

13

Käyttöliittymän suunnittelu ja testaus

Kolmessa edellisessä luvussa on käsitelty käyttöliittymän suunnittelussa huomioitavia asioita. Tässä luvussa käänämme katseen käyttöliittymän suunnittelemisen prosessiin.

Olemme aikaisemmissa luvuissa tunnistaneet joukon käyttäjystävällisyyttä edistäviä käyttöliittymän suunnittelutavoitteita. Näitä ovat:

- vuorovaikutuksen toteutus käyttäjälle tutuilla ja hänen tarpeisiinsa liittyvillä käsitteillä,
- keskustelujen rakenteen ja tietosisällön toteutus käyttötilanteen tarpeiden ja prioriteettien mukaisesti,
- käyttäjän käsittelemien objektien näkyvyys,
- käyttäjän toimintamahdollisuuksien näkyvyys,
- toimenpiteiden peruolettavuus,
- käyttäjän käsitemallin johdonmukaisuus, sekä
- monimutkaisissa sovelluksissa käyttäjän käsitemallin kerroksittaisuus

Lista alkaa olla kattava, kun näihin lisätään vielä

- käyttötapojen yhdenmukaisuus ja johdonmukaisuus sekä
- informaation järjestämisen- ja esitystapojen yhdenmukaisuus ja johdonmukaisuus.

Nämä kaksi viimeistä tavoitetta eivät toteudu itsestään! Ennen käyttöliittymän yksityiskohtaista suunnittelua täytyy tietää ne yhdenmukaiset ja johdonmukaiset yleiset periaatteet, joiden mukaan käyttöliittymä toimii. Nämä periaatteet kirjataan suunnitteludokumentteihin sekä sovelluksen tylioppaaseen.

Tuskin kenelläkään on sellaista aivokapasiteettia, että hän pystyisi tyhjältä pöydältä kirjoittamaan nämä käyttöliittymän periaatteet. Yleisten periaatteiden pohtiminen on paljon helpompaa, jos pohtimisen raaka-aineena on hyviä esimerkkitapauksia. Tarvitaan siis prototyyppisiä tai ainakin paperiluonnoksia käyttöliittymän keskeisistä osista.

Käyttöliittymän periaatteiden määrittely on välttämättä luonteeltaan iteratiivista. On syytä varautua siihen, että nämä iteraatiokierrokset jatkuvat vielä tuotteen julkaisun tai systeemin käyttöönoton jälkeenkin uusien versioiden puitteissa.

Suorakäsittelykäyttöliittymän periaatesuunnittelu

Ohjelmistotekniikassa suunnittelumenetelmät jaetaan yleensä osittaviin menetelmiin (top-down) ja kokoaviin menetelmiin (bottom-up). Käyttöliittymien suunnittelua voisi näistä kummastakin poikkeavasti luonnehtia sanoilla ”keskeltä ulospäin”. Ensin suunnittelulle luodaan tärkeitä periaateratkaisuista kova ydin, jonka ympärille sitten aletaan kääriä yksityiskohtaisia ratkaisuja.

Tämä suunnittelun keskeltä ulospäin –luonne korostuu suorakäsittelyyn perustuvan käyttöliittymän suunnittelussa. Ennen käyttöliittymän yksityiskohtaista suunnittelua tarvitaan monenlaisia periaateratkaisuja:

- Mitä objekteja käyttäjälle esitetään?
- Mitkä ovat näiden objektien käyttäjälle näkyviä ominaisuuksia?

- Miten objektit ja niiden ominaisuudet esitetään käyttäjälle?
- Miten vuorovaikutuksen tilat esitetään käyttäjille?
- Mitä työkaluja käyttäjälle on tarjolla?
- Miten työkalut toimivat?

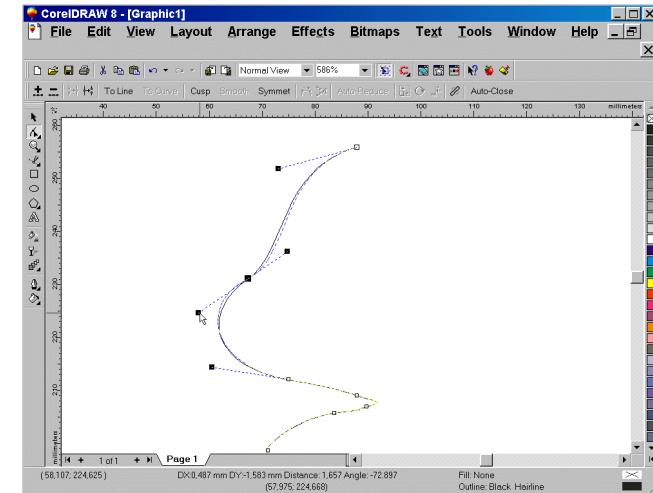
Käyttöölyttymän periaatesuunnittelu on viimeinen tilaisuus löytää asioiden sisällä usein piilevä yksinkertaisuus! Jos tässä vaiheessa päädytään tarpeettoman monimutkaisiin ratkaisuihin, monimutkaisuus alkaa pahimmassa tapauksessa ruokkia itseään. Jos monimutkaisuuden taltuttamiseen ja hallitsemiseen on uhrattu paljon työtä ja vaivaa, kukaan ei halua ajatella, että tämän kaiken vaivan tulokset pantaisiin roskiin jonkin yksinkertaisemman perusratkaisun tieltä!

Käytettävyyssivisio periaatesuunnittelun apuna

Kirjan toisessa luvussa käsittelimme käytettävyyssivioita: millaista sovelluksen käyttö olisi, jos tekniikka ei asettaisi mitään esteitä? Kun ensin on jonkinlainen ihanneratkaisu tiedossa, siitä on sitten helppoa askeltaa taaksepäin, kunnes tullaan mahdollisen piiriin.

Käytettävyyssivisio on hyvä työkalu myös suorakäsittelyn periaatesuunnitteluun. Vision haussa kannattaa usein mennä rohkeasti teknologian rajoitusten tuolle puolelle.

Kuvitellaan esimerkiksi, että olemme suunnittelemassa vektorigrafiikan piirtämiseen tarkoitettua ohjelmaa. Esimerkiksi Corel Draw käyttää mielivaltaisten muotojen kuvaamiseen niin sanottuja Bezier-käyriä:



Bezier-käyrät ovat matemaattinen menetelmä mielivaltaisten muotojen kuvaamiseen. En ole koskaan selvittänyt itselleni niiden takana olevaa matemaattista teoriaa. Olen käytännössä oppinut, että voin laittaa viivoihin tartuntapisteitä, joita voi sekä siirtää että kiertää. Näihin tartuntapisteisiin liittyy kahvoja, joiden venyttäminen vaikuttaa viivan muotoon.

Minun ajatusmallissani on siis aukkoja, mutta viivan oikea muoto löytyy yleensä kohtuullisella määrällä yrityksiä ja erehdyksiä.

Olemme aikaisemmin todenneet, että käyttäjän kanssa pitäisi viestiä hänelle tutuilla käsitteillä. Voisimmeko löytää vektorigrafiikkaohjelmalle käytettävyyssivision, joka ei sisällä Bezier-käyrien kaltaisia matemaattisia käsitteitä?

Visioinnissa täytyy katsoa rohkeasti tekniikan rajoitusten tuolle puolen. Niinpä kuvittelemme itsemme virtuaaliavaruuteen! Siellä on helppoa havaita, että emme tarvitse viivojen muokkaamiseen muuta kuin kaksi virtuaalista kättä. Voimme kuvitella, että viivat ovat kuin rautatankoja, joita voimme siirtää, kiertää eri asentoihin sekä venyttää ja taivuttaa haluamistamme kohdista.

Käytettävyyvisio on hyvin yksinkertainen: kaksi kättä, jotka tekevät viivoille asioita, jotka ovat tuttuja jokapäiväisestä elämästä. Miten sitten askellamme tästä taaksepäin mahdollisen piiriin?

Virtuaalimallisuuden tutkijat ovat kehittäneet datasilmälasia, joilla käyttäjä voi nähdä ympärillään asioita, jotka ovat olemassa vain tietokoneen muistissa. He ovat myös kehittäneet datakäsineitä, joilla voi tarttua virtuaalisiin esineisiin. Mutta nämä ovat vielä kalliita, harvinaisia ja vielä parantelua kaipaavia tekniikoita. Ehkäpä askellamme niiden ohitse!

Jospa panisimme piirtämishjelman työkalupalettiin rajattoman pinon käsiä? Käyttäjä voisi poimia paletista käsityökaluja ja sijoittaa niitä viivalle haluamiinsa kohtiin. Kun hän vääntää, kiertää tai venyttää hiirellä yhtä kättä, muut kädet tukevat ja rajoittavat viivan venymistä ja vääntymistä.

Tässä meillä on ratkaisu, joka on toteutettavissa nyt! Se perustuu käyttäjälle hyvin tuttuihin käsitteisiin: käsiin. Tämä yksinkertaisuuden hakemisessa tarvittiin katseen kohottamista nykyisistä ratkaisuista.

Tämän esimerkin tarkoituksena ei siis ollut keksiä uutta piirtämishjelmaa vaan havainnollistaa käytettävyyvision käyttöä suunnittelussa: Ensin siis loikataan niin kauas kuin uskalletaan ja osataan. Sitten askeltaan takaisin mahdollisen piiriin.

Tässä esimerkissä käytettävyyvisiota käytettiin suorakäsittelysovelluksen käyttöliittymän ratkaisumallin suunnitteluun. Käytettävyyvisio voi olla hyödyllinen väline myös palvelusovellusten kehittämisessä. Niiden kohdalla käytettävyyvisio koskee kuitenkin usein koko liiketoimintaprosessin uudelleenrakentamista niin, että siitä tulee käyttäjälle mahdollisimman yksinkertainen. Tällainen käytettävyyvisio toteutetaan pääasiassa systeemisuunnittelun tasolla. Käyttöliittymän suunnittelussa se on taustalla, mutta ei suoraan mukana suunnittelussa.

Keskustelukäytön periaatesuunnittelu

Keskustelukäyttöisten sovellusten periaatesuunnittelussa käyttöliittymän jaottelu toiminnallisiin osiin on keskeinen asia. Käyttäjä näkee tämän jaottelun tulokset sovelluksen valikkorakenteissa. Sovelluksen ja valikkojen

rakenteet pitäisi siis rakentaa käyttäjän käsitteiden ja tavoitteiden mukaisiksi – eikä esimerkiksi toteutusteknisten käsitteiden mukaisiksi.

Sovelluksen jaottelun lisäksi käyttöliittymän suunnittelijat kohtaavat paljon pieniä ja keskikokoisia asioita, jotka on toteutettava yhdenmukaisella ja johdonmukaisella tavalla.

Taitaa olla mahdotonta antaa tyhjentävää listaa niistä asioista, jotka käyttöliittymässä pitäisi standardoida tai ohjeistaa. Eri tyyppisissä sovelluksissa standardoidaan erilaisia asioita.

Standardoimisen mahdollisuuksia ja ohjeistamisen mahdollisuuksia tulee sovelluksen kehitystyössä jatkuvasti esille. Projektin vetäjän tehtäviin kuuluu järjestää näiden asioiden kerääminen sekä päätösten valmistelu ja vahvistaminen.

Seuraavassa on esimerkkinä joitain alueita, joilla käyttöliittymän suunnittelussa voidaan tarvita yhtenäisiä periaatteita.

Visuaalinen yleisilme on ensimmäinen asia, jonka sovelluksen uusi käyttäjä kohtaa. Visuaalisen viestinnän ammattilaiset osaavat laatia ja ohjeistaa sellaisen ulkoasun, jonka imagolliset viestit tukevat yrityksen viestintätavoitteita. Epäyhtenäinen ilme viestii laadun puutetta – vaikka sovellus olisi muuten kuinka hyvin suunniteltu.

Terminologian standardointi on tärkeää. Tarvitaan standardit muun muassa

- käsiteltävien tietojen nimistä, sekä
- tapahtumien ja toimenpiteiden nimistä.

Tiedon esitystapoja koskevia standardeja tarvitaan esimerkiksi

- formatoinnista ja maskeista,
- värin käytöstä ja
- fonteista.

Vakiomuotoisilla dialogeilla hoidetaan saman tyyppisinä toistuvia tehtäviä. Tietojen haussa ja selauksessa käytettävät dialogit ovat tästä tyyppinen esimerkki. Standardoitavia asioita ovat mm.

- dialogin toimintatavat,
- valikkojen rakenne,
- tietojen sijoittelu ja ulkoasu,
- komentopainikkeiden sijoittelu ja ulkoasu sekä
- oiko- ja pikanäppäinten käyttö.

Käyttöliittymien kehitystyön eteneminen

Käyttöliittymien kehittäminen etenee syklisesti:

Suunnitteluvaiheessa suunnitellaan jokin osa tai osia käyttöliittymästä. Suunnittelun tulokset kuvataan tai toteutetaan sellaiseen muotoon, jossa niitä voidaan testata ja arvioida.

Palautevaiheessa suunnittelun tuloksista hankitaan projektiryhmän ulkopuolista palautetta

- pyytämällä arvio käyttöliittymäviestinnän asiantuntijalta, tai
- tekemällä tai teettämällä käytettävyydestejä.

Millaista testausta?

Käytettävyyden testausmenetelmien kirjo on varsin laaja. Esimerkiksi tieteellisesti pätevän periaatteellisen tiedon hankkiminen testaamalla vaatii korkeaa erikoisosaamista. Vihjeiden hankkiminen parannuksia tarvitsvista asioista ei onneksi ole yhtä vaativaa.

Projektinvetäjät kammoavat myöhään projektissa ilmeneviä suuria muutostarpeita. Esimerkiksi valikkojen järjestyksen parantelu ei aiheuta projektin loppupuolellakaan harmaita hiuksia. Suuret käyttöliittymän periaatteita ja arkkitehtuuria koskevat remontit sen sijaan panevat aikataulut ja budjetit venymään ja paukkumaan.

Ongelmia paljastavasta käytettävyydestestauksesta on sitä enemmän hyötyä, mitä aikaisemmin sen tuloksia saadaan. Toimivien prototyyppienkin aikaansaaminen vie aikaa. Siksi testaaminen on parasta aloittaa **pikatesteillä**, joissa käytetään pelkkiä suunnittelun käyttöliittymän kuvia. Käyttöliittymän toiminnallisuutta ei näin päästä juurikaan testaamaan, mutta sopivalla kysymyksenasettelulla voidaan saada monista asioista hyödyllistä informaatiota.

Pikatestin kulku

Pikatestauksessa käytetään käyttöliittymän näköiskuvia. Ne voidaan laatia vanhanaikaisesti piirtämällä esimerkiksi fläppitaululle, tai sitten ne voidaan laatia käyttämällä jonkin ohjelmointivälineen lomake-editoria sekä tarpeen mukaan kuvaeditoria. Tietokoneella tehty käyttöliittymän kuva voidaan heijastaa projektorilla valkoiselle tussitaululle.

Koekäyttäjälle kerrotaan ensin testin tarkoitus. Hänelle korostetaan, että tässä ei testata hänen taitojaan vaan suunnitellun ohjelman ymmärrettävyyttä. Sitten hänelle kerrotaan lyhyt – mieluiten vain muutaman lauseen mittainen – taustatarina, joka kertoo, että miksi hän on juuri nyt käyttämässä tietokonetta ja tätä sovellusta.

Sitten käyttäjälle esitellään lähtötilanne. Se selostetaan sovellusalueeseen liittyvillä käsitteillä. Esimerkiksi nettikaupan testissä voidaan kertoa, että olet löytänyt tuoteluettelosta Domino-keksit. On kuitenkin varottava millään lailla kuvaamasta niitä ohjelman käytön toimenpiteitä, joilla lähtötilanteeseen on tultu.

Koekäyttäjälle kerrotaan tehtävä – kuten esimerkiksi ”Osta kolme pakettia Domino-keksejä”. Tehtävän annossa on varottava antamasta vihjeitä keinoista. Koehenkilölle ei esimerkiksi pidä sanoa, että ”Valitse ostoskoriisi kolme pakettia Domino-keksejä”. Tämä antaisi jo vihjeitä sekä välineistä (ostoskori) että toimenpiteistä (valitse ostoskoriin...).

Koekäyttäjältä kysytään, että miten hän lähtisi toteuttamaan tehtävää. Häntä kehoitetaan tutkimaan käyttöliittymää kaikessa rauhassa.

Jos käyttäjä valitsee ensimmäisen toimenpiteen oikein, kokeen johtaja vaihtaa seinälle uuden kuvan, jossa on hänen toimenpiteittensä tulokset. Jos käyttäjä ei valitse oikeaa toimenpidettä, häneltä voidaan kysyä, että miksi hänestä tuntui, että valittu toimenpide oli oikea. Häneltä voidaan myös kysyä, että harkitsiko hän jotain muuta toimenpidettä. Jos sekään ei ole oikea, häneltä voidaan pyytää vielä yhtä arvausta. Viimeistään kolmen arvauksen jälkeen on syytä mennä eteenpäin, ettei käyttäjä stressaannu tilanteesta.

Joissain tapauksissa kannattaa auttaa käyttäjä oikean langan päähän. Jos hän pystyy seuraamaan tätä lankaa vaivatta, tiedetään, että ongelma on vain langan pään esille asettamisessa. Jos esimerkiksi ollaan testaamassa työkaluihin perustuvaa suorakäsittelysovellusta, käyttäjälle voidaan osoittaa oikea työkalu. Sitten kysytään, miten hän suorittaisi tällä työkalulla annetun tehtävän.

Testissä edetään seuraavaan vaiheeseen joko niin, että käyttäjä valitsee oikean toimenpiteen tai niin, että testin ohjaaja kertoo tämän toimenpiteen. Sitten testin johtaja vaihtaa seinälle toimenpiteen jälkeistä tilannetta esittävän käyttöliittymän kuvan. Se toimii lähtötilanteena seuraavan toimenpiteen valinnalle.

Pikatestin pitäisi testata vain yhtä asiaa. Testit ovat tyypillisesti vain parin kolmen toimenpiteen mittaisia. Tavoitteeseen vievä prosessi voidaan hyvin keskeyttää jonkin osatehtävän loppuun. Esimerkiksi nettiruokakaupan testissä tämä yksi testattava asia voisi olla tuotteen poimiminen ostoskoriin. Testin lopuksi käyttäjältä voidaan kuitenkin kysyä vielä käsitystä siitä, että miten edettäisiin.

Esimerkkimme ruokakaupassa käyttäjältä voisi vielä kysyä, että ”kun olet saanut ostokset kerättyä, mikä on seuraava toimenpide?” Tässäkin on oltava varovainen, ettei kysymykseen sisälly vihjeitä. Kysymyksessä ei pidä puhua tilauksen vahvistamisesta, kassalle menosta tms.

Kun testitehtävä on suoritettu, käyttäjältä voidaan vielä kysyä hänen tulkintojaan esillä olevista sellaisista asioista, jotka eivät liity tehtävään. Häneltä voidaan esimerkiksi pyytää muutamia nopeita arvauksia

joidenkin valikkovalintojen merkityksistä, tai häntä voidaan pyytää tulkitsemaan joidenkin esillä olevien objektien tilaa.

Jos käyttöliittymän kuva on heijastettu tussitaululle, joitain käyttäjän toimenpiteitä voidaan simuloida tussikynällä. Jos käyttäjän täytyy syöttää jokin tieto numero- tai tekstikenttään, sitä voidaan simuloida kirjoittamalla tieto taululle kentän kohdalle. Erilaisia valintakontrolleja, kuten valintaruutuja ja valintapainikkeita (radionappeja) voidaan myös käsitellä tussilla ja pyyhintälapulla.

Hiirellä osoittamista voidaan simuloida sormella, karttakepillä tai laserosoitimella. Jos käyttöliittymässä on tarkoitus käyttää kelluvia opasteita, testin johtaja voi simuloida niitä suullisesti: kun käyttäjä osoittaa jotain kontrollia, testin johtaja sanoo siihen liittyvän kelluvan opasteen tekstin. Improvisointi ei tässä ole vaarallista, kunhan lausutut ohjeet pidetään lyhyinä – korkeintaan kolmen sanan mittaisina.

Testin valmistelu

Hanki ainakin kolme **koehenkilöä** per testi. Valmiiden käyttöliittymien ongelmien etsinnässä käytettävissä laajemmissa testeissä optimaalinen koehenkilöiden määrä on yleensä viisi tai kuusi. Pikatestissä sen sijaan tutkitaan, että ovatko käyttöliittymän perusajatukset käyttäjälle ymmärrettäviä. Jos ne eivät ole sitä, tämä alkaa näkyä jo muutaman koehenkilön jälkeen.

Laajemmissa testeissä koekäyttäjät pyritään löytämään todellisesta käyttäjäpopulaatiosta. Pikatestien idea on, että niitä voitaisiin tehdä suunnittelun alkuvaiheessa melko nopeasti toteutettuina niin useasti kuin on tarpeen. Niinpä koehenkilöiden helppo ja nopea saatavuus on niin tärkeää, että heidän edustavuudestaan on ehkä syytä joiltain osin tinkiä. Mukana testauksessa on syytä olla ainakin joitain varsinaisen käyttäjäpopulaation edustajia. Heitä voi kuitenkin täydentää esimerkiksi sellaisilla työtovereilla, jotka eivät ole tekemisissä projektin kanssa.

Pikatestauksessa koekäyttäjää voi myös jonkin verran kierrättää, kunhan heidän uusiokäyttönsä ei kohdistu samoihin toimintoihin, joita he ovat aikaisemmin käyttäneet.

Suunnittele ensin testitehtävä: mikä on lähtötilanne ja mitä käyttäjän pitäisi saada aikaan. Pikatestissä tavoitteen pitäisi olla korkeintaan parin kolmen toimenpiteen päässä. Tavoitteen pitäisi olla sellainen, että käyttäjän täytyy siihen päästäkseen tulkita oikein käyttöliittymässä esillä olevia asioita.

Laadi tehtävälle lyhyt **taustatarina**.

Laadi suunnitellusta **käyttöliittymästä näköiskuva** joko vanhanaikaisesti piirtämällä tai sitten käyttämällä jonkin ohjelmointivälineen lomake-editoria sekä tarpeen mukaan piirtämiseen tarkoitettua kuvaeditoria, kuten Corel Draw'ta tai Powerpointia.

Jos tehtävässä tarvitaan useampia ikkunoita, tee niistä kaikista näköiskuvat. Jos tehtävän suorittaminen vaatii valikkovalintoja, tee myös kuvat, jossa tarvittava valikko on avattu.

Laadi **lähtötilanteen kuvaus**.

Laadi **lista kysymyksistä**, jotka esitetään koekäyttäjälle tehtävän suorittamisen jälkeen.

Testin kirjaus ja analyysi

Laajemmissa käytettävyytstesteissä koko testin kulku tallennetaan yleensä videolle. Näin voi tehdä pikatesteissäkin, mutta yleensä pikatestit pyritään toteuttamaan mahdollisimman kevyellä koneistolla. Kaikkein kevyin ratkaisu on tehdä muistiinpanot muistinvaraisesti heti testin jälkeen. Pikatestien pitäisi olla niin lyhyitä, että tämä tapahtumien kirjaaminen välittömästi jälkikäteen ei rasita muistia kohtuuttomasti.

Pikatesti etenee kysymysten ohjaamana kapeassa putkessa. Niissä haetaan vastausta oikeastaan vain yhteen kysymykseen: kuinka helposti käyttäjä pääsee sisälle käyttöliittymän käsitelmalleihin. Tämä näkyy siten, että käyttäjä tulkitsee oikein käsiteltävien objektien tilat sekä

toimintamahdollisuutensa. Tätä voi arvioida esimerkiksi seuraavalla skaalalla:

1. Käyttäjä ei kyennyt hahmottamaan tilannetta ja toimintamahdollisuuksia.
2. Käyttäjä tarvitsi useimmissa askelissa merkittävästi ohjausta.
3. Käyttäjä löysi oikeat toiminnat ja tulkinnat enimmäkseen vasta toisella tai kolmannella yrityksellä.
4. Käyttäjä tarvitsi alkuun vihjeen, mutta valitsi sen jälkeen oikeat toimenpiteet itsenäisesti ja nopeasti
5. Käyttäjä valitsi oikeat toimenpiteet itsenäisesti ja nopeasti.

Jos näin arvioitujen koetulosten keskiarvo on lähempänä kolmesta kuin nelosta, käyttöliittymään kannattaa todennäköisesti etsiä uudenlaista ratkaisumallia.

Ketkä tekevät pikatestauksen?

Jos projektin omat suunnittelijat testaavat omia suunnitelmiaan, on olemassa vaara, että he pyrkivät ennemminkin hankkimaan vahvistusta ideoilleen kuin etsimään niistä heikkouksia. Projektin ulkopuolelta tulevat testaajat pystyvät tunteettomammin paneutumaan heikkouksien etsimiseen. Ulkopuolisen testaajan käyttö puoltaa siis paikkaansa projektin jossain vaiheessa. Eräät asiat puoltavat kuitenkin sitä, että suunnittelun alkuvaiheen pikatestit suorittaisi kuitenkin pääasiassa projektin oma väki.

Pikatestejä pitäisi tehdä melko paljon. Kun jotain on luonnosteltu, sitä pitäisi päästä nopeasti testaamaan. Suunnittelu- ja testikierrosten pitäisi seurata toisiaan välittömästi. Tämä on paljon helpompaa, jos testaajat on omasta takaa, ja koehenkilötkin saadaan pääasiassa lähipiiristä.

Kannattaa kuitenkin harkita ulkopuolisen asiantuntijan ottamista silloin tällöin mukaan pikatestin suunnitteluun ja suorittamiseen. Hänellä saattaa saada hyödyllisiä uusia näkökulmia ja ohjeita testin suunnitteluun ja suorittamiseen sekä tulosten tulkintaan.

Testaus ääneen ajattelun menetelmällä

Ääneen ajattelu on yleisin käytettävyydestestauksen menetelmä. Nimitys tulee siitä, että käyttäjää pyydetään videonauhoitettavassa testissä puhumaan jatkuvasti ääneen. Hänen toivotaan kertovan mm.

- mitä hän etsii,
- miksi hän etsii,
- miten hän tulkitsee esillä olevia asioita,
- miksi hän valitsee tekemänsä toimenpiteet.

Toiset ovat parempia ääneenajattelijoita kuin toiset. Koehenkilöiksi pitäisi löytää koehenkilöitä, jotka voisivat olla kehitettävän systeemin tai tuotteen tyypillisiä käyttäjiä. Olisi myös hyvä, jos he olisivat luontaisia supliikkihenkilöitä!

Testin käsikirjoitus

Testille laaditaan kirjallinen käsikirjoitus, joka annetaan käyttäjälle. Seuraavassa on esimerkki siitä, että minkälaisella käsikirjoituksella voitaisiin esimerkiksi testata Microsoftin esitysmateriaaliohjelman Powerpointin perusasioita.

Powerpoint-käytettävyydesti

Yleiset ohjeet

Edempänä on testitehtävien sarja. Kustakin testitehtävästä kerrotaan, *mitä* sinun pitäisi tehdä. Ohjeet eivät kerro, että *miten* sinun pitäisi tehdä nämä asiat. Tarkoituksena on testata ohjelman ymmärrettävyyttä sekä etsiä asioita, jotka tuottavat käyttäjille vaikeuksia.

Puhu työskennellessäsi! Kerro testin ohjaajalle, mitä kysymyksiä pohdit ohjelmaa käyttäessäsi! Kerro, mitä teet ja miksi! Kerro, mitä odotit tapahtuvaksi, jos ohjelma toimii muuten kuin odotit.

- Tällainen ruutu ilmaisee, että kyseessä on suoritettavaa tehtävää koskeva toimintaohje. Kun olet saanut tehtävän tehtyä, merkitse ruutuun rasti.

Tilanne

Olet uuden Wimpaimen tuotemyyntipäällikkö. Käsillä on kokous, jossa sinun pitäisi kertoa ensimmäisen vuosineljänneksen myynnistä.

Sinulla on käytössäsi PC, johon on asennettu testattava ohjelma.

Testitehtävät

Lähtötilanne

Ohjelma on käynnistettynä edessäsi, mutta yhtään esitelmää ei ole auki.

1. Tutustuminen työympäristöön

- Katso läpi sovelluksen päävalikko sekä esillä olevat painikkeet

2. Uusi esitelmä

- Kirjoita uuden esitelmän otsikkokalvon otsikoksi "Wimpaimen myynti odotusten mukainen"

3. Tekstikalvo

- Kirjoita seuraavan kalvon otsikoksi "Kilpailutilanne kiristyy"
- Kirjoita samalle kalvolle seuraavat alakohdat
 - Hillman
 - Uusi X1-malli
 - Uusi versio GL-3:sta
 - Miller
 - Uusittu A2

4. Pylväsdiagrammi

Esitä budjetoidun ja toteutuneen myynnin pylväskaaviona.

- Anna seuraavalle kalvolle otsikoksi "Myyntin kehitys".
- Luo kalvolle pylväskaavio, joka esittää seuraavat tiedot

	Tammi	Helmi	Maalis
Toteutunut	512	567	778
Budjetoitu	500	600	700

- ❑ Lisää kalvolle selitys helmikuun odotettua huonommalle myyntitulokselle. Käytä 18:n pisteen kokoista kursivoitua tekstiä. Kirjoita: "Pohjois-Amerikan edustaja vaihtui".
- ❑ Kytke teksti ja helmikuun pylväät toisiinsa nuolella

5. Kalvojen järjestyksen muuttaminen

- ❑ Vaihda toisen ja kolmannen kalvon paikkoja keskenään.

6. Tulostus

- ❑ Tulosta esitelmä.

7. Lopuksi

- ❑ Esitelmää ei tarvitse tallentaa.

Testin kulun taltiointi

Testit siis videoidaan. Joissain ammattimaisissa käytettävyysslaboratorioissa käytetään kahta kameraa, joista toinen kuvaa näppäimistöä ja toinen ruutua. Nämä kuvat yhdistetään editointivaiheessa.

Jos näppäinten näkyvyydestä tingitään, ratkaisut yksinkertaistuvat huomattavasti. Testi voidaan kuvata yhdellä kameralla, joka kohdistetaan kuvaruutuun tai sen käytössä olevaan osaan.

Saatavilla on myös ohjelmia, jotka tallentavat kuvaruudun tai sen osan tapahtumat suoraan kovalevylle videomuotoon. Myös ääni voidaan taltioida, jos PC:ssä on mikrofoni.

Eräs tällainen ohjelma löytyy osoitteesta www.techsmith.com.

Nämä PC:n videonauhoittimet asettavat kovia vaatimuksia levytilalle ja prosessorin teholle. Testin suunnittelijan täytyy testata, että taustalla pyörivä nauhoitus ei hidasta merkittävästi testattavaa ohjelmaa. Ennen testiä täytyy varmistaa, että levyllä on tilaa pitkäksikin venyvän testin videotallennukselle.

Testien valmistelu

Laadi käsikirjoitus.

Hanki koehenkilöt. Alan kirjallisuus on liikuttavan yksimielinen siitä, että viisi tai kuusi käyttäjää on optimaalinen koehenkilöiden määrä yhdessä testissä. Kuudennen koehenkilön jälkeen uusia ongelmia löytyy enää hyvin harvaksen.

Valmistele testiympäristö. On syytä kiinnittää erityistä huomiota siihen, että systeemin lähtötilanne on kaikilla sama. Jos käyttäjä esimerkiksi muuttaa jotain asetuksia, tai hän tallentaa tietokantaan tai levyille jotain, on alkuperäinen tilanne palautettava ennen seuraavan testin alkua.

Laadi lista kysymyksistä, jotka esitetään tehtävien suorituksen jälkeen.

Testaa testi! Ensimmäisen koehenkilön tarkoitus ei ole testata sovellusta vaan testin toimivuutta. Tämä testin koekäyttö selvittää, että

- ymmärtääkö koekäyttäjä käsikirjoituksen ja muun ohjeistuksen halutulla tavalla, ja että
- testiympäristö ja koejärjestelyt toimivat.

Testin analyysi

Pikatestien tavoite oli todeta, ovatko sovelluksen perusratkaisut ymmärrettäviä. Testaus ääneen ajattelulla tuottaa enemmän yksityiskohtiin painottuvaa tietoa siinä vaiheessa, kun ohjelma on ainakin joiltain osin jo käytettävässä kunnossa.

Testeissä syntyy tyypillisesti useiden tuntien verran videotallennuksia. Koko projektiryhmän ei kannata kahlata niitä kaikkia läpi. Jonkun on syytä poimia nauhoitteista kohdat, joissa käyttäjillä oli ongelmia. Editointilaitteisto tai –ohjelmisto antaa mahdollisuuksia ryhmitellä ja karsia videopätkiä uudelleen. Jos esimerkiksi jossain tietyssä tehtävässä oli usealla käyttäjällä ongelmia, nämä voidaan esittää peräkkäin.

Ennen yhteistä palaveria joku voi tehdä yhteenvetotilastoa eri osatehtävien onnistumisprosentteista ja suoritusajoista.

Testien analyysi on varsin suoraviivaista: katsotaan, että missä käyttö ei suju odotusten mukaisesti. Arvioidaan ongelmien vakavuudet. Priorisoidaan ongelmat ja päätetään niiden korjaamisesta.

Projektin koko suunnittelutiimille voisi olla opettavaista katsoa yhteenvedojen lisäksi ainakin yksi testi kokonaisuudessaan läpi.

Muut testausmenetelmät

Edellä esitetyt menetelmät ovat välineitä käyttöliittymien ongelmien etsimiseen. Testejä voi käyttää muihinkin tarkoituksiin. Esimerkiksi tehokkuuden optimointi on tärkeää, jos kehitetään vaikkapa asiakaspalvelun järjestelmää, jossa toimenpiteisiin menneet ylimääräiset sekunnit kertautuvat ja syövät toiminnan tehokkuutta. Käytettävyydestit voivat auttaa viilaamaan jokaisen turhan katseen siirron ja käden liikkeen.

Tällaisissa testeissä vertaillaan erilaisia vaihtoehtoja. Vertaileminen edellyttää selviä numeroita, jotka kuvaavat erilaisten vaihtoehtojen tehokkuutta.

Tällaisten mittauksiin perustuvien vertailevien testien tekeminen vaatii jo niin korkeaa käytettävyydestestauksen erikoisosaamista, että se jää tämän kirjan aihepiiriin ulkopuolelle.

Aiheesta enemmän kiinnostuneet voivat hakea tietoa esimerkiksi Jacob Nilsenin teoksesta ”Usability Engineering”. Tämä teos teki testaamisen mahdollisuuksia tunnetuksi 90-luvun alkupuolella, ja se on vieläkin lukemisen arvoinen.

Suomenkielisillä markkinoilla Sinkkosen, Kuoppalan, Parkkisen ja Vastamäen teos ”Käytettävyyden psykologia” käsittelee testaamista laajemmin kuin tämä kirja.

Miten hankitaan palautetta?

Mikä sitten on käytettävyydestestauksen ja asiantuntija-arvioiden oikea sekoitussuhde? Tähän ei ole yleispäteviä nyrkkisääntöjä.

Olemme tässä kirjassa nähneet useampia esimerkkejä sellaisista käytettävyyden ongelmista, joihin testaus puree huonosti. Testeissä

käyttäjälle annetaan tarkkaan suunniteltuja tehtäviä. Puutteet käyttäjän toiminnan ymmärryksessä rajaavat myös testien suunnittelua. Jos joitain käytön tilanteita ei ymmärretä, niitä ei osata myöskään testata.

Projektin ulkopuolinen käyttöliittymäviestinnän asiantuntijan voi katsoa käytettävyyttä käyttöliittymän yksityiskohtia laajemmasta näkökulmasta. Hänen täytyy tarkastella käyttöliittymien lisäksi analyysivaiheen tuottamaa tietoa käyttäjän toiminnasta.

Käytettävyydestit tuovat hyvin esiin käyttöliittymien ongelmia. Mutta korkea laatu on muutakin kuin suoranaisten ongelmien poissaolo. Jos tavoitteena on käyttäjälle mieluisa käyttökokemus, täytyy paranneltavaa etsiä sellaisistakin asioista, joihin ei liity suoranaisia ongelmia.

Käyttöliittymäviestinnän asiantuntija voi siis auttaa projektia

- osoittamalla sellaisiakin käytettävyyden ongelmia, jotka tulevat huonosti esille testaamisessa, sekä
- ehdottamalla vielä parempia tapoja tehdä sellaisiakin asioita, joihin ei liity suoranaisia ongelmia.

Tarvitaanko ulkopuolisten tekemiä käytettävyydestejä, jos projekti teettää päteviä asiantuntija-arvioita, ja projektin omalla väellä on ainakin hieman kokemusta pikatestauksesta sekä ääneen ajatteluun perustuvasta testausmenetelmästä?

Vastaus riippuu siitä, kuinka varmoja halutaan olla lopputuloksen laadusta. Ulkopuoliset testauksen asiantuntijat etsivät ja löytävät ongelmia useimmiten tehokkaammin kuin projektin omasta väestä otetut testaajat. Jos sovelluksen käytettävyys vaikuttaa yrityksen menestykseen, kannattaa sovellus testauttaa ulkopuolisilla asiantuntijoilla ainakin kerran projektin kuluessa.

Yhteenvedo

Sovelluksen täytyy tarjota käyttäjälle yhtenäiset ja johdonmukaiset käyttö- ja esitystavat. Ne kirjataan osittain sovelluksen tyylioppaaseen, ja

osittain muihin suunnitteludokumentteihin. Tyylioppaan kehittäminen on tuotteen tai systeemin elinkaareissa jatkuva prosessi.

Suorakäsittelyyn perustuvien käyttöliittymien suunnittelussa täytyy aluksi tehdä suuria periaateratkaisuja. Käyttöliittymävisio on eräs keino etsiä ideoita näihin ratkaisuihin.

Käyttöliittymien kehitystyössä tarvitaan suunnittelu- ja toteutustyön lisäksi palautetta. Sitä saadaan käytettävyytestauksesta ja/tai käyttöliittymäviestinnän asiantuntijoiden arvioinneista. Näiden palautemuotojen oikea sekoitussuhde täytyy etsiä tapauskohtaisesti.

Suunnittelun alkuvaiheissa voidaan pikatestauksella hankkia nopeaa palautetta ratkaisujen ymmärrettävyydestä. Kun sovellus alkaa olla ainakin joiltain osin käyttökunnossa, ääneen ajatteluun perustuva testimenetelmä voi antaa tietoa käyttöliittymän yksityiskohtien ymmärrettävyydestä ja toimivuudesta.