

Ohjelmistotuotantomenetelmien kehittyminen 1950-luvulta nykypäivään

Lauri Suomalainen

Kandidaatintutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 20. lokakuuta 2013

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Lauri Suomalainen			
Työn nimi — Arbetets titel — Title			
Ohjelmistotuotantomenetelmien kehittyminen1950-luvulta nykypäivään			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Kandidaatintutkielma	20. lokakuuta 2013	5	
Tiivistelmä — Referat — Abstract			
Tiivistelmä			
Avainsanat — Nyckelord — Keywords			
avainsana 1, avainsana 2, avainsana 3			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Sisältö

1	Johdanto	1
2	Peruskäsitteistöä	2
3	Ohjelmistotuotannon alkutaival	4
4	Perinteiset ohjelmistotuotantomenetelmät	4
5	Inkrementaaliset ja iteratiiviset menetelmät	4
6	Prototyypaus	4
7	Ketterät menetelmät	4
8	Ohjelmistojen kehitys tänään ja huomenna	4
	Lähteet	4

1 Johdanto

Tämä kandidaatintutkielma tarkastelee ohjelmistotuotantomenetelmien kehittymistä ohjelmistokehittämisen alkuaajoista nykypäivään ja lähitulevaisuuteen. Se käsittelee erilaisia ohjelmistotuotantomenetelmiä kronologisesti. Jokaisen menetelmän kohdalla pyrin vastaamaan seuraaviin kysymyksiin:

- Mistä ohjelmistotuotantomenetelmässä on kyse?
- Miksi sitä käytetään/käytettiin ja mitä hyötyä siitä on/oli?
- Mitkä olivat sen heikkoudet?

Luku 2 esittelee lyhyesti keskeisimmät ohjelmistotuotantoon ja tuotantomenetelmiin liittyvät käsitteet.

Luku 3 käsittelee ohjelmistotuotannon alkuaikoja, jolloin tietokoneet itsessään olivat vielä verrattain uusi ilmiö ja suuret ohjelmistot ja niiden tuottaminen oli vähäistä. Alun alkaen laitteiston rajallisuus sitoi myös ohjelmistojen mahdollisuuksia, mutta laitteiston kehittyessä syntyi tarve organisoidummalle ohjelmistojen kehittämiselle.

Luku 4 käsittelee perinteistä ohjelmistotuotantomallia eli niin sanottua "vesiputousmallia". Tarkastelen mallin peruseräjäiteita, sen nousemista aikansa keskeisimmäksi ohjelmistotuotantomenetelmäksi, sen onnistumisia ja epäonnistumisia sekä sen heikkouksia ja siihen kohdistettua kritiikkiä.

Luku 5 tarkastelee inkrementaalisia ja iteratiivisia ohjelmistotuotantomenetelmiä sekä erityisesti RUP:ia eli Rational Unified Processia.

Luku 6 keskittyy prototyyppaamiseen ohjelmistotuotantomenetelmänä. Erityisesti tarkastellaan spiraalimallia, jota voidaan pitää vesiputousmallin kehittyneempänä versiona.

Luvussa 7 otetaan tarkasteluun ketterät ohjelmistotuotantomenetelmät. Vaikka ketterä ohjelmistokehitys on kattokäsitteenä monelle erilaiselle menetelmälle kuten XP ja Kanban, käsittelen Scrumia tyyppiesimerkkinä, sillä se on kaikista ketteristä menetelmistä suosituin ja tunnetuin.

Luku 8 on tutkielman yhteenveto ja tarkastelen siinä myös mahdollisia lähitulevaisuuden kehityssuuntia.

Ohjelmistotuotannon eri osa-alueita tarkastellaan tutkielmassa ohjelmistotuotantomenetelmiä määrittävinä piirteinä. Tämä tarkoittaa sitä, että tarkasteltaessa esimerkiksi miten vaatimusmäärittely toteutetaan jossain tietyssä ohjelmistotuotantomenetelmässä, keskitytään prosessin konkreettisen toteutuksen sijasta sen asemaan ja erityispiirteisiin menetelmän kontekstissa.

2 Peruskäsitteistöä

Software eli ohjelmisto käsittää tietokoneohjelman tai -ohjelmia sekä kaiken niihin liittyvän informaation ja materiaalin kuten tietokannat ja dokumentaation.

Tietokonelaitteisto eli hardware käsittää tietokoneen fyysiset osat kuten prosessorin ja kovalevyn. Laitteistoa tarvitaan ohjelmistojen suorittamiseen ja laitteisto tarvitsee toimiakseen toimintaohjeet matalan tason tietokoneohjelmoina. Käytännössä tietokoneohjelmistot ja -ohjelmat sekä tietokonelaitteisto eivät ole käyttökelpoisia yksinään, vaan kumpakin tarvitaan toisen järkevään käyttöön.

Termi software engineering, suomeksi ohjelmistotuotanto, alkoi esiintyä kirjallisuudessa 1960-luvun puolivälissä. Termi itsessään on ollut usein keskustelun ja väittelyn kohteena ja ohjelmistotuotannon kuulumista insinööritaitoihin on kyseenalaistettu. [2, 3, 5] Watts S. Humphrey on määritellyt ohjelmistotuotannon tarkoittavan kurinalaista laadukkaiden ohjelmistojen tuottamista hyödyntäen niin luonnontieteellisiä, matemaattisia kuin insinööritaitojenkin periaatteita ja käytänteitä[4]. IEEE Computer Society määrittelee termin viittaavan kurinalaiseen, systemaattiseen ja arvioitavissa olevaan lähestymistapaan ohjelmistojen tuotannossa, käytössä ja ylläpidossa[1]. Ilkka Haikala ja Jukka Märijärvi tulkitsevat määrittelyjen tarkoittavan ohjelmistotyötä, jonka tuloksena syntyvät järjestelmät täyttävät käyttäjiensä kohtuulliset toiveet ja odotukset ja tämän lisäksi valmistuvat laadittujen aikataulujen ja kustannusarvioiden puitteissa[3].

Ohjelmistotuotantoon kuuluvat kaikki ohjelmistotuotantoprosessin osa-alueet. Haikala ja Märijärvi [3] määrittelevät ne seuraavasti:

Määrittely sisältää asiakasvaatimusten analyysin ja niistä johdetaan ohjelmistovaatimukset.

Suunnittelu pitää sisällään ohjelmiston määrittelyssä jäsennehtyjen toiminnallisuuksien ja ominaisuuksien suunnittelun

Toteutus tarkoittaa ohjelmiston ohjelmointia sekä testauksen toteutusta

Testaus pyrkii karsimaan ohjelmistosta ohjelmointivirheitä ja muita vikoja. Tyypillisiä testaustapoja ovat yksikkötestaus, integraatiotestaus ja debuggaus.

Dokumentointi käsittää ohjelmistoprojektin aikana tuotettavan kirjallisen materiaalin, kuten projektisuunnitelmat, testaussuunnitelmat ja jopa ohjelmakoodin kommentoinnin.

Käyttöönotto ja **ylläpito** ovat asiakkaan ongelmien ratkomista, virheiden korjaamista ja tarvittaessa uusien ominaisuuksien lisäämistä.

Laatujärjestelmällä ja **laadunvarmistuksella** on tarkoitus taata, että ohjelmisto täyttää käyttäjän ja asiakkaan toiveet ja odotukset.

Projektinhallinta on työkalu ohjelmistotuotantoprojektin organisointiin. Suuret ohjelmistoprojektit koostuvat usein useasta rinnakkain tai peräkkäin etenevistä osaprojekteista ja tällöin niiden järjestelmällinen hallinta voi olla keskeistä koko projektin onnistumisen kannalta.

Tuotteenhallinta: Usein kaupallisella ohjelmistolla on useita eri konfiguraatioita jolloin se voidaan aina räätälöidä yksilöllisesti kullekin asiakkaalle sopivaksi. Tuotteenhallinnan tarkoitus on varmistaa, että asiakkaalla on tarvitsemansa toimiva versio ohjelmistosta.

Ohjelmistotuotantomenetelmä on koko ohjelmistotuotantoprosessin kattava viitekehys, joka ohjaa prosessin osa-alueitten käytännön toteutusta.

- 3 Ohjelmistotuotannon alkutaival
- 4 Perinteiset ohjelmistotuotantomenetelmät
- 5 Inkremetaaliset ja iteratiiviset menetelmät
- 6 Prototyypaus
- 7 Ketterät menetelmät
- 8 Ohjelmistojen kehitys tänään ja huomenna

Lähteet

- [1] Abran, Alain, Moore, James W., Bourque, Pierre, Dupuis, Robert ja Tripp, Leonard L.: *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE, 2004. <http://www.swebok.org/>, ISO Technical Report ISO/IEC TR 19759.
- [2] Grier, David Alan: *Software Engineering: History*. Teoksessa *Encyclopedia of Software Engineering*, sivut 1119–1126. 2010.
- [3] Haikala, I. ja Märijärvi, J.: *Ohjelmistotuotanto*. Korkeakoulu-sarja. Satakku, 2003, ISBN 9789521404863. <http://books.google.fi/books?id=xIVaAAAACAAJ>.
- [4] Humphrey, W. S.: *The software engineering process: definition and scope*. Teoksessa *Proceedings of the 4th international software process workshop on Representing and enacting the software process*, ISPW '88, sivut 82–83, New York, NY, USA, 1988. ACM, ISBN 0-89791-314-0. <http://doi.acm.org/10.1145/75110.75122>.

- [5] Mahoney, Michael S.: *Finding a History for Software Engineering*. IEEE Annals of the History of Computing, (1):8–19, ISSN 1058-6180. <http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1278847>.