

Käyttöliittymälähtöinen vaatimusmäärittely



Sari A. Laakso vastaa käyttöliittymäsuunnittelun ja käytettävyyden ope-
tustarjonnasta Helsingin
yliopiston tietojenkäsit-
telytieteen laitoksella
ja kehittää käyttöliitty-
mälähtöisiä menetelmiä
sekä yliopistolla että
Interacta Designissa.
sari@interacta.fi

Vaatimusmäärittelyn keskeiset osat voidaan saada näkyviksi ja testauskelpoisiksi käyttöliittymän näyttökuvien avulla. Käyttäjien työntekoon perustuva käyttöliittymäsuunnittelu tuo konkretiaa vaatimusten selvittämiseen, ja osa hyväksymistestauksesta saadaan siirrettyä projektin alkuun.

Nykyisten vaatimusmäärittelyjen ongelmia

Ohjelmiston määrittely projektin alussa on usein hankalaa. Kun vaatimuksia on kirjattu tilaa-
jan toiveiden pohjalta ja järjestelmää määriteltä, tilaajan ongelmaksi tulee se, että vaatimusmäärittelystä on mahdotonta nähdä, millainen ohjel-
misto näiden vaatimusten pohjalta syntyy. On epäselvää, soveltuuko se siihen työhön, jota sillä on tarkoitus tehdä.

Ohjelman toimittajalle ongelmat tulevat näkyviin tyypillisesti vasta projektin loppuvaiheissa, kun toimittajan mielestä ohjelmisto on valmis, mutta tilaaja löytää siitä yhä lisää koodattavaa. Vaatimusmäärittely voi olla monesta kohdasta niin avoin, ettei siitä selviä, mitä alkuperäiseen tarjoukseen oikeastaan sisältyi.

Pahin ongelma voi kuitenkin olla se, mitä toimittaja ei välttämättä koskaan näe ja mistä tilaajakin saattaa kuulla vain mutkan kautta: Työn tekeminen juuri käyttöönotetulla ohjelmalla onkin hyvin mutkikasta ja virhealtista, eikä kaikkia työtehtäviä välttämättä voida tehdä sen avulla ollenkaan. Käyttäjät ja kouluttajat keksivät nerokkaita tapoja ongelmien kiertämiseksi, jotta edes osa työstä olisi mahdollista tehdä uudella järjestelmällä.

Ongelmien syynä on monesti se, ettei ohjelmistoprosessi sisällä johdonmukaista menettelyä järjestelmän määrittelemiseksi, vaan määrittely perustuu pitkälti muistiinpanoihin niistä mielikuvista, joita tilaajan edustajilla on järjestelmästä, sekä tekniikan ehdoilla laadittuihin toteutusratkaisuihin. Kehitysprosessista saattaa puuttua kokonaan järjestelmällä tehtävän työn selvittäminen ja työn tukemisen suunnittelu – vaikka järjestelmän

olemassaolon tarkoitus on juuri työntöön helpottaminen.

GUIDe-mallin käyttöliittymälähtöinen määrittely

Käyttöliittymälähtöiseen määrittelyyn perustuvassa GUIDe-prosessimallissamme vaatimusmäärittelyn ongelmia ratkaistaan systemaattisen käyttöliittymäsuunnittelun avulla, joka tehdään käyttäjien työn ehdoilla. Ennen kuin määrittelyvaihe päättyy, tilaaja näkee käyttöliittymän näyttökuvasarjoista, millainen ohjelmisto on tulossa, ja järjestelmää voidaan koekäyttää käyttöliittymän avulla (esimerkki kuvassa 1). Erilaisia toteutusmahdollisuuksia arvioidaan myöhemmin käyttöliittymän asettamia vaatimuksia vasten.



Kuva 1. Ote PDA-aikataulujärjestelmän näyttökuvasarjasta.

Käyttäjien työprosesseja selvitetään mm. perehtymällä nykykäytäntöä kuvaaviin prosessikaavioihin, haastattelemalla käyttäjiä heidän työpaikoillaan tai seuraamalla vierestä, kun he tekevät työtään. Käyttöliittymäratkaisu tuotetaan simuloimalla yhden käyttäjän työntekoa kerrallaan ja piirtämällä simuloinnin aikana tietoja ja toiminnallisuutta näkyviin sitä mukaa kuin niitä työssä tarvitaan. Simulointipohjaisen GDD-suunnitteluprosessin (Goal-Derived Design) ideana on käyttää suunnittelussa samaa menettelyä kuin järjestelmän käyttöönottovaiheessa, jolloin käyttäjä ryhtyy tekemään työtään ja etenee kohti tavoitettaan askel askeleelta. Näin syntyy näyttökuvia, joissa käyttäjällä on tilanne päällä: hän

näkee käyttötilanteessa tarvittavan tietosisällön ja kulkee mahdollisimman suoraviivaisen polun tavoitteensa saavuttamiseksi.

Toisin kuin tekstimuotoisten vaatimusmäärittelydokumenttien tapauksessa, nyt voimme itse simulointitestata työn tekemistä tai mennä kohdekäyttäjien luo käyttöliittymäkuvien kanssa ja tehdä käyttäjien työtä vaihe vaiheelta paperikuvien tai esimerkiksi PowerPoint-näyttökuvasarjojen avulla. Kun työtä yritetään oikeasti tehdä käyttöliittymän avulla, voimme nähdä, kuinka käyttäjät reagoivat jokaiseen outoon käytäntöön, puuttuvaan toiminnallisuuteen tai puutteellisiin tietoihin, jotka haittaisivat heidän työtään, jos järjestelmästä tulisi tällainen.

GUIDe-mallissa osa hyväksymistestauksesta on tavallaan siirretty projektin alkuun. Ne määrittelyvaiheen virheet ja puutteet, jotka on mahdollista saada kiinni ilmankin koodaamista, otetaan kiinni käyttöliittymäsunnittelun ja työnkulutestauksen avulla.

Vaatimusten irrottaminen käyttöliittymäkuvista

Kun vaatimuksia on käyttöliittymäratkaisujen kiinnittämisen avulla tuotu näkyviksi ja testattu, ne voidaan irrottaa käyttöliittymäkuvista. Toimimme siis moniin nykykäytäntöihin nähden päinvastaisessa järjestyksessä: ensin käyttöliittymä ja siitä käsin käyttötapauksia, tietomalli ja käyttäjien työn kannalta hyvä rajausta järjestelmälle.

Käyttöliittymässä näkyvät toiminnalliset vaatimukset voidaan dokumentoida esimerkiksi UML-

käyttötapauksina. Kuvassa 2 on ote NCC Rakennus Oy:lle laaditun käyttöliittymän perusteella kirjoitetusta UML-käyttötapauksesta.

Myös käyttäjälle näkyvät tietosisältövaatimukset voidaan poimia tietomalliin näyttökuvista. Osa käyttöliittymän tietosisällöstä saadaan toisista järjestelmistä, joten näistä kohdista syntyy rajapintavaatimuksia. Kuvassa 3 on ote käyttöliittymäkuvasta, josta näkyy, mitkä tiedot pitäisi saada toisesta järjestelmästä.

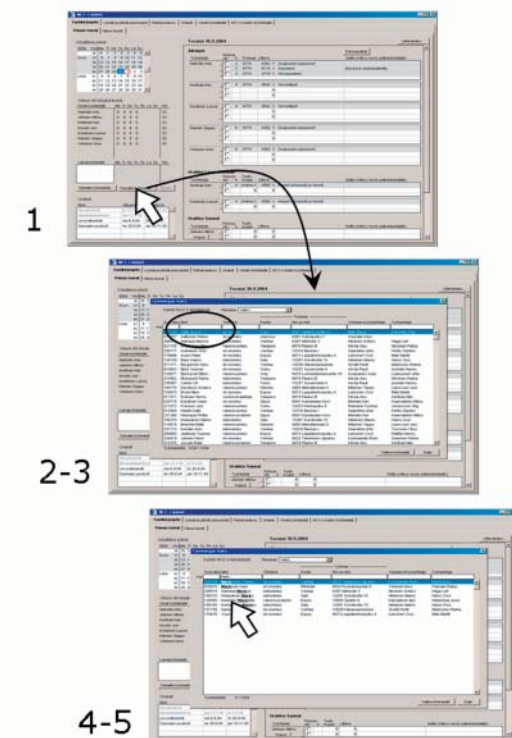
Miksi käyttöliittymä ja työnkulut ensin?

Osa toiminnallisista vaatimuksista määräytyy sen perusteella, miten osuvan käyttöliittymäratkaisun pystymme laatimaan käyttäjän työhön. Esimerkiksi jos yritämme ilman käyttöliittymää määritellä parturin ajanvarausjärjestelmän toimintoja, eräs aivan ilmeinen mieleen tuleva vaatimus olisi haku (ja hakusivu), jolla parturin asiakas voisi hakea vapaita aikoja päivän, parturin nimen tai esimerkiksi toimenpiteen mukaan.

Jos taas tuotamme käyttöliittymän käyttötilanteiden perusteella, saatamme pystyä tukemaan asiakkaiden käyttötilanteita paljon paremmin esimerkiksi laatimalla ajanvarauksista lukujärjestysmuotoisen kalenterinäytön ja näyttämällä kaikkien parturien yhden päivän ajat rinnakkain. Tällaisessa ratkaisussa erillistä hakua ei tarvita, vaan sen sijaan pitäisi tehdä kalenterinäytöt ja antaa käyttäjän vaihtaa näkyvissä olevaa päivää kalenterista. Nämä ovat eri toimintoja kuin alkuperäinen hakusivu.

Kuva 2. Käyttöliittymän toimintalogiikkaa käyttötapauksena. (Näyttökuvien esimerkkidata on kuvitteellista.)

Käyttöliittymän kuvasarja



UML-käyttötapaus

T12. Poimi lainatyöntekijä

Tavoite: Uusi lainatyöntekijä on saatu työmaan työntekijäluetteloon.

Käyttäjärooli: Työnjohtaja

Esiehdot: *Tuntikirjanpito*-välilehti valittuna.

Tyypillinen tapahtumien kulku (T)

1. Käyttäjä painaa *Toiselta työmaalta*.
2. Järjestelmä avaa *Työntekijän haku* -ikkunan. Hakuehto on tyhjä, ja taulukossa näkyvät kaikki työntekijät.
3. Käyttäjä kirjoittaa hakuehtoa. (A1)
4. Jokaisen näppäinpainalluksen jälkeen järjestelmä poistaa näkyvistä ne rivit, jotka eivät vastaa hakuehtoa, ja järjestää rivit: alkuun ne rivit, joissa hakuosuma on sanan alussa. Ikkunan alareunan rivimäärä vastaa näkyvissä olevien rivien lukumäärää. Taulukon ylin rivi on koko ajan valittuna. (A2)
5. Käyttäjä valitsee taulukosta työntekijän.

Projektsuunnittelujärjestelmästä noudettavaa tietoa:
litteranumerot, nimet, kuvaukset, tavoitetunnit, ennusteet.

Käyttöliittymäratkaisuja laatimalla voidaan löytää myös uusia mahdollisuuksia työprosessien suoraviivaistamiseksi. Esimerkiksi jos olemme määrittelemässä sairaalan lääkemääräysjärjestelmää, saatamme työnkulkuselvityksissä havaita, että reseptin laatimiseen liittyvässä päätöksentekokohdassa lääkäri käyttää tietoa sekä aiemmista resepteistä että hoitokertomuksesta. Käyttöliittymäsunnittelussa tämä johtaa siihen, että nämä kaikki tiedot pyritään keräämään yhteen näkymään. Tämän seurauksena lääkärin työnsä sujuvuutta saattaa kadota tarve vaihtaa järjestelmästä toiseen tai hänen työskentelystrategiansa voi tehostua, jos päätöksentekokohdat pystytään jakamaan esimerkiksi lääkärinkierron kannalta parempiin kohtiin.

Määrittelyvaiheen alussa kannattaa käydä seuraamassa edes muutaman tunnin ajan parin käyttäjän työskentelemistä heidän työpisteissään. Työssä eteen tulevat tilanteet (esimerkiksi potilaan vastaanotto tai asiakkaan ongelman käsittely) kirjoitetaan muistiin testitapauksiksi. Jo yhdessä työpäivässä on mahdollista käydä tapaamassa paria käyttäjää ja dokumentoida keskeisimmät tapaukset, joiden avulla saadaan konketriaa määrittelytyöhön. Digikuvilla on hyvä havainnollistaa tilanteita muillekin projektiryhmälle.

Vaikei käyttöliittymäkuvia olisi mahdollista päästä testaamaan käyttäjillä, määrittelypala- vereihin kannattaa sisällyttää käyttäjien työn simulointia. Joku osallistujista valitsee yhden työtilanteen ja näyttää näyttökuvaluonnosten avulla vaihe vaiheelta, miten käyttäjä tekee työtään. Nyt mahdolliset ongelmat alkavat paljastua, kun käyttäjien työtä tuntevat henkilöt saattavat huomata

Käyttöliittymän näyttökuvien simulointi työnkulkujen avulla on konkreettinen työkalu myös eri ratkaisuvaihtoehtojen väliltä valitsemista varten. Jos määrittelijöillä on epäselvyyttä tai erimielisyyksiä siitä, kumpi esille tulleista vaihtoehdoista olisi parempi (esimerkiksi pitäisikö olla erillinen hakusivu tai pitääkö myös peruskäyttäjillä olla oikeus muokata tietoja vai ei), molemmista ratkaisuvaihtoehdoista voidaan laatia käyttöliittymäluonnokset ja testata niihin liittyvien töiden tekemistä rinnakkain. Jos toinen ratkaisu tukee selvästi paremmin käyttäjien työtä kuin toinen, se yleensä näkyy rinnakkain asetetuista työnkulkukuvauksista ja näyttökuvasarjoista selvästi.

Käyttöliittymälähtöisillä menetelmillä ei tarvitse lähteä heti korvaamaan nykyisiä määrittelykäytäntöjä kokonaan, eivätkä ne edes kata koko vaatimusmäärittelyä. Kuitenkin jo täydentämällä nykyisiä käytäntöjä käyttöliittymälähtöisillä menetelmillä voidaan parantaa määrittelyjä ja jopa lyhentää määrittelyyn kuluva aikaa, kun epäselvien vaatimusten arvailemista ja pohdiskelua voidaan korvata käyttäjien työnkulkuselvityksillä ja simulointitestauksella. Vähäisilläkin resursseilla tehtyjen työnkulkuselvitysten ja niiden pohjalta laadittujen käyttöliittymäratkaisujen avulla saadaan täsmennettyä ja validoitua vaatimuksia. Mitä paremmin käyttäjien työ tunnetaan ja uusi järjestelmä osataan siihen sovittaa, sitä enemmän hyötyä ja säästöjä sen avulla saadaan käyttöön-
oton jälkeen.

Systemityö 2/2006