaaton kanssa.
Matti Vuori

Lukijalle



Matti Vuori

Hyvää syksyä!

Kädessäsi on taas uusi lehden numero. Tällä kertaa ekassa katsellaan monelta kantilta asioiden arviointia, mikä on jatkuvasti tärkeämpää, koska asioiden kriittisyys ja laatuvaatimukset kasvavat jatkuvasti – ja testauspainotteisessa kulttuurissa on syytä ymmärtää kaikenlaisia asioiden arvioinnin menettelyjä.

Toinen juttu reflektoi testaajan olemusta ja työtä vertailemalla sitä tutkijan vastaaviin. Kummallakin ryhmällä on tietysti tärkeä roolinta laatuun liittyvässä ihmisten verkostossa ja osaamisten kudoksessa. Mutta ihmiset ja työt ovat erilaisia ja juuri siitä kumpuaa ammatillisen maailmamme rikkaus ja toimivuus.

Kivaa on se, että useimmiten jokainen numero on saanut mukaan uudenkin kirjoittajan. Ville Laitila kirjoitti tähän numeroon mielenkiintoisen tekstin softan kehittämiseen ja versionhallintaan ja muuttuvien asioiden testaukseen liittyen.

Miellyttäviä ja ideoita antavia lukuhetkiä!

Suomessa 1.10.2019

Muutama ajatus asioiden arvioinnista

Matti Vuori

Kaikki toiminta perustuu arviointiin. Kyllä, sillä maailma on täynnä ideoita ja impulsseja. Toiminta lähtee siitä, kun niitä puntaroidaan ja tehdään valintoja. Tuotekehityksestäkin sanotaan, että ideat eivät ratkaise, vaan niiden toteutus. Aiheen käsittely yleisellä tasolla on tärkeää siksi, että eri tilanteissa tarvitaan erilaisia toimintamalleja. Parhaita käytäntöjä ei ole. Niinpä on osattava leipoa uniikkiin toimintatapaan juuri sille parhaat arvioinnin tavat. Siihen tarvitaan arvioinnin syvällistä yleistä ymmärrystä.

Monenlaista arviointia

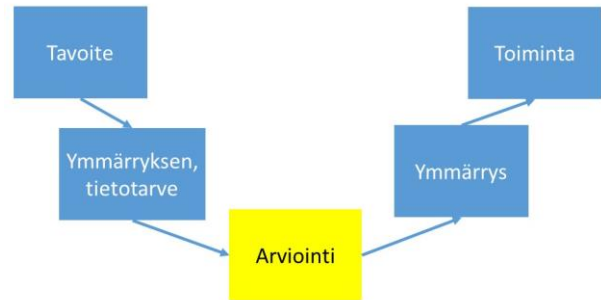
Matkalla ideasta lopputulemaan tarvitaan monenlaista tietoista tai tiedostamatonta arviointia, esim.:

- Olisiko tämä idea sopiva meille?
- Vastaako idea asiakkaiden tarpeita?
- Mitä vaikutuksia sillä on?
- Onko aikaansaannoksemme laatu hyvää?
- Mitä pitäisi parantaa?
- Täytämmekö pakolliset vaatimukset?
- Missä on riskejä? Miten suuria?

Ja organisaation elämässä on samoin monenlaisia kysymyksiä:

- Miten fiksu on tapamme toimia?
- Täytämmekö laatu- tai turvallisuusstandardin vaatimukset?
- Miten selviämme tästä kriisistä?

Arviointi tukee aina jotain tavoitetta ja sen luonne riippuu arvioinnin kontekstista. Käsiyöläinen arvioi työnsä jälkeä ja työn kohteen muotoutumista jatkuvana taitopohjaisena prosessina. Riskianalyysin tekijällä taas on toisenlainen, tietopohjainen näkökulma, jossa systemaattisesti käydään läpi kohdetta eri näkökulmista formaalina prosessina. Kriittinen ajattelu on aina lähtökohta arvioinnille ja jo sen piilevä muoto.



Arvioinnin avulla eteenpäin.

Erilaisia lopputulemia

Arvioinnin arvo isoin arvo on sen lopputuloksessa. Projektitoiminnassa ja arvioivassa konsultoinnissa syntyy usein halutut tiedot sisältävä määrämuktoinen ja -kielinen raportti. Joskus asioista annetaan virallinen lausunto tai tehdään epävirallinen lyhyempi arvio. Joskus vain asioita kommentoidaan. Viimeisessä tapauksessa ei taustalla yleensä on systemaattista arviointia.

Taiteiden alueilla kirjoitetaan usein "kritiikki", joka on vapaamuotoinen ja saattaa olla itsessään lähes kaunokirjallinen teos. Sellainen tukee diversin ajattelun esilletuloa. Kritiikkejä pitäisikin kirjoittaa enemmän muillakin toiminnan alueilla.

Lopputulema ei toki ole vain artefakta, vaan siihen liittyy asian esittely – arvioinnin kokonaisprosessista riippuen – jolla voi olla suurempi merkitys kuin dokumenteilla. Arvioinnin tuloksena syntyy aina jotain ajatuksia, jotka ehdotetaan pantavaksi toimeen tai annetaan ainakin ymmärtää niin. Siihen ei riitä pelkkä neutraali raportointi, vaan lopputuloksen pitää "myydä" ajatukset ja herättää tai jopa muuttaa ihmisten tunteita. Presentaatio ja keskustelu antavat siihen aina hyvän mahdollisuuden. Arvioinnin kokonaisprosessin viimeinen vaihe onkin se, kun saadaan yhteinen signaali seuraavista stepeistä tai muu yhteinen kokemus uudesta ymmärryksestä.

Arvioinnin sisältö on ymmärrettävä oikein

Joskus näkee kerrottavan, että esim. organisaation tietoturva-auditointi on kyllä tehty ja siksi onkin yllätys, että kräkkerit iskivät. Silloin on ymmärretty arvioinnin skooppi väärin. Laatujärjestelmänkin auditointi arvioi vain laatutoimintaa, mutta se ei vielä kerro yhtään mitään siitä, onko jonkin nimenomaisen tuotteen testaus ollut pätevää.

Tämä on arvioijan ja tilaajan yhteine haaste: pitää viestiä ja ymmärtää, mitä ollaan arvioimassa ja mitä ei, ja mistä siis arvioinnin tulo kertoo.

Monikriteeristä rikkautta

Valitun skoopin sisällä hyvä arviointi on monikriteeristä ja tarkastelee kaikkia relevantteja tekijöitä valitussa skoopissa ja valitulla abstraktiotasolla. Niinpä käyttöliittymän arviointiin ei riitä käytettävyyden arviointi eikä projektisuunnitelman arviointiin sen toteutettavuuden ja riskien arviointi. Monikriteerisyys paljastaa arvioijan ymmärryksen ja tarjoaa arvioinnin hyödyntäjille rikkaan kokemuksen sekä oppimista. Kun toiminta perustuu aina suurimmalta osalta historiaan, voi arviointi olla yksi katalyytti uusien asioiden tuomiseen mukaan ajatteluun. Tietoturvasuuskun oli alkuun uusi juttu, jota ei ole helpolla saatu projektien agendaan.

Tietenkin monikriteerinen arviointi voidaan palastella, jolloin saadaan kriittisiin asioihin fokusoitu ote. Mutta vaikka käyttöliittymästä arvioitaisiinkin käytettävyyttä, saavutettavuutta ja toiminnallista laatua, pitää katsoa kokonaisuutta ja huolehtia, ettei kokonaisuuden arviointiin jää reikiä.

Arviointi on oppimistilaisuus

Samalla, kun arviointi tukee primääriä tavoitetaan, sillä on muita etuja. Arviointi on tilaisuus oppia tekemisen kohteesta tai tekemisestä syvästi. Kun vaikka tietoteknisen systeemin arkkitehtuuria arvioidaan pienessä ryhmässä, siinä käydään yleensä läpi systeemin toimintaa ja erilaisia sen kohtaamia skenaarioita koko systeemin elinkaaren aikana. Luodaan siis jo kytkentää tulevaisuuteen.

Arviointi on sateenvarjo monenlaisille aktiviteeteille

Arviointi on käsitteenä oleellinen. Esim. testauskulttuurissa puhutaan paljon vain testauksesta, vaikka haasteena on tuotteen arviointi, missä testauksella on oma roolinsa, mutta vain oma roolinsa. Tarvitaan muutakin: analyysiä, tarkastamista, tekijöiden subjektiivisen näkemyksen selvittämistä, mittausta, jne. Sama pätee työelämän kokeilutoimintaan. Kokeilu on vain yksi tapa viedä asioita eteenpäin. Toiminnallisena se on tärkeä murros monen organisaation byrokraattiseen kulttuuriin, mutta kehityksen soisi johtavan rikkaampaan ja syvällisempään asioiden arviointiin - ja kokeilunkin tekemiseen laadukkaasti.

Arvioinnin sisäanatomiaa

Arvioinnissa arvioija tarkastelee kohdetta tai sen representaatiota. Representaatio korvaa kohteen jollain esityksellä, mallilla, kuvauksella, suunnitelmalla jne... ja samalla siihen liittyy tietoa kohteen toiminnasta. "Korvikkeita" on monenlaisia. Toiminta kuvautuu kenties vain tiettyjen skenaarioiden kautta.

Taulukko: Eri asioiden korvaavia esityksiä.

Asia	Esitys, korvike
Tavoitteen määrittely	Informaatio
Tuote, järjestelmä, kohde	Representaatio (malli, kuvaus)
Ympäristö	Representaatio (malli, kuvaus)
Käyttäjä	Syntetisaatio (persoonat, stereotyyppit, roolikuvaukset)
Toiminta	Skenaario, simulaatio

Tarkastelu on välineellistä ja erityisiä välineitä ovat arviointitekniikat, tarkistuslistat, testausmenetelmät, metriikat jne... Kaikki tämä tapahtuu organisaation ja kontekstin lausuttujen ja lausumattomien sääntöjen ja oletusten puitteissa. Ne voivat määrittää "sallitut" arviointitavat, odotetun yhteistyön prosessissa, asiat joita tarkastellaan - ml. organisaation tabut, joista ei saa tai osata

puhua. Yleensä on niin, että mitä ammattimaisempaa toiminta on, sitä enemmän arvostetaan systemaattista asioiden arviointia.

Arviointi on exploroivaa

Arviointi voi joskus kulkea mekaanisesti A:sta Ö:hön, mutta silloinkin pitää kussakin luokassa löytää todisteita ("evidenssiä") asioiden tilasta tarttumalla pieniin johtolankoihin ja antamalla niiden kuljettaa tarkastelua eteenpäin. Tilanne on saman tyyppinen organisaatiokulttuuria arvioiden ja tehdessä tietokoneohjelman tutkivaa testausta. Joskus väitetään, että se havaintojen teko juontuu kokemuksesta, mutta pätevässä, luotettavassa arvioinnissa se perustuu metodiseen lähestymistapaan: onko tilanteessa jotain erityistä, ristiriitoja, puuttuuko näyttöä, esiintyykö häiriöitä, onko jotain keskittymiä... Eri tilanteisiin on havaintojen tueksi omia heuristisia periaatteitaan.

Tarkistuslistojen etuja

Tarkistuslistat ovat oleellisia, sillä ihmisen mieli on hatara ja kaikki olennainen ei aina pysy mielessä. Tarkistuslistan käyttö auttaa siinä. Ryhmytyksessä toisaalta tarkistuslista auttaa orientoitumaan siihen, minkä tyyppisiä asioita ollaan etsimässä. Haasteena on sitten se, ettei rajoituta vain listalla oleviin. Siihen auttaa tarkastelun näkökulman ja tekniikan vaihtaminen, eli...

Tarkistuslistoja on tietysti erilaisia. Heuristiikat muodostavat oman tarkistuslistansa. Tuotteiden arvioinnissa pitää olla konseptitasolla omat listansa ja detaljisuunnittelua varten toinen setti.

Riippumattomuus

Arvioinnin perusidea on "objektiivisuus". On oleellista, että arvioija pystyy mentaalisesti irtautumaan kohteesta ilman, että antaa omien haasteiden tai paineiden vaikuttaa arviointiin. Esimerkiksi tietosuoja koskevaan arviointiin ei saa vaikuttaa se, että ohjelmoijille ovat tietokantoihin liittyvät asiat vähän vieraita. Ja jos tehdään uutuustuotetta, tiimissä voi olla vääristynyt kuva sen hyvyydestä, mikä voi tuottaa arvioijalle vinoutuneen lähtökohdan, jolloin arviointia ei tehdä asianmukaisesti ja sen tuloksia tulkitaan väärin.

Joskus arviointia käytetään vakavien asioiden hyväksymiseen – onko esimerkiksi systeemi

riittävän turvallinen – ja silloin on arvioijan syytä olla riittävästi irti kohteen tuottavasta organisaatiosta. Siihen on olemassa eri tasoja, riippuen kohteen riskeistä.

Riippumattomuutta vaativissa tilanteissa, joihin kuuluvat myös erilaiset organisaation toiminnan sertifiointit, tehdäänkin yleensä oma esiarviointi, jonka pitäisikin olla olennaisempi, sillä ulkopuolinen arvioija kannattaa kutsua vasta sitten, kun on suhteellisen varmaa, että arviointi tulee menemään läpi.

Näkökulma rajaa ja antaa perspektiivin

Arvioinnilla on aina jokin näkökulma, mikä antaa sille raamit. Selvitetään esim. toiminnallista laatua ja tietoturvasuutta. Tämä auttaa keskittymään asiaan, mahdollistaa asiantuntemuksen hyödyntämisen ja ennen kaikkea: on tiedossa, mitä on arvioitu ja mitä ei. Samalla se merkitsee sitä, että tarvitaan aina jonkinlainen teoria siitä, millaista olisi hyvä kohde, esim. hyvä projekti, hyvä käyttöliittymä. Ne teoriat vaihtelevat. Kypsyysmalleilla on erilainen teoria kuin ketterydestä lähtevällä arvioinnilla. Ja kummassakin tapauksessa teorian on syytä olla pätevä ja tilanteeseen sopiva. Kyse on siis tarkasteltavan asian laadun elementtien ymmärtämisestä kontekstissa.

Arvioinnille on aina eduksi, jos se käyttää erilaisia menettelyjä, koska ne aina paljastavat kohteesta jotain erilaista. Esimerkiksi projektiriskejä arvioitaessa on hyvä käyttää tarkistuslistoja ja toisaalta käydä projektin suunnitelmaa läpi vaihe vaiheelta. Testauksessa on sama juttu. Testauksessa erilaiset testaustekniikat täydentävät toisiaan.

Arviointi edellyttää kokonaisuuden eloon herättämistä

Käytettävyyttä ei voi arvioida tarkastelemalla vain käyttöliittymää, sillä on kyse sen ja käyttäjän suhteesta. Eikä sekään riitä, sillä se suhde on relevantti vain tavoitteellisessa toiminnassa. Siksi tarvitaan skenaarioita, joissa käyttöä tarkastellaan.

Sama pätee tietotekniikan toiminnallisuuteen. Tarvitaan skenaarioita, joissa liikkuu dataa realistisella tavalla ja jossa käyttäjällä on tietynlainen tavoite ja tapa. Tutkiva testaus perustuu usein

juuri tähän (eli on löytänyt käytettävyydestäuksen ikivanhoja oppeja...).

Arkkitehtuuriakaan ei voi arvioida tarkastelemalla staattista oliota, vaan on mentaalisesti altistettava se erilaisille skenaarioille.

Ja ratikkatyömaan vaikutuksia voi arvioida vain miettimällä, mitä kaupungissa tapahtuu, kun remontti jossain alkaa, jatkuu, ja toivottavasti joskus loppuu.

Eloon herättämiseen tarvitaan ennen kaikkea ajattelua. Aikajanan, ympäristön ja vuorovaikutuksen mallintamista, kuvittelua. Ja sitten simulointia. Mutta simulointi ei auta, ellei sitä käytetä pätevästi.

Arviointi on aistillista

Arvioinnin oletetaan aina perustuvan järkeen ja logiikkaan. Mutta siinä on aina aisteilla iso rooli. Tutkivassa testauksessa testaajan on tarkkana oireille: jokin näyttää hassulta, tuntuu, että ohjelma hidastuu jossain, kovalevyn ääni kertoo oudosta kuormittumisesta. Moni testaustekniikka vertailee systemaattisen prosessin osana saatuja tuloksia visuaalisesti johonkin referenssiin (oikeaksi tunnetut tulokset). Käyttäjäkokemuksen testauksessa reflektoidaan koehenkilöiden aistinvaraisia ja tunnetason kokemuksia. Datapohjaisessa testauksessa (ml. koneoppivien systeemien testaus) pitää jo silmällä arvioida miltä data näyttää. Siinä se "ihmisoppiva" systeemi havainnoi asioiden luonnetta. Toiminnan laatua arvioidessa etsitään nenällä huonojen tapojen "lemuja" (ok, tässä se ilmaus on lähinnä vertauskuvallinen). Toiminnan virtausta tarkasteltaessa osaava silmä näkee, missä virtaus pätkii. Organisaatiokulttuurin tuntee kaikilla aisteilla.

Ihmisen aistit ovat aina asioiden arvioinnissa vahvassa roolissa, olipa käytössä mitä menetelmiä tai välineitä tahansa.

Olemme tapojen ja maailmankuvamme vankeja

Historia ja perinne ohjaavat ajatteluamme. Esimerkiksi ohjelmistotuotteiden laadunvarmistus on alkanut toiminnallisuudesta ja vasta myöhemmin ovat mm. käytettävyys ja tietoturvallisuus tulleet mukaan. Uudet asiat ovat aina mukana viiveellä ja

vähän lisäkkeinä ja niiden arviointiin panostetaan liian vähän.

Menetelmien valinta riippuu arvioijan maailmankuvasta. Esimerkiksi yhteiskunnan systeemien vaikutusten arvioinnissa talousihmiset haluavat vääntää kaiken numeroiksi, olipa sille pohjaa tai ei. Ja arviointiin päätyy vain asioita, joita osataan vääntää numeroiksi. Jos taas arvioija olisi systemaattisesti ajatteleva sosiologi, arvioinnin genre ja tyyli olisivat hyvin erilaiset. Oliko esim. Tampereen ratikan vaikutustenarvioinnissa käsitelty pitkän työmaa-ajan vaikutuksia kaupungin vetovoimaan, kansalaisten viihtyvyydelle ja keskustan liike-elämälle? Ei. Vaikutusten arviointi on riskianalyysiä ja siinä pitää olla samantyyppinen ote.

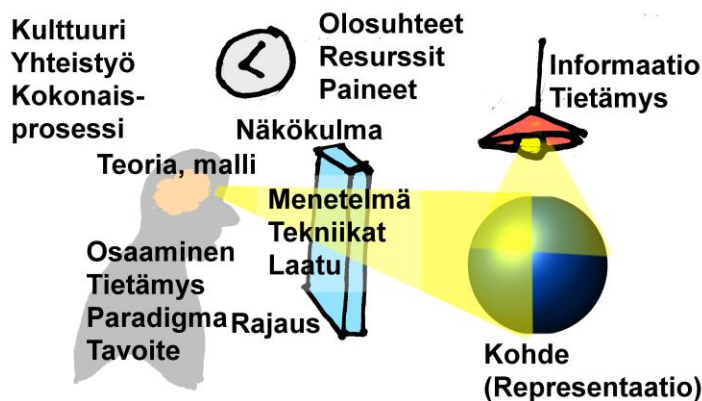
Laatuarvioinneissa meitä rajaa laatuksityksemme. Laadun maailma laajenee koko ajan ja organisaatioajattelu on aina yhden pykälän jäljessä. Toiminnallinen laatu on laadun stereotypia ja esim. käyttäjäkokemuksen syvällisiä tasoja hallitaan huonosti.

Testaustoimintaa arvioidessa voi vastinpintana olla jokin kypsyysmalli, vaikka organisaatiotutkimuksessa on pitkään ymmärretty, että jokaisen organisaation pitää löytää omaan tilanteeseen parhaat tavat. Niinpä arvioinnin vastinpintana pitäisikin olla syvälinen ymmärrys organisaation kontekstin tarpeista ja mahdollisuuksista. Se toisaalta edellyttää arvioijaa ja kypsää organisaatiota, jotka pystyvät näkemään niitä mahdollisuuksia – niitä ei voida nyt lukea arviointimenetelmän manuaalista. On ikävää, jos arviointi panostaa yksikkötestauksen menetelmiin eikä kokeilutoiminnan ja innovoinnin testauskäytäntöihin tai käyttäjäkokemuksen testaukseen, jos organisaation menestys on enemmän kiinni niistä.

Tällaiset käsitysten kuplat ovat vaarallisia arvioinnille ja sen kohteelle.

Diversi osaaminen tarpeen

Näkökulmia lisää aina arvioijaan liittyvä diversi teetti. Eli siis se, että kun arviointia tehdään ryhmätyönä, on hyvä, jos ryhmässä on erilaisia ihmisiä. Osaltaan se syntyy jo siitä, että mukana on ihmisiä, jotka edustavat eri näkökulmia – joku tuottajaa, joku asiakasta, joku ympäröivää infraa, jne...



Arvioinnin mikrosysteemi

tunnistat asiantuntijan":

https://www.mattivuori.net/julkaisuluettelo/liitteet/mista_tunnistat_asiantuntijan.pdf

Yksilötyönä tehtävässäkin arvioinnissa pitää hyväksyä ihmisten erilaiset ajattelutavat. Kun arvioijalla on päässään teoria kohteesta, niitä teorioita on erilaisia. Ja hänellä on myös teoria arvioinnista. Jollekin tuote on selkeä kokonaisuus, josta järjestyksessä ruksataan standardin luettelemat piirteet arvioinnissa läpi. Toiselle se on lähtökohtaisesti uniikki olio, jonka olemus täytyy ensin tunnistaa ja sitten tarttua ensin olennaisuuksiin.

Pitää tiedostaa ja hyväksyä, mikä on arvioijan teoria, kun hänet päästää töihin. Tosi asiantuntija osaa katsella asiaa muutamankin teorian kautta.

Jos arviointia tehdään asiantuntijatyönä, on olennaista asiantuntijan substanssiosaaminen. Esim. käytettävyyden analysointi on usein tällaista. Mutta ryhmätyö-arvioinnissa on tärkein taito prosessin fasilitointi; substanssiasiantuntemus löytyy silloin ryhmästä.

Onko arviointi tieteellistä?

Arviointi soveltaa osin tieteen saavutuksia käyttäessään jotain teoriaa ohjaamaan ymmärrystä arvioitavista asioista ja arviointiprosessista. Mutta varsinaisten asioiden suhteen on yhtä tärkeää unohtaa tieteellisyyden rajat ja käyttää kaikkia havainnoinnin, ajattelun ja johtopäätösten kykyä, mitä ihmisestä ja ryhmästä löytyy.

Mistä tunnistan hyvän arvioijan?

Jos ajatellaan, että arvioinnissa korostuu monenlainen asiantuntemus, voi arvioijan piirteitä suhteuttaa asiantuntijan piirteisiin (muistettakoon kuitenkin, että arvioinnissa tarvitaan yleensä myös muunlaisia piirteitä). Vanha juttuni "Mistä

Tekoöly arvioijana

Tai siis nykyään koneoppiva systeemi, joka tietysti voi hyödyntää sääntöpohjaisuuttakin, mitä ei enää aina tykätä kutsua tekoölyn kenttään kuuluvaksi. Kutsutaan näitä kumpaakin nyt kuitenkin tekoölyksi. Esimerkiksi liiketoimintasuunnitelmasta kone voi helposti tarkastaa, että kaikki olennaiset osuudet ovat mukana, eli että mitään olennaista ei puutu. Tämä onnistuu sääntöjä ja luonnollista kieltä prosessoimalla. Seuraava taso on tarkastaa, että asioiden kuvaukset ovat sellaisia kuin aiemmissa hyvissä suunnitelmissa. Mutta! Siihen tarvitaan paljon dataa. Jokainen liiketoiminta-suunnitelma on uniikki – uniikki aika, idea, ympäristö; uniikit assetit, uniikit riskit. Ja jokainen suunnitelma liikeideaksi ja sitä toteuttavaksi toimintajärjestelmäksi pitää aina sovittaa olosuhteisiin ja organisaatioon. Pitää siis olla uniikki, kenties jopa luova. Luovien ratkaisujen arviointiin ei tekoöly pysty. Sillä ei ole kylliksi dataa eikä perspektiiviä eteenpäin. Kaikki data on historiasta.

Arviointiin tarvitaan uniikkiudelle herkkä, osaava ihminen.

Tekoöly voi suunnilleen tarkastaa asioiden vastavuuden pakollisiin vaatimuksiin – kunhan se on niidenkin suhteen ajantasalla on hallitsee niiden tulkinnat, eikä vain kirjainta.

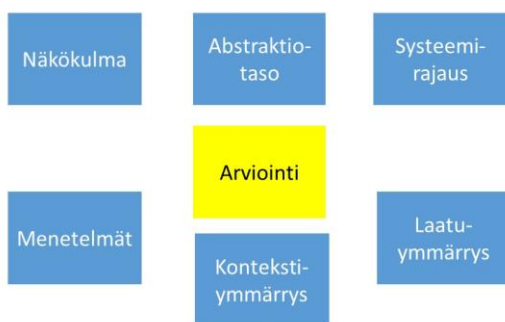
Ensin asia kartalle

Arvioinnilla on usein prosessi, joka kulkee tarkasteltavan prosessin rinnalla. Sen ensimmäinen askel on asemointi: millainen tehtävä, prosessi, tilanne on kyseessä? Onko se uudenlainen, riskialtis, vanha tuttu? Sen luokittelun perusteella

voidaan päättää, millaista arviointia tarvitaan jatkossa. Esimerkiksi projektien alussa ne "laitetaan kartalle" ja sovitaan sitten millaista riskienhallintaa, ml. riskianalyytit, tullaan tekemään. Myöhemmin tilannekatsaukset toistuvat toisen tyyppisinä.

Prosessiin liittyy näkökulmien ohella monitasoisuus. Turvallisuuskriittisiä järjestelmiä arvioidaan usealla tasolla, jotka myös vastaavat systeemin rakenteellisia tasoja. Katsellaan koodia, softa-rauta-järjestelmää, käyttöliittymää ja kokonaisuutetta sen käyttöympäristössä. Tällainen tasorakenne auttaa jäsentämään ajatuksia ja luomaan lopputulos, jossa mahdollisilla ongelmilla on monta turvaverkkoa.

Samalla se merkitsee sitä, että asioilla on erilaisia rytmejä. Kokonaisuutta ei arvioida jatkuvana prosessina, vaan rytmisesti, pysähtyen ja keskittyen. "Miten elefantti syödään? Pala kerrallaan" – niinpä, mutta elefantin "elefanttiutta" ei voida arvioida pala kerrallaan.



Tartuntapintoja kohteeseen.

Arvioinnin organisatorinen taso

Edellä on ollut kyse arvioinnin ydinprosessista tai tiedollisesta prosessista. Yhtä oleellinen on arvioinnin organisatorinen prosessi. Esim. toiminnan laatua arvioitaessa pitää organisaation olla mukana arvioinnissa, jotta se on hedelmällinen.

Organisatorinen prosessi lähtee arvioinnin tarpeen huomaamisesta jonkin signaalin, rutiinin tai oivalluksen, perusteella. Sitten pitää organisoida. Joskus arvioinnit ovat erikseen projektoituja ja silloin sovitaan tehtävät ja huolehditaan resursoinnista ja viestinnästä. Arvioinnin jälkeen taas palataan vahvasti organisaatiotasolle, kun mietitään, mitä arvioinnin tulosten perusteella pitäisi tehdä.

Arvioinnin menestystekijöitä

- Pätevä arviointiperusta.
- Osaava arvioija.
- Toteutus tilanteeseen sopivan nopeasti (palautesykli arvioinnin tarvitsijoille).
- Resurssien käyttö.
- Tulosten hyödynnettävyys.
- Muutama eri menetelmä tai näkökulma.
- Riittävä psykologinen riippumattomuus.

Arvioinnin uhkia

Arviointi ei aina ole ihan priimaa. Sille on monenlaisia uhkia, esim. tällaisia:

- Heikko laatukulttuuri. Arviointia ei tehdä tai tehdään vain tavan vuoksi.
- Heikko etiikka. Arviointi tehdään valkopesemään asia.
- Heikko arviointiosaaminen.
- Heikko substanssiosaaminen.
- Arvioijan sellainen substanssiin liittyvä ajattelutapa, joka ei vastaa kohteen haasteita.
- Vain yksi menetelmä.
- Kapea skouppi.
- Arviointi ei lähde kohteen olennaisista piirteistä tai yritä tunnistaa niitä.
- Liian raskas tai hidas prosessi.
- Liian pintapuolinen ja kevyt prosessi.
- Väärä ajoitus. Tehdään liian myöhään.
- Arviointien tuloksia ei osata hyödyntää.
- Ei ole rutiinia ja arviointi on kiinni ihmisistä.
- Kriteerit piilotetaan.

Arvioinnin pitää olla avointa

Joskus arvioinnin kriteerit pidetään salassa. Tätä esiintyy silloin, kun arviointi on kaupallistettu ja kriteerit ovat liikesalaisuus. Mutta esimerkiksi työpaikan arviointi on merkityksetöntä, ellei arviointiperustaa kerrota. Toinen avoimuuden alue on arviointiprosessi. Vaikkapa tutkivaa testausta on vaikea pitää luotettavana ja hintansa arvoisena, ellei sen palvelukuvaukseen ja lopputuloksiin kuulu esittely siitä, mitä tullaan tekemään, millä perusteluilla, ja mitä oikeasti tehtiin.

Esimerkki arvioinnin rikkaudesta – käytettävyyden arviointi

Esimerkkinä arvioinnin monimuotoisuudesta olkoon tuotteen kehittämisen aikainen käytettävyyden arviointi, millä on monia tapoja ja tavoitteita.

Referenssi, standardi: Kilpailijatuotteiden arvioinnilla päästään kartalle. Käytettävyys, sen elementit, odotettu taso ja mahdollisuudet ovat aina suhteellisia ja sen suhteellisuuden maailma paljastuu nykyisten kilpailijatuotteiden arvioinnilla.

Konsepti, kokonaisuus: Konseptivaiheessa ja vaatimusmäärittelyssä arvioidaan analyttisesti ja katselmoiden perusideoita ja sitä, mitä ollaan hakemassa.

Arviointiperustan jalostus: Kun ymmärretään, mitä suunnilleen ollaan tekemässä, syntetisoidaan arviointinäkökulmien pohjaksi käyttäjiä mallintavat persoonat ja käyttöä mallintavat skenaariot.

Toteutuskonseptien vertailu: Varhaisessa vaiheessa luonnostellaan muutama vaihtoehtoinen konsepti, joita vertaillaan – siis arvioidaan erillisesti tai suhteessa toisiin – ja yhden kanssa jatketaan.

Ulkoinen ääni: Focus group päästetään kenties arvioimaan ideoita.

Suunnittelun arviointi: Luonnoksia ja prototyyppisiä arvioidaan käyttämällä eritasoisia tarkistuslistoja, heuristiikkoja ja työn analyysin kaltaista skenaarioiden systemaattista analyttistä läpikäyntiä. Testausta tehdään kokonaisuudelle ja yksittäisille asioille (esim. symbolien preferenssi- ja ymmärrettävyydestit).

Toimijoiden jatkuva reflektio: Muotoilijat ja käyttöliittymäsuunnittelijat arvioivat itse jatkuvasti työtään ja käyvät dialogia suunnitelmista ja toteutuksista.

Toteutusten jatkuva arviointi: Uusien toimintojen käytettävyyttä analysoidaan ja testataan ainakin kevyesti. Systemin vähän kypsyessä kokonaisuuden käytettävyydestejä jatketaan (mutta niin varhain, että voidaan vielä muuttaa suunnitelmia).

Erityisalueet: Saavutettavuutta on tarkasteltu alusta asti. Ennen julkistusta sen vastaavuus

vaatimukseen (julkishallinnon käytössä) vielä tarkastetaan.

Ulkoinen ääni: Pilottiasiakkaat saavat kokeilla systeemiä ja kertoa siitä palautetta. Pilotti-asiakkaiden kanssa tehdään käytön havainnointia ja kerätään käytöstä lokia, joita analysoidaan.

Porttitilanteet: Ennen julkistusta tehdään julkistuskatselmointi, jossa arvioidaan tuotteen julkistusvalmius.

Käytönaikainen arviointi: Käyttäjäpalautetta ja mediaseurantaa tehdään ja saatua arvioidaan.

Jne...

Esimerkissä näkyy kokonaisuus, jossa arviointi on koko ajana läsnä. Sen muoto vain vaihtelee sen mukaan, mitä kulloinkin tarvitaan, mikä tuo rytmiä, mille aika on kypsä.

Lopuksi

No niin, jutussa on ihmetelty arviointia eri suunnista ideana tukea arvioinnin yleistä ymmärrystä, jotta uudistuvassa maailmassamme osattaisiin tehdä uusenlaisille kohteille, uudenlaisissa ympäristöissä oivia arviointeja.

Arviointi näyttää usein simppeleiltä ja sitä se voi ollakin professionaalisessa kulttuurissa, mutta kun arviointia tehdään uudenlaisissa konteksteissa perusasiat saattavat unohtua.

Toivottavasti juttu muistutti arvioinnin olennaisuuksista ja antoi impulsseja lukijan oman ympäristön arviointitoiminnan kehittämiseen.

Matti Vuorella on kokemusta monenlaisesta asioiden formaalista arvioinnista: teollisuuden turvallisuusanalyysit, tuotteiden käytettävyyden arviointi, laatujärjestelmän arviointi, projektiriskien arviointi, tieteellisten julkaisujen arviointi, opinnäytteiden arviointi jne... Hän on paitsi tehnyt arviointia, myös kehittänyt siihen menetelmiä ja niiden tueksi eksaktisti ziljoonia tarkistuslistoja ja kouluttanut näitä asioita.

Liite: Arviointiin liittyvä luokituslistoja

Tavoitteita

- Tarkastaa vastaavuus vaatimukseen, lakeihin, direktiiveihin, standardeihin
- Hyväksyntä, akkreditointi, sertifiointi
- Saada todenne

- Laadun arviointi
- Perustason luominen kehittämiselle
- Vertailu
- Oppiminen, kehittyminen, parantaminen
- Edistymisen mitta

Menettelyjä

- Tarkastaminen, auditointi
- Näytteenotto
- Empiirinen testaus
- Hyväksymistestaus
- Kokeilu
- Itsearviointi
- Mittaus
- Pilotointi
- Havainnointi
- Kysely, haastattelu
- Oma kokemus

Arviointiperusta

- Lait, direktiivit, standardit
- Arviointimalli
- Kypsyysmalli
- "Parhaat käytännöt"
- Vaatimukset, määrittelyt
- Kontekstin implisiittiset vaatimukset
- Tavoitteet

Kohteita – ylipään jonkin konseptin kaikki elementit

- Organisaation strategia, toiminta, liiketoiminta, toiminto, viestintä
- Arvot, politiikat, kulttuuri, ilmapiiri, etiikka
- Kyvykkyys, suorituskyky, tuottavuus
- Projekti
- Palvelu
- Tuote, prototyyppi, MVP
- Teknologia
- Artefakta, suunnitelma, malli, kuvaus
- Idea, ajatus, ehdotus

Arviointi osana toimintaa

- Työn kiinteänä osana
- Prosessin vaiheena
- Prosessin näkökulmana
- Prosessia tarkastelevana
- Tilattuna arviointina
- Toiminnan kokonaisuuden tarkastamisena
- Katselmuksena, katselmointina

Rytmi

- Jatkuva
- Prosessin rytmin mukainen
- Vuosikello

Mitä yhteistä on tutkijoilla ja testaajilla?

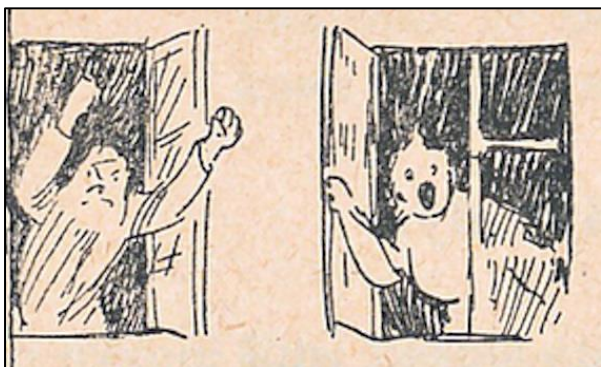
Tuula Pääkkönen, Kansalliskirjasto

Kun on tehnyt töitä testaajien ja tutkijoiden kanssa alkaa hiljalleen huomata pieniä piirteitä, jotka ovat molemmille ryhmille samanlaisia. Vaikka työalue tai kiinnostuksen kohde on erilaista, tavoitteissa ja peruseriaatteissa on paljon samaa, jotta saadaan hyvä tutkimus, tuote tai palvelu.

Roolin tavoitteet

Testausgurut usein korostavat, että **testaaja** pyrkii parantamaan lopputuotetta, saamaan tietoa tuotteesta, yleisimmin etsimällä vikoja tai nostamalla kysymyksiä tuotteesta sen eri vaiheissa. Tavoitellen sitä että lopputuote tai -palvelu on parempi kuin aiemmin.

Tutkija pyrkii taas löytämään tietoa, käyttämällä muodollisesti jäsennettyjä tutkimuskysymyksiä, tieteellisiä menetelmiä ja saatavilla olevaa dataa, tutkimuskysymysten ratkaisemiseksi. Tavoitteena ymmärtää kiinnostuksen kohdetta ja vaikka tehdä maailmaa tältä osin paremmaksi paikaksi.



Kuva 1. Tutkija vai testaaja? (Ote Rudolf Koivun piirroksesta, Vallio (1918)).

Alussa oli...

Tutkijan tutkimus lähtee yleensä uteliaisuudesta*, hän törmää alallaan johonkin asiaan, josta haluaa

tietää lisää, ja jossa tuntuu olevan aukko nykyisessä tietämyksessä. Eli syntyy tutkimuskysymys, uudesta asiasta, tai uudesta näkökulmasta vanhaan kysymykseen. Testaajalla tilanne on sama*, voi olla vanhan palvelun uusi versio, aivan uusi palvelu tai uusi domain ja kysymyksiä alkaa heti muotoutua. Testaaja lähtee siis esimerkiksi tutkivan testauksen avulla haravoimaan aluetta, ja saamaan mielikuvaa mitä ohjelmistossa on meneillään ja onko erityis-alueita, joita pitäisi erityisesti tutkia.



Kuva 2. SMART tavoitteet, kuvan lähde Wikimedia Commons CC BY-SA 4.0)

Tutkijan tutkimuskysymyksen pitää olla sopivan muotoinen – sen pitää olla tarpeeksi tarkasti määritelty, mitattavissa tai jotenkin taklattavissa niin, että valitussa tapauksessa tutkija voi kysymykseen vastata. Tutkimuskysymyksen pitää olla toisaalta myös rajattu, jotta muutamaan niistä voi tutkimusartikkelin sisällä vastata. Tutkimuskysymys toimii tavoitteena, joka auttaa myös artikkelin lukijaa pitämään mielessä, millä ajatuksella tutkija on asiaan lähtenyt.

Testaajalla taas on testauksen malli mielessään. Tehtävänä on sovellus ja sen nykyiset toiminnot, tai uudet hienot toiminnot tai ihan uusi sovellus, ja dokumentaatiota on tai on vaihtelevasti lähtötilanteessa. Niinpä, saatetaan tehdä joko ihan testi-suunnitelma, muistilista, tai katsotaan muuttuvat kohdat, joita voidaan katsoa vastaamaan tutkimus-

kysymystä – mikä kaikki toimii ja mikä ei? Peruskysymyksiä, joita sitten voidaan kysyä: ”Toimiiko kaikki?” Testaajankin pitää miettiä mitä hän nyt pystyy testaamaan ja suhteuttamaan sen käytettävissä olevaan aikaan, joten pitää pohtia vaikka riskialttiimpia kohtia, tai mistä asioista voi olla lainsäädäntöä tai millaisista virheistä tulee isoimmat lööppiotsikot. Molemmat pohtivat työnsä alkustrategiaa, olemassa olevan tiedon perusteella.

**) Välihuomio. Pitää toki muistaa, että joskus tutkimuskohteet voivat löytyä myös organisaation tavoitteiden tai (tutkimus)projektien tavoitteista, jotka ovat menneet rahoittajalta tai organisaatioissa erillisestä ideaprosessista eteenpäin toteutusvaiheeseen. Oli tavoitteen takana sitten ideologia tai maallisempi ’leipä’ niin miellyttävämpää on jos teemasta saa itsensä kiinnostuman, jolloin tekeminen ei tunnu niin raadolliselta työltä.*

Suhde tietoon

Tutkijat tarvitsevat dataa. Mitä parempaa dataa sen parempi, ja usein alussa tutkimus lähteekin datan ja aineiston keräämisellä. Aineistoa voidaan saada aiemmista tutkimuksista, kollegoilta, omalta organisaatiolta tai data voidaan kerätä itse. Dataa voidaan saada monesta lähteestä, esimerkiksi voidaan käyttää haastatteluja, koeasetelmia tai vaikka tilastotietoja kerättyinä toiseen tarkoitukseen. Hyvä data onkin kullon arvoista, ja tutkijat pitävät siitä kyllä kiinni, koska siitä voi saada lisätietoa seuraaviinkin tutkimuksiin. Sekin kuinka dataa kerätään riippuu siitä mitä tutkitaan, jolloin pitää olla varma että kerätty data vastaa ennalta-asetettuun tutkimuskysymykseen. Data tosin yksinään ei riitä, vaan sitä pitää käsitellä, muovata ja järjestää siten, että haluttujen tutkimuskysymyksiin saadaan vastaus.

Testaajat taas keräävät tietoa testattavasta kohteesta, eri puolilta ja eri ominaisuuksista. Onpa joskus niin, että testaaja saattaa olla se ensimmäinen joka alkaa koostamaan kaikkea tuotteen aineistoa yhteen, kun hän alkaa pohtimaan testitapauksia tai suunnittelemaan tutkivan testauksen matkareittiään. Joskus tuote kertoo tiettyjä asioita, ja organisaatio ja sen toimintatavat toisia. Mihin osiin ei haluta koskea? Miksi? Millaisia keskusteluja vilisee, missä voisi olla ohjelmiston ’ajatuksessa’

aukkoja, joissa viat voivat piileskellä tai mitkä voisivat hämmentää käyttäjää?

”Ei näin kalliissa puhelimessa voi tällaista vikaa olla”

Yllä oleva on lause, jonka kuullut pari kertaa kadulla ohikulkijoilta. Hilpeää tai surullista, mutta tuotteen hinta ei välttämättä korreloi vikamäärän kanssa ja siksi testaus on suhteellisen samanlaista toimintaympäristöstä riippumatta. Kalliimpi tuote saattaa tarkoittaa isompia voimavaroja kehitysorganisaatiossa, mutta toisaalta silloin ’tutkimuskohde’ on laajempi eli focus pitää sitten levittää laajemmalle.

Vapaus ja vastuu

Suomessa puhutaan tutkijan vapaudesta tutkia aihetta, joka häntä kiinnostaa. Lisäksi pidetään tärkeänä sitä, että huolimatta ulkoisen rahoituksen etsinnästä, niin tutkimuksen pitäisi kuitenkin olla neutraalia ja riippumatonta. Väliverronen (2019) antaa lukuisia esimerkkejä, kuinka tutkijan vapautta on ulkomailla rajoitettu esimerkiksi poliittisten rahoituspäätösten avulla, tai viestintäohjeilla... Tutkimuksen vaikutukset siis tunnustetaan ja siksi niitä halutaan aika ajoin kontrolloida, jotta tarina taas joku toinen haluaa esittää pysyy yhtenäisenä.

”Testaamisen arvostuksessa pitäisi työstää sitä, että kun jotakin löydetään, testaajat eivät joutuisi tappelemaan siitä, että heitä kuunnellaan” / Maaret Pyhäjärvi (Korhonen, 2019).

Testaajille viestin välittämisen vaikeudet ovat myös usein mielessä. Kuinka esittää huonot uutiset rakentavassa hengessä ja siten että ollaan samassa veneessä. Jos aina olet se, joka tuo huonoja uutisia, niin pitää varautua tuskan kiljahduksiin, kun jotakin löytyy – tai toisaalta jos jokin jää löytymättä, niin yhtäläillä sanomista voi tulla. Siksi, kuten tutkijakin, pitää ”omat kotiläksyt” olla tehty, ja olla varustautunut tiedoilla ja menetelmillä, jolloin voi taapertaa pikaviestimen ääreen ja avata keskustelun joskus kivuliaasta aiheesta. Joskus vian voi raportoida kahdeksalla merkillä, joskus se voi vaatia esseen ja parin vuoden nalkutuksen.

Raportointi

Tutkijoiden työn raportointi tyypillisesti tapahtuu tutkimusartikkeleiden myötä, tuotokset kirjataan

ylös tietyllä tieteellisen julkaisun odottamalla tavalla ja lähetetään lehden katselmointiryhmän ruodittavaksi. Artikkelissa kuvataan mitä on tehty, miksi, millaista aineistoa on käytetty ja millaisilla menetelmillä tutkimusta on viety eteenpäin. Artikkelissa kuvataan siis tutkimuksen 'tarina' näillä elementeillä, joka vie suuriin johtopäätöksiin ja ajatuksiin siitä mitä voisi tutkia seuraavaksi. Joskus tuotokset saatetaan julkaista myös tviittinä, blogitekstinä tai muuna raporttina riippuen keitä tutkimuksen raportoinnilla haluttaisiin tavoittaa.

Testaajat taas kertovat mitä ovat testanneet, miten ja mitkä lopputulokset. Ehkä voisi jopa uskaltaa sanoa, että vikaraportteja ja tietoja annetaan itse testausprosessin aikana, jolla voidaan viestiä kehitykselle mitä tarvitsisi tehdä ja miten ja jotta palautesykli kehitykselle on mahdollisimman paras. Testaus ei välttämättä ota sen suuremmin kantaa mitä jollekin huomatuille asialle tehdään, vaan kertoo huomatuut faktat ja voi luottaa siihen että asiat tulevat korjattua. Joskus tosin, jotkin viat voivat vaatia 'bug advocacy' eli lobbausta jotta eniten kaivelevat viat saadaan korjattua.

Lopuksi

Tutkijoiden töitä on hauska seurata, koska siitä löytää niin paljon samanlaista testaustyöhön. Tietysti erojakin on, jota pitää ehkä miettiä jatkotekstissä. Tuima työasenne ja tarkka ote huomattuihin asioihin ehkä molemmilla oleellista. Lisäksi oleellista on ainainen uteliaisuus ja kiinnostus uusiin asioihin.

Kirjoittaja toimii Kansalliskirjastossa tietojärjestelmä-asiiantuntijana. Kokemusta tietojärjestelmien kehityksestä on piirun yli 20 vuoden ajalta, testauksesta ohjelmistokehitykseen.

Lähteet

Korhonen, S. (2019). Utelias virheiden viestinviejä. Tietoviikko 8/2019, s. 30-33.

Vallio, K. (1918). Prinsessa kultakutri / lapsille kirjoittanut Kari Vallio ; kuvat piirtänyt R. Koivu. <https://digi.kansalliskirjasto.fi/teos/binding/1985420?page=86>

Dataa testausta tehostamaan – vai astutaanko kehittäjien varpaille?

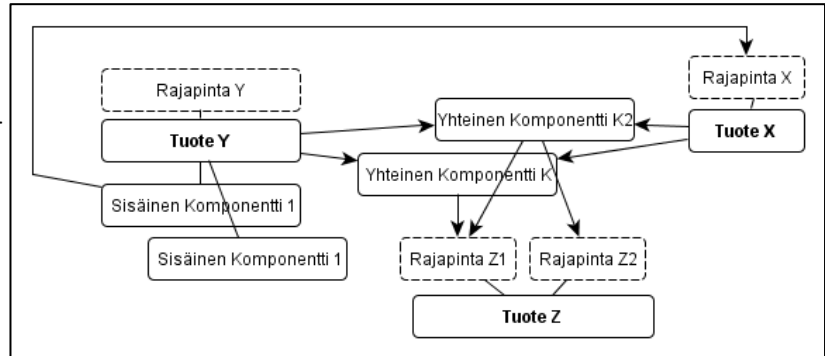
Ville Laitila, Softagram

Tyypillisesti ohjelmistokehityksessä käytetään JIRA:n kaltaisessa työkalua läpinäkyvyyden parantamiseksi. Mikäli läpinäkyvyyttä tarvitaan siitä, mitä muutoksia ollaan kussakin julkaisussa julkaisemassa, käytäntönä usein on lisätä merkintä, joka kertoo missä versiossa kyseinen muutos on tarkoitus julkaista. Näin julkaisuista päättävä, esim. tuotemistaja, voi hahmottaa mitä julkaisuun on menossa, määrittää testatarpeen ja tiedottaa asiakkaita julkaisun vaikutuksista.

Monimutkaisten kokonaisuuksien haasteita

Pienekköissä sovelluksissa harvemmin on tähän liittyvää kitkaa mutta jos kyse on useamman komponentin muodostamasta alustasta ja sen päälle rakentuvien sovellusten kokonaisuudesta, riski kasvaa siitä, että julkaisuun liittyvät merkinnät eivät ole ajan oikein merkitty. Tämä voi johtaa siihen, että julkaisu tehdään virheellisin tiedoin, ja tullaan julkaisseksi jotain, joka ei ole kunnolla testattu, tai vielä edes tarkoitettu julkaistavaksi. Lisäksi yksi seuraus voi olla se, ettei ymmärretä mitä julkaistiin, ja se sitten yllättääkin myöhemmin.

Erityisen hankalaksi ongelma muodostuu, kun useamman jaetun komponentin parissa tehdään muutoksia eri tuotteiden ominaisuuksia ja korjauksia toteuttaessa. Tällöin riippuvuuksien hallinta on tärkeää. Jos tuotteisiin liittyy myös takaisinpäin yhteensopivuuden vaatimuksia, vaikeuskerroin kasvaa. Kehittäjän näkökulmasta kaikki voi näyttää täysin toimivalta: koodi kääntyy, muutokset on tehty kaikkiin tarvittaviin paikkoihin. Kuitenkin on mahdollista, että koodimuutos rikkoo tietyn palvelun rajapinnan (esim. ylläolevan kuvan tuote Z muuttuu niin että rajapinnan Z2 yhteensopivuus (takaisinpäin) särkyä) ja vaatii tuolloin saman-



aikaiset päivitykset kaikkiin sitä käyttäviin tuotteisiin (koska yhteiset komponentit ovat osana tuotteita X ja Y, jotka altistuvat muutokselle).

Merkkaus versiohallintaan?

Tämä ihmisiin perustuva prosessi on hyvin helposti särkyvä. Muutosten julkaisussa kyse on ihmisten muistista, ja mitä pidempi julkaisusykli on, sitä todennäköisempää on, että jotain muutosten vaikutuksista unohtuu ja hallinta julkaisusta heikkenee. Mitä seuraisi, jos merkittäisiin (git tag) kaikkiin Git-repositoryihin komponenttien julkaisuversiot ja julkaisukandidaattien versiot per tuote? Käyttämällä git log -komentoa voisi sitten helposti selvittää mitkä koodimuutokset olisivat menossa seuraavaan julkaisuun, olettaen että komento ymmärretään ajaa sopivan laajasti kaikille repositoryille erikseen. Kun ymmärretään mitä koodimuutoksia tiedoston, commit-viestin ja omaisuus-tunnisteiden (JIRA ID:t) tarkkuudella, alkaa olla paljon luotettavampi dataan perustuva tietämys käsillä siitä, mitä ollaan julkaisemassa. Useinhan tiimit kehittävät yhteisiä komponentteja samanaikaisesti, ja yhteistä päähaaraa käytettäessä yhden tuotteen julkaisuun saattaa joutua mukaan muiden tuotteiden takia tehtyjä muutoksia. *Feature flag* -konseptia hyödyntäen uusia toimintoja voidaan viedä tuotantopäiväkirjoihin pois kytkettyinä (jotta toiminto ei toisi suurta riskiä koko käyttäjäjoukkoille vaan se olisi kytkettävissä päälle koemielessä pienemmälle joukkoille). Tällöin

kuitenkin poiskytketyksi väitetty uusi toiminto aiheuttaa silti muutosriskiä tuotteen julkaisuun.

Sivuutetaanko kehittäjiä?

Tässä on kuitenkin dilemma: kun perinteinen malli korostaa kehittäjien vastuuta informoida julkaisuista, datapohjainen lähestymistapa lisää läpinäkyvyyttä mutta osittain sivuuttaa kehittäjät tiedon tarpeiden selvittämisestä. Loukataanko tässä kehittäjiä, kun heidän vastuualuettaan rajataan ja julkaisua viedään kohti anonymimpää datapohjaista mallia? Nostaako datan hyödyntäminen kehittäjien kynnystä informoida julkaisusta päättäviä? Vai voidaanko informointi muuttaa täysin työkalupohjaiseksi, perustuen kurinalaiseen commit-viestien käyttöön (joissa informoidaan muutokseen liittyvän ominaisuuden tunniste) ja tagien käyttöön?

Datapohjaisessa lähestymistavassa vaarana on aina valheellisen turvallisuuden tunteen saaminen. Kun on vasara, kaikki alkaa näyttää nauiloilta. Ohjelmistoalan erityshaaste on siinä, että usein syy – seuraus -suhteissa ketjut ovat pitkiä. Esimerkkinä tällaisista pidemmistä ketjuista on tyyppinen REST-rajapinnan endpointien-käsittelijän (controller) käyttäytymisen muuttuminen ilman että itse käsittelijän koodia on muutettu. Kun controller käyttää ulkopuolista luokkaa tai kirjastoa pyynnön käsittelyn osana, niihin tulevat muutokset voivat aiheuttaa rajapinnan yhteensopivuusrikkomuksen. Tämä koodiriippuvuuksien verkosto tuo muutosvaikutusten analysointiin sellaista vaikeutta, johon ei vielä tekoäly kykene tuomaan juurikaan apua.

Uusia mahdollisuuksia

Omien kokemuksieni mukaan tämä malli ei vie kehittäjiltä lusikkaa suusta eikä myöskään luo liikaa valheellista hallinnan tunnetta. Kehittäjien palaute kuvatun kaltaisesta menetelmästä on ollut yleensä positiivinen. Se sitä paitsi mahdollistaa sen, että kiireellisten julkaisujen (hot fix) tekeminen on helpompaa ja luotettavampaa, tagien ollessa versionhallinnassa sitä varten käytettävissä. Lisäksi se tuo mahdollisuuden ajaa staattisia analysointoreita julkaisutestauksen yhteydessä juuri julkaisun muutoksille (yleensä suurissa projekteissa on täysin mieleton ajatus ajaa niitä koko projektin lähdekoodille koska löydösten läpikäyntiin menisi valtavasti aikaa). Staattisen analyysin avulla voi

selvitä muuttuneista tiedostoista ohjelmavikoja ja julkaisun kannalta kriittisiä muita tietoja (kuten vaikutuksia yhteensopivuuslupauksen pitämiseen). Riippuvuusanalysointien avulla voidaan selvittää sekä staattisen että dynaamisen analyysin keinoin mitkä muut osat (esim. mitkä tuotteet, mitkä tuotteen käyttöliittymän osat) altistuu muutoksille. Tämä tieto on arvokasta julkaisun testauksessa, mutta se ei saa liiaksi ohjata testausta. Tutkiva testaus on silti tarpeen. Ohjelmistoala on kuitenkin vasta nuori, ja kehittyneempiä työkaluja ei suuresti vielä käytetä, vaikka käytännöt ovat jo pitkälti versionhallintatyökalun myötä vakiintuneet.

Data voisi auttaa julkaisutestausta muutenkin. Julkaisutestausta helpottamaan voisi tuoda myös resurssin joustava säätö: särkyvimmat (eniten korjauksia niissä tai niiden lähiympäristössä) ja kriittisimmät (riippuvuusmäärien perusteella, *page rank* -algoritmin kaltaisesti) voitaisiin pisteyttää riskipistein. Julkaisun muutettujen kooditiedostojen riskipisteiden summa suhteutettaisiin aikaisempien julkaisujen vastaavaan metriikkaan, ja sen perusteella voisi suunnitella testauksen resurssintia julkaisua ajatellen. Ota yhteyttä minuun, jos aihe kiinnostaa!

Ville Laitila on ohjelmistokehittäjä, perheenisä ja yrittäjä. Hänen erikoisaluetta on ohjelmistoriippuvuuksien analysointi, mallinnus ja ohjelmistojen visualisointi. Hän on työskennellyt ohjelmistokehityksessä vuodesta 2000 lähtien. Hän perusti Softagram Oy -nimisessä koodin analysoinnin alueelle keskittyvän yrityksen vuonna 2013 ja työskentelee siinä kehitystehtävissä ja CTO:na. Softagram tarjoaa pilvepalvelua ja on-premises -ratkaisua ohjelmistomuutosten jatkuvaan analysointiin. Email: ville.laitila@softagram.com

Muistojen tiellä... ensimmäinen kohtaamiseni testiautomaaton kanssa

Matti Vuori

Ensimmäinen oma kokemukseni testiautomaatiosta oli noin kolme vuosikymmentä sitten. Kehittelin silloin VTT:llä ohjelmaa, joka teki työtapa-turmatiedoista erilaisia tilastoja varsinkin erilaisten pylväsdiagrammien muodossa. Usealle firmalle tehtiin palvelutoimintana kerran vuodessa nippu raportteja. Alustana oli MS-DOS ja ohjelmointikielenä C. Siihen aikaan ei ohjelmistotestaus ollut kauhean tunnettua, osattua tai edes puhuttua. Ja jos puhuttiinkin, se ei kuulunut kauas, kun viestinnän maailma oli aikailalla erilainen, joten käytettävissä ei ollut kauheasti niitä kuuluisia parhaita käytäntöjä tai framewörkkejä. Siihen aikaan minäkään en ymmärtänyt softien oikeaoppisesta testaamisesta yhtään mitään.

Raporttien tekeminen oli robottimaista touhua, kuten sen ajan tehokkailla tekstipohjaisilla käyttöliittymillä yleensäkin. Puolisokkoja valikoista valinta cursorinäppäimillä, enter ja taas takaisin. Eli siis fyysistä toistotyötä. (Sellainen käyttöliittymä olisi optimaalinen robotisoitava.)

Ensimmäinen askel automaatioon tuli, kun tein ohjelmaan makrosysteemin, jolla sai tehtyä ison nipun raportteja nappia painamalla. Sillä oli näppärä ajaa diagrammeja asiakkiden ohella testidatalle, kun ohjelmaa kehiteltiin. Toinen askel oli huomata, että regressioita oli helppo huomata visuaalisesti vertaamalla hyväksi tiedettyjä kaavioita juuri tulostetuihin (ei, regressio-termi ei ollut käytössä)! Ei tarvitse vertailla numeroita, kun silmällä näkee erot suoraan. Näppärää, mutta alkeellistako? Itse asiassa nykynäkökulmasta kenties modernia:

* Taustalla robusti sovellukseen rakennettu automaatio erillisen härvelin sijaan.

* Ei tarvetta monimutkaiselle verifiointikomponentille - "oikein tehtynä" kaavioita vertailtaisiin jollain työkalulla, jollainen vaatisi kohtalaisen paljon kehittämistä ja psykologisista syistä muuttuisi pian isoksi työkaluprojektiksi.

* Ihmisen ja koneen kykyjen hyödyntäminen tilanteessa tarkoituksenmukaisella tavalla.

* Testauksen vanhan esteettisen ideaalin mukaan vertailtaisiin graafeille tuotettua dataa näytteinä ja graafinpiirtoa erikseen ja sellaisen testien rakentamisessa on aina oma iso ja turhauttava työnsä - nykyisin moni pitääkin tolkullisena kokonaisvaltaisempaa verifiointia, mitä näemmä kutsutaan mm. approval-testeiksi. Tuloksena on testien minimointi ja oikea maksimaalisesti hyödyttävä testauspiste.

Oppeina siis: yksinkertaisuus, tilanteeseen sopiva automatisoinnin taso, yhdistetään koneen ja ihmisen kyvykkyksiä tasapainoisesti, lähtökohtana testattavuuden parantaminen muutenkin toimintaa hyödyntävillä tavoilla.

TestausOSY
Finnish Association of Software Testing