

4

Työskentelytavat kuntoon

Tässä luvussa etsimme sovelluskehityksen johtamisen ja käytettävyyssasioiden välistä suhdetta. Katsomme ensin asioita sovelluskehitysyksikön johtamisen näkökulmasta ja sitten projektin johtamisen näkökulmasta.

Sovelluskehitystä tekevän yksikön vetäjän tehtäviin kuuluu kehittää yksikkönsä kykyä tehdä käyttäjätavallisia sovelluksia. Hänen täytyy:

- kehittää yksikkönsä yleistä tietämystä ja ymmärrystä käytettävyyden asioista,
- värvätä ja/ tai kouluttaa yksikköönsä käytettävyyden erikoisosaajia,
- tarvittaessa luoda yhteistyösuhteita ulkopuolisten käytettävyyden asiantuntijoiden kanssa, sekä
- edistää sellaista toimintakulttuuria, jossa ihmiset osaavat ja haluavat hyödyntää toisten ihmisten näkemyksiä ja palautetta.

Käytettävyydestä kasvatetaan kaikkien osaamista

Viestinnässä onnistuminen vaatii monenlaisia taitoja ja kykyjä. Tärkein näistä taidoista on kyky katsoa omaa viestintää muiden ihmisten silmin. Hyvä viestinnän tekijä osaa tavallaan astua itsensä ulkopuolelle pohtiessaan, minkälaisilla eri tavoilla viesti voidaan kokea ja ymmärtää.

Toisilla on tähän paremmat luontaiset taipumukset kuin toisilla. Kyseessä on kuitenkin taito, jossa kaikki voivat kehittyä paremmiksi. Käytettävyyden testaukseen osallistuminen on tähän ylivoimaisen hyödyllinen keino.

Edellisessä luvussa totesimme, että lehden teon eri osa-alueiden osaajat tuntevat toistensa osaamisalueita sen verran, että he osaavat hyödyntää tehokkaasti toistensa osaamista. Esimerkiksi graafiset suunnittelijat ymmärtävät jutun rakentamisen ja dramaturgian kysymyksiä, ja toimittajat puolestaan ymmärtävät, miten typografia ja grafiikka voivat auttaa tarinan kertomista.

On tietenkin toivottavaa, että käyttäjätavallisten sovellusten kehittämisessä päästään samanlaiseen hyvään yhteistyöhön kuin lehtialalla. Tämä edellyttää yleensä, että nostetaan kaikkien sovelluskehitykseen osallistuvien tietämystä käytettävyyssasioista.

Tämä kirja edustaa yhden henkilön kirjoitushetken näkemystä kaikille sovelluskehittäjille tarpeellisen yhteisen perustietämyksen sisällöstä. En kuitenkaan kehota ketään uskomaan kritiikittä, että sovelluskehityksen osapuolten pitäisi hallita juuri nämä ja vain nämä asiat. Sovelluskehitysyksikön käytettävyyssasiantuntijan täytyy muodostaa itselleen oma perusteltu käsitys siitä, mitä hänen työtoveriansa pitäisi tietää käytettävyydestä. Hänen täytyy myös kertoa sovelluskehityksen johdolle, miten sovelluskehittäjät koulutetaan aiheeseen esimerkiksi

- alan julkisilla kursseilla,
- konsulttien pitämällä yrityskohtaisesti räätälöidyillä kursseilla,
- yrityksen sisäisissä koulutustilaisuuksissa sekä

- käytettävyydesteihin osallistumisella.

Käytettävyyden osaajien täytyy ymmärtää systeemisuunnittelua

Käytettävyys on erikoisosaamisen alueena nuori, vaikka sillä onkin vankkoja juuria monilla vanhemmilla aloilla. Ensimmäiset juuri tälle alalle koulutetut asiantuntijat ovat tulleet työelämään 90-luvulla. Tätä ennen alalle on tullut pioneereja monilta eri aloilta. Esimerkiksi tulijat käyttäytymistieteiden piiristä ovat tuoneet mukanaan käytettävyyden testaamisen menetelmien osaamista sekä paljon havaitsemista ja ajattelua koskevaa tutkimustietoa. Tulijat graafisen suunnittelun sekä kirjoitetun viestinnän piiristä taas ovat tuoneet mukanaan muun muassa vuosisatojen aikana kertynyttä painetun viestinnän valtavaa osaamispäätömaa.

Jo ennen käytettävyyden oppien ja käytettävyydestauksen yleistymistä monet atk-suunnittelijat ja sovelluskehittäjät ovat erikoistuneet käyttöliittymäviestinnän kysymyksiin. Tämä on usein tapahtunut hankkimalla omatoimisesti tietoa mm. psykologian ja viestinnän tutkimustuloksista sekä graafisen suunnittelun ja typografian periaatteista.

Monenlaisista lähtökohdista voi kehittyä hyviä käytettävyyden osaajia ja asiantuntijoita. On aivan luonnollista, että samanlaisista lähtökohdista tulevat muodostavat keskenään alalle koulukuntia. On myös luonnollista, että näiden koulukuntien välillä on painotuseroja menetelmissä ja prioriteeteissa. Mutta jos joku väittää omaa koulukuntaansa ainoaksi oikeaksi ja autuaaksi tekeväksi, tällaiset puheet kannattaa jättää omaan arvoonsa. Käytettävyys on niin monitahoinen asia, että siihen ei ole kenelläkään yhdellä henkilöllä tai edes yhdellä koulukunnalla kaikkia tarvittavia eväitä. Tärkeintä on kyky yhteistyöhön!

Tässä kirjassa painotetaan systeemisuunnittelun laadun vaikutusta käytettävyyteen. Käytön suunnittelun ja systeemisuunnittelun välillä on hyvin tärkeä rajapinta, joka on saanut alan kirjallisuudessa vain vähän huomiota. Tämä rajapinta täytyy saada toimimaan hyvin. Mitä paremmin

käyttöliittymäviestinnän ja käytettävyyden osaajat ja asiantuntijat ymmärtävät systeemisuunnittelun käsitteitä ja työtapoja, sitä paremmin he osaavat kommunikoida systeemisuunnittelijoiden kanssa ja kysyä heiltä hyviä ja oikeita kysymyksiä. Tällaisten valmiuksien kehittäminen saattaa vaatia sovelluskehityksen johdon huomiota.

Uhkina ego ja arvovalta

Sovelluskehityksen vetäjän suurin haaste on luoda sellainen toimintakulttuuri, jossa ihmiset kykenevät hyödyntämään toistensa osaamista ja panosta ilman, että egot ja arvovaltakysymykset pääsevät väliin.

Kun olen lehtialalla osallistunut jutuntekijöiden koulutukseen, meitä on neuvottu pitämään egomme irti jutunteossa. Tekijän pitäisi suhtautua kirjoittamaansa tekstiin kuin raaka-aineeseen, joka lähtee tuotantolinjalla eteenpäin. Ei pidä ottaa itseensä, jos teksti sitten muuttuu edetessään tuotantoprosessissa.

Sovelluskehityksen prosessit ovat erilaisia kuin lehdenteon, mutta niihinkin on parasta olla sotkematta egoaan. Sovellukset ovat viestintä-tuotteita, ja viestinnässä on kyse ajatusten välittämisestä toisen ihmisen pääkoppaan. Niinpä kaikki palaute, minkä tekijä voi saada toisten ihmisten pääkopasta, voi auttaa häntä tekemään parempaa viestintää.

Silloinkin, kun muut esittävät selvästi vääriä tai huonolta tuntuvia ehdotuksia, niiden takana saattaa olla jokin aivan mielekäs huolenaihe, joka kannattaa kaivaa esille.

En halua tällä ollenkaan sanoa, että asioista pitäisi päättää jonkinlaisella huutoäänestyksellä. Useimmiten on parasta, että yksi henkilö kerää ajatuksia ja näkemyksiä monilta tahoilta, ja muodostaa sitten näistä aineksista ratkaisukokonaisuuden. Käytön suunnittelulle kannattaa nimetä vastuuhenkilö, jolla on hyvät tiedot käyttöliittymien suunnittelusta sekä paras mahdollinen ymmärrys käyttäjien tarpeista. Sitten muiden täytyy luottaa siihen, että hän on todella paras henkilö tekemään juuri näitä ratkaisuja. Mitä turvatumaksi hän tuntee

päättäjävaltansa, sitä helpompaa hänelle on hyödyntää muiden ihmisten ajatuksia.

Lehtialalla lähdetään siitä, että kukaan ei saa kertalaakista kaikkea kohdalleen. Jos jutun työstäjät löytävät siitä parantelemisen mahdollisuuksia, kenenkään mieleen ei juolahtaisi syyttää alkuperäisen tekstin tekijää ammattitaidon puutteesta.

Kun sovelluksen käytettävyys paranee työstämisen ja testauksen myötä, siitä ei myöskään pidä vetää sellaista johtopäätöstä, että se olisi alun perin tehty ammattitaidottomasti. Tekijöiden ammattitaitoa voidaan arvioida vasta työstämisen lopputuloksen perusteella.

Oikea asenne tähän prosessiin on myös osa ammattitaitoa! Yksikön ilmapiirille ja työn tuloksellisuudelle on myrkkyä, jos joku käyttää työstämisen tuloksia työtoveriensa ammattitaidon mustamaalaamiseen.

Omien ajatusten kyseenalaistaminen ja meriittien etsiminen muiden ajatuksista ei ole useimmille ihmisille luonteenomainen tapa toimia. Aika harvat ihmiset ovat tässä suhteessa kypsiä aloittaessaan työuraansa.

Mikään ei opeta viestinnän tekijälle nöyryyttä niin kuin oman viestinnän tulosten tutkiminen. Voi esimerkiksi olla silmiä avaava kokemus lukea jonkun toisen muistiinpanoja siitä, mitä olet jossain kokouksessa hänen mielestään sanonut.

Käytettävyyden testaus on parasta kasvatusta sovellusten tekijöille. Hyvä testi voi antaa hyvin konkreettista palautetta käyttöliittymäviestinnän onnistumisesta.

Tyylioppaat ja muu ohjeistus

Alan kirjallisuudessa käsitellään usein niin sanottuja **tyylioppaita**. Niiden tarkoitus on toimia jonkinlaisena käyttöliittymän suunnittelun pysyväsuhjeistona.

Tyylioppaan pitäisi kertoa, mikä on juuri tälle tuotteelle tai systeemille ominaista. Alan kirjallisuudessa tyylioppaisiin neuvotaan usein sisällyttämään myös hyvien käyttöliittymien yleisiä periaatteita.

Sovelluskehitysyksiköllä täytyy toki olla suunnitelma siitä, miten nämä yleiset periaatteet koulutetaan sovelluskehittäjille. Niitä ei kuitenkaan kannata kerrata tyylioppaassa. Siitä ei pidä tehdä käyttöliittymien suunnittelun ja laatimisen oppikirjaa! Sinne ei kannata kirjata sellaisia asioita, jotka löytyvät alan kirjallisuudesta.

Tyylioppas on yleensä tuote- tai systeemikohtainen. Yrityskohtaiselle tyylioppaalle on harvemmin tarvetta. Kun uuden tuotteen tai systeemin kehittäminen käynnistyy, sen tyylioppaan pohjaksi voidaan yhdistellä aikaisempien tuotteiden tai systeemien oppaita.

Tyylioppaan ohjeet on syytä jakaa vaatimuksiin ja suosituksiin – ja vaatimusten lista on parasta pitää mahdollisimman lyhyenä! Esimerkiksi sovellusten ulkoasun tunnusomaisien piirteiden noudattamisesta on hyviä syitä pitää kiinni. Tyylioppas ei kuitenkaan saisi kahlita uusien, entistä parempien ratkaisujen kehittämistä ja käyttöönottoa.

Tyylioppaan tekijöille ensimmäinen on vaikein. Aikaa menee usein tuhattomasti oppaaseen sisällytettävien asioiden miettimiseen. Niinpä kannattaa vakavasti harkita, että firman ensimmäisen tyylioppaan laatimiseen osallistuu joku ulkopuolinen asiantuntija, joka on tehnyt niitä aikaisemminkin.

Projektin vetäjän haasteet

Projektin vetäjä kohtaa työssään paljon samoja asioita kuin sovelluskehitysyksikön vetäjä, mutta konkreettisemmassa muodossa.

Sovelluskehityksen vetäjää kiinnostaa osaajien ja asiantuntijoiden rekrytointi sekä yksikön valmiuksien yleinen kehittäminen. Projektin vetäjää kiinnostaa oman projektin miehitys sopivalla ja riittävällä osaamisella.

Sovelluskehityksen vetäjää kiinnostaa viestinnän ja yhteistyön valmiuksien kehittäminen. Projektin vetäjän taas on otettava lopullinen vastuu siitä, että yhteistyö toimii juuri hänen projektissaan niin, että esimerkiksi systeemisuunnittelun tekijät ja käytön suunnittelijat ymmärtävät toisiaan riittävän hyvin.

Projektin alkaessa sen vetäjän on päätettävä seuraavista käytettävyyteen liittyvistä asioista:

- Miten tehtävät ja vastuut jaetaan projektin sisällä?
- Minkälaisilla menettelytavoilla käyttöä suunnitellaan?
- Tehdäänkö käytettävyydesteitä? Kuinka usein? Missä projektin vaiheissa?
- Käytetäänkö projektin ulkopuolista käytettävyyden asiantuntijaa kommentoijana? Kuinka usein? Missä projektin vaiheissa?

Vastuunjaon palapeli

Käytön suunnittelussa tarvitaan niin monenlaisia taitoja, että ne kaikki eivät yleensä mahdu yhden ihmisen päähän:

Sovelluksen yleisilme viestii aina mielikuvia ja imagoa. Visuaalisen viestinnän osaajia (graafisia suunnittelijoita) tarvitaan luomaan sovellukselle yhtenäinen ilmiä, joka viestii yrityksen viestintätavoitteiden mukaista imagoa.

Typografialla voidaan ohjata käyttäjän katsetta sekä jäsentää käyttöliittymän sisältöä. Aihe kuuluu graafisten suunnittelijoiden osaamisalueisiin, mutta typografian perustiedot auttavat sovelluskehittäjiä tekemään yhteistyötä graafisten suunnittelijoiden kanssa. Luku 12 kertoo, mitä jokaisen sovelluskehittäjän on hyvä tietää typografiasta.

Käyttöliittymän toiminnan suunnittelun opit nojaavat osin Donald Normanin kirjan ”The Psychology of Everyday Things” lähtökohtiin. Sovellusten käyttöliittymien erityiskysymyksistä on myös hyvin paljon tutkimustietoa. Monet käyttöliittymäviestintään erikoistuneet sovelluskehittäjät hallitsevat nämä asiat, samoin kuin monet alalle erikoistuneet graafiset suunnittelijat. Norman itse on psykologi – niinpä alalle erikoistuneet psykologit ovat myös näistä asioista perillä.

Käytettävyyden testaus antaa sovelluksen kehittäjille kaikessa viestinnässä tärkeää palautetta. Testaus on erikoisala, jossa voi kehittyä vaikka kuinka pitkälle. Osaamiskynnys hyödylliseen testaukseen ei kuitenkaan ole ylettömän korkea. Siitä huolimatta useimmissa projekteissa kannattaa käyttää testaamiseen (myös) ulkopuolista asiantuntijaa. Omia aikaansaannoksiaan testaava projektiryhmän jäsen ei ehkä etsi paranneltavaa samalla antaumuksella kuin projektin ulkopuolelta tuleva testaaja.

Systeemisuunnittelu tuottaa käytön suunnittelussa tarvittavan ymmärryksen käytön tilanteista. Jos eri henkilöt tekevät systeemisuunnittelua ja käytön suunnittelua, heidän täytyy kyetä hyvään yhteistyöhön. Mitä paremmin he ymmärtävät toistensa työtä, sitä paremmat mahdollisuudet heillä on luoda käyttäjätavallinen sovellus. Myös onnistunut ja hyödyllinen käytettävyydestestaus nojaa siihen käyttötilanteiden ymmärrykseen, jonka systeemisuunnittelu tuottaa.

Kirjoittajia tarvitaan, jos käyttöliittymään sisältyy yhtään informatiivista ainesta, kuten esimerkiksi tuotteen tai palvelun kuvauksia tai toimintaohjeita. Ammattitaitoiset kirjoittajat tuottavat selkeästi luettavaa tekstiä. Jos kuitenkin käyttöliittymän halutaan markkinoivan esimerkiksi jotain tuotetta tai palvelua, projekti tarvitsee markkinointikirjoittamisen osaajaa.

Näiden osaajien lisäksi tarvitaan

- suunnitteluun osallistuvia käyttäjien edustaja tai edustajia,
- sovellusalueen asiantuntijoita, sekä
- käyttäjien edustajia koehenkilöiksi käytettävyydesteihin.

Suunnitteluun osallistuvat käyttäjien edustajat eivät kelpaa käytettävyydestien koehenkilöiksi, koska he omaksuvat työn kuluessa niitä suunnittelijoiden ajatusmalleja, joiden oikeellisuutta käytettävyydestien pitäisi tutkia. Heidän tehtävänsä on auttaa suunnittelijoita löytämään käyttöliittymäviestinnälle oikea sisältö. Koehenkilöt taas auttavat tarkistamaan, että tämän sisällön esitystapa toimii hyvin.

Eri henkilöillä saattaa olla erilaisia yhdistelmiä näistä taidoista. Myös kunkin henkilön osaamisen taso on huomioitava, kun projektin mieltymyksen palapeliä kasataan.

Tarvitaan tuoreita silmiä!

Lehdenteossa vastuu jutusta siirtyy työstämävaiheessa tekijältä toimittajalle – viestinnän ammattilaiselle. Tekijä saa kyllä yleensä tarkistaa, ettei lopulliseen tekstiin ole työstämisen yhteydessä pujahtanut asiavirheitä. Vastuu jutun lopullisesta asusta on kuitenkin selkeästi työstäjällä.

Huono lehtijuttu ei yleensä vaikeuta merkittävästi lukijoiden elämää. Mutta sovelluksen virheet ja puutteet vaikeuttavat aivan varmasti käyttäjien elämää! Yritysten toiminta ja ihmisten työ riippuu usein siitä, että sovellukset toimivat täsmälleen oikein ja kattavat käyttäjän tarpeet. Niinpä lehden teon toimintamalleja ei voi aivan sellaisenaan kopioida sovelluskehityksen maailmaan. Sovellusta tehtäessä vastuun ja päätäntävällän pitäisi olla siellä, missä on paras ymmärrys niistä tilanteista ja tarpeista, joita sovellus palvelee.

Lehdenteossa tarvitaan työstämistä, koska me kaikki olemme jossain määrin sokeita oman viestintämme puutteille. Tuoreet silmät näkevät heti sellaisia asioita, joita tekijä ei huomaa.

Tuoreet silmät voivat auttaa myös käyttöliittymäviestinnän tekemistä. Projektin suunnittelutyöhön osallistumaton asiantuntija voi tarjota projektin käyttöön hyödylliset tuoreet silmät. Hän pystyy varmasti tekemään parannusehdotuksia. Hän kuitenkin tietää yleensä käyttäjien tarpeista vähemmän kuin systeemisuunnittelijat ja/tai käytön suunnittelijat. Niinpä hän ei ole oikea henkilö tekemään käyttöä koskevia päätöksiä. Hänen tehtävänsä on auttaa käytön suunnittelun vastuuhenkilöä omilla huomioillaan ja ehdotuksillaan.

Monet projektinvetäjät joutuvat pohtimaan, tarvitaanko systeemisuunnittelijan ja graafisen suunnittelijan väliin käyttöliittymäsuunnittelijaa. Tiedon siirtäminen yhdestä päästä toiseen

lisää tietenkin aina kitkaa projektin työskentelyyn. Mitä pitemmälle käyttäjien tarpeiden paras asiantuntija pystyy viemään käytön suunnittelun, sitä vaivattomammin projekti yleensä sujuu.

Muunkinlaiset työnjaot saattavat toimia hyvin. Esimerkiksi graafisen suunnittelijan rooli voi olla suurempi, jos hänellä on hyvät tiedot käyttöliittymän toiminnan suunnittelun periaatteista, ja hän kykenee osallistumaan systeemisuunnitteluun.

Protoilu voi viedä hakoteille

Käytön suunnittelu on erottamaton osa systeemisuunnittelua. Nykyisissä suunnittelumenetelmissä korostuvat käyttötilanemallit (use cases) sekä käyttötarinat (scenarios). Ne ovat systeemin tietojen ja toimintojen löytämisen ja havainnollistamisen välineitä.

Käytön havainnollistamiseen tarvitaan jonkinlaisia prototyypppejä. Näiden käytöllä voi olla kolmenlaisia tavoitteita:

- **Systeemin tietosisällön** protoilulla haetaan ymmärrystä siitä, mitä tietoja systeemi ja käyttäjät käsittelevät.
- **Keskustelujen sisällön** protoilulla pyritään siihen, että käyttäjän ja systeemin keskustelujen sisältö on oikea kussakin tilanteessa.
- **Käyttöliittymän protoilulla** pyritään saamaan käyttöliittymän yksityiskohdat kohdalleen.

Nykyiset sovelluskehittimet tarjoavat varsin kehittyneitä välineitä tietosisällön protoiluun. Niillä saa nopeasti ja helposti luotua tietokannan, ja myös tietokannan määrittysten muuttaminen on yleensä helppoa. Oletuslomakkeilla pääsee nopeasti syöttämään tietokannan tauluihin testiaineistoa. Peruslomakkeet tietokannan selaamiseen ja hakuihin sekä tietojen editointiin syntyvät helposti.

Tällaisessa työskentelytavassa on käytettävyyteen liittyviä vaaroja! Olen pariinkin otteeseen tehnyt IT-alan lehdille PC:n kirjanpito-ohjelmien vertailutestejä. Useimmat näistä ohjelmista keskustelivat käyttäjän kanssa

kirjanpidosta kirjapidon käsitteillä. Jotkut ohjelmat kuitenkin keskustelivat käyttäjän kanssa kirjapitotietokannan editoinnista. Niissä luodaan ja poistetaan tietueita sen sijaan että kirjattaisiin tositteita. Myös valikkojen hierarkia ja tietojen jäsenitys noudattavat tietokannan rakennetta, kun niiden pitäisi noudattaa käyttäjän tehtävien rakennetta.

Sovelluskehittäjien käyttäjille on houkuttelevaa tehdä ensin kaikille tietokannan tauluille selauslomakkeet, ja lähteä sitten rakentamaan prototyyppiä parantelemalla askel kerrallaan näitä selauslomakkeita erilaisiin tilanteisiin ja tarkoituksiin. Tämä tie voi kuitenkin johtaa yllä kuvatun kaltaiseen tietokantaeditoriin – ja huonoon käytettävyyteen.

Tietosisällön protoilu etenee yleensä rinnakkain keskustelujen sisällön protoilun kanssa. Kun niiden avulla on saatu selvä käsitys keskustelujen sisällöstä, on aika katsoa tuorein silmin keskustelujen toteutusta. Lopullinen käyttöliittymä voidaan suunnitella vasta, kun keskustelujen tavoitteet ja sisältö ymmärretään täysin. Keskustelujen sisällön selvittämiseksi tehdyt protot eivät saisi hämärtää näkemystä, kun lopullista käyttöliittymää suunnitellaan!

Missä vaiheessa testataan?

Projektin vetäjän on myös otettava kantaa käytettävyydestä: milloin ja kuinka paljon?

Kaikille kokeneille projektinvetäjille on yhteistä halu löytää ongelmat ja virheet mahdollisimman aikaisessa vaiheessa. Kannattaisiko siis käytettävyydestäkin ryhtyä mahdollisimman aikaisessa projektin vaiheessa?

Totesimme juuri yllä, että on parasta ensin selvittää keskustelujen sisältö, ja vasta sitten lähteä suunnittelemaan keskustelujen toteutusta! Testaus soveltuu varmasti hyvin projektin siihen vaiheeseen, jossa keskustelujen sisältö on selvillä, ja sen pohjalta on tehty ensimmäiset paperiluonnokset tai protot keskustelujen toteutuksesta.

Mutta kannattaako testata keskustelujen sisällön selvittämiseksi tehtyjä protoja? Ehkä niitä testaamalla voidaan selvittää keskustelussa

käytettyjen käsitteiden ymmärrettävyyttä sekä tehtävien etenemisjärjestykseen liittyviä asioita. Projektiryhmään kuuluvat käyttäjien edustajat pystyvät kuitenkin usein antamaan näissä asioissa hyvää palautetta ilman testejäkin. Määrittelyvaiheessa testaamiseen pitäisi olla todella hyvät syyt, ja testin työmenetelmät on syytä pitää kevyinä.

Mihin käytettävyydestä soveltuu?

Käytettävyyden testauksella saadaan hyödyllistä ja konkreettista palautetta erityisesti siitä, miten käyttäjät

- tulkitsevat käyttöliittymää sekä
- löytävät tarvitsemaansa tietoja ja toimintoja.

Testaus voi mm. auttaa selvittämään, että

- kuinka helppoa tai vaikeaa on ryhtyä käyttämään sovellusta, ja
- kuinka tehokasta työskentely sovelluksella on.

Kummatkin voivat olla sovellukselle ratkaisevan tärkeitä kysymyksiä. Esimerkiksi nettipankilla pitäisi olla hyvin alhainen oppimiskynnys, koska käyttäjät eivät halua lukea ohjeita hoitaakseen raha-asioitaan. Maksujen hoitamisen tehokkuuden pitäisi olla vanhaa menetelmää parempi, jotta esimerkiksi minä siirtyisin nettipankin käyttäjäksi.

Edellä on useaan kertaan todettu, että projektissa tarvitaan ulkopuolisen asiantuntijan tuoreita silmiä osoittamaan käytön ongelmia ja tekemään parannusehdotuksia. Asiantuntijalta saa usein palautetta nopeammin ja halvemmalla kuin testaamalla. Milloin siis pitäisi testata, ja milloin luottaa asiantuntijan palautteeseen?

Huomattava panostus testaamiseen on paikallaan erityisesti silloin, kun halutaan olla varmoja.

Kirjan toisessa luvussa todettiin, että IT-johdolla pitäisi olla selvä käsitys siitä, mitkä ovat yrityksen menestykselle tärkeitä toimintoja tai käyttökynnyksiä. Projektin vetäjällä pitäisi olla omasta sovelluksestaan vastaava käsitys: mitkä kynnykset on saatava mataliksi, ja mitkä toiminnot on saatava tehokkaiksi.

Käytettävyys ja vaatimusten määrittely

IT-alalla on yleistä, että yritys tilaa itselleen sovelluksen tai jopa kokonaisen tietojärjestelmän joltain ulkopuoliselta toimittajalta. Tilaussopimus perustuu yhteiseen ymmärrykseen siitä, mitä ollaan toteuttamassa. Tämä yhteinen ymmärrys kuvataan dokumentilla, jota on tapana kutsua vaatimusmäärittelyksi.

Vaatimusten yksityiskohtaisuuden taso saattaa vaihdella kovastikin. Joskus vaatimusmäärittely sisältää varsin yksityiskohtaiset kuvaukset käyttöliittymäviestinnästä. Joskus taas pyritään jättämään tilaa toteuttajien ammattitaidolle pitämällä määrittelyt mahdollisimman yleisluontoisina.

Ajatus käytettävyysvaatimuksista voi tuntua houkuttelevalta: määritellään käytettävyyden tavoitteet, ja annetaan toteuttajien päättää, että miten niihin päästään.

Voiko hyvää käytettävyyttä kiteyttää yksiselitteisiin vaatimuksiin? Jos vaaditaan, että käytön täytyy olla vaivatonta ja ymmärrettävää, niin miten voidaan yksiselitteisesti todeta, että vaatimus on täytetty?

Alan kirjallisuus tarjoaa vastaukseksi käytettävyydestä. Se on melko pätevä, mutta hyvin vaativa keino. Sekä käytettävyysvaatimusten määrittelyssä että niiden testaamisessa on parasta käyttää kokeneita testaamisen asiantuntijoita.

Vaatimusmäärittelyssä voidaan esimerkiksi kuvata koekäyttäjille tarjottavat testitilanteet, sekä asettaa onnistumisprosentteja sekä suoritusajoja koskevia vaatimuksia.

- Monet asiat voivat tuottaa ongelmia:
- Testataanko olennaisia ja tärkeitä asioita?
- Miten hyvin testitulokset kuvaavat käytettävyyden todellista tasoa?
- Tuleeko kaikki olennainen ja tärkeä testattua?
- Ovatko vaatimukset kohtuullisia?

Systeemin toimittajalle vaatimusmäärittelyt ovat keino tietää mihin sitoutuu. On erittäin vaikeaa arvioida, mitä vaadittuihin käytettävyystestien tuloksiin pääseminen käytännössä vaatii – tähän ei oikein ole mitään yleisesti tunnettuja ja hyväksytyjä oppeja. Käytettävyysvaatimukset sisältävät siis toimittajalle huomattavia riskejä, jotka on huomioitava käyttämällä hinnoittelussa korkeita varmuuskertoimia.

Käytettävyysvaatimukset ovat niin järeä ja vaativa keino, että valistunut tilaaja käyttää niitä hyvin valikoiden. Tilajalla pitäisi olla selvä käsitys tärkeysjärjestyksestä: mitkä kynnykset on saatava alhaisiksi, ja mitkä toiminnot on saatava tehokkaiksi.

Yhteenveto

Systeemisuunnittelun ja käyttöliittymäviestinnän välillä on käytettävyyden kannalta hyvin tärkeä rajapinta. Valistunut sovelluskehitysyksikön vetäjä edistää viestintää ja yhteistyötä tässä rajapinnassa kehittämällä

- systeemisuunnittelijoiden osaamista käytettävyysasioissa sekä
- käyttöliittymäviestinnän osaajien ja asiantuntijoiden osaamista systeemisuunnittelussa.

Projektin alkaessa sen vetäjän on päätettävä seuraavista käytettävyyteen liittyvistä asioista:

- Miten tehtävät ja vastuut jaetaan projektin sisällä?
- Minkälaisilla menettelytavoilla käyttöä suunnitellaan?
- Tehdäänkö käytettävyystestejä? Kuinka usein? Missä projektin vaiheissa?
- Käytetäänkö projektin ulkopuolista käytettävyyden asiantuntijaa kommentoijana? Kuinka usein? Missä projektin vaiheissa?