

Ohjelmistojen vaatimusmäärittelyn nykytila ja tulevaisuuden suuntauksia

Hannu Räisänen

Helsinki 9.12.2004

Seminaarialustus

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

| | |
|--|----------|
| 1 Johdanto | 1 |
| 2 Ohjelmistojen vaatimusmäärittelyn kehittyminen 1990-luvulla | 1 |
| 2.1 Vaatimusten ja vaatimusmäärittelyn uusi luonne | 2 |
| 2.2 Vaatimusmäärittelyn inhimillinen ulottuvuus | 3 |
| 2.3 Vaatimusten epätäydellisyys | 3 |
| 3 Vaatimusmäärittelyn nykytila teollisuudessa | 4 |
| 4 Vaatimusmäärittelyn tulevaisuuden huomionkohteita | 5 |
| 4.1 Formaalien menetelmien etsiminen ympäristön kuvaamiseen | 6 |
| 4.2 Laajoille markkinoille suunnattujen ohjelmistojen vaatimusmäärittely | 6 |
| 4.3 Ohjelmistoarkkitehtuurin vaikutus vaatimusmäärittelyyn | 7 |
| 4.4 Vaatimusten priorisointi | 7 |
| 4.5 Vaatimusten uudelleenkäyttö | 7 |
| 5 Yhteenveto | 8 |
| Lähteet | 8 |

1 Johdanto

Vaatimusmäärittely on ohjelmistotuotantoon välttämättömänä kuuluva prosessi, jonka tarkoituksena on löytää ja hallita tietokonepohjaisen järjestelmän (järjestelmä, ohjelmisto) vaatimuksia. Tässä kirjoitelmassa kuvataan vaatimusmäärittelyn nykytilaa. Tarkempia selvityksiä vaatimusmäärittelyn määritelmästä, osa-alueista ja keskeisistä tekniikoista saa seminaarin muista kirjoitelmista.

Kun halutaan tarkastella jonkin ilmiön nykytilaa, on välttämätöntä katsoa ajassa sekä eteen- että taaksepäin. Toisessa kappaleessa kuvataan ohjelmistojen vaatimusmäärittelyn kehitystä 1990-luvulla sekä niitä uusia tuulia, joita 90-luku toi tullessaan. Kolmannessa kappaleessa kerrotaan uudesta vaatimusmäärittelyn nykytilaa teollisuudessa kartoittaneesta tutkimuksesta. Neljännessä kappaleessa esitellään arvioita siitä, mitkä tulevat olemaan vaatimusmäärittelyn tärkeimpiä huomion kohteita lähitulevaisuudessa. Viidennessä kappaleessa on lyhyt yhteenveto.

2 Ohjelmistojen vaatimusmäärittelyn kehittyminen 1990-luvulla

Ohjelmistotuotannon ammattilaisten ja tutkijoiden mielenkiinto on kohdistunut korostetulla tavalla eri aikoina ohjelmistojen elinkaaren eri vaiheisiin [Sid94]. 1960-luvulla suurin mielenkiinto kohdistui ohjelmointiin. 1970-luvulla keskityttiin suunnitteluun ja 1980-luvulla päähuomion kohteena olivat spesifikaatiot. Luonnollisesti todellisuudessa kaikki nämä aiheet ovat olleet aina tärkeitä, mutta juuri näinä aikoina kukin aiheista varasi paikkansa omana tutkimuskohteenaan.

Vaatimusmäärittelyä alettiin tutkia kehittää jo 1970-luvulla, mutta vasta 1990-lukua voidaan pitää vaatimusmäärittelyn vuosikymmenenä. Vuosikymmenen alussa pidet-

tiin sarja kansainvälisiä tapaamisia ja vuosikymmenen puolivälissä aloitti alaan keskittynyt kansainvälinen tieteellinen lehti [Tev]. Tämä tutkimus toi vuosikymmenen loppuun mennessä esiin muutamia perustavaa laatua olevia uusia ajatuksia vaatimusmäärittelystä ja sen luonteesta. Useat näistä ajatuksista kytkeytyvät vahvasti toisiinsa. [Sid94, BN00]

2.1 Vaatimusten ja vaatimusmäärittelyn uusi luonne

Tärkeimpänä uutena ajatuksena on hylätty vanha määritelmä, jonka mukaan "vaatimukset kertovat mitä ohjelmisto tekee ottamatta kantaa siihen, miten se sen tekee". Määritelmän suurin ongelma on ajatuksen kapeus. Kautta aikojen vaatimusmäärittelyn tärkeä vaihe on ollut kaikkien sidosryhmien tunnistaminen ja sitä kautta niiden näkökulmien huomioiminen vaatimuksia kartutettaessa ja muokattaessa. Tähän ajatukseen sisältyy paradoksi: on väistämätöntä, että jonkin sidosryhmän "mitä" on toiselle "miten". Edelleen, kaikki ohjelmiston kuvaukset ovat tietyn tasoisia abstraktioita. Myös tällöin törmätään tilanteeseen, jossa yhden abstraktiotason "mitä" on toisen "miten"[Sid94].

Joseph Goguen esittää vaatimukset informaationa ja muistuttaa, että kaikki informaatio on sidoksissa esittäjäänsä ja muuhun ympäristöön. Juuri ympäristö määrittelee tarkemmin vaatimusten merkityksen [JS96]. Näin ollen vaatimusmäärittelyssä ei tulisi keskittyä kuvaamaan tuotettavaa ohjelmistoa (tai järjestelmää), vaan kuvata koko ympäristöä kahdelta kannalta. Ensinnäkin sitä, millainen ympäristö on nykyään ja mitkä asiat siinä ovat nyt tosia ja toisaalta sitä, millaiseksi tuotettavan ohjelmiston halutaan ympäristön muuttavan [ZJ97].

Vanha vaatimuksen määritelmä ei kuitenkaan ole täysin käyttökelvoton. Se voidaan esittää uudessa, ehkä hiukan tarkemmassa muodossa: "Vaatimukset kertovat, mitä ohjelmiston ja ympäristön rajapinnasta voidaan tarkastella, ei mitään itse

ohjelmistosta"[ZJ97].

2.2 Vaatimusmäärittelyn inhimillinen ulottuvuus

Edellä kuvataan kuinka juuri ympäristö on tärkein tarkasteltava kokonaisuus, eikä vaatimusmäärittelyä voida tehdä eristyksissä niistä organisaation ja sosiaalisen ympäristön vaikutteista, joiden keskellä tuotettavan järjestelmän tulisi toimia [BN00]. Kun otetaan ympäristö tarkastelun kohteeksi, voidaan varmistua siitä, että sekä sosiaaliset että tekniset asiat tulee otetuksi huomioon. Vaatimukset tulevat ilmi pääasiassa vaatimusmäärittelijän ja ohjelmiston sidosryhmien sosiaalisena vuorovaikutuksena. Tämä on pidemmälle vietyä ajattelua, kuin pelkkä epämääräinen "sidosryhmien huomioon ottaminen". Nykyään yleiset menetelmät, kuten kyselyt ja haastattelut eivät yksin ole riittäviä menetelmiä saamaan esiin ympäristön äänetöntä sovellusalueen tuntemusta [JS96].

Vaatimusmäärittelyä (sekä vaatimusten esiinkaivelu että myöhempää muokkausta) voidaan tehdä monellakin eri tavalla, mutta kaikissa muissa kuin kirjallisiin tietolähteisiin perustuvissa tekniikoissa tärkeää on määrittelijän kyky kommunikoida tietolähteenä toimivan henkilön tai henkilöiden kanssa. Tekniikoita, joissa ihmisen ja hänen toimintansa ymmärtäminen on tärkeää ovat mm. aivoriihet, diskurssianalyysit, etnografia, haastattelut, sovelluksen yhteissuunnittelu (JAD, joint application design), käyttäjän tarkkailu sekä käyttötapausten ja skenaarioiden käyttö. [Hat]

2.3 Vaatimusten epätäydellisyys

Yksi syy ohjelmistojen lisäävien elinkaarimallien käyttöönottoon oli ymmärrys siitä, että yhdellä projektinomaisella vaatimusmäärittelyvaiheella ei voida löytää kaikkia olennaisia vaatimuksia. Silti useat lähteet korostavat vaatimusmäärittelyvaiheen tuotteena syntyvien vaatimusten täydellisyyden, ristiriidattomuuden ja oikeellisu-

den tärkeyttä. Kuitenkin suurimmalle osalle ohjelmistotuotannon ammattilaisista eri syistä epätäydelliset vaatimukset ovat osa arkipäivää. [JS96, BN00, van00]

1990-luvulla ymmärrettiin, että vaatimusmäärittelyn ammattilaisten täytyy ottaa vakavasti tarve analysoida ja ratkaista vaatimusten ristiriitaisuuksia, edistää sidosryhmien välistä neuvottelua ja käsitellä epä johdonmukaisia malleja. Tätä tarvetta varten kehitettiin vaatimusten hallintaa, erityisesti jäljitettävyyttä. Jäljitettävyyden on tärkeässä osassa, kun ohjelmistoon tehdään lisäyksiä tai vaatimukset muuttuvat [Pal].

Muuttuvien vaatimusten seurauksena syntyivät myös niin kutsutut kevyet prosessimallit, esimerkiksi XP (Extreme Programming) [Bec99]. XP:n lähtökohtana on vaatimusten jatkuva muuttuminen siten, että laajaa ja pitkäkestoista vaatimusmäärittelyä täydellisten vaatimusten löytämiseksi ja niihin perustuvaa suunnittelua ei kannata tehdä ainakaan perinteisessä muodossaan.

3 Vaatimusmäärittelyn nykytila teollisuudessa

Colin Neill ja Phillip Laplante tutkivat kyselyn avulla vaatimusmäärittelyn tilaa teollisuudessa maaliskuussa 2002 [CJN03]. He lähettivät 1519 ohjelmistotuotannon eritasoiselle ammattilaiselle pyynnön vastata www-pohjaiseen 22 kysymystä sisältäneeseen kyselyyn. Kyselyyn oli vastannut määrääjässä 194 henkilöä. Tekijät esittävät tutkimuksen tulokset, mutta eivät esitä mitään johtopäätöksiä. Tulokset ovat kuitenkin mielenkiintoisia ja antavat mahdollisuuden arvioida vaatimusmäärittelyn nykytilaa. Seuraavassa esitän muutamia valittuja kohtia tutkimustuloksista.

- 35% vastaajista käyttää vesiputousmallia.
- Vesiputousmalli on suosituin elinkaarimalli projektin kestosta ja vastaajan asemasta riippumatta. Vain yli kaksi vuotta kestävässä projekteissa lisäävä malli

oli vesiputoumalla suosituimpi.

- 60% käyttää jonkinlaista prototyypitystä.
- Yli 50% käytti skenaarioita ja käyttötapauksia vaatimusmäärittelyn aikana.
- 33% ei käytä mitään erityistä menetelmää vaatimusten analysointiin ja mallintamiseen. 70% prosenttia heistä uskoi valmiin tuotteen täyttävän loppukäyttäjän tarpeet ja 63% uskoi loppukäyttäjän pitävän tuotetta helppokäyttöisenä.
- Formaalit menetelmät ovat harvinaisia, vain 7% käytti niitä. Formaaleja menetelmiä käyttäneet pitivät lopputuotteen laatua heikompana kuin epäformaaleja menetelmiä käyttäneet.
- 59% teki jonkinlaisia tarkastuksia vaatimuksille.
- 52% piti organisaationsa panostusta vaatimusmäärittelyyn riittämättömänä.
- Vain 40% vastaajista uskoi projektin ylittäneen aikataulun tai budjetin.

4 Vaatimusmäärittelyn tulevaisuuden huomionkohteita

Suurin osa seuraavassa esitetyistä mahdollisista vaatimusmäärittelyn huomionkohteista perustuvat 90-luvulla esitettyihin uusiin ajatuksiin ja toisaalta toisiin tutkimuskohteisiin. Toinen tärkeä vaatimusmäärittelyn tulevaisuutta ohjaava tekijä on valmiskomponenttien ja ohjelmistojen tuoteperheiden suosion jatkuva kasvu.

4.1 Formaalien menetelmien etsiminen ympäristön kuvaamiseen

Useisiin vaatimusmäärittelyn malleihin on kehitetty formaaleja esityksiä. Useat näistä malleista kuvaavat suurelta osin ohjelmiston nykyistä tai tulevaa käyttäytymistä. Määrittelyn pääpainon siirtyessä kohti ympäristön kuvaamista on välttämätöntä kehittää formaaleja menetelmiä myös ympäristön kuvaamiseen. Näiden mallien tulisi kyetä käsittelemään myös epätäydellisiä, epäyhtenäisiä ja kehittyviä vaatimuksia. Formaalien mallien tulisi ottaa huomioon myös tuotettavan ohjelmiston (tai erityisesti ohjelmistokomponentin) ympäristöönsä kohdistamat oletukset. Näin niiden siirtäminen uuteen ympäristöön ja sulauttaminen toiseen ohjelmistoon tai ohjelmistoperheeseen helpottuu huomattavasti. [BN00]

Ihmisläheisiin (esim. etnografisiin) tekniikoihin perustuvat vaatimusten kartuttamismenetelmät tuottavat laajaa ja monipuolista tietoa ympäristöstä. Nämä menetelmät eivät kuitenkaan käy yksiin nykyisten sovellusalueen ominaisuuksia kuvaavien formaalien tekniikoiden kanssa. Tulevaisuudessa on välttämätöntä kaventaa ihmisläheisten menetelmien ja niiden tuotteita kuvaavien formaalien analyysi- ja spesifiikaatiomenetelmien välistä kuilua. [BN00]

4.2 Laajoille markkinoille suunnattujen ohjelmistojen vaatimusmäärittely

Suurin osa nykyään tuotettavista ohjelmistoista tehdään yleisille markkinoille. Tällöin sen vaatimuksia ei voida kartuttaa tuotteen tilanneen asiakkaan ympäristön perusteella. Eräs tapa kartuttaa vaatimuksia on tutkia jonkin sovellusalueen ongelmia ja keksiä niihin ratkaisuja. Tässä tapauksessa tarkempi vaatimusmäärittely tehdään vasta perinteistä myöhemmässä vaiheessa saattaen sisältää tietoa mm.

markkinatilanteesta, tuotettavan ohjelmiston koosta, mahdollisten ominaisuuksien joukosta ja tuotteen sopimisesta tuottavan organisaation tuotestrategiaan. Perinteinen vaatimusmäärittely ei juuri tue näiden ongelmien käsittelyä. Tutkijat ovat alkaneet kiinnittää asiaan laajempaa huomiota vasta viime aikoina. [JS96]

4.3 Ohjelmistoarkkitehtuurin vaikutus vaatimusmäärittelyyn

Ohjelmistoarkkitehtuureita on viime vuosina tutkittu runsaasti. Tutkimus on keskittynyt niiden olemukseen, mallintamiseen, käyttäytymisen kuvaamiseen ja toteutukseen. Vähemmälle huomiolle on jäänyt tietyn arkkitehtuurivalinnan vaikutus nykyisiin ja tuleviin vaatimuksiin sekä niiden priorisointiin ja evoluutioon. Erityishuomion ansaitsevat tuoteperheet ja vaatimusten vaihtelu tuoteperheen eri tuotteissa. [BN00]

4.4 Vaatimusten priorisointi

Kilpailulliset tekijät ovat muodostaneet suuren rajoitteen ohjelmiston tuottamiseen käytettävälle ajalle. Yleensä tuotteen markkinoille saamista nopeutetaan pienentämällä julkaisun kokoa ominaisuuksien kustannuksella. Tämä pakottaa määrittelijät luokittelemaan vaatimukset toivottuihin ja pakollisiin ominaisuuksiin. Sopivien vaatimusten karsiminen tai muuttaminen saattaa mahdollistaa ohjelmiston toteuttamisen valmiiden komponenttien avulla. Tutkimus erityisesti nykyisissä olosuhteissa tapahtuvasta vaatimusten priorisoinnista on vähäistä. [JS96]

4.5 Vaatimusten uudelleenkäyttö

Vaatimusten eroavaisuus toisiaan muistuttavien sovellusalueiden välillä on yleensä vähäisempää kuin toteutusten eroavaisuus [van00]. Siksi onkin yllättävää, että tekniikat uudelleenkäytettävien vaatimusten löytämiseksi, yhdistämiseksi ja sopeut-

tamiseksi ovat saaneet vain vähäistä huomiota verrattuna muuhun uudelleenkäytön tutkimukseen. Erityisen ongelmallista on jo löydettyjen tekniikoiden soveltuminen käytäntöön ja niiden skaalautuvuus. Erittäin suuri hyöty uudelleenkäytettävistä vaatimuksista saataisiin valmiskomponenttien tuottamisessa ja valinnassa. Uudelleenkäytettävät vaatimukset mahdollistaisivat vaatimusviitemallien luomisen eri sovellusalueille. [van00, BN00]

5 Yhteenveto

Ohjelmistojen vaatimusmäärittely koki suuria muutoksia 1990-luvulla. Myös itse vaatimusten määrittelyn muuttaminen ympäristön kuvaamiseksi tuotteen ominaisuuksien kuvaamisen sijaan on nähty tarpeelliseksi. Samalla tärkeäksi on muodostunut ihmisläheisten tekniikoiden käyttäminen vaatimusten kartuttamiseksi ja riittävän sovellusalueymmärryksen muodostamiseksi.

Tulevaisuudessa vaatimusmäärittelyn tutkimuksen pääpaino kohdistuu niihin muutoksiin, joita 1990-luku toi mukanaan. Toinen tärkeä tuleva tutkimuskohde on nykyisten suurten tutkimusalueiden, kuten ohjelmistoarkkitehtuurien, tuoteperheiden ja uudelleenkäytön sovittaminen vaatimusmäärittelyprosessiin.

Kuilu tutkimuksen ja käytännön välillä on suuri. Erityisesti formaalien menetelmien käyttö suhteessa niiden tutkimukseen on vähäistä.

Lähteet

- Bec99 Beck, K., *Extreme Programming Explained: Embrace Change*. Addison Wesley, 1999.

- BN00 Bashar Nuseibeh, S. E., Requirements engineering: A roadmap. *Proceedings of the conference on The future of Software engineering*, 2000.
- CJN03 Colin J Neill, P. A. L., Requirements engineering: The state of the practice. *IEEE Software*, 20,6(2003), sivut 40–45.
- Hat Hatakka, T., Vaatimusten esiinkaivelu.
- JS96 Jawed Siddiqi, M. C. S., Requirements engineering: The emerging wisdom. *IEEE Software*, 13,2(1996), sivut 15–19.
- Pal Palo, M., Requirements traceability.
- Sid94 Siddiqi, J., Challenging universal truths of requirements engineering. *IEEE Software*, 11,2(1994), sivut 18–19.
- Tev Tevanlinna, A., Requirements engineering – concepts and relations.
- van00 van Lamsweerde, A., Requirements engineering in the year 00: a research perspective. *Proceedings of the 22nd international conference on Software engineering*. ACM Press, 2000, sivut 5–19.
- ZJ97 Zave, P. ja Jackson, M., Four dark corners of requirements engineering. *ACM Trans. Softw. Eng. Methodol.*, 6,1(1997), sivut 1–30.