



Irmeli Sinkkonen on Adage Oy:n kehittämisjohtaja. Irmeli on käytettävyyden pioneereja Suomessa. Hän myös kouluttaa käyttäjäkeskeisen suunnittelun käyttöä verkkoprojekteissa. Ennen käytettävyytutkimukseen siirtymistään hän on toiminut neljännesvuosisadan suunnittelijana ja suunnittelun esimiehenä.



Janne Tompuri on Adage Oy:n toimitusjohtaja. Hänellä on yli kymmenen vuoden työkokemus verkkosivustojen ja selainpohjaisten sovellusten suunnittelusta ja projektijohtamisesta.

Sovelluksia verkkoon – miten käytettävyyden?

World Wide Web on kokemassa historiallista muutosta siirryttäessä toisiinsa linkitetyistä hypertekstidokumenteista selainpohjaisiin sovelluksiin. Uudet teknologiat, kuten AJAX ja avoimet sovellusrajapinnat, mahdollistavat rikkaan vuorovaikutuksen käyttäjän ja sovelluksen välillä sekä useita tietolähteitä ja toimintoja yhdistäviä sovelluksia. Käytettävyyden kannalta muutos tuo haasteita. Uudet teknologiat rikkovat Webin perinteisen vuorovaikutusmallin. Toisaalta uusi teknologia mahdollistaa käytettävyyttä tukevia ratkaisuja, joiden avulla useita perinteisiä käytettävyyssongelmia voidaan välttää. Verkkosovelluksen käytettävyys voidaan varmistaa hyödyntämällä käyttäjäkeskeisiä suunnittelumenetelmiä.

Sivustoja ja sovelluksia

Verkkoon rakennettavat järjestelmät ovat perinteisesti olleet sivustoja, hypertekstijärjestelmien jälkeläisiä, joiden merkitys on tiedon esittämisessä ja rakennearkkitehtuuri perustuu tuotteen informaation arkkitehtuurille. Verkkosivustojen kehittämis- ja ylläpitovastuu on tyypillisesti jossain muualla kuin yrityksen tai yhteisön tietohallinnolla. Tyypillinen ja luonteva verkkosivustosta vastaava osapuoli on viestintäyksikkö. Viestintävastuulliset ovat ymmärtäneet keskimäärin erittäin hyvin käytettävyyden tärkeyden ja käytettävyysskoulutukseen osallistuvista on arviolta alle 5% IT-ammattilaisia, kun yli 90% on viestintävastuullisia, kuten tuotepäälliköitä ja graafikoita.

Varsin pitkään on osana suurelle yleisöllekin tarkoitettuja verkkosivustoja ollut lyhyehköjä toimintaprosessin pätkiä, suljetuissa verkoissa on vaativampia sovelluksia. Käytettävyytutkijat ovat olleet jo huolissaan siitä, että samalla kun

monessa yrityksessä tuotteiden toimintalogiikka monimutkaistuu, niin käyttöliittymät ”puuroutuvat” ja toimintaidea on kateissa, mikä on aina merkki puutteellisesta käyttöliittymän määrittelytyöstä.

Uusien teknologioiden ansiosta verkkosovellukset saadaan toimimaan lähes yhtä luontevasti kuin työpöytäsovellukset. Käytössä on esimerkiksi sivun osien dynaaminen päivitys, heti tehtävät syötteiden tarkistukset ja objektien raahaus. Verkkoympäristö taas antaa sovellukselle mahdollisuuksia, joita työpöytäsovelluksissa ei ole ollut. Ongelmatilanteisiin voidaan rakentaa parempi tuki, tuotteeseen monipuolisempi ja värikkäämpi käyttöliittymä. Käytettävissä on paremmat tiedon visualisointimahdollisuudet, paremmat yhteydet sovelluksesta toiseen ja monipuolisemmat yhteystavat käyttäjien välillä. Käyttöliittymistä voidaan myös tehdä helposti versiot mobiililaitteisiin ja toteuttaa erikieliset versiot.

Vaikka uusissa verkkosovelluksissa on kyse samasta asiasta kuin missä tahansa sovelluksessa, kuten tiedon säilyttämisestä, syötöstä, tulostamisesta ja päivityksistä sekä näyttämisestä, niin uudet sovellukset ovat entistä rikkaampia, monipuolisempia ja monimutkaisempia. Tämä tarkoittaa sitä, että jos käyttäjä ja käyttäjän tarpeet jätetään sivurooliin tuotekehityksessä, lopputulos voi olla lähes käyttökelvoton.

Usea yritys miettii paraikaa vanhojen työpöytäsovellusten siirtämistä verkkoon esimerkiksi vanhojen sovellusten huonon käytettävyyden takia. Käyttöliittymäsuunnitteluun uudet tekniikat antavat mahdollisuuksia, joita ei ole ennen ollut sivustoissa eikä työpöytäsovelluksissa. Vastassa on nyt kaksinkertainen haaste: miten saada käy-

tettävyyks kuntoon samalla kun sovellukset ja vuorovaikutustavat monipuolistuvat. Suunnittelijalta vaaditaan tämän kaltaisissa projekteissa paljon: asiantuntemusta niin informaatioarkkitehtuurin, käyttöliittymän kuin myös vuorovaikutuksen suunnittelussa.

Prosessi

Tuotekehitysprosessit eivät ole vakiintuneet vuosien saatossa. Viimeisen 40 vuoden kuluessa on kehitetty lukuisia menetelmiä kehittämistyön vakaannuttamiseksi ja teollistamiseksi. Useiden tuotekehitysmenetelmien ongelmat johtuvat puuttuvista lähtötiedoista: ei olla selvillä siitä, miksi tuotetta kehitetään; miksi joku käyttää tai ei käytä tuotetta; millainen tuote sopii tiettyyn työhön tai toimintatapaan. Saati ketkä tuotetta käyttävät tai keiden halutaan sitä käyttävän. Aikataulut eivät pidä, koska kukaan ei tarkkaan tiedä, mitä ja millaisia toimintoja tuotteeseen tarvitaan. Tuotteen toimittaja ei tiedä, paljonko kehitystyöhön kuluu aikaa, koska hänellä ei ole tarkkaa käsitystä tuotteen sisällöstä ja monimutkaisuudesta.

Ketterissä tuotekehitysmenetelmissä on nostettu kädet ylös ja luovuttu ajatuksesta, että tilaaja pysyisi tarkasti tietämään, mitä oikeasti haluaa, jolloin toimittaja vastaavasti tietää vielä vähemmän, mitä on myymässä. Menetelmä perustuu tekemiseen ja tehdyn työn toistuvaan korjaamiseen. Kyse on siis tuotteen rakentamisesta iteroimalla, mutta melko raskaalla ja kalliilla tavalla. Toisaalta ketterien menetelmien avulla hallitaan kehitystyön aikana muuttuvia vaatimuksia ja mahdollisesti nopeutetaan tuotteen valmistumista, mutta mitä vähemmän näitä muutoksia tulee toimintatapaan, sitä nopeampi prosessi on. Pahimmillaan ketterät menetelmät hidastavat tuotekehitystä.

Käyttäjakeskeisessä suunnittelussa ajetaan takaa samaa peikkoa, mutta menetelmä on osittain sama, osittain toinen. Tuote rakennetaan iteroimalla aivan kuten ketterissä menetelmissä, mutta iterointi on huomattavasti kevyempää ja tiheämpää sekä luotettavammin tuotetta eteenpäin vievää. Siinä iteroidaan mahdollisimman kevyin välinein käyttöliittymäprototyyppiä eli verkotuotteen käyttäjälle näkyvää osuutta. Tämä tarkoittaa sekä tuotteen toimintatapaa, rakennetta, terminologiaa että ikkunamalleja. Iterointikierrosten osana evaluoidaan ratkaisut käytettävyyks-tein tai asiantuntija-arvioin. Tällöin jatkotyö ei ole pelkästään tilaajan edustajan osaamisen varassa, vaan perustuu tietoon ja tutkimukseen.

Toinen tavallinen sovellusten kehittämistapa on muokattu vesiputousmalli, jossa tuote tehdään vaihe vaiheelta lähtien liiketoiminnallisista tarpeista. Vesiputousmalli on luonteva tapa edetä tuotekehityksessä. Kannattaa kuitenkin tiedostaa, että tämä edelleen mukana sinnittelevä prosessimalli on kehitetty aikana, jolloin käyttäjän panos

eli syötteet saatiin mukaan määrämuotoisten reikäkorttien avulla ja operaattorit pyörittelivät heksalukuja päässään ja korjasivat ongelmatilanteissa suoraan koneen muistissa olevaa koodia. Malli vaatii käyttöliittymäsuunnittelun osalta suuria muutoksia toimiakseen.

Oli tuotekehitysmenetelmä mikä tahansa, tuotekehitysprosessi kannattaa käytännössä jakaa kahdeksi projektiksi. Ensimmäisessä määritellään ja toteutetaan käyttöliittymäprototyyppi. Toisessa projektissa toteutetaan tuote. Tuote siis ensin konseptoidaan ja varsinainen toteutus tehdään vasta kun kaikilla on selvä näkemys siitä, millainen tuotteesta tehdään. Toteutustyön kustannukset ja kesto kannattaa laskea vasta kun ensimmäinen projekti eli tuotteen käyttöliittymä on valmis, ja kun sekä tilaaja että toimittaja tietävät mistä kauppaa ollaan tekemässä.

Määrittelymenetelmät

Käyttäjakeskeisen sovelluskehityksen menetelmiä on useita. Menetelmät ovat perusrakenteeltaan hyvin samanlaisia, käsittäen seuraavat vaiheet.

1. Käyttäjätiedon keruu
2. Tiedon analysointi, mallintaminen ja tiivistäminen
3. Työn uudelleen suunnittelu
4. Prototyyppien teko ja testaus
 - a. rakennesuunnittelu
 - b. sivusuunnittelu
 - c. vuorovaikutussuunnittelu
5. Visuaalisen suunnittelun yhdistäminen
6. Käytettävyyks-testit
7. Käyttöliittymäohjeistojen ym. tuottaminen

Tunnetuimmat menetelmät eli *Contextual Design* (CD) ja *Scenario-Based Design* (SBD) eroavat lähinnä siinä, että CD muistuttaa analyysimenetelmissään perinteellisempiä määrittelymenetelmiä, kun taas SBD nojaa tekstimuotoisiin skenaarioihin ja niiden vaiheittaiseen kehittämiseen.

On tärkeää, että käyttöliittymän määrittelyn osana selvitetään liiketoiminnasta vastaavien kanssa, ketkä tulevat käyttämään tuotetta ja mitä heidän halutaan sillä tekevän. Seuraavaksi selvitetään millaisia käyttäjät ovat, miten uusi tuote voisi liittyä heidän toimintatapoihinsa, millainen on heidän toimintatapansa nyt ja miten sitä voidaan parantaa, millaisia ongelmia heillä on nyt työssään sekä millaisessa ympäristössä he toimivat ja haluavat toimia. Tämän vaiheen tärkeimmät menetelmät ovat haastattelut, havainnoinnit ja ryhmäkeskustelut.

Selvitykset analysoidaan, päällekkäisyydet puretaan ja asiat ryhmitellään samankaltaisuus-analyysillä, ja merkityksellisimmistä asioista

tehdään tarpeellisia lisäkuvauskuksia. Contextual Design -menetelmässä käytetään tähän flow-mallia, sekvenssimallia, fyysistä ympäristömallia, kulttuurimallia ja artefaktimallia, SBD:ssä käytetään eritasoisia täydentäviä skenaariokuvauskuksia (ongelmaskenaariot → aktiviteettiskenaariot → informaatio- ja vuorovaikutusskenaariot). Näitä malleja ja kuvauskuksia voi täydentää perinteisillä malleilla, mutta on huomattava, että nyt kuvataan käyttäjiä ja käyttäjien tekemisiä, ei rooleja.

Lopuksi

Käyttäjakeskeinen suunnittelu puree sekä tuotteen käytettävyyden parantamiseen, että tuotekehitystyön suurimpien ongelmien karsimiseen. Tämä on johdonmukaista, koska kehitysprosessin viivästyksset ja etenkin yllätyksset ovat seurausta huonosti tehdystä pohjatyöstä. Jos halutaan käyttää ketteriä menetelmiä, on hänenkin, joka vastaa asiakkaan puolella tuotteen kehittämisestä, tehtävä nämä selvityksset ja tutkimuksset – kunnolla.

Käytettävyyden huomioonottamista tuotekehityksessä ”syytetään” siitä, että projektit venyvät entistä pitemmiksi. Käyttäjakeskeisessä tuotekehityksessä käytetään kieltämättä enemmän aikaa määrittelyvaiheeseen kuin ei-käyttäjakeskeisissä. Toisaalta hyvin suunniteltu on jo melkein tehty. Vaihtoehtona on, että osa asioista vain jätetään kokonaan selvittämättä ja suunnittelematta.

Käytännössä projekteihin menee usein jopa vähemmän aikaa, kun tiedetään mitä ollaan tekemässä eikä tehdä liikaa tai liian vähän piirteitä tuotteeseen. Pois jää myös kalliita ja hyödyttömiä väittelyitä siitä, tarvitaanko jotain ominaisuutta vai ei, ja millainen sen olisi oltava. Samoin vähe-nee toimittajan resurssien hukkakäyttö ja turhat muutokset, kun tuotteen toiminta on valmiiksi mietitty. Suunnittelijan ja toteuttajan työn ikävimpiä tilanteita ovat tapauksset, joissa asiakkaan eri edustajat tai osastot antavat keskenään ristiriitaisia tai puutteellisia ohjeita suunnittelijalla ja toteuttajalle.

Käyttöliittymissä monipuoliset mahdollisuudet tarkoittavat käytännössä myös monipuolisia ongelmia. Mitä paremmat mahdollisuudet on rakentaa käyttäjälle hieno käyttäjäkokemus, sitä suuremmat mahdollisuudet on myös epäonnistua pahasti. Esimerkkinä erilaisten tehosteiden, korostusten ja huomiota kiinnittävien animointien ylitsepursuava hyödyntäminen verkkopalvelussa. Nämä tekevät kokonaisuuden sekavaksi ja vievät suunnitteluenergian helposti toisarvoiseen. Uusien tekniikoiden antaessa yhä monipuolisemmin uusia hyödyllisiä mahdollisuuksia alkaa käyttöliittymäsuunnittelu olla yhä enemmän erikoistuneiden suunnittelutiimien työtä.

Mikäli tuotteiden helppokäyttöisyydestä ei huolehdi, käyttäjien eriarvoisuus lisääntyy: koke-neeet käyttäjät pystyvät hyödyntämään uusia ominaisuuksia entistä paremmin ja kokematto-milla voi olla entistä enemmän hankaluksia.

Käyttäjakeskeisiin menetelmiin voidaan yrityk-sissä siirtyä vaiheittain ja ne voivat täydentää niin vesiputousmallia kuin ketteriä menetelmiä. Oleel-lista on kuitenkin tehdä käyttöliittymäsuunnittelu loppuun asti huolellisesti, jolloin projektista tulee ennustettavampi sekä lopputulokseltaan, kustan-nuksiltaan että aikataulultaan.

Lisätietoja artikkelissa viitatuista menetelmistä

Beyer, H., Holtzblatt, K.: Contextual Design: A Customer-Centered Approach to Systems Designs, John Wiley & Sons, New York 1996;

Holtzblatt, K., Wendell, J. Wood, S.: Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design, Elsevier Science & Technology 2004

Carroll, J. Scenario-Based Design: Envisioning Work and Technology in System Development, John Wiley & Sons, 1995; Rosson, M.B.,

Carroll, J: Usability Engineering , Academic Press 2002

CDstä löytyy napakka yhteenveto esimerkiksi http://en.wikipedia.org/wiki/Contextual_design#Work_modeling