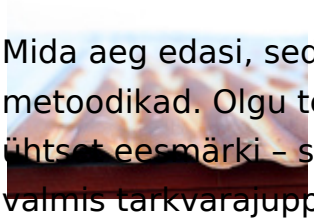
Kas kasutajakeskne disain ja paindlik tarkvaraarendus on võimelised ühe katuse all elama?

13 aastat tagasi - 08.01.2013 Autor: Roger Puks



Mida aeg edasi, seda kindlamini kinnitavad tarkvaraarenduses kanda agiilsed metoodikad. Olgu tegu Scrum-i või Extreme Programming-ga, jagavad nad kõik ühtset eesmärki – saada võimalikult paindlikult ning mõistliku ressursikuluga valmis tarkvarajupp, mis vastaks täpselt kliendi ärinõudmistele. Kui aga klassikalised arendusmetoodikate puhul näeb klient töö lõpptulemust alles mitme kuu möödudes, siis agiilsete metoodikate puhul on esialgne variant valmis juba umbkaudu nädalaga.

See aga asetab UX-disainerid pisut kehva olukorda – kuidas leida aega niivõrd kiire arendusprotsessi jooksul kasutatavusloomele, mis on juba iseenesest vägagi aeganõudev protsess?

Kas oleks tark tegu eelmainitu kogu arenduskäigust hoopis välja jätta ning loota oma õnnele või tuleks otsida variante, kuidas kasutatavusspetsialiste kaasata ilma liigset ajakulu tekitamata?

Alustada tuleks eelkõige spekuleerimisest, miks UX-disainerid agiilsete metoodikate puhul üleüldse kiputakse mängust välja jätma. Klientide jaoks on olulised muud prioriteedid: kärpida time-to-market aega võimalikult palju ning korrigeerida rakendust vastavalt turuolukorrale. Seetõttu on üsna keeruline hakkama saada samade tarkvaranõuetega kogu arendusprotsessi käigus.

Hea rakenduse eelduseks on paindlikkus kogu projekti käigu jooksul, mistõttu on sunnitud mitmed prograammeerijad kahtlema kasutatavustestimise vajalikkuses – miks tekitada projektis pudelikaelaefekt kui selle tulemused ei pruugi kestagi?Sestap hakatigi otsima võimalusi, loomaks vähemate ressursside abil etemat tarkvara. Lahendus saabus metoodikate näol, mis keskendusid koleme printsiibile – olla kiire, adaptiivne ning inimkeskne. Agiilsed metoodikad muutusid aja jooksul üha populaarsemaks, kuna tulemus oli kliendi jaoks märksa rahuldavam – vähem ressursse, kiirelt valmiv tulemus ning kliendi pidev kursishoidmine arendusprotsessi käekäiguga.

Nagu mainitud, on üheks agiilsete metoodikate põhiprintsiibiks inimkesksus. Loogiline järeldus oleks, et keskne roll on ka UX-disainil. Paraku on olukord aga vastupidine. Kui jutt on agiilsest tarkvaraarendusest, on üsna tavapärane, et arendusmeeskond koosneb üksiti… arendajatest. Kõrvale on lisaks kasutajaliidesespetsialistidele jäänud ka tarkvaratestijad, seega ei kannata vaid UI pool, vaid kogu tarkvara kvaliteet üldiselt. Tegu on arendusmetoodikatega, milles kannavad keskset ning ainsat rolli arendajad.

Lisaks eelmainitule tulevad siinkohal mängu ka UX-disaini ning agiilsete arendusmetoodikate kui protsesside ajalised kestvused. Kuigi mõlema näol on tegu iteratsioonidest koosnevate protsessidega, on arendusetappide pikkus kardinaalselt erinev. Kui agiilne arendusmeeskond on võimeline iga tööpäeva lõpus tagant esitama kliendile mingi tasemeni töötava rakenduse, jõuab sama ajaperioodiga UX-disainer ehk fookusgrupi paika panna, kellele tuginedes hakata kasutajaliidest looma.

Fakt on aga see, et lõpp-kokkuvõttes on just nimelt positiivne kasutajakogemus see, mis toob loodava rakenduse esile teiste omasuguste seast. Projekti tähtaeg on aga igal juhul fikseeritud ning selle muutmine tähendaks juba kliendi jaoks kahju - kuidas sellisel juhul käituda?

Eelkõige tuleks vaadata UX-disainereid pigem liitlaste kui protsessi aeglustajatest vaenlastena – nende töö eesmärk on siiski rakenduse kvaliteeti tõsta. Mis kasu on ajalisest võidust, kui selle hinnaks on rakendus, mis käivatatakse vaid üks kord ning seejärel hakatakse alternatiive otsima?

Kuidas aga kaasata üsnagi nõudlikud kasutajaliidesespetsialistid agiilsesse arendusprotsessi? Kas selleks on ka võimalusi, mis ei nõua olemasolevate metoodikate kardinaalset ümbertöötamist või protsessi ajalist pärssimist? Üheks alternatiivseks lahenduseks UX-probleemile oleks arendajate koolitamine antud vallas. Agiilsesse projektimeeskonda, kuhu kuuluvad enamasti üksiti back-end ning front-end arendajad, on keeruline juurde tekitada terve hulk eraldiseisvaid spetsialiste. Sestap võiks iga projektiliige end viia võimalikult hästi kurssi erinevate praktikate ja trendidega, tunda tehnoloogilisi võimalusi ning piiranguid, oskama rakendada erinevaid teada-tuntud heuristikaid. See ei pruugi tegelikkuses olla nii keeruline ettevõtmine kui esialgu tundub – alustada võib kasvõi erinevate erialakirjanike teoste lugemisest, olgu nendeks Jakob Nielsen, Steve Krug või Alan Cooper.

Alternatiivne ettepanek eelmainitule oleks kasutajatestide planeerimine paralleelselt iteratsioonide paikapanemisega. Antud juhul mängiks olulist rolli just

nimelt eeltöö – samal ajal kui enne esmaste iteratsioonide algust planeeritakse ette arendusprotsess, oleks võimalik leida juba ettenägelikult hulk inimesi, kelle seast hakata otsima tulevasi testijaid. Sobiva profiiliga inimesi ei tohiks olla keeruline leida, eeldades, et loodav rakendus ei ole suunatud vaid väga kitsale sihtgrupile.

Abiks on antud juhul sotsiaalmeedia, olemasolev kliendibaas, kuid väga edukalt võib ekspluateerida ka kõrvalkontoris töötavaid inimesi või juhuslike koridoriskõndijaid. Ainus määrav tingimus oleks antud juhul nende eriala - tuleks silma peal hoida, et testgruppi ei satu ühtegi arendajat või veebidisainerit, kes suudavad iga loodava rakenduse kohta midagi kriitilist välja mõelda.

Testimisprotsessi tuleks agiilse projekti korral aga kaasata võimalikult vähe kasutajaid. UX-disaini eesmärk on tavapäraselt olnud detailse kuvandi loomine. Iga aspekt kasutajaliidesest peab olema paika pandud, mis tähendab, et selleni jõudmiseks tehtav töö võib võtta omajagu aega. Kui tavapäraselt kaasatakse kasutajaliideste testimisse umbkaudu 5-10 inimest, siis agiilsete projektide puhul oleks ehk targem alustada pisut väiksemate eesmärkidega – testida kasutuskeskkonda kolme inimese peal korraga.

Nn. agiilse testimise eesmärk pole kõikide kitsaskohtade avastamine kõikide kasutusstsenaariumite korral, vaid pigem kriitiliste kasutatavusvigade, mis võivad teha tarkvara kasutamise jätkamise võimatuks, kõrvaldamine. UX-spetsialist suudab läbi viia testimissessioonid koos tulemuste analüüsi ning disainiettepanekute tegemisega kolme inimesega juba ühe päeva jooksul, rohkema arvu inimeste puhul suureneb ajahulk juba oluliselt. Lisaks sellele on väärtuslikum testida erinevate inimeste peal erinevaid kasutajaliideseid kui ühte lõplikku kuvandit piiratud hulga inimeste peal.

Pisut uuendusliku meetmena võib proovida ka projektimeeskonda kuuluvaid programmeerijaid kasutatavustestimisel ära kasutada. Ühegi UX-spetsialisti jaoks pole võõras olukord, kus arendusmeeskond näeb teda kui vaenlast, kelle ainueesmärgiks on kogu arendusprotsessi aeglustamine ning senitehtu kritiseerimine. Sellisest mentaliteedist tuleks üle saada – alustades näiteks lihtsalt meeskonda kuuluva koodikirjutaja paigutamine erinevatesse UX-disainiprotsessi käivatesse rollidesse - esmalt vaatlejana, siis moderaatorina. Lisaks faktile, et selline tegevus aitab kaasa eelmainitud projektimeeskonna harimisele UX-disaini valdkonnas, suudavad arendajad peale iseseisvalt läbiviidud sessiooni koheselt sisse viia rakenduses ilmnenud puudujäägid, mida järgmisel stand-up koosolekul juba ülejäänud inimestele esitleda.

Pisut vähem aeganõudvam pääsetee oleks lihtsalt ekspertauditi või järelevalve tellimine.

UX-disainerite töömeetodeid on palju ning kogu loomeprotsess ei pea keskenduma ainiti kasutatavustestidele. Kui meeskonnas leidub kogenud UX-eksperte, on väga võimalik iga uue väljalaske tagant lasta kõnealusel isikul sellele lihtsalt pilk peale heita ning jagada pisut heureetilisi näpunäiteid. Ekspert ei pea tingimata olema keegi, kes osaleb pidevalt kogu meeskonna töös ning viibib kohapeal. Vastupidi - tegelikkuses ei ole tarvis isegi, et ekspert viibiks meeskonnaga samas riigis.

Pisut nutikas oleks asjatundja just nimelt palgata erinevast ajavööndist - see võimaldaks tal visata tehtud tööle pilk peale samal ajal kui arendusmeeskonna töö seisab ning esitada oma raport juba järgmiseks hommikuks.

Üks on kindel – agiilsed metoodikad ning kasutajakeskne disain on kaks asja, mis ei ole arendusmaastikult lähiajal kadumas.

Täpselt sama kindel on tegelikkuses ka asjaolu, et ideaalse lõpptulemuse nimel tuleb mõlemal leida ühiseks koostööks sobivaim viis. Seeläbi on võimalik end rakendada arendus- ning UX-meeskonnal ning projektid lõppevad märksa edukamalt.

Projekti kulukuse alandamine ei tohi tulla kasutajate kaasamise arvelt – sedasi võidetakse küll ajaliselt, kuid unustatakse, et ilma adekvaatse kasutuskeskkonnata minetab kogu tegevus oma eesmärgi.

ROGER PUKS

- Lahendused