

Veebi kujunduse elutsükkel

Veebileht, veebirakendus on põhimõtteliselt samasugune toode nagu mistahes tarkvara. Kuigi lihtsama veebilehe loomise ja hooldamisega saab hakkama pea iga inimene, kellel on tehnilist taipu ja silma kujunduseks, eeldab korraliku veebirakenduse loomine lisaks kujundamisele ja arendusele (sisu loomine, koodi kirjutamine jms) siiski ka korralikku planeerimist, analüüsi, testimist ning hilisemat ülalpidamist. Veebirakendusel on nagu igal teisel tootel oma elutsükkel, milles osalevad erinevate rollidega inimesed. Veebirakenduste loomine on reeglina meeskonnatöö kuna eeldab väga paljude erinevate teadmiste ja oskuste rakendamist (kujundamine, programmeerimine, pilditöötlus, heli- ja videotöötlus, animeerimine jne).

Rollid veebiarendusmeeskonnas

Veebiarendusmeeskonnad võivad olla erineva suurusega alates üksikisikust lõpetades suurte mitmekümne või isegi mitmesaja liikmelistega. Igal meeskonnaliikmetel on täita kindel roll. Rollide nimed võivad erinevates meeskondades olla erinevalt nimetatud, sama rolli võib täita mitu inimest ja ühel inimesel võib olla täita mitu rolli kuid väga üldiselt jagunevad meeskonnaliikmed juhtideks (*manager*), disaineriteks (*designer*), arendajateks (*developer*) ja dokumenteerijateks (*documenter*). Natuke täpsemalt võiks

Projektijuht (*project manager*) koordineerib projekti läbiviimist, jagab meeskonnaliikmetele ülesandeid, hoolitseb vajalike ressursside kättesaadavuse eest, organiseerib kohtumisi ja koostööd kõigi osapoolte vahel (klient, disainer, arendaja jne). Sageli on juht abiks kui mõni projekti osa on ajakavast maha jäänud.

Veebidisainer (*web designer*) loob toote visuaalse poole, kogub ja loob kasutatava graafika jne. Suuremates meeskondades on disaineri ülesanded kitsamalt jagatud:

- Graafikadisainer (*web graphics designer*) loob ja töötleb graafikafaile (illustratsioonid, bännerid jms), optimeerib neid veebi jaoks.
- Multimeediumidisainer (*web multimedia designer*) tegeleb heli ja videoklippide loomise ja töötlemisega, sageli ka graafikaga. Väiksemas meeskonnas võib see roll olla ühendatud graafikadisaineri rolliga, suuremas aga võib heli ja video jaoks olla eraldi inimene.
- Kasutajaliidese disainer (*user interface designer*) hoolitseb, et veebileht oleks mugavalt kasutatav (navigatsioon jms).
- Kasutajakogemuse disainer (*User eXperience designer*) on uus roll, tegeleb parima kasutajakogemuse loomisega.

Veebiprogrammeerija (*web programmer*) ehk arendaja (*developer*) loob loogika (koodi), mis töötab kujunduse taga ning hoolitseb, et rakendus teeks seda, mida nõutud.

Dokumenteeriija (*documenter*) koostab kõik projektiga seotud dokumendid, protokollib koosolekuid, hoolitseb, et kogu dokumentatsioon oleks kõigile meeskonnaliikmetele kättesaadav.

Testija (*tester*) testib loodud rakendust, selle osi, otsib võimalikke vigu ja probleeme ning annab nende kohta infot disaineritele ja arendajatele.

Telliija ehk klient (*client*) ei osale küll igapäevaselt meeskonna töös kuid ideaalis on ta tihedas kontaktis projekti juhi ning disaineritega, et loodav veebileht vastaks tema soovidele ja ootustele.

Lisaks võib meeskonda kuuluda veel hulk erinevaid spetsialiste, kes hoolitsevad näiteks tekstide (sisu) kirjutamise eest, andmebaaside loomise eest, meeskonda võib kuuluda fotograaf jne.

Suuremate, vastutusrikkamate projektide puhul kuulub meeskonda ka valdkonnaspetsialist, kes peab jälgima, et veebileht vastaks käsitletava valdkonna vajadustele, nõuetele. Väiksemate projektide puhul on valdkonnaspetsialisti rollis sageli tellija ise.

Elutsükkel

Igasuguse toote elutsükli moodustavad kõik temaga seotud tegevused kavandamisest valmimise, hilisema hoolduse ja täiendamise ja lõpuks kasutusest kõrvaldamiseni. Üldiselt vaadeldakse tarkvara puhul tema loomist ja täiendamist ning kuna täiendamine kordab väiksemas mahu mõningaid loomise etappe, siis tavaliselt räägitakse tarkvara ehk siis meie puhul veebilehe loomise elutsüklist.

Veebilehe loomise etappe (nagu mistahes tarkvara puhul) loetletakse erinevates allikates erinevate nimedega kuid sisult on tegemist alati järgmistega:

- nõuete väljaselgitamine (*requirements*);
- kavandamine (*design*);
- teostamine (*implementation*);
- valideerimine (*validation*);
- tugiteenus (*support*).

Ükski meeskond ei saa korralikku veebilehte luua ilma kõiki neid etappe läbimata.

Hõlbustamaks töö planeerimist, on välja töötatud mitmeid erinevaid mudeleid, kuidas toote loomise elutsükli läbida (koskmudel (*waterfall*), spiraalmudel (*spiral*), tähtmudel (*star*) jpt).

Praegusel ajal kasutatakse väga palju nn ADDIE mudelit või selle erinevaid variatsioone.

Tegemist on lihtsa, loogilise mudeliga, mille järgimine aitab tööd planeerida ja juhtida.

ADDIE mudeli igas faasis saadav tulemus on lähtematerjaliks järgmisele faasile.

- Analüüsi (*analysis*) faasis selgitatakse "probleem", püstitatakse eesmärgid, selgitatakse tingimused (nõuded).
- Disaini (*design*) faasis valitakse strateegia, vahendid, meedia jms.
- Arendusfaasis (*development*) luuakse toode vastavalt disainifaasis tehtud otsustele.
- Rakendusfaasis (*implementation*) testitakse toote prototüüpe, valmistatakse lõplik toode ja võetakse see kasutusele (luuakse juhendmaterjalid, koolitatakse kasutajad jne).
- Hindamisfaasis (*evaluation*) testitakse ja hinnatakse toodet (kasutajate tagasiside).



Joonis 1 ADDIE arendusmudel

Hindamist viiakse tegelikult läbi praktiliselt kõikides etappides, igas etapis toimub töö iteratiivselt, et saada parim tulemus.

Järgnevalt vaatleme veebilehe loomise elutsükli etappe just sellest mudelist lähtudes.

Analüüs

Veebilehe loomine algab kokkuleppes tellijaga. Tavaliselt avaldab tellija soovi saada mingisugust veebilehte või veebirakendust, esitab teatud tingimused, lähteülesande ja selle põhjal püütakse teha hinnapakumist. Reeglina on vaja tellijale esitada täpsustavaid küsimusi, et paremini hinnata ees ootavat tööde mahtu ja iseloomu.

Kokkuleppe saavutamise järel algab täpsem analüüs, mida klient soovib, mida see tegema peab, millised on nõudmised väljanägemisele jne. Kliendi soovide väljaselgitamine ning tema

ideede rakendamine kujunduses suurendab rahulolu tulemusega ja vähendab hilisemaid kulutusi kasutajatoele ning veebirakenduse hooldamisele.

Toimub nõuete kogumine (*requirements gathering*). Selgitatakse välja:

- funktsionaalsed nõuded (*functional requirements*) – mida loodav veebileht või veebirakendus peab tegema;
- andmete nõuded (*data requirements*) – milliseid andmeid loodav süsteem vajab, kasutab,;
- kasutuskõlblikkuse nõuded (*usability requirements*) – puudutavad tulevasi kasutajaid, nende eeldatavaid kogemusi ja oskusi, millele süsteem peab vastama.

Nõuete kogumiseks kasutatakse kõige enam intervjuud kliendiga, vesteldakse, küsitletakse.

Kasulik on kliendiga vestlusesse kaasata mitu meeskonnaliiget (aitab tagada, et kliendi soovidest saadakse õigesti aru), kindlasti peab kohal olema disainer.

Eeltööna enne intervjuud, saab tutvuda valdkonnaga, mida loodav veebileht käsitleb, tutvuda teiste sama valdkonna veebilehtedega, tellija vana veebilehega (kui see on olemas).

Alustatakse veebilehe eesmärkide selgitamisega. Otsitakse vastuseid küsimustele:

- Millist informatsiooni peab loodav veebileht edastama?
- Millised komponendid on olemas kõigil edukatel veebilehtedel?
- Milliseid aspekte peab loodava veebilehe kujundamisel veel arvestama?

Väga täpselt tuleb välja selgitada veebilehe sihtgrupp, kes hakkavad seda külastama/kasutama. Püüdke leida vastuseid küsimustele:

- Kes hakkavad loodavat veebilehte külastama? Millised on nende vajadused? See info aitab otsustada, millist informatsiooni on vaja veebilehele lisada.
- Miks inimesed seda veebilehte külastama peaksid? Millistes sotsiaalsetes oludes võimalikud külastajad elavad, töötavad?
- Millist dialoogi, milliseid sõnumeid soovivad külastajad veebilehelt saada?
- Millist tehnoloogiat kasutada külastajatega suhtlemiseks?

Koostöös kliendiga on vaja otsustada, milliseid komponente veebilehel kasutada, millised nende seas on kriitilise tähtsusega, millised lihtsalt kasulikud ja milliseid võiks lisada, kui nende jaoks ressursse jätkub. On hulk kliente, kes väga täpselt teavad ja nõuavad, mida leht peab sisaldama ja mida mitte, on ka selliseid, kelle ainus konkreetne soov on saada endale veebileht. Kliendile saab pakkuda erinevaid võimalusi, komponente kuid mitte neid peale suruda.

Kindlasti tuleb uurida ka kliendi soove loodava veebilehe välimuse osas. Milline on kliendi logo (kui see on olemas), kas on olemas nn firmavärvid ning kas neid on tarvis kasutada, milliseid kirjastiile eelistatakse jne.

Tähelepanu tuleb pöörata ka sellele, millised vajadused ja võimalused on kliendi veebilehe majutamiseks (*hosting*). Täpsemad tingimused (kui palju kettaruumi vajatakse, milliseid tehnoloogiaid peab veebiserver toetama) saab määrata alles pärast disainiprotsessi kuid juba analüüsi ajal on tarvis sellele mõelda. Võimalik, et klient soovib veebilehte majutada kindla teenusepakkuja juures või oma serveris, mis võib seada tehnilisi piiranguid, mida peab juba algusest peale arvesse võtma.

Välja tuleb püüda selgitada:

- kui palju on tarvis kettaruumi (arvestada tuleb ka laiendamisvaru);
- mis tehnoloogiaid tuleb kliendi soovitud veebilehe loomisel kasutada;
- mis tüüpi faile soovitakse veebilehel kasutada (mõned veebimajutuse pakkujad seavad teatud failivormingutele piiranguid);
- Kui suur on planeeritav külastatavus, millise andmeliiklusega peaks arvestama.

Analüüsi tulemusena koostatakse loodava veebilehe spetsifikatsioon, kuhu pannakse kirja kõik nõuded, tingimused, mis on välja selgitatud ning loodava veebilehe kirjeldus. Nõuete kirjapanekuks on loodud ka erilisi abstraktseid keeli kuid kasutada võib ka naturaalselt keelt, vältida tuleks vaid slängi.

Kui spetsifikatsioon on kliendiga kooskõlastatud, algab disainietapp.

Disain

Kõigepealt tuleb läbi mõelda veebilehe struktuur, kuidas sisu organiseerida. Kõik kliendiga kokku lepitud komponendid ja veebilehe sisu jagatakse kategooriateks, hierarhiasse.

Arendusmeeskond valib spetsifikatsioonist lähtudes eesmärkide saavutamiseks kõige paremini sobivad ning nõuetele vastavad tehnoloogiad.

Järgmine samm on juba veebilehe välimuse planeerimine. Disainer alustab lihtsatest eskiisidest ning jõuab lõpuks täpsete prototüüpideni, millel on paigas visuaalne külg: navigatsioonivahendid, komponentide paigutus.



Joonis 2 Näide veebilehe kujunduse eskiisist

Luuakse prototüübid, mis tulevast veebilehte kirjeldavad. Keerukamate tegevuste sooritamise kohta luuakse nn kasutajalood – pannakse detailset ja sammhaaval kirja, kuidas kasutaja tegevusi sooritab.

Disaini ja arendusmeeskond loob koostöös mitmeid erinevaid ideekavandeid ja kujundusversioonide eskiise, millede seast koostöös kliendiga valitakse välja kõige paremini kliendi vajadustele vastav variant.

Prototüüpimine

Looge veebilehestiku disaini illustreeriv mudel (sisukaart jms), see võimaldab kliendil juba kujundusprotsessi varajases staadiumis tagasisidet anda kui muudatusi on veel väga lihtne teha. Hiljem lisage juba ka kasutajaliidese prototüüp (eskiis, joonistus), mis võimaldab kliendil aimu saada, milline veebileht tegelikkuses välja nägema hakkab.

Tüüpiliselt tuleb läbida mitu tööringi: kujundada, hinnata tulemust kliendiga suheldes, ümber töötada jne.

Tavaliselt kasutatakse disaini illustreerimiseks järgmist tüüpi mudeleid:

- sisukaart (*site map*) – reeglina tekstipõhine ülesehituse kirjeldus;
- süžetahtel (*storyboard*) – veebilehestiku graafiline skeem, võib olla paberile joonistatud, loodud mistahes graafikaprogrammiga, genereeritud mõne veebiredaktori (*HTML editor*) poolt;

- sõrestikmudel (*wireframe*) – keht, mis näitab loodavaid veebilehti. Iga leht on joonistatud informatsiooniplokkidena, mis näitavad kus ja kuidas paiknevad erinevad komponendid. Selliselt esitatakse objektide visuaalne paigutus (*layout*) kuid mitte lõplik välimus.

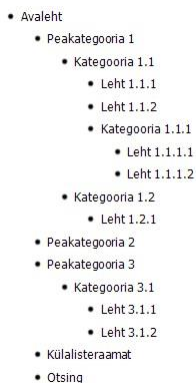
Sisukaart

Sisukaart (*site map*) on üks kriitilise tähtsusega ja laialt kasutatav vahend veebidisainis. Kuna sisukaart näitab veebilehestiku üldist struktuuri ja hierarhiat, siis kasutatakse neid veebilehestiku üldise ülesehituse ja navigatsiooni kavandamisel.

Alustada tuleks lihtsast, jämedakoelisest eskiisist, lisades sinna lihtsalt esmaselt planeeritud, olulised lehed. Vältida tuleks detailse info lisamist sisukaardile. Sisukaart muutub ja täieneb projekti edenedes.

Sisukaardid on enamasti tekstipõhised kuid vahetevahel ka graafilised. Iga lehe kohta pannakse kirja pealkiri ning talle viitav link (URL). Töö käigus võivad disainerid sinna lisainfot kirja panna (näiteks lehe olulisus terviklikus veebilehestikus, viimase muutmise aeg, kui sageli seda lehte muudetakse jne).

Vägagi sageli lisatakse HTML sisukaart valmis veebilehestikule. See on eriti kasulik, kui veebilehestik sisaldab paljusid lehti, lihtsustab navigatsiooni ning võimaldab ka otsingumootoritel veebilehestikku läbi uurida.



Joonis 3 Tüüpiline veebilehestiku sisukaart

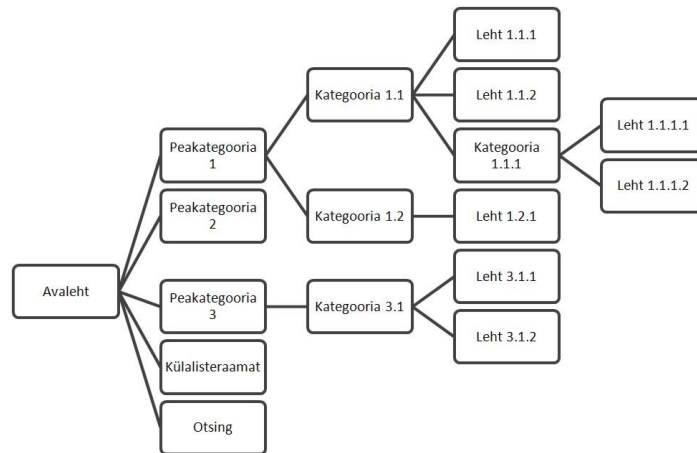
Süžetahvel

Süžetahvli (*storyboard*) kasutatakse lisaks sisukaardile aitamaks teha otsuseid disaini, tehnoloogia aga ka eelarve osas. Hea süžetahvel pakub kõigile projektiga seotud inimestele selget ülevaadet loodavast veebilehestikust.

Süžetahvil ei ole näha kõiki üksikuid veebilehestiku lehti kuid see katab peamised lehestiku funktsionaalsed osad. Reeglina pannakse iga lehe kohta kirja pealkiri ja mõned märkmed tema sisu kohta.

Ka süžetahvel ei sisalda mingit infot lehtede visuaalse disaini kohta kuid võimaldab saada ettekujutuse, millised põhielemendid igal lehel olemas on.

Süžetahvli võib sarnaselt sisukaardile luua pliiatsiga paberile joonistades või mõnda graafikaprogrammi kasutades.



Joonis 4 Tüüpiline süžeetahvel

Sõrestikumudel

Sõrestikumudel (*wireframe*) veebidisainis on veebilehele paigutatavate elementide skemaatiline esitus. Võttesõnaks sõrestikumodeli puhul on kiirus, neid kasutatakse uute ideedega eksperimenteerimiseks, testimiseks ja sobitamiseks aga mitte kujunduse detailseks kuvamiseks.

Kliendiga suhtlemisel lasevad sõrestikumodelid kliendil keskenduda lehe ülesehitusele (*layout*) laskmata end disainielementidel (pildid, kirjastiilid jt) oluliselt kõrvale kallutada. Sõrestikumodeli loomisel kasutatakse lihtsaid graafilisi kujundeid (mitte reaalseid pilte), lisatakse neile pealkiri. Sõrestikumodelil peavad nähtaval olema kõik tähtsad veebilehe elemendid.



Joonis 5 Tüüpiline veebilehe sõrestikumudel

Stsenaarium

Stsenaarium (*scenario*) on verbaalne prototüüp, mis kirjeldab loodava veebilehe ülesehitust. Siia alla kuuluvad ka kasutajalood (*user story*) – tegevuste, valikute ja tulemuste verbaalsed kirjeldused.

Visuaalne disain

Pärast veebilehestiku struktuuri ja üksikute lehtede ülesehituse (komponentide paigutus) fikseerimist saab asuda visuaalse disaini juurde. Valitakse kirjastiil pealkirjadele ja tekstile,

kasutatava graafika stiil ja värviskeem. Lähtepunktiks on veebilehe sõrestikmodel (*wireframe*).

Prototüüpidena kasutatakse siinkohal joonistusi (lihtsatest visanditest detailsete värvipiltideni) ja ekraanipilte. Visandid on sageli käsitsi tehtud, värvipiltide loomisel kasutatakse kujundusprogramme.

Ka visuaalsest disainist luuakse mitmeid erinevaid versioone, millede seast koos kliendiga sobivaim välja valitakse.

Detailne prototüüp

Pärast seda, kui kõik osapooled on disainiprotsessis osalenud, valmib veebilehestiku detailne prototüüp, süžetahvlile (*storyboard*) ja sõrestikmodelile on lisandunud hulk detaile, valminud on visuaalne kujundus.

Kokku on pandud kogu materjal, millest lähtudes saavad arendajad soovitud veebilehe luua. Prototüüpi hinnatakse koos kliendiga ja selle heakskiitmisel algab arendusprotsess.

Arendus

Arendusfaasis toimub koostatud spetsifikatsioonile vastava veebilehe loomine: programmeerimine, andmebaaside loomine, rakenduse sidumine visuaalse kujundusega, animatsioonide, graafika, heli- ja videoklippide loomine jms.

Üks olulisemaid asju on kogu arendusprotsessi jooksul keskenduda kliendi ja kasutaja vajadustel!

Disain pole kunagi ideaalne, veatu. Arendusprotsess peab olema iteratiivne – korduv loomine, ülevaatamine ja muutmine kuni tulemus vastab kliendi ja kasutaja soovidele. Üldiselt on soovitatav ka arendusfaasis kliendiga võimalikult tihedat koostööd teha. Klienti ei huvita iga koodirida või loodud pilt kuid mingit funktsionaalsust pakkuvate komponentide valmimisel on mõistlik neid kohe kliendile näidata.

Iteratiivse arendusega võib ka juba olemasolevaid veebilehti edasi arendada tehes regulaarselt väikseid muudatusi. Muudatusi tuleb teha tsüklitena jättes alati alles osad, mis "töötavad" ning asendades sobimatuid.

Rakendamine

Üleminek arendusfaasist rakendamisfaasi pole väga selgelt määratletav. Rakendusfaasis toimub testimine ja vigade leidmisel nende parandamine (mis on sisuliselt arendamine). Pärast arendusprotsessi algab testimine ja täiustamine (*refinement*), mis on samuti iteratiivne protsess. Selle käigus otsitakse ja parandatakse tehnilisi vigu, kasutuskõlblikkuse (*usability*) vigu, vaadeldakse lehe avanemiseks kuluvat aega ja vajadusel optimeeritakse rakendust.

Testimisprotsess hõlmab mitmeid erinevaid hindamisi, näiteks:

- Heuristiline läbivaatus (*heuristic evaluation*) – eksperdid vaatavad üle loodava veebilehe kasutajaliidese (*user interface*) ning hindavad seda kasutuskõlblikkuse (*usability*) seisukohalt lähtudes üldtunnustatud kasutuskõlblikkuse (*usability*) printsiipidest. Erinevad inimesed märkavad erinevaid vigu, seetõttu on ühel inimesel üksinda heuristilist läbivaatust teha, tavaliselt soovitatakse kasutada kolme kuni viit eksperti. Leitud vead tuleb loomulikult kõrvaldada.
- Korrektuur (*proofread*) – testitakse kõiki rakenduse interaktiivseid elemente (ekraanivormid, dialoogiboksid jms). Vajadusel õpetatakse klienti kõiki rakenduse osi

kasutama. Vaadatakse üle õigekiri ja grammatika. Kontrollitakse kas kõik lingid avanevad ja viitavad õigetele lehtedele.

- Mitteformaalne testimine, valjusti mõtlemine (*think aloud*) – veebirakendust kasutavad kaks inimest koos, üks tegutseb ja arutleb valjusti oma tegevuste üle (avaldab mõtteid, tekkivaid küsimusi jne), teine teeb märkmeid. Sellisel moel tulevad probleemid kiiresti ilmsiks sest protsess hõlmab reaalset tööd rakendusega. Testkasutajaks peab olema inimene, kes pole seda rakendust enne näinud ega tea selle eesmärki, otstarvet. Testkasutajale tuleks aega anda 30 minutit, kui ta vajab rohkem aega, siis vajab rakendus ümbertegemist!

Testimisprotsessi järel tuleks veebirakendus publitseerida testserverile, mis pole kõigile avalikult kättesaadav, et ka klient saaks tulemust testida.

Rakendusfaasi käigus koostatakse ka rakenduse dokumentatsioon (juhendmaterjalid jms), vajadusel koolitatakse kasutajaid (kliendi esindajad, tulevased veebilehe administraatorid jne).

Vajadusel tuleb teha veel täiendavaid muudatusi ning alles seejärel kui ka klient on tulemuse heaks kiitnud publitseeritakse veebirakendus avalikuks kasutamiseks avalikku serverisse.

Veebileht on valmis. Siinkohal sageli elutsükkel katkestatakse!

Hindamine

Hindamisfaas on see, mis toimub pärast veebilehe valmimist ja publitseerimist. Selle käigus toimub formaalne kasutuskõlblikkuse hindamine (*formal usability studies*) – protsess, mille käigus analüüsitakse reaalseid kasutusandmeid inimese ja süsteemi koostöö kohta. Kõige tavapärasemateks andmeallikateks on veebiserveri logid, veebikasutuse analüsaatorid (näiteks Google Analytics). Sellist hindamist saabki läbi viia alles pärast veebilehe publitseerimist (aga näiteks ka testkasutamise ajal testserveril) ning võib olla vägagi kallis. Hea on see, et pole vaja otsida testkasutajaid või eksperte, kasutada on andmed reaalsete igapäevaste kasutajate käitumise kohta veebilehel.

Hindamisfaasis tuntakse huvi kasutajate tagasiside vastu, uuritakse, kuidas ollakse veebilehega rahul.

Hindamisfaas jääb sageli väga lühikeseks või seda ei toimuigi. Tavaliselt ilmneb hulk võimalikke probleeme alles pikema aja jooksul reaalse kasutamise käigus ning ka siis oleks tarvis parandusi teha, seda aga tihti peale arendaja ja kliendi vahel kokku ei lepita.

Mõistlik oleks juba esimeste kokkulepete sõlmimisel kliendiga juhtida tähelepanu ka võimalikule vajadusele hiljem täiendusi ja parandusi teha ning selle tegevuse tingimustes kokku leppida. Samas on selline hilisem arendustöö arendajale tülikas sest projektiga seotud meeskond on juba hõivatud teiste projektidega või lausa laiali läinud.