



Antti Martikainen on käytettävyyttä ja sovelluskehitysmenetelmiä päivittäisessä työssään pohtiva Realizer by SYSOPENDIGIA -tuotteen tuotepäällikkö. Realizer on mallinnuspohjainen suunnittelu- ja toteutusympäristö selainpohjaisten sovellusten rakentamiseksi.

Pätevillä protoiluvälineillä potkua vaatimusmäärittelyyn

Oikeiden vaatimusten löytäminen IT-hankkeen budjetoinnin ja aikataulutuksen pohjaksi on vaikeaa. Väärät vaatimukset puolestaan johtavat usein vakaviin laatuongelmiin erityisesti käyttäjien tarpeiden näkökulmasta. Näihin ongelmiin voidaan pureutua projektointia työtä edeltävällä, kattavasti käytettävyyteen ja käyttöliittymään keskittyvällä vaatimusmäärittelyvaiheella, jossa hyödynnetään tarkoituksenmukaisia välineitä.

Tutkimukset osoittavat, että vaiheistettuja sovelluskehitysmenetelmiä käytettäessä jopa 40 prosenttia IT-hankkeiden elinkaarikustannuksista aiheutuu vääristä vaatimuksista. Voitaneen siis todeta, että alkuvaiheen tekemiseen pitäisi löytää uusia lääkkeitä; oikeat vaatimukset pitää saada esiin aikaisemmassa vaiheessa kehityshankkeen elinkaarta, jotta projektit saadaan suunnattua oikein. Ketterät menetelmät lupaavat parannusta, mutta niiden soveltaminen on usein sopimusteknisesti haastavaa niihin liittyvän jaetun riskin vuoksi. Tässä tekstissä pohditaan mahdollisuuksia luoda vaiheistetuille hankkeille laadullisesti parempi lähtökohta, tehokkaan käyttöliittymälähtöisen vaatimusmäärittelyvaiheen avulla.

Prototypointityön apuna on usein muihin tarkoituksiin suunnitellut välineet

Käyttöliittymäprototyyppijä on jo pitkään hyödynnetty vaatimusten löytämisen välineenä. Esimerkiksi paperiprototyyppit ovat nopeasti tuotettavissa, ja niiden avulla voidaan tehdä myös käyttäjätestausta. Muita usein käytettäviä ei-tekniisiä välineitä ovat muiden muassa MS PowerPoint, sekä Visio. Mitään edellä kuvatuista välineistä ei olla kuitenkaan lähtökohtaisesti suunniteltu käyttöliittymäprototyyppien rakentamiseen, mikä näkyy niiden käyttömahdollisuuksien rajallisuutena. Erityisinä ongelmia voidaan mainita niillä tuotettujen prototyyppien tarjoama rajallinen laatu sidosryhmäkommunikaation ja todellisen käyttäjäkokemuksen testauksen kannalta, sekä puuttellinen kytkentä sovelluskehitysprosessin muihin tuotoksiin.

Teknisiä välineitä hyödynnettäessä ongelmaksi puolestaan muodostuu – niissä ketterämissäkin tapauksissa – heikko tuottavuus ja vastaavasti kyvyttömyys skaalautua laajojen hankkeiden käyttöliittymien kattavaan (esimerkiksi poikkeustapaukset huomioivaan) ja interaktiiviseen prototypointiin siten, että kehitystyö ja kustannukset pysyisivät aisoissa.

Entäpä jos otetaan kuvaan mukaan välineet, jotka on lähtökohtaisesti suunniteltu nimenomaan laadukkaiden ja interaktiivisten käyttöliittymäprototyyppien kustannustehokasta tuottamista varten. Mikä voisi olla tällaisten – käyttöliittymien nykyistä tehokkaamman simuloinnin mahdollistavien – välineiden tuottama lisäarvo?

Käyttöliittymien simulointivälineillä oikeat vaatimukset esiin

Tarkennetaanpa simulointivälineiden tarjoama hyötypotentiali hypoteesin muotoon: sovelluskehityksen riskejä voidaan nykyisestään pienentää siten, että palvelun täsmälliset vaatimukset – käytön onnistumisen ja siten asiakasorganisaation liiketoiminnallisen hyödyn kannalta – haetaan esiin koko järjestelmän toiminnot kattavan



Hyvän prototyyppivälineen ominaisuuksia

- Nopeus ja ylläpidettävyyys todellisen kaltaisten ja interaktiivisten prototyyppien tuottamisessa
- Prototyypit käytettävissä oikeilla laitteilla (esim. puhelin, PC)
- Monimutkaistenkin käyttötapauksen simulointi poikkeustapauksineen mahdollista
- Helppo jaeltavuus, ei kynnystä kokeiluun esim. asennusvaatimusten muodossa

interaktiivisen ja todellisen kaltaisen prototyypin kautta, ennen varsinaisen määrittelytyön, teknisen suunnittelun ja toteutuksen aloitusta. Puhutaan siis prototyyppistä, joka mahdollistaa hankkeesta keskustelemisen tehokkaasti niin sisäisesti kuin kumppaneiden kanssa, ja myös mahdollistaa – ainakin soveltuvin osin – lopullisen kaltaisen käyttäjäkokeuksen testaamisen. Tällainen malli tuntuisi melko luontevasti mahdollistavan vaatimusten – ja budjetin ja aikataulun – perustelun kiinnittämisen teknistä suunnittelua ja toteutusta varten. Ai niin, ja asiakaskin tietäisi jo etukäteen mitä tulee saamaan.

Entäpä sitten tällaiset simulointivälineet, onko niitä? Maailmalla – vähemmän Suomessa – on tosiaankin etenkin selainpohjaisten sovellusten kehityksessä alettu hyödyntämään välineitä, joiden ideana on luoda kehitysvaiheen kommunikaatiolle ja käyttäjäpalautteelle mahdollisimman konkreettinen pohja, nopeasti tuotettavan ja lopullisen tuntuisen käyttöliittymän käytön kautta. 100M\$ vuodessa Web-projekteihin sijoittava Sprint säästi ensimmäisenä ”simulaatiovuotenaan” 50% (17M\$) vuosittaisiin muutospyyntöihin varatusta budjetistaan (Forrester research: Show, Don't Tell: How High-Fidelity Prototyping Tools Improve Requirements Gathering, 2005). Uudenlaisilla menetelmillä ja välineillä ollaan järjestelmien käyttäjilleen tarjoaman laadun suhteen päästy – ainakin paikoin – huikesiin tuloksiin.

Toimintamallit ja välineet murrosvaiheessa

Käyttöliittymäsimuloinnin mahdollistavien välineiden markkina on nyt elinkaarensa alkuvaiheessa. Esimerkkinä markkinoilla olevasta tuotteesta voi mainita amerikkalaisen iRise-välineen. Omassa Realizer by SYSOPENDIGIA –tuotteessamme käyt-

töliittymien tehtävämallinnuspohjaisuus kytkee prototyyppivaiheessa tuotetut ja simuloidut selainkäyttöliittymät suoraan tekniseen toteutusvaiheeseen. Prototyyppi ei siis ole poisheittävä asia, vaan lähtökohtaisesti osa teknistä projektitoimitusta. Näin pyritään nopeuttamaan kokonaisprosessien läpivientiaikoja ja siirtämään käytettävyyden eteen alkumetreillä tehtävä työ muuttumattomana hankkeiden lopputuloksiin.

Käyttäjälähtöisyys on joka tapauksessa nousevassa kurssissa. Ja niin on myös tarve tehostaa käytettävyytyötä ja sen vaikutusta sovellusten laatuun. Tarvitaan paitsi menetelmiä ja ajatuksia, myös entistä tehokkaampia työvälineitä, jotka mahdollistavat kattavien ja laadukkaiden prototyyppien kustannustehokkaan kehityksen ja käytettävyysspanostusten nykyistä konkreettisemmän integroinnin ohjelmistokehitystyöhön. Lisäksi kaivataan uskallusta muuttaa vanhoja toimintamalleja, monilla toimialoilla myös tietojärjestelmien ostajan näkökulmasta.

