

Matti Vuori

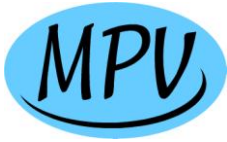
Leanista ja testauksesta

Lean on ”kevyt tuotantotapa”, joka on kotoisin vuosikymmenten takaa Toyotan autotehtaalta. Se saavutti 1990-luvulla suosiota valmistavassa teollisuudessa ja viime vuosina sitä on alettu tuoda ohjelmistokehitykseenkin. Nyt on hyvä aika pohtia, mitä se voisi tarkoittaa testauksen kannalta. Siksi ohessa muutamia reflektioita asiasta kiinnostuneille.

T Testaukseen liittyvät reflektiot on merkitty T-liikennemerkillä.

Sisällysluettelo

Mitä "Lean" on?.....	2
Lean on monenlaista.....	4
Lean on monta erilaista mutaatiota ihmisten mielissä	4
Toyotan tuotantojärjestelmän filosofia ja toiminta	5
Suomalainen valmistavan teollisuuden lean.....	8
Lean ohjelmistokehitys.....	8
Leanin ja agilen suhde	10
Miten "lean" saadaan aikaan?.....	12
Diagnoosi.....	13
Liinin testaajan piirteet	13
Liinin testaustoiminnan filosofia	14
Liinin testaustoiminnan toimintajärjestelmä	14
Hukattoman testaustoiminnan piirteitä	15
Lopuksi	16
Lähteitä.....	17



Mitä "Lean" on?

Lean oli alun perin "lean production", nykyisin yksinkertaisesti "lean" edustaa näkemystä hyvästä "kevyestä" toimintatavasta. Sitä on sovellettu paljon valmistavassa teollisuudessa, mutta myös palvelualoilla ja nyt se on tulossa ohjelmistokehitykseenkin.

Lean on peräisin Toyotan autotehtaalta, jossa tunnistettiin länsimaille uusia piirteitä ja jotka syntetisoitiin "lean production"-paradigmaksi. Toyota oli kehittänyt uudenlaista ajattelua, jolla se pystyi olemaan nopea ja joustava tuotannossaan (nopea valmistettavan mallin tai sen konfiguraation vaihto, nopea valmisitus), edullinen (mahdollisimman vähäiset pääoma- ja virhekustannukset), tehokas (virheetön, optimoitu tuotanto ja logistiikka, ei oppimiskäyrää) ja laadukas (virheetömät tuotteet).

Kaikki tämä oli radikaalisti uutta eurooppalaisille 1990-luvun alussa, vaikka Toyota oli kehittänyt toimintaansa jo vuosikymmeniä.

Lean production koskee (Warnecke & Hüser 1993) eniten tuotekehitystä, hankintoja, tuotantojohtoa ja jossain määrin myynnin jälkeisiä palveluja. Eli voitaisiinkin puhua vaikka "lean managementista" tai "lean industrystä", koska valmistus on vain osa, jota käsite koskee! Niinpä nimi lyhennettiin pelkäksi leaniksi.

Lean productionille on tyypillistä (Bullinger et al 1993):

- parannettu koordinaatio
- organisaation ja teknologian integraatio

Ja siksi suuri tehokkuus. Koko nimi tulee siitä, että ollaan tehokkaampia pienemmillä resursseilla.

Mistä nämä sitten syntyvät? (Kts. tarkemmin Bullinger et al 1993, Warnecke & Hüser 1993 ja Womack et al 1990.)

Lähtökohdat ja yleistavoitteet:

- Hylätty perinteisen massatuotannon näkemys pelkän teknologian ylivoimasta ja teknologiavetoisesta kehityksestä. Inhimilliset resurssit koetaan määräävänä tuotantotekijänä.
- Asiakaskeskeinen laatuajattelu.
- Kohdentuminen markkinoilla oleviin tarpeisiin ja asiakkaaseen.
- Näkemys: aina on parannettavaa. Vain paras on kylliksi hyvää.
- Jatkuva tuotantotekijöiden kehittäminen (jatkuva parantaminen, koulutustoiminta).
- Ennaltaehkäisevä laadunhallinta: kehitetään prosessin laatua eikä testausta. Laatu integroituu prosesseihin.
- Alihankinta tähtää pitkäjänteiseen kehitykseen. Alihankkijat integroituu osa tuotantoprosessia.

Johtamisnäkemys:

- Hajautettu johtaminen ja vastuu.
- Tiimit.
- Funktionaalisen (toiminnan mukaan jaetun) organisaation sijaan prosessi-orientoitunut. Suuntautuneisuus asiakkaiden tilausten (toiveiden) täyttämiseen.
- Koko tuotanto nähdään yhtenä prosessina.

Tuotantotekniikka:

- Joustavaa tekniikkaa.
- Tehdään kokonaisuuksia.
- Suunniteltu tuotantotoiminnan kokonaisuuden kannalta.
- Prosessipohjainen tuotannonohjaus ja logistiikka (materiaalien ja tuotteiden oikea-aikainen siirtely).



Tuotantohenkilöstö:

- Laajoja työtehtäviä.
- Kaikki tuntevat ryhmän kaikki tehtävät.
- Aktiivista, jatkuvaa oman työn tehokkuuden ja laadun parantamista kehittämistyöryhmissä.
- Tietoisia asiakkaasta (sekä ulkoinen asiakas että seuraava työvaihe).

Toimintatavat:

- Käytössä työkaluja analyttisyyden ja luovuuden tukemikseksi.
- Muutoksia aiheuttamalla tuotantojärjestelmää "ravistellaan" kohti optimitasoa.
- Tuotanto on mukana tuotesuunnittelussa.

Jne...

Yksi yleinen teema on "hukka". Sen rooli on leanissa tärkeä, mutta ei mitenkään muusta eroavan kriittinen (ainakaan nykyään, kun tehtaot, tekivät autoja tai softaa, ovat "luonnostaan" ihan eri tyyliä kuin toisen maailmansodan jälkeen). Leanin hukkalajeja ovat:

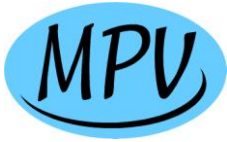
- Ylituotanto (tehdään tavaraa varastoon enemmän kuin on tarvetta)
- Odottaminen (edellisessä työvaiheessa ei kyetä toimittamaan tavaraa – asiakas odottaa softaa, testaaja testattavaa...)
- Tarpeettomat liikkeet (ml. softan siirtäminen tarpeettomasti eri osapuolten välillä)
- Kuljetus (auton osien turha liikuttaminen trukilla paikasta toiseen... ohjelmia ei onneksi tarvitse lähettellä postitse CD-kiekolla ja megatavut liikkuvat yleensä ripeästi. Silti turha softan siirtäminen on virhe.)
- Yliprosessointi tai vääränlainen prosessointi (turha hinkkaus, jos laatu on jo riittävän hyvä)
- Tarpeettomat varastot (kaikenlaiset keskeneräisen työn välivarastot ovat hyvin hankalia, kun tehdään fyysisiä tuotteita. Ja digitaalisen tiedonkin varastot kasvavat...)
- Virheet. (!!!)
- Hyödyntämätön henkilöstön luovuus. (Menetetty mahdollisuudet!)

Mutta ennen kaikkea leanin avainsana on positiivinen virtaus, flow. Sitä etsitään ensisijaisesti! Positiivisuus ja usko huomiseen ovat leanin tunnusmerkkejä – eikä lamamasennus, jossa ankeana etsitään prosessin nurkista pieniä säästöjä.

Lean ei ole patologinen hukanpoistokone. Olennaista on kokonaisuus – ja sen vuoksi saatetaan lisätäkin hukkaa! Esimerkiksi: työntekijä ajattelisi heti, että työntekijä, jolla on välillä loppoaikaa, joutuu odottamaan tekemistä ja se on hukkaa. Pitää saada käyttöaste sataan prosenttiin! Mutta ihmiset eivät toimi sillä tavalla.

Kokonaisuuksissa kaikki vaikuttaa kaikkeen.

Lean on kokonaisuus, josta ei voi ottaa vain joitain paloja sanoen, että tässä on "lean"...



Lean on monenlaista

Lean on monta erilaista mutaatiota ihmisten mielissä

Leanin historia on siis Toyotan autotehtaalla, jossa se muodosti kokonaisvaltaisen johtamis- ja tuotantojärjestelmän. Kun lean tuotiin teollisuuteen, se oli omanlaisensa olio – paljon unohdettiin, mutta mukaan tuotiin lisää akuutteja asioita ja niitä jäsennettiin mm. eurooppalaisten tapojen mukaisesti (esim. johtamisen ja tuotantoperiaatteiden selkeämpi erottelu). Taas ohjelmistokehitykseen viettäessä teksti on taas muuttunut ja moni kriittisen tason alkuperäinen viesti on väistämättä muuttunut. On hyvä, että ajattelu kehittyi, mutta jos se kehittyi väkisin vääntäen, olosuhteet jatkolle eivät ole parhaat mahdolliset.

Usein esimerkiksi leanissa nostetaan esille hukkan vähentäminen ja se vaikuttaa siivoustalkoilta, vaikka se on vain pieni detalji leanissa. Joskus taas nostetaan ”flow” ja logistiikka pääasemaan.

Tätä lean siis ei ole: mekanistista management-by-perkele-hukkajahtia, jossa esimerkiksi testaus ja dokumentointi ovat koko ajan vaarassa joutua viemäriin pesuveden mukana. Lean on päinvastoin intellektuaalista ja rikasta toimintaa, joka ei hae nopeita säästöjä, vaan pitkän tähtäimen optimia.

Kotitehtävä: mieti, miksi hukkajahti korostuu niin vahvasti ja enemmän kuin laatujohtaminen?

Pick your lean!

Erilaisia leaneja on siis liikkeellä ja prosessi-ihmisten kannattaa katsella kaikkia niitä avoimin silmin. Kuitenkin, jotkin näistä leanin olemuksista ovat parempia kuin toiset:

- Johtamisjärjestelmä ja toiminnan laadun paradigma (+++)
- Yksi Toyotan tuotantojärjestelmän piirre (+++)
- Siitä tunnistettu paradigma tuotannon johtamiseen (+++)
- Strategia tuotannon kehittämiseen (++)
- Näkökulma tehokkaaseen toimintaan (+++)
- Tapa yksinkertaistaa tuotantoa ja humanisoida työtä (++)
- Rationalisoinnin ajatuskehikko (???)
- Säästötoimien kainalosauva (- - -)
- Tapa ”tieteistää” ketterän ohjelmistokehityksen puutteita (- - -)

Toyotan tuotantojärjestelmän filosofia ja toiminta

”Toyotan tapa” on siis se johtamis- ja tuotantojärjestelmän kokonaisuus (Toyotalla: Toyota Production System, TPS), josta lean alun perin tunnustettiin ja jäsenettiin. Sen keskeiset periaatteet on esitetty kuvattu The Toyota Way (suom. Toyotan tapaan), mutta nykyään tietenkin tiivistetysti Wikipediassa.

Toyotan tavan 14 periaatetta (vain otsikkotasolla):

Osa I: Pitkän tähtäimen filosofia

1. Tee päätöksiä pitkän tähtäimen filosofian pohjalta, jopa lyhyen tähtäimen taloudellisten tavoitteiden kustannuksella.

T Pitkä tähtäin tuo ihmisille stabiilin mission, jota ilman ei organisaatio jaksaa. Toyotalla on tavoitteena olla oppiva organisaatio, joka pystyy mukautumaan muutoksiin ja menestyy jatkossakin. Juuri tämä on testauksen keskeinen haaste sekä toimintana että organisaatioina.

Osa II: Oikea prosessi tuottaa oikeat tulokset

2. Luo jatkuva prosessinvirtaus tuodaksesi ongelmat esille.

- Ylituotanto (tehdään tavaraa varastoon enemmän kuin on tarvetta)
- Odottaminen (edellisessä työvaiheessa ei kyetä toimittamaan tavaraa – asiakas odottaa softaa, testaaja testattavaa...)
- Tarpeettomat liikkeet (ml. softan siirtäminen tarpeettomasti eri osapuolten välillä)
- Kuljetus (auton osien turha liikuttaminen trukilla paikasta toiseen... ohjelmia ei onneksi tarvitse lähettellä postitse CD-kiekkolla ja megatavut liikkuvat yleensä ripeästi. Silti turha softan siirtäminen on virhe.)
- Yliprosessointi tai vääränlainen prosessointi (turha hinkkaus, jos laatu on jo riittävän hyvä)
- Tarpeettomat varastot (kaikenlaiset välivarastot ovat hyvin hankalia, kun tehdään fyysisiä tuotteita. Ja digitaalisen tiedonkin varastot kasvavat...)
- Virheet.
- Hyödyntämätön henkilöstön luovuus. (Menetetty mahdollisuus!)

T Virtauksessa työn kulku on jatkuvaa ilman odotuksia. Softan kulku testaukseen ja testausvaiheiden läpi on sujuvaa ja automatisoitua. Niin kuin se nykyään pyrkii olemaan – ja ketterässä toiminnassa usein sujuvampaa kuin ei-ketterässä. Tehokkaat asiakirjamallit, tietojärjestelmien käynnistäminen projekteille yms. ovat toiminnan tärkeää virtausta. Ympäristöjen hallinta ja softien asennus ja konfigurointi testiympäristöihin on asia, jossa virtaus ei usein ole vielä optimaalista.

T Virtaus punnitaan katkoskohdissa ja kun asiat muuttuvat. Tekniikan pitää olla joustavaa niin kokoonpanolinjalla kuin testauksessa. Kevyet tietojärjestelmät, jotka saadaan kaikkien käyttöön hetkessä eikä kuukaudessa, testausohjelmien vapaa saatavuus, testiympäristöjen helppo konfiguroitavuus jne...

T Miten virtaus muka nostaa asiat esille? Tiedon virtaus, viestintä, on se virtauksen laji, joka tekee tämän! Ja viestintä on aina yksi ohjelmistokehityksen ja testauksen murheenkrynejä.

T Hukan poisto on aina hyödyllistä, vaikka se ei olekaan pääasia. Tuplatastaus, priorisointi, turha suunnittelutyö – suunnitellaan asioiden testaus kerran; suunnitellaan testaustavat kerran, eikä aina uudestaan. Virheet ovat hukkaa. Testausprosessin virheitä tunnustetaan esimerkiksi katselmoineissa ja analysoimalla prosessin läpäisseitä ohjelmiston virheitä.

3. Käytä imujärjestelmiä välttääksesi ylituotantoa.

T Ideana on tarjota virtauksen ”alavirtaan” prosessin asiakkaille sitä, mitä ne haluavat, koska ne haluavat ja oikein annosteltuna. Eli testataan sellaista, mihin on tarvetta. Määrä-asiassa korostuu priorisoinnin (vakavuus, riski) tuoma mahdollisuus ottaa vastaan tärkeitä asioita vain kohtuullinen määrä. Mutta ajoitus on oleellinen asia – ainakaan testauksen tuottama tieto ei saa myöhästyä...

4. Tasapainota työmäärää (työskentele mieluummin kuin kilpikonna, älä kuin jänis)

T Työmäärä ja ihmisten jaksaminen on tärkeää. Testauksessa työmäärän vaihtelu on perinteinen ongelma. Hopealuotia asiaan ei tietenkään ole, mutta siihen on mahdollista vaikuttaa ihan tavallisilla testauksen kehittämisen keinoilla. Toyotalla hommaa tehtäisiin siten, että tuotetta aletaan valmistaa, kun tilauksia on riittävästi, jolloin päästää suoraan täyteen vauhtiin. Testauksessa ei sellainen odotusperiaate oikein toimi... Joka tapauksessa testaukseen sopii hyvin olla kilpikonna hyperaktiivisen jäniksen sijaan.

5. Luo kulttuuri, jossa pysähdytään korjaamaan ongelmia, jotta laatu saataisiin kuntoon heti ensimmäisellä kerralla.

T Toyotalla laatu on ykkösasia ja testauksen pitäisi olla samaa mieltä. Olennaista tässä ei ole lupa pysäyttää liukuhintaa, vaan se, että saadaan tieto asioiden toimimattomuudesta: jos testausympäristö ei toimi tai tietoliikenne tökkii tms. – asiat pitää viedä heti päälliköille ja ryhtyä toimenpiteisiin. Ongelmiin pitää varautua ja tietää heti, miten toimitaan kun niitä ilmenee. Ja mitä tehdään, kun huomataan, että yksikkötestaus onkin unohtunut? Sama juttu. Ongelmat on korjattava heti.

T Virheet korjataan siellä, missä ne syntyvät eikä niitä siirretä eteenpäin. Jokaisessa softan tekemisen vaiheessa tunnustetaan virheet ja korjataan ne mahdollisimman pian. Eli vaatimusmäärittelyn virheet etsitään mahdollisimman hyvin siinä vaiheessa ja yksikkötestaus pyrkii löytämään ne virheet, jotka on mahdollista löytää. Tässä viedään järkevyyssajattelua normaalia pidemmälle, sillä ideana on oikeasti täydellisyys. Mutta on ymmärrettävä, että kaikilla testaustasoilla ja -vaiheilla on merkityksensä.

T Ongelmien analysointi ja poistaminen, jotta ne eivät toistu, on oleellista. Bugien analysointi on vielä kovin heikkoa. Kaikki virheet olisi periaatteessa mahdollista analysoida kunnolla ja ryhtyä toimiin niiden välttämiseksi. Käytännössä kannattaa aloittaa tämä työ bugeista, joiden elinkaari on pitkä... eli hyväksymistestauksen ongelmista ja vastaavista. Lessons learned on prosessin ongelmien tunnustamisen paikka. Jatkuva toiminnan ongelmien raportointi ja käsittely. Laatuihmiset pitää ottaa enemmän mukaan projekteihin!

6. Standardoidut tehtävät ja prosessit ovat jatkuvan parantamisen ja työntekijöiden voimauttamisen perusta.

T Vakaat, koetellut prosessit ovat kaiken toiminnan peruslähtökohta, niin myös testauksen. Mutta kun jotain on standardoitu, sitä pitää voida parantaa. Ja sitten se uusi paras tapa standardoidaan läpi organisaation. Tässä tarvitaan projektit ylittävää prosessityötä.

7. Käytä visuaalista tuotannonohjausta, jotta ongelmat eivät jää piiloon.

T Graafien, liikennevalojen yms. käyttö lisääntykin jatkuvasti. Toyota korostaa, että tekstuaalisten raporttien pitää olla tiiviitä (yhdeksi sivulle – faksiaikana max. A3). Ketterien tiimien käyttämä seinätaululla oleva tehtävien hallinta lappuja aina seuraavaan sarakkeeseen siirtämällä on puhdasoppista leania ja testauksellekin sovitettavissa.

8. Käytä ainoastaan luotettavaa, perusteellisesti testattua teknologiaa, joka palvelee ihmisiä ja prosesseja.

T Tämä pätee sekä testausteknologiaan että toiminnan infrastruktuuriin, joka on yksi yleinen ongelma-alue. Mahdollisten testausohjelmistojen määrän lisääntyessä on niiden arviointiin panostettava, jotta löydetään pitkällä tähtäimellä palvelevia ratkaisuja. Lähtökohtana pitää yleensä olla ensin manuaalinen ratkaisu, joka sitten välineistetään ja automatisoidaan.

Osa III: Tuota arvoa organisaatiolle kehittämällä ihmisiä

9. Kasvata johtajia, jotka tuntevat työn perusteellisesti, noudattavat filosofiaa ja opettavat sitä muille.

T Testauksenkin johtajia – aloittaen testauspäälliköistä -- pitäisi kasvattaa omasta talosta. Tämä merkitsee ”organisatorista virtausta” ja oman oppimisen kumuloitumista. Periaate näkyy systemaattisessa ihmisten kasvattamisessa erilaisia urapolkuja pitkin.

10. Kehitä poikkeuksellisen eteviä ihmisiä ja tiimejä, jotka noudattavat yrityksen filosofiaa.

T Filosofian relevantteja lajeja ovat arvojen yms. lisäksi ainakin laatu- ja testauspolitiikka. Toyota korostaa eri toimintoja yhdistäviä tiimejä ratkomaan yhteisiä ongelmia. Testauksen näkökulmasta tällaiset ovat esim. laatutiimejä, prosessikehitystiimejä ja vastaavia. Ihmisiä pitää opettaa tiimityöhön. Testauksessa tämä opetustarve korostuu ketterässä kehityksessä ja sen tiimeissä.

11. Kunnioita yhteistyökumppaneilla ja alihankkijoilla laajennettua verkostoa tarjoamalla heille haasteita ja auttamalla heitä kehittymään.

T Testauksenkin alihankkijat ovat oman toiminnan laajennus ja pitää ottaa vahvasti mukaan yhteiseen toimintaan. Haastaminen ja tiukat vaatimukset ovat merkki kunnioituksesta!

Osa IV: Jatkuva juuriongelmiin ratkaiseminen kehittää organisaation oppimista

12. Mene itse paikan päälle katsomaan ymmärtääksesi tilanteen perusteellisesti.

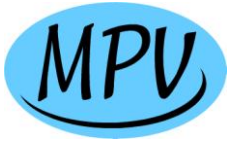
T Johtajien pitää selvittää asioiden oikea laita. Testauksessa tämä merkitsee esim. sitä, että ei pelkästään lueta testiraportteja, vaan jutellaan ohjelmiston tilanteesta testaajien kanssa. Ja mikäpä estää kokeilemasta pahimpia bugeja. Testaajien pitää luoda johdolle mahdollisuus siihen. Testausalihakijoiden työtä voidaan käydä seuraamassa, eikä vain tekemässä perinteisiä auditointeja.

13. Tee päätöksiä hitaasti yksimielisyyden pohjalta kaikkia vaihtoehtoja perusteellisesti harkiten; toteuta päätökset nopeasti.

T Päätöksiä tehtäessä pitää olla vaihtoehtoja, joista valitaan paras rauhassa. Niiden mielipiteitä pitää kuunnella, joihin päätöksen vaikuttavat ja on löydettävä konsensus. Esimerkki: päätökset testauspolitiikasta, testiympäristön varustelusta, vikakannasta, organisoitumisesta. Ennen kaikkea on käytettävä aikaa sen miettimiseen, mitä on tärkeintä testata hyvin tarkasti. Se ei ole ketterää työtä, vaan systemaattista aivotyötä. Toyotallakin tämä periaate pätee kaikkeen päätöksentekoon, lähtien tuotekonseptin suunnittelusta, joten testauksellakin on ”varaa” vaatia tilaa joka suuntaan.

14. Tee yrityksestäsi oppiva organisaatio väsymättömän itsearvioinnin ja jatkuvan parantamisen avulla.

T Kun testauksen toimintajärjestelmä on kunnossa, kehitys ei pääty siihen, vaan alkaa siitä!



Suomalainen valmistavan teollisuuden lean

Metalliteollisuuden keskusliiton julkaisussa Lean-toiminta, suomalaisten yritysten kokemuksia, Veikko Kajaste ja Timo Liukko esittelevät lean-toiminnan periaatteet:

Johtamisperiaatteet

- Ihmiset tekevät tuloksen yhteistyössä
- Selkät tavoitteet, mittarit ja seuranta
- Pitkälle menevä tulosvastuullinen delegointi ja hajautettu organisaatio
- Monitaitoinen ja yritteliäs henkilöstö
- Asiakkaan, omistajan ja henkilöstön yhteinen etu

Näissä näkyy suomalainen teollinen jämäkkä konsensus-ajattelu. Se on tärkeä viesti myös 2000-luvun lean-sovelluksille. Asiat pitää lokalisoida kulttuuriin monella tasolla: kieli, käytännöt, politiikka.

Toimintatavat

- Perustana asiakkaalle tuleva lisäarvo
- Huomion kiinnittäminen kokonaisuuteen
- Jatkuva kustannusrakenteen keventäminen
- Tiedonkulun suoruus ja avoimuus
- Jatkuva oman toiminnan kehittäminen
- Joutavat ja nopeat toimitukset
- Henkilöresurssien järkevä yhdistäminen nykyaikaiseen tuotantotekniikkaan

Kokonaisuuden merkitys sanotaan suuremmin kuin Toyotalla. Syy tähän lienee se, että Toyotalla on ollut vähemmän vaaraa paikallisista optimoinneista kuin Suomessa, jossa yksikkö, osasto ja tiimi ovat huonompia tekemään yhteistyötä toisen vastaavan tai vaikka alihankkijan kanssa. Ja koska 1990-luvulla elettiin kaiken päälle tulosjohtamisen aikaa (mikä jatkuu toki edelleen eri muodossa), jokainen pyrkii optimoimaan oman tuloksensa.

Koska kyse oli paljolti alihankintoihin perustuvasta kokoonpanotyöstä, logistiikka ja just-in-time (JIT; aiemmin JOT) olivat leimaavia "lean"-piirteitä

T Testauksenkaan ei pitä ajatella omaa etuaan, vaan ohjelmistokehityksen ja organisaation ja asiakkaan etua.

T Tiedonkulku on asia, joka koskee testausta kahdessa mielessä. Sen tuottaman tiedon pitää olla hyvää ja sen pitää myös saada hyvää tietoa.

Lean ohjelmistokehitys

Tietenkään näin monimuotoista asiaa ei voi esittää muutamassa sivussa – jokainen toyotalainen hämmästelisi sellaista – vaan siihen tarvitaan hyvä kirja, esim. keskeisten gurujen Mary & Tom Poppendickin *Implementing Lean Software Development: From Concept to Cash*.

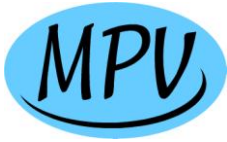
Jyvälle asioista pääsee lyhyemmän, seitsemään sivun tekstin avulla:

<http://www.poppendieck.com/papers/LeanThinking.pdf>

...ja keskeiset konkreettiset näkemykset ohjelmisto-Leanin valtavirran periaatteista esittelee selkeimmin teeman Wikipedia-sivu: http://en.wikipedia.org/wiki/Lean_software_development

Mutta kuitenkin, ohjelmistokehitys-leanin keskeiset periaatteet ovat (tässä Poppendickien kirjan mukaan; Wikipediassa vastaava lista on hieman eri muodossa):

- Eliminoi hukka.
 - Perinteiset hukkalajit, mutta yksilöidymmin: turha ”kirnuaminen” testauksen ja korjaamisen parissa, turhat vaatimukset, viipeet, byrokratia, huono viestintä.
 - Infrastruktuurissa voi olla paljon hukkaa. Vaikkapa isoja järjestelmiä bugien käsittelyyn. Jos bugeja ei tehtäisi, niitä järjestelmiäkään ei tarvittaisi!
 - Kuitenkin on yllättävää, että hukasta tehdään pääasia, kun se on Toyotallakin mukana vain pienessä ja ei-itseisarvoisessa roolissa. Leanissa on taustalla myös se vanha laatuviisaus, että patologioille herkkiä asioita ei koskaan pidä mennä kehittämään suoraa! Eli jos halutaan tehokkuutta, ei pidä kehittää tehokkuutta, vaan laatua. Tämä pätee suoraan hukkajahtiin. Mutta vanha viisaus ei ole aina muotia...
 - Testauksen hukkia käsiteltiin hieman jo edellä.
- Rakenna laatu sisään
 - Laatua tuotetaan ensimetreiltä alkaen. Softa on kunnossa kaikin tavoin kaikilta piirteiltään; teknologiat ovat luotaisesti laadukkaita.
 - Bugit pitää torjua ennakolta, eikä korjata.
 - Kun bugi löytyy, sen syy poistetaan – ideana on se, että tietty bugi ei sitten toistu enää koskaan
 - Tämä laatuviisaus viittaa rikkaaseen laadunvarmistuskulttuuriin, jossa myös testaajat ovat ”testausta” laajemmassa roolissa ja auttavat muita suunnittelemaan ja toteuttamaan erinomaisen kokonaisuuden.
- Luo tietämystä
 - Vaiheittainen kehittäminen, kokeilu; softan tekeminen on oppimisprosessi.
 - Tämä oppiminen on projektin puitteissa tapahtuvaa, eikä lean-kulttuurin pitkäjänteistä organisaation oppimista.
 - Testaus on yksi tapa oppia järjestelmän käyttäytymisestä.
- Viivytä sitoutumista
 - Ideana on se, että tietämys kasvaa koko ajan ja siksi päätökset kannattaa tehdä mahdollisimman myöhään tai toiminnot voidaan vaikka pudottaa pois vaikka loppumetreillä.
 - On huomattava, että suunnittelu ei ole sama asia kuin sitoutuminen. Leanissa suunnitellaan huolella, mutta ymmärretään, että suunnitelmien muuttuminen on ihan normaalia. Ja että nippeliasioihin ei pidä sitoutua liian aikaisin.
 - Testauksessa pätee sama periaate.
- Toimita nopeasti
 - Tämä on suoraan ketterän toiminnan periaate. Asiakkaalle halutaan antaa arvoa niin pian kuin mahdollista. Ja toinen periaate: softaa pitää toimittaa niin nopeasti, että asiakas ei ehdi muuttaa mieltään... (!)
 - Perinteinen lean lähtisi siitä, että nopeuden eteen ei uhrata muita asioita – tärkeimpänä laatua. Silloin toimitusajankohdan määrittää a) asiakkaan pyyntö ja b) se, milloin tuote on toimituskuntoinen
 - Testauksessa ohjelmistokehittäjät ja muut sidosryhmät ovat asiakkaita ja testauksen tuottama tieto on toimitus. Mitä nopeammin saadaan jaettua testauksen ”palautetta”, sitä parempi. Ja siihenhän pyritäänkin: ripeää testausta ja viestintää ja tiedon jakamista reaaliaikaisia kanavia pitkin.



- Kunnioita ihmisiä
 - Toiminta menee aseman edelle.
 - Tiimille suorat asiakasyhteydet ja päättäväisyyttä projektin asioista. Tämä on tiimijattelun ja leanin perinteinen malli ja suora kopio ketterästä kehittämisestä.
 - **T** Kunnioitettakoon testauksessakin samoilla periaatteilla. Kaikki ovat tärkeitä, mutta tärkeintä on se, mitä tekoja tehdään kokonaislaadun eteen, eivätkä esimerkiksi sertifikaattidiplomit tai virkaikä. Kunnioituksen pohjalla ei missään nimessä saa olla mekanistinen mittausmenetelmä.
- Optimoiminen kokonaisuus
 - Optimoidaan koko softan tekemisen arvoketju tilauksesta toimitukseen.
 - Softa on enemmän kuin osiensa summa! Kokonaisuuksia ei ymmärrä, jos katselee vain omaa pientä palastaan. Kokonaisuuden ymmärtäminen vaatii viestintää ja aikaa.
 - **T** Tämä on testauksessa oleellista. Siksi pitääkin tunnustaa tuotteenkäyttäjät, tarkoitus, käyttötavat, ympäristö jne., vaikka kaikkea ei olisi missään kuvattunakaan. Testauksessa on kasvanut ymmärrys kokonaisvaltaisten ”putkien” testaamisesta ja kaikkien ohjelmiston laatuominaisuuksien tunnistamisesta ja varmistamisesta.

Poppendickien ajattelussa on virkistävää historian tunteminen ja klassisten laatujohtamisen ajattelumallien ja menettelyjen näkyminen (”virkistävää tuulahdus 1990-luvulta!”). Ei siis pidä ajatella, että ohjelmistokehityksen lean olisi vain edellämainittujen periaatteiden luettelo; sen rivien välissä on paljon muutakin, mutta ohjelmistokehityksen erilaisista konteksteista johtuen niitä ei voida esitellä geneerisellä tavalla kovin suoraviivaisesti – eikä varsinkaan tässä muistiossa. Kuitenkin, tällaisia ajattelun piirteitä esiintyy pitkien heidän kirjaansa:

- Tervejärkisyys
- Asioiden pitäminen kevyinä
- Juurisyyden poistaminen (poistetaan bugit, niin kaikki muu on paljon helpompaa)
- Systemaattisuus ja analyyttisten menetelmien käyttö
- Tapakulttuurin kyseenalaistaminen, näkökulmien avoimuus
- Luottaminen onnistumiseen

Nämä kaikki ovat alkuperäisen leanin olennaisia piirteitä.

Leanin ja agilen suhde

Agilea väitetään usein leaniksi ja toisinpäin. Tampereen teknillisen yliopiston ja Pirkanmaan Tietojenkäsittely-yhdistyksen Projektinhallintapäivillä vuonna 2009 James Coplien murskasi hienosti tämän käsityksen (esitys on osoitteessa http://www.cs.tut.fi/tapahtumat/projektinhallinta09/JimCoplien_12082009.pdf)

Seuraavassa on muutama olennainen pointti esityksestä.



Aluksi leanin ja agilen piirteiden vertailua:

Lean	Agile
Toimijat kutsutaan	Kaikki mukaan
Leanissa ajatellaan	Ketterässä toimitaan (ketterässä EI ajatella)
Leanissa katsellaan eteenpäin	Agilessa taaksepäin
Korkea läpäisykyky	Vähäinen latenssi
Yksi aikajänne	Monta aikajännettä
Suunniteltu	Reaktiivinen
Kehittyvät maailman mallit	Jaetut maailman mallit
Prosessi	Ihmiset

Agilen refaktorointi on oikeasti hukkaa! Mutta suunnittelun aikainen lisätyö ei ole.

Erilaisilla malleilla on erilaiset parhaat sovellusalueet

Coplienin mukaan agile sopii komplekseihin systeemeihin, joita on vaikea hallita, jotka ovat enemmän kuin osiensa summa. Mutta lean sopii komplisoituihin systeemeihin, joissa on vain paljon osia, mutta ne ovat ymmärrettäviä (toim. huom: kuten vaikka henkilöauton valmistaminen....) Niissä kokonaisuus = osiensa summa. (Huom! Suomessa ei ole hyviä toisistaan eroavia termejä complex:lle ja complicated:lle).

Ei ole olemassa yhtä johtamismallia, joka sopii kaikkiin tilanteisiin! ...Mutta ihmiset haluavat yksinkertaisia paketteja, joita voi myydä, kouluttaa...

Agilen ja leanin suhtautuminen muutamiin asioihin

Standardit

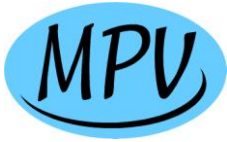
- Agile: tutki ja sovelle; jokainen saa tehdä mitä vaan, olet voimautettu
- Lean jos on ongelma, tutki standardeja etukäteen – standardoidut työtavat voimauttavat työntekijää (tämä on Toyota Wayn periaate 6. Toyota Way on kuvattu kirjassa "The Toyota Way", mutta tiivistelmä siitä on Wikipedian sivulla http://en.wikipedia.org/wiki/The_Toyota_Way)

Ajattelu:

- Agile: Ota muutos vastaan
- Lean: Tee päätös hitaasti (koska ihmiset ajattelevat hitaasti) ja etsi konsensus ja vasta päätöksen jälkeen toimi nopeasti (Toyota Way:n periaate 13)

Erikoistuminen

- XP: Ei erikoistumista, ei koodin omistajuutta. Scrum: Moniosaava tiimi
- Lean: Käytä vuosia jokaisen yksilön kasvattamisessa syvälliseen osaamiseen (Toyota Wayn periaate 10). Siis vahvaa erikoistumista.



Muutostyöt (rework):

- Agile: Refaktorointi kompensoi arkkitehtuurisuunnittelun puutteita – mutta eihän se onnistu käytännössä
- Lean: Uudelleentyöstäminen suunnittelussa antaa lisäarvoa; uudelleen tekeminen tuotannossa on hukkaa. Prototyypit ovat järkevää lisätyötä.

Milloin pitää tehdä päätöksiä:

- Agile: Aikaiset päätökset ovat usein vääriä, joten tee päätös niin myöhään kuin mahdollista
- Lean: Jos päätöksen jättää niin myöhäiseen hetkeen, että se vaikuttaa toisiin päätöksiin, se tuottaa lisätyötä. Päätöksiä ei saa lykätä.

<End of Coplien...>

Leanin periaatteet tuovat kuitenkin toimintaan sitä metatason kyvykkyyttä, jota ketteryys tarvitsee! Ne ovat aivan traditionaalaisia hyvän organisaation piirteitä: osaaminen, oppiminen, virheiden poistaminen... Kun homma on kunnossa, toimintajärjestelmä on eheä ja hyvin suunniteltu ja tuotantojärjestelmä on kevyt, muutokset ovat helppoja.

Miten "lean" saadaan aikaan?

Lean näyttää kokoelmalta, jossa on lähes kaikki hyvinä pidetyt yrityksen piirteet. Suuri osa niistä on jo pitkään ollut eurooppalaisessa "ihannekapsäkissä" (esimerkiksi Warnecke & Hüser 1993). Integroituneisuus on tärkeää, sillä "yhden asian liikkeet" joutuvat käytännössä aina integraatio-ongelmiin, kun ne törmäävät yrityskulttuurillisiin epäsovittomuuksiin. Mutta voidaanko sitten "lean"ia ottaa kokonaisuudessaan käyttöön, sillä sehän vaihtaisi kerralla koko yrityskulttuurin!

On olemassa myös muita integroituja malleja, joissa on otettu mm. pohjoismaisia kulttuurillisia reunaehtoja huomioon. Lean on kuitenkin hyvä esimerkki, koska se on hyvin dokumentoitu, ja maailmalla koeteltu ratkaisu ja paikallisten kehitysten pohja.

Mutta missä määrin lean voi olla elävää todellisuutta yrityksissä? Jos yrityksessä väitetään sovellettavan leania, se voi johtua kolmesta syystä:

- 1) Se on vähitellen orgaanisesti kasvanut yritykseen ja "leimattu" ulkopuolisten toimesta (esimerkiksi Toytota voisi kuulua tähän kastiin).
- 2) Se on hyvin huolellisesti, tietoisesti ja kokonaisvaltaisesti kehitetty yritykseen.
- 3) Lean ei ehkä olekaan todellisuutta yrityksessä. Siitä voi olla joitakin piirteitä olemassa, mutta käsite on vain lainattu kuvaamaan pyrkimyksiä joustavuuteen, asiakassuuntautuneisuuteen ja tehokkuuteen.

Kakkoskohta on kiinnostava. Miten sellainen, joka jossain ympäristössä on syntynyt orgaanisesti, voidaan toisessa ympäristössä kasvattaa samanlaiseksi edes korkean tason periaatteidensa osalta? Ja jos voidaan, onko se paras ratkaisu?

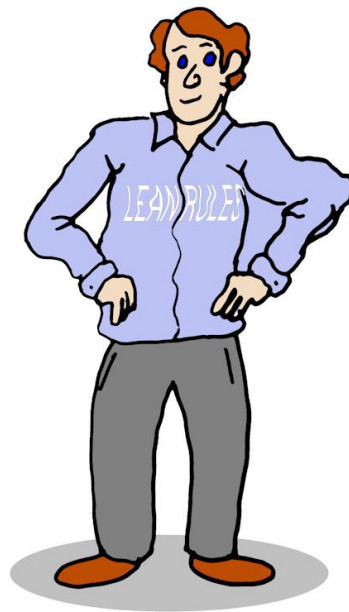
T Testauksen näkökulmasta leaniin pätee sama periaate kuin automatisointiin. Kumpaakaan ei pidä alkaa harrastaa ennen kuin toiminnan laatu on kunnossa ja tarkkaan tiedetään mitä ollaan tekemässä. Muuten ei homma onnistu ja riskitaso on liian suuri.

Diagnoosi

Nyt on aika vetää yhteen, millaiset leanin piirteet ovat relevantteja testauksen näkökulmasta. Eli millaiset piirteet ovat vahvistamisen arvoisia ja auttavat testaaajien pärjäämistä leanin kontekstissa – ja kenties muutenkin parantavat testauksen pitkän tähtäimen suorituskykyä. Huom! Tämä luku on vasta ensimmäinen versio aiheesta ja kattaa vain joitakin leanin elementtejä.

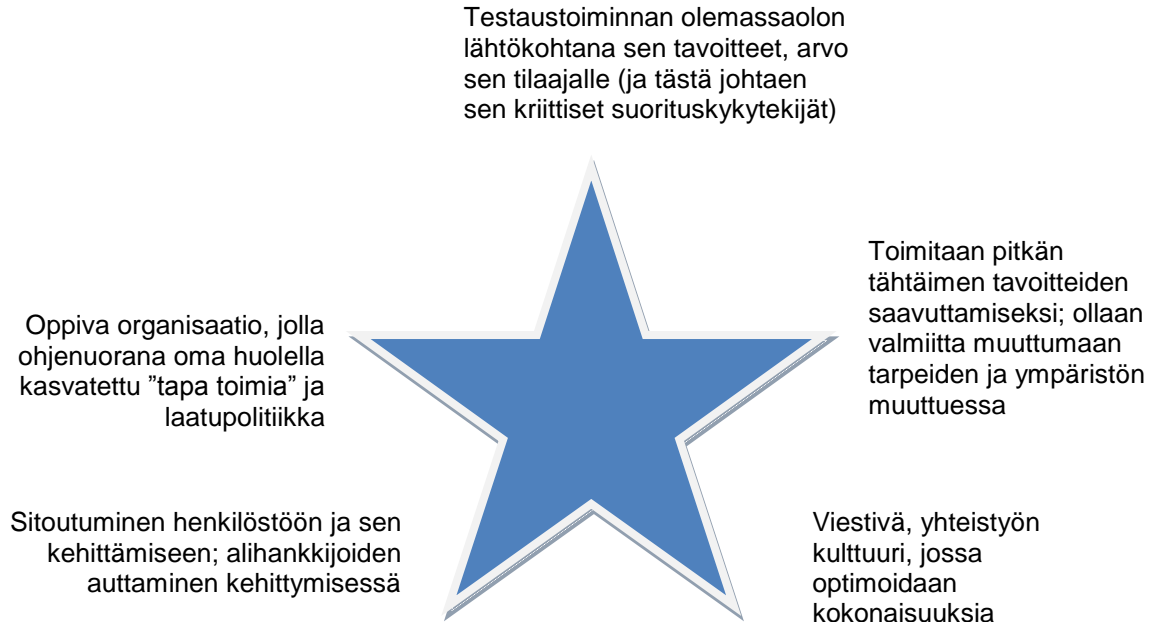
Liinin testaaajan piirteet

- Asiakaslähtöinen (ottaa huomioon ulkoiset ja sisäiset asiakkaat)
- Laatutietoinen
- Holistinen
- Tiimipeluri
- Sitoutunut

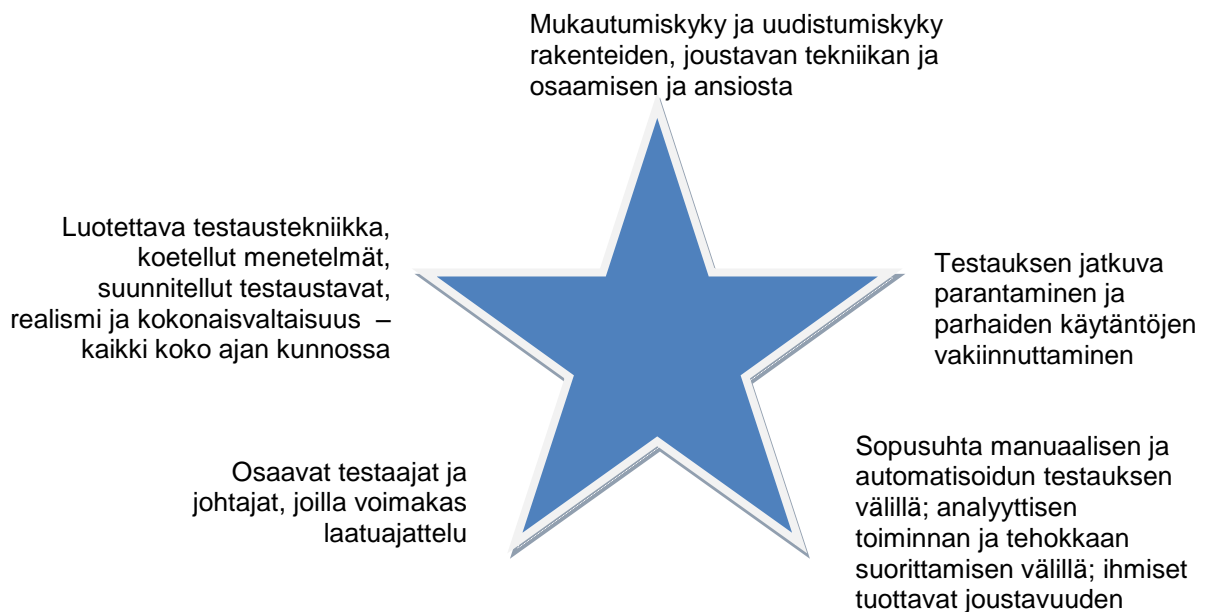


- Osaava
- Toiminnan kehittäjä, puuttuu heti ongelmiin
- Harkitseva ja analyttinen; samaan aikaan käytännöllinen
- Nopea
- Hyvät välineet

Liinin testaustoiminnan filosofia



Liinin testaustoiminnan toimintajärjestelmä



Hukattoman testaustoiminnan piirteitä

Originaalin leanin hukkalajien reflektointia testauksen näkökulmasta. Taulukko ei ole kattava, vaan lähinnä esimerkkejä. Kussakin prosessissa ja organisaatiossa on omat hukkinsa, jotka pitää tunnistaa avoimin mielin ja silmin.

Toyotalaisen leanin hukkia	Testauksessa	Eväitä virtaukseen
Ylituotanto	Testataan ja raportoidaan kerralla liikaa virheitä, eikä niitä ehditä korjata	<ul style="list-style-type: none"> • Testaus on jatkuvaa, bugeja ja muita havaintoja raportoidaan jatkuvasti ja tasapuolisesti eri kehittäjille • Johdon raportointia kevennetään siten, että olennaiset asiat ehditään käsitellä
Odottaminen	<ul style="list-style-type: none"> • Testaus käynnistyy hitaasti. • Prosessin vuoksi testauksen tuloksia joudutaan odottamaan • Testausresurssit eivät riitä • Ympäristöjä tai protoja ei ole • Ympäristöjen avaamisessa testaukseen viiveitä • Testaajat odottavat testattavaa • Asioiden edistyminen odottaa katselmointia ja hyväksyntää 	<ul style="list-style-type: none"> • Valmiit ajatukset siitä, miten testataan • Testauksen ja kehittämisen rinnakkaisuus • Softaa toimitetaan testaajille inkrementteittäin, mahdollisimman pian (kuitenkin siten, että sen testaaminen onnistuu) • Testaukseen varataan riittävät resurssit ja niistä pidetään kiinni • Tiimillä mahdollisimman itsenäiset ympäristöt • Katselmointien ja hyväksynnän nopeuttaminen; tiimille päätösvalta menettelyistä
Tarpeettomat liikkeet	<ul style="list-style-type: none"> • Testausprosessissa tarpeettoman monia vaiheita • Testaajilla monta projektia tai liian erilaisia testustehtäviä 	<ul style="list-style-type: none"> • Testausprosessi mahdollisimman yksinkertainen (mutta ei liian yksinkertainen!) • Testaajien työsuunnittelulla keskittymistä tukeva kuormitus
Tarpeeton kuljetus	<ul style="list-style-type: none"> • Testiympäristöjen suuri määrä • Tarpeettomat testausalihankinnat • Huono osapuolten pääsy ympäristöihin • Hitaat tietoliikenneyhteydet 	<ul style="list-style-type: none"> • Testiympäristöjen konfigurointi eri tarpeisiin • Tehokkaampi ympäristöjen hyödyntäminen ja yhteiskäyttö • Tietoliikenteen nopeuttaminen • Datat ja ohjelmien esiprosessointi generoinnin tai muuntamisen sijaan • Asennettavuuden ja konfiguroinnin parantaminen
Yliprosessointi tai vääränlainen prosessointi	<ul style="list-style-type: none"> • Sekundaaristen softan alueiden liiallinen painotus • Liiallinen ajankäyttö raporteihin ja suunnitelmiin 	<ul style="list-style-type: none"> • Testauksen priorisointi ja dynaaminen ohjaus • Raporttien automaattinen generointi • Valmiiksi pohditut lähestymistavat • Tehokkaat asiakirjamallit • Testimateriaalien uudelleenkäyttö

Toyotalaisen leanin hukkia	Testauksessa	Eväitä virtaukseen
Tarpeettomat varastot	<ul style="list-style-type: none"> Eri ympäristöissä liikkuu buildeja, joiden tila ja tarkoitus ei ole selvä Suunnitelmia, ohjeita, testitapauksia, skriptejä yms. ei hallita Keskeneräisiä suunnitelmia yms. odottamassa katselmointia ja hyväksyntää Koneissa, levyillä ja tietokannoissa on vanhaa taukkaa, mikä hidastaa tai tukkii tai aiheuttaa häiriöitä Testikoneiden käyttö muihin tarkoituksiin 	<ul style="list-style-type: none"> Buildien merkintä, merkintöjen ymmärrettävyys Ympäristöjen siivous Prosessissa olevien buildien järkevä vähentäminen Katselmointiprosessien nopeuttaminen; testaaajille päätösvaltaa työtavoista ilman ulkoista katselmointia Testausartefaktien versionhallinta tietojärjestelmillä Siivous- ja arkistointitalkoot projektien ja niiden pitkien vaiheiden päätteeksi
Virheet.	<ul style="list-style-type: none"> Testausprosessin häiriöt ja ongelmat Infrastruktuurin ongelmat Testauksen suunnittelun virheet Viestinnän virheet Jne... 	<ul style="list-style-type: none"> Prosessin robustiuden kehittäminen Infrastruktuurin valvonta ja hallinta Tietojärjestelmien professionaalinen (ITIL-tyylinen) ylläpito Jne...
Hyödyntämätön henkilöstön luovuus.	<ul style="list-style-type: none"> Toimintaa ei paranneta yhteistyössä Ei ole lupaa tai tapaa antaa ideoita 	<ul style="list-style-type: none"> Testauksen jatkuva parantaminen Parannusideoiden kerääminen Lessons learned Kyselyt Vastaantulevien ongelmien välitön analysointi – pysähtyminen!

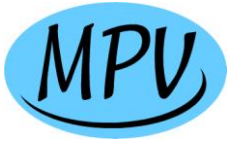
Lopuksi

Lean on oikeastaan klassista **laatujohtamista** ja tarjoaa kaivatun vastapainon ketterälle toiminnalle. Samalla se tukee parhaita ketterän maailman periaatteita ja käytäntöjä ja tekee samaa myös muille järkeville paradigmoille – esimerkiksi laatujohtamisa- ja kypsyysoajattelu.

Lean on mitä korkeinta **humanismia ja idealismia** karussa maailmassa! Toyota näki, että oppiva organisaatio, älyllinen yhteistyö jatkuvassa parantamisessa ja täydellistämässä, itsensä ja alaisten kehittäminen, sopiva työtahti, koneet ihmisen palvelijana jne... ovat tapa saavuttaa tuloksia.

Lean ei ole menetelmä, vaan joukko periaatteita. Lean tullaan ohjelmistokehityksessä ymmärtämään kunnolla vasta 2020-luvulla, sillä ohjelmistokehityksen maailman kypsymiselle tarvitaan muutenkin vielä vuosia.

Mutta lean on vaarallista, sillä se antaa paljon mahdollisuuksia väärinkäytölle. Esimerkiksi lyhyen tähtäimen hukkajahti on leanin mutaatio, joka voi olla hyvin vahingollinen. Leanin tähtäin on aina ensin pitkällä aikajänteellä ja yksittäisen hukkaintoilijan sijaan päätökset tehdään harkitusti yhteistyössä, päätöksenteon apuvälineitä käyttäen.



Lähteitä

Balancing the tension between lean and agile? James Coplien, Scrum Training Institute. Esitys Tampereen teknillisen yliopiston ja Pirkanmaan Tietojenkäsittely-yhdistyksen Projektinhallintapäivillä vuonna 2009 (http://www.cs.tut.fi/tapahtumat/projektinhallinta09/JimCoplien_12082009.pdf)

Veikko Kajaste ja Timo Liukko. 1994. Lean-toiminta -- suomalaisten yritysten kokemuksia, Metalliteollisuuden keskusliitto MET 6/94.

Likert, Jeffrey K. 2004. The Toyota Way. 14 management principles from the world's greatest manufacturer. (Suom. Toyotan tapaan).

Poppendick, Mary & Tom. 2007. Implementing Lean Software Development: From Concept to Cash. (Heillä on muitakin kirjoja; tämä nähdään usein hyväksi kokonaisteokseksi, joka päivittää joitakin vuosia aiempia ajatuksia, mutta ei mene vielä liian erikoistuneisiin teemoihin.)

Poppendick, Mary. Principles of Lean Thinking. <http://www.poppendieck.com/papers/LeanThinking.pdf>

Seven Principles of Lean Software Development. <http://agilesoftwaredevelopment.com/leanprinciples>

Tuominen, Kari. 2010. Lean. Kohti täydellisyyttä. readme.fi. 168 s.

Warnecke, H.J. & Hüser, M. 1993. Lean production. 9 s.

Wikipedia-sivut:

- The Toyota Way http://en.wikipedia.org/wiki/The_Toyota_Way
- Lean manufacturing http://en.wikipedia.org/wiki/Lean_manufacturing
- Toyota Production System http://en.wikipedia.org/wiki/Toyota_Production_System
- Lean Software Development http://en.wikipedia.org/wiki/Lean_software_development

Womack, J.P.; Dones, D.T. & Roos, D. 1990. The machine that changed the world. New York, Rawson Associates, MacMillan Publishing Company. 323 s.