

# Ohjelmistotuotanto

Matti Luukkainen ja ohjaajat Antti, Pooki, Riku, Sini, Taneli

syksy 2024

Luento 9

25.11.2024

## Kurssipalaute

- ▶ Kurssipalaute
  - ▶ Kurssilla lopussa kerättävän palautteen lisäksi ns. jatkuva palaute <https://norppa.helsinki.fi>

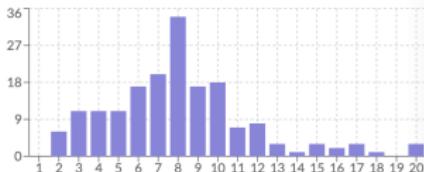
# Kurssipalaute

- ▶ Kurssipalaute
  - ▶ Kurssilla lopussa kerättävän palautteen lisäksi ns. jatkuva palaute <https://norppa.helsinki.fi>
- ▶ laskarien työläys?
- ▶ tavoitteena
  - ▶ viikot 1 ja 2: 8 tuntia
  - ▶ viikko 3: 6 tuntia
  - ▶ viikot 4-7: 4 tuntia

## part2

submissions: 180  
hours average: 8.0  
exercises average: 11.2

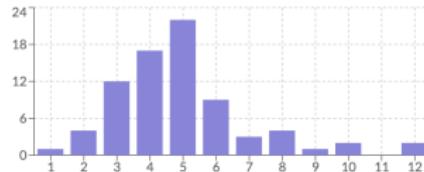
### used hours



## part4

submissions: 78  
hours average: 5.0  
exercises average: 5.8

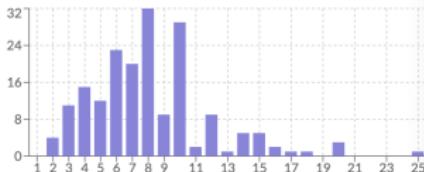
### used hours



## part1

submissions: 188  
hours average: 8.1  
exercises average: 15.6

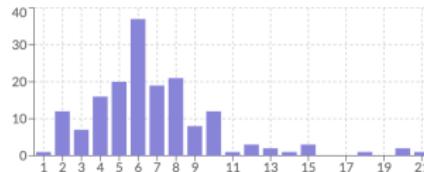
### used hours



## part3

submissions: 171  
hours average: 6.8  
exercises average: 6.9

### used hours



## Kurssin viimeiset 3 viikkoa

- ▶ Miniprojektit jatkuvat
  - ▶ tällä ja ensiviikolla asiakastapaaminen

## Kurssin viimeiset 3 viikkoa

- ▶ Miniprojektit jatkuvat
  - ▶ tällä ja ensiviikolla asiakastapaaminen
- ▶ Loppudemot (jokainen ryhmä osallistuu toiseen demoista)
  - ▶ ke 11.12. klo 12-14 B123
  - ▶ to 12.12. klo 12-14 CK112
  - ▶ ilmoittautuminen käynnissä

## Kurssin viimeiset 3 viikkoa

- ▶ Miniprojektit jatkuvat
  - ▶ tällä ja ensiviikolla asiakastapaaminen
- ▶ Loppudemot (jokainen ryhmä osallistuu toiseen demoista)
  - ▶ ke 11.12. klo 12-14 B123
  - ▶ to 12.12. klo 12-14 CK112
  - ▶ ilmoittautuminen käynnissä
- ▶ Normaali luento ti 3.12.
- ▶ Vierailuluennot
  - ▶ ti 26.11. Kristiina Vainio Houston Inc ja Ville Nordberg Trail openers
  - ▶ ma 2.12. Jami Kousa ja Aleksandr Tereshchenko Unity
  - ▶ ma 9.12. Kasper Hirvikoski Unity ja Luomus Fossil Database Team
  - ▶ ti 10.12. Hannu Kokko Elisa

# Kurssin viimeiset 3 viikkoa

- ▶ Miniprojektit jatkuvat
  - ▶ tällä ja ensiviikkolla asiakastapaaminen
- ▶ Loppudemot (jokainen ryhmä osallistuu toiseen demoista)
  - ▶ ke 11.12. klo 12-14 B123
  - ▶ to 12.12. klo 12-14 CK112
  - ▶ ilmoittautuminen käynnissä
- ▶ Normaali luento ti 3.12.
- ▶ Vierailuluennot
  - ▶ ti 26.11. Kristiina Vainio Houston Inc ja Ville Nordberg Trail openers
  - ▶ ma 2.12. Jami Kousa ja Aleksandr Tereshchenko Unity
  - ▶ ma 9.12. Kasper Hirvikoski Unity ja Luomus Fossil Database Team
  - ▶ ti 10.12. Hannu Kokko Elisa
- ▶ Koe torstaina 19.12. klo 7-14 moodlessa (koeaika 4h)

# Ohjelmiston elinkaaren vaiheet

## Ohjelmiston elinkaaren vaiheet

- ▶ Riippumatta tyylistä ja tavasta jolla ohjelmisto tehdään, ohjelmistojen tekemiseen kuuluu
  - ▶ vaatimusten analysointi ja määrittely
  - ▶ suunnittelu
  - ▶ toteutus
  - ▶ testaus/laadunhallinta
  - ▶ **ohjelmiston ylläpito**

## Ohjelmiston elinkaaren vaiheet

- ▶ Riippumatta tyylistä ja tavasta jolla ohjelmisto tehdään, ohjelmistojen tekemiseen kuuluu
  - ▶ vaatimusten analysointi ja määrittely
  - ▶ suunnittelu
  - ▶ toteutus
  - ▶ testaus/laadunhallinta
  - ▶ **ohjelmiston ylläpito**
- ▶ Ohjelmistot ovat suurimman osan elinajastaan ylläpitovaiheessa

## Ohjelmiston elinkaaren vaiheet

- ▶ Riippumatta tyylistä ja tavasta jolla ohjelmisto tehdään, ohjelmistojen tekemiseen kuuluu
  - ▶ vaatimusten analysointi ja määrittely
  - ▶ suunnittelu
  - ▶ toteutus
  - ▶ testaus/laadunhallinta
  - ▶ **ohjelmiston ylläpito**
- ▶ Ohjelmistot ovat suurimman osan elinajastaan ylläpitovaiheessa
- ▶ Jos ensimmäinen versio julkaistaan nopeasti, ovat ketterät ohjelmistoprojektit “jatkuvassa” ylläpitovaiheessa

## Muutama ylläpitovaiheen kannalta oleellinen asia

- ▶ Varmuuskopiointi, ja niiden palautus

## Muutama ylläpitovaiheen kannalta oleellinen asia

- ▶ Varmuuskopiointi, ja niiden palautus
- ▶ Sovelluksen lokit
- ▶ Analytiikka
- ▶ Virheiden monitorointi

# Sovelluksen lokit

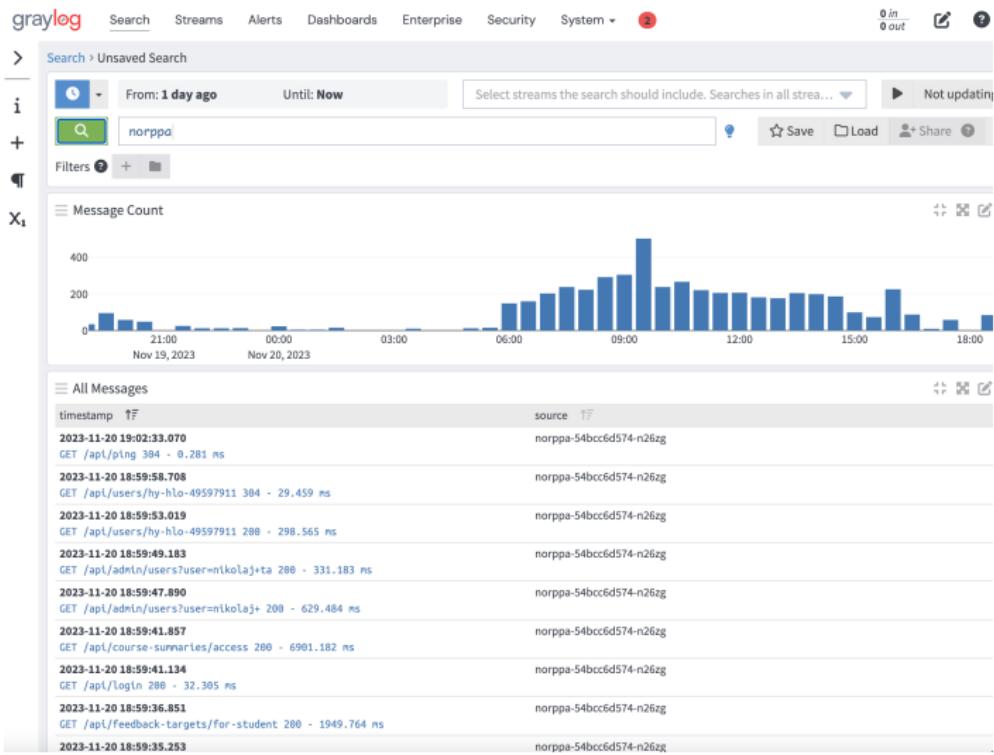
- Sovellusten tulee tulostaa lokiviestejä erilaisista mielenkiintoisista tilanteista

```
80      const processingTime = (Date.now() - processingStart).toFixed(0)
81      const totalTime = (Date.now() - loopStart).toFixed(0)
82      loopStart = Date.now()
83
84      logger.debug(`[UPDATERLOOP]`, {
85        url,
86        offset,
87        items: currentData.length,
88        requestTime,
89        processingTime,
90        totalTime,
91      })
92
93      count += currentData.length
94      offset += limit
95
96    } catch (e) {
97      if (!e.isLoggedIn) {
98        logError('Unknown updaterloop error:', e)
99      }
100    }
101  }
102
103  const duration = Date.now() - start
104  logger.info(
105    `[UPDATER] Updated ${count} items at ${((duration / count).toFixed(
106      4,
107    ))ms/item}, total time ${((duration / 1000).toFixed(2))s}`,
108  )
109
110 module.exports = mangleData
```

func mangleData

# Sovelluksen lokit: Graylog

- Sovellusten lokit tulee kerätä paikkaan, mistä niitä helppo tarkastella



# Analytiikka: Grafana

► Lokeja voidaan hyödyntää erilaiseen analytiikkaan



# Virheiden monitorointi: Sentry

► Virhetilanteista voidaan muodostaa hälytyksiä

norppa-sentry

+ Add a bookmark

Sentry APP 3:47 PM Today

Error

Request failed with status code 504

PALAUTE-6 via Send a notification for new issues | Today at 3:47 PM

Resolve... Ignore Select Assignee...

Sentry APP 4:38 PM

TypeError

(n=g.slice()).splice is not a function. (In '(n=g.slice()).splice(r,1);' '(n=g.slice()).splice' is undefined)

PALAUTE-G via Send a notification for new issues | Today at 4:38 PM

Resolve... Ignore Select Assignee...

Sentry APP 6:20 PM

N+1 Query

```
db - INSERT INTO "feedback_target_logs"  
("id","data","feedback_target_id","user_id","created_at","updated_at") VALUES  
(DEFAULT,$1,$2,$3,$4,$5) RETURNING  
"id","data","feedback_target_id","user_id","created_at","updated_at","user_id","feedback_tar  
get_id";
```

PALAUTE-19 via Send a notification for new issues | Today at 6:19 PM

Resolve... Ignore Select Assignee...

Sentry APP 6:42 PM

ApplicationError

# Virheiden monitorointi: Sentry

Issues > PALAUTTE-G

TypeError handleChange(@mui/material/ToggleButtonGroup/ToggleButtonGroup)

● (n=g.slice()).splice is not a function. (In 'n=g.slice().splice(r,1);', 'n=g.slice()' is undefined)

Details Activity User Feedback Attachments Tags All Events Merged Issues Replays

Event ID: d7980f80 Nov 20, 2:38 PM

Tags: @helsinki.fi, Mobile Safari Version: 17.1.1, iOS Version: 17.1.1, iPhone Model: iPhone

browser: Mobile Safari 17.1.1, browser.name: Mobile Safari, device: iPhone, device.family: iPhone, environment: production, handled: yes, level: error, mechanism: instrument, os: iOS 17.1.1, os.name: iOS, release: e3906c94a555, url: https://norppa.helsinki.fi/course-summary, user: id:hy-hlo-1456049

Stack Trace

TypeError  
(n=g.slice()).splice is not a function. (In 'n=g.slice().splice(r,1)', 'n=g.slice()' is undefined)

mechanism: instrument, handled: true, function: addEventListener, handler: bound Ht, target: EventTarget

JS  
./node\_modules/@mui/material/ToggleButtonGroup/ToggleButtonGroup.js in handleChange at line 117:16

```
112     }
113     const index = value && value.indexOf(buttonValue);
114     let newValue;
115     if (value && index >= 0) {
116       newValue = value.slice();
117       newValue.splice(index, 1);
118     } else {
119       newValue = value ? value.concat(buttonValue) : [buttonValue];
120     }
121     onChange(event, newValue);
122   };

```

./node\_modules/@mui/material/ToggleButton/ToggleButton.js in <object>.onClick at line 123:15

## Moar buzzwords

2000-luvun alussa **Ketterä/Agile**

## Moar buzzwords

2000-luvun alussa **Ketterä/Agile**

==>

## Moar buzzwords

2000-luvun alussa **Ketterä/Agile**

==>

2010-luvulla **Lean**

Lean: taustaa

## Lean: taustaa

- ▶ Syntynyt Toyotan tuotannon ja tuotekehityksen menetelmistä

## Lean: taustaa

- ▶ Syntynyt Toyotan tuotannon ja tuotekehityksen menetelmistä
- ▶ Japanissa 2. maailmansodan jälkeen jälleenrakennusbuumi, mutta pääoma ja raaka-aineita niukalti

## Lean: taustaa

- ▶ Syntynyt Toyotan tuotannon ja tuotekehityksen menetelmistä
- ▶ Japanissa 2. maailmansodan jälkeen jälleenrakennusbuumi, mutta pääoma ja raaka-aineita niukalti
- ▶ Huomataan että laadun parantaminen nostaa tuottavuutta
  - ▶ mitä vähemmän tuotteissa ja tuotantoprosesseissa ongelmia, sitä enemmän tuottavuus kasvaa

## Lean: taustaa

- ▶ Syntynyt Toyotan tuotannon ja tuotekehityksen menetelmistä
- ▶ Japanissa 2. maailmansodan jälkeen jälleenrakennusbuumi, mutta pääomaa ja raaka-aineita niukalti
- ▶ Huomataan että laadun parantaminen nostaa tuottavuutta
  - ▶ mitä vähemmän tuotteissa ja tuotantoprosesseissa ongelmia, sitä enemmän tuottavuus kasvaa
- ▶ Japanilaisiin yrityksiin tuli vahva *laatua korostava kulttuuri*

## Lean: taustaa

- ▶ Syntynyt Toyotan tuotannon ja tuotekehityksen menetelmistä
- ▶ Japanissa 2. maailmansodan jälkeen jälleenrakennusbuumi, mutta pääomaa ja raaka-aineita niukalti
- ▶ Huomataan että laadun parantaminen nostaa tuottavuutta
  - ▶ mitä vähemmän tuotteissa ja tuotantoprosesseissa ongelmia, sitä enemmän tuottavuus kasvaa
- ▶ Japanilaisiin yrityksiin tuli vahva *laatua korostava kulttuuri*
- ▶ Resurssien niukkuus johti *Just In Time (JIT)* -tuotantomalliin
  - ▶ aloitetaan tuotteen valmistus vasta kun ostaja tehtyy tilauksen
  - ▶ vastakohtana perinteinen massatuotanto: tehdään paljon tuotteita varastoon

# *Just In Time (JIT) -tuotantomalli*

## *Just In Time (JIT)* -tuotantomalli

- ▶ Pyrkimys saada tuote tilauksen jälkeen mahdollisimman nopeasti kuluttajalle:
  - ▶ *Lyhyt läpimenoaika* (lead time) tilauksesta toimitukseen

## *Just In Time (JIT)* -tuotantomalli

- ▶ Pyrkimys saada tuote tilauksen jälkeen mahdollisimman nopeasti kuluttajalle:
  - ▶ *lyhyt läpimenoaika* (lead time) tilauksesta toimitukseen
- ▶ Etuja massatuotantoon verrattuna
  - ▶ *läpimenoaika lyhyt*: laatuongelmat paljastuvat nopeasti
  - ▶ asiakkaiden muuttuviin tarpeisiin oli helppo vastata

## *Just In Time (JIT) -tuotantomalli*

- ▶ Pyrkimys saada tuote tilauksen jälkeen mahdollisimman nopeasti kuluttajalle:
  - ▶ *lyhyt läpimenoaika* (lead time) tilauksesta toimitukseen
- ▶ Etuja massatuotantoon verrattuna
  - ▶ *läpimenoaika lyhyt*: laatuongelmat paljastuvat nopeasti
  - ▶ asiakkaiden muuttuviin tarpeisiin oli helppo vastata
- ▶ Massatuotanto pyrkii optimoimaan yksittäisten työntekijöiden ja koneiden työpanosta
  - ▶ ideaali käyttöaste on koko ajan 100%

## *Just In Time (JIT) -tuotantomalli*

- ▶ Pyrkimys saada tuote tilauksen jälkeen mahdollisimman nopeasti kuluttajalle:
  - ▶ *lyhyt läpimenoaika* (lead time) tilauksesta toimitukseen
- ▶ Etuja massatuotantoon verrattuna
  - ▶ *läpimenoaika* lyhyt: laatuongelmat paljastuvat nopeasti
  - ▶ asiakkaiden muuttuviin tarpeisiin oli helppo vastata
- ▶ Massatuotanto pyrkii optimoimaan yksittäisten työntekijöiden ja koneiden työpanosta
  - ▶ ideaali käyttöaste on koko ajan 100%
- ▶ JIT:issä optimoinnin kohde tuotteen *läpimenoaika*
  - ▶ eliminoidaan kaikki **hukka** (waste), joka ei edesauta työn nopeampaan virtausta tilauksesta asiakkaalle

# Toyota production system (TPS)

## Toyota production system (TPS)

- ▶ Vuodesta 1965 alkaen Toyota alkoi kutsua toimintatapaansa Toyota Production Systemiksi

## Toyota production system (TPS)

- ▶ Vuodesta 1965 alkaen Toyota alkoi kutsua toimintatapaansa Toyota Production Systemiksi
- ▶ Ensimmäiset englanninkieliset julkaisut vuodelta 1977

# Toyota production system (TPS)

- ▶ Vuodesta 1965 alkaen Toyota alkoi kutsua toimintatapaansa Toyota Production Systemiksi
- ▶ Ensimmäiset englanninkieliset julkaisut vuodelta 1977
- ▶ MIT:in tutkijat alkoivat 1980-luvun lopussa tutkimaan TPS:ää
  - ▶ Ianseeraavat nimikkeen *lean-* (eli *virtaviivainen*) tuotanto

# Toyota production system (TPS)

- ▶ Vuodesta 1965 alkaen Toyota alkoi kutsua toimintatapaansa Toyota Production Systemiksi
- ▶ Ensimmäiset englanninkieliset julkaisut vuodelta 1977
- ▶ MIT:in tutkijat alkoivat 1980-luvun lopussa tutkimaan TPS:ää
  - ▶ Ianseeraavat nimikkeen *lean-* (eli *virtaviivainen*) tuotanto
- ▶ 1990 ilmestynyt kirja *The Machine That Changed the World* toi leanin laajempaan tietoisuuteen

## Toyota production system (TPS)

- ▶ Vuodesta 1965 alkaen Toyota alkoi kutsua toimintatapaansa Toyota Production Systemiksi
- ▶ Ensimmäiset englanninkieliset julkaisut vuodelta 1977
- ▶ MIT:in tutkijat alkoivat 1980-luvun lopussa tutkimaan TPS:ää
  - ▶ Ianseeraavat nimikkeen *lean-* (eli *virtaviivainen*) tuotanto
- ▶ 1990 ilmestynyt kirja *The Machine That Changed the World* toi leanin laajempaan tietoisuuteen
- ▶ Toyota alkoi 2000-luvulla käyttämään sisäisesti nimitystä lean

## Toyota production system (TPS)

- ▶ Vuodesta 1965 alkaen Toyota alkoi kutsua toimintatapaansa Toyota Production Systemiksi
- ▶ Ensimmäiset englanninkieliset julkaisut vuodelta 1977
- ▶ MIT:in tutkijat alkoivat 1980-luvun lopussa tutkimaan TPS:ää
  - ▶ Ianseeraavat nimikkeen *lean-* (eli *virtaviivainen*) tuotanto
- ▶ 1990 ilmestynyt kirja *The Machine That Changed the World* toi leanin laajempaan tietoisuuteen
- ▶ Toyota alkoi 2000-luvulla käyttämään sisäisesti nimitystä lean
- ▶ Viimeisen 25 vuoden aikana on ilmestynyt suuri määrä kirjoja, jotka kuvaavat Toyotan tuotantojärjestelmää
  - ▶ Ehkä paras Jeffrey Liker: *The Toyota Way* (2001)

## Lean tuotanto ja tuotekehitys

- ▶ Alussa lean oli Toyotalla *tuotantoa* (production) optimoiva toimintatapa, nykyään myös *tuotekehitystä* (development)

## Lean tuotanto ja tuotekehitys

- ▶ Alussa lean oli Toyotalla *tuotantoa* (production) optimoiva toimintatapa, nykyään myös *tuotekehitystä* (development)
- ▶ Tuotanto ja tuotekehitys luonteeltaan hyvin erilaisia ja niihin sovellettavat lean-käytänteet eroavatkin paikoin

# Lean tuotanto ja tuotekehitys

- ▶ Alussa lean oli Toyotalla *tuotantoa* (production) optimoiva toimintatapa, nykyään myös *tuotekehitystä* (development)
- ▶ Tuotanto ja tuotekehitys luonteeltaan hyvin erilaisia ja niihin sovellettavat lean-käytänteet eroavatkin paikoin
- ▶ Sovellettu monille aloille, myös ohjelmistotuotantoon
  - ▶ Mary ja Tom Poppendieck: *Lean software development, an agile toolkit (2003)*

## Lean tuotanto ja tuotekehitys

- ▶ Alussa lean oli Toyotalla *tuotantoa* (production) optimoiva toimintatapa, nykyään myös *tuotekehitystä* (development)
- ▶ Tuotanto ja tuotekehitys luonteeltaan hyvin erilaisia ja niihin sovellettavat lean-käytänteet eroavatkin paikoin
- ▶ Sovellettu monille aloille, myös ohjelmistotuotantoon
  - ▶ Mary ja Tom Poppendieck: *Lean software development, an agile toolkit* (2003)
- ▶ Scrum saanut vaikutteita leanista
  - ▶ kehittäjät Ken Schwaber ja Jeff Sutherland tunsivat leanin

# Lean tuotanto ja tuotekehitys

- ▶ Alussa lean oli Toyotalla *tuotantoa* (production) optimoiva toimintatapa, nykyään myös *tuotekehitystä* (development)
- ▶ Tuotanto ja tuotekehitys luonteeltaan hyvin erilaisia ja niihin sovellettavat lean-käytänteet eroavatkin paikoin
- ▶ Sovellettu monille aloille, myös ohjelmistotuotantoon
  - ▶ Mary ja Tom Poppendieck: *Lean software development, an agile toolkit* (2003)
- ▶ Scrum saanut vaikutteita leanista
  - ▶ kehittäjät Ken Schwaber ja Jeff Sutherland tunsivat leanin
- ▶ Leania sovelletaan nykyään laajasti
  - ▶ terveydenhuoltoon, pankkitoimintaan, yliopiston hallintoon, ...
  - ▶ vaikea enää sanoa mitä lean tarkoittaa...

Lean TPS:n mukaan...

## Lean TPS:n mukaan...

- ▶ Leania havainnollistaa *lean thinking house*

# Lean thinking house

Sustainable shortest lead time, best quality and value (to people and society), most customer delight, lowest cost, high morale, safety		
<p><b>Respect for People</b></p> <ul style="list-style-type: none"><li>- don't trouble your 'customer'</li><li>- "develop people, then build products"</li><li>- no wasteful work</li><li>- teams &amp; individuals evolve their own practices and improvements</li><li>- build partners with stable relationships, trust, and coaching in lean thinking</li><li>- develop teams</li></ul>	<p><b>Product Development</b></p> <ul style="list-style-type: none"><li>- long-term great engineers</li><li>- mentoring from manager-engineer-teacher</li><li>- cadence</li><li>- cross-functional</li><li>- team room + visual mgmt</li><li>- entrepreneurial chief engineer/product mgr</li><li>- set-based concurrent dev</li><li>- create more knowledge</li></ul> <p><b>14 Principles</b></p> <ul style="list-style-type: none"><li>long-term, flow, pull, less variability &amp; overburden, Stop &amp; Fix, master norms, simple visual mgmt, good tech, leader-teachers from within, develop exceptional people, help partners be lean, Go See, consensus, reflection &amp; kaizen</li></ul>	<p><b>Continuous Improvement</b></p> <ul style="list-style-type: none"><li>- Go See</li><li>- kaizen</li><li>- spread knowledge</li><li>- small, relentless</li><li>- retrospectives</li><li>- 5 Whys</li><li>- eyes for waste<ul style="list-style-type: none"><li>* variability, overburden, NVA ... (handoff, WIP, info scatter, delay, multi-tasking, defects, wishful thinking..)</li></ul></li><li>- perfection challenge</li><li>- work toward flow (lower batch size, Q size, cycle time)</li></ul>
Management applies and teaches lean thinking, and bases decisions on this long-term philosophy		

## Lean TPS:n mukaan...

- ▶ Leania havainnollistaa *lean thinking house*

## Lean TPS:n mukaan...

- ▶ Leania havainnollistaa *lean thinking house*
- ▶ Leanilla on
  - ▶ tavoite (**goal**)
  - ▶ perusta (**foundation**)
  - ▶ kaksi peruspilaria (**pillars**) ja
  - ▶ joukko näitä tukevia periaatteita (*14 principles ja product development -periaatteet*)

## Lean TPS:n mukaan...

- ▶ Leania havainnollistaa *lean thinking house*
- ▶ Leanilla on
  - ▶ tavoite (**goal**)
  - ▶ perusta (**foundation**)
  - ▶ kaksi peruspilaria (**pilars**) ja
  - ▶ joukko näitä tukevia periaatteita (*14 principles ja product development -periaatteet*)
- ▶ Lisäksi olemassa joukko leania tukevia työkaluja
  - ▶ näistä kuuluisin lienee *Kanban*

## Lean: tavoite, perusta ja pilarit

### ► Tavoite (**goal**)

- kestävä ja nopea tapa edetä “ideasta asiakkaalle myytyn tuotteeseen”
- työntekijötä ja yhteistyökumppaneita riistämättä
- ylläpitäen korkea laatutaso ja asiakastyytyväisyys

# Lean: tavoite, perusta ja pilarit

## ► Tavoite (**goal**)

- ▶ kestävä ja nopea tapa edetä “ideasta asiakkaalle myytyn tuotteeseen”
- ▶ työntekijöitä ja yhteistyökumppaneita riistämättä
- ▶ ylläpitäen korkea laatutaso ja asiakastyytyväisyys

## ► Perusta (**foundation**)

- ▶ syvälle yrityksen kaikille tasolle juurtunut lean-toimintatapa
- ▶ mitä johtajat soveltavat ja opettavat alaisille

# Lean: tavoite, perusta ja pilarit

- ▶ Tavoite (**goal**)
  - ▶ kestävä ja nopea tapa edetä “ideasta asiakkaalle myytyn tuotteeseen”
  - ▶ työntekijöitä ja yhteistyökumppaneita riistämättä
  - ▶ ylläpitäen korkea laatutaso ja asiakastyytyväisyys
- ▶ Perusta (**foundation**)
  - ▶ syvälle yrityksen kaikille tasolle juurtunut lean-toimintatapa
  - ▶ mitä johtajat soveltavat ja opettavat alaisille
- ▶ Lean rakentuu kahden peruspilarin varaan
  - ▶ jatkuvan parantaminen (**continuous improvement**)
  - ▶ ihmisten kunnioittaminen (**respect for people**)

# Jatkuvan parantaminen (continuous improvement)

## Jatkuvan parantaminen (continuous improvement)

- ▶ The root of the Toyota Way is to be **dissatisfied with the status quo**; you have to ask constantly
  - ▶ “*Why are we doing this?*”

## Jatkuvan parantaminen (continuous improvement)

- ▶ The root of the Toyota Way is to be **dissatisfied with the status quo**; you have to ask constantly
  - ▶ “*Why are we doing this?*”
- ▶ Oleellista
  - ▶ **jatkuvan oppimisen** ilmapiiri ja
  - ▶ ympäristö joka ei pelkästään hyväksy vaan **rohkaisee muutokseen**

Ihmisten kunnioittaminen (respect for people)

## Ihmisten kunnioittaminen (respect for people)

- ▶ Jatkuvaan oppimisen ja parantamisen ilmapiiri mahdollinen vaan, jos toiminta perustuu *ihmisten kunnioittamiseen*.
  - ▶ työntekijöitä kuunnellaan, mentoroidaan ja vastuutetaan
  - ▶ työympäristö mielekäs ja turvallinen
  - ▶ työtehtävät mielekkäitä, annetaan mahdollisuus urakehitykselle

## Ihmisten kunnioittaminen (respect for people)

- ▶ Jatkuvaan oppimisen ja parantamisen ilmapiiri mahdollinen vaan, jos toiminta perustuu *ihmisten kunnioittamiseen*.
  - ▶ työntekijöitä kuunnellaan, mentoroidaan ja vastuutetaan
  - ▶ työympäristö mielekäs ja turvallinen
  - ▶ työtehtävät mielekkäitä, annetaan mahdollisuus urakehitykselle
- ▶ Periaatteen kattaa myös alihankkijat ja loppuasiakkaat
  - ▶ alihankkijoiden kanssa rakennetaan aitoja partneriukseja, joista molemmat hyötyvät sekä opettaa alihankkijoille lean-ajattelua

# Jatkuva parantaminen – arvo ja hukka

## Jatkuva parantaminen – arvo ja hukka

- ▶ TPS:n kehittäjä Taiichi Ohno kuvaa *jatkuvan parantamisen periaatetta* seuraavasti:
  - ▶ We are looking at **time line**, from the moment **customer gives us an order to the point where we collect the cash**
  - ▶ And we are reducing the time line by **reducing the non-value-adding wastes**

## Jatkuva parantaminen – arvo ja hukka

- ▶ TPS:n kehittäjä Taiichi Ohno kuvaa *jatkuvan parantamisen periaatetta* seuraavasti:
  - ▶ We are looking at **time line**, from the moment **customer gives us an order to the point where we collect the cash**
  - ▶ And we are reducing the time line by **reducing the non-value-adding wastes**
- ▶ Keino päästä nopeampaan tuotantosykliin on siis eliminoida *arvoa* (value) tuottamattomia asioita eli *hukkaa* (waste)

## Jatkuva parantaminen – arvo ja hukka

- ▶ TPS:n kehittäjä Taiichi Ohno kuvaa *jatkuvan parantamisen periaatetta* seuraavasti:
  - ▶ We are looking at **time line**, from the moment **customer gives us an order to the point where we collect the cash**
  - ▶ And we are reducing the time line by **reducing the non-value-adding wastes**
- ▶ Keino päästää nopeampaan tuotantosykliin on siis eliminoida *arvoa* (value) tuottamattomia asioita eli *hukkaa* (waste)
- ▶ Mitä arvo ja hukka tarkoittavat?
  - ▶ arvo: asiat ja työvaiheet, mistä asiakas on valmis maksamaan
  - ▶ hukka: kaikki tuotantoon liittyvä, mikä ei tuota asiakkaalle arvoa

## Jatkuva parantaminen – arvo ja hukka

- ▶ TPS:n kehittäjä Taiichi Ohno kuvaa *jatkuvan parantamisen periaatetta* seuraavasti:
  - ▶ We are looking at **time line**, from the moment **customer gives us an order to the point where we collect the cash**
  - ▶ And we are reducing the time line by **reducing the non-value-adding wastes**
- ▶ Keino päästää nopeampaan tuotantosykliin on siis eliminoida *arvoa* (value) tuottamattomia asioita eli *hukkaa* (waste)
- ▶ Mitä arvo ja hukka tarkoittavat?
  - ▶ arvo: asiat ja työvaiheet, mistä asiakas on valmis maksamaan
  - ▶ hukka: kaikki tuotantoon liittyvä, mikä ei tuota asiakkaalle arvoa
- ▶ Lean tunnisti alunperin 7 lähdettä hukalle (**muda**)

## Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Over-production** ylituotanto
  - ▶ Tuotteita tulee valmistaa ainoastaan siinä määrin mitä asiakas on niitä tilannut

## Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Over-production** ylituotanto
  - ▶ Tuotteita tulee valmistaa ainoastaan siinä määrin mitä asiakas on niitä tilannut
- ▶ Ohjelmistojen ylimääräiset toiminnallisuudet voidaan ajatella olevan ylituotantoa
  - ▶ Tutkimuksien mukaan 64% ohjelmistojen toiminnallisuksista on joko ei ollenkaan tai ainoastaan erittäin harvoin käytettyjä

## Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Over-production** ylituotanto
  - ▶ Tuotteita tulee valmistaa ainoastaan siinä määrin mitä asiakas on niitä tilannut
- ▶ Ohjelmistojen ylimääräiset toiminnallisuudet voidaan ajatella olevan ylituotantoa
  - ▶ Tutkimuksien mukaan 64% ohjelmistojen toiminnallisuksista on joko ei ollenkaan tai ainoastaan erittäin harvoin käytettyjä
- ▶ **In-process inventory** välivarastointi
  - ▶ Tähän kategoriaan kuuluu osittain tehty työ ja sen säilöminen

# Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Over-production** ylituotanto
  - ▶ Tuotteita tulee valmistaa ainoastaan siinä määrin mitä asiakas on niitä tilannut
- ▶ Ohjelmistojen ylimääräiset toiminnallisuudet voidaan ajatella olevan yli tuotantoa
  - ▶ Tutkimuksien mukaan 64% ohjelmistojen toiminnallisuksista on joko ei ollenkaan tai ainoastaan erittäin harvoin käytettyjä
- ▶ **In-process inventory** välivarastointi
  - ▶ Tähän kategoriaan kuuluu osittain tehty työ ja sen säilöminen
- ▶ Ohjelmistotuotannossa
  - ▶ tarkka vaatimusmäärittely ominaisuuksille joita ei vielä toteuteta
  - ▶ valmis koodi mikä ei ole vielä testattu tai otettu käyttöön
  - ▶ koodi joka toteuttaa asiakkaan ehkä tulevaisuudessa haluamia toiminnallisuuksia

# Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Extra processing** liikatyö
  - ▶ prosessin pakottamat turhat työvaiheet
  - ▶ “pyörän keksiminen uudelleen”
  - ▶ liian hyvä laatu (?)

## Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Extra processing** liikatyö
  - ▶ prosessin pakottamat turhat työvaiheet
  - ▶ “pyörän keksiminen uudelleen”
  - ▶ liian hyvä laatu (?)
- ▶ **Transportation** tarpeeton materiaalin siirtely

# Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Extra processing** liikatyö
  - ▶ prosessin pakottamat turhat työvaiheet
  - ▶ “pyörän keksiminen uudelleen”
  - ▶ liian hyvä laatu (?)
- ▶ **Transportation** tarpeeton materiaalin siirtely
- ▶ Ohjelmistotuotannossa ns “handoff”, eli jos ohjelmisto
  - ▶ määritellään ja toteutetaan erillisten tiimien toimesta
  - ▶ toteutetaan ja viedään tuotantoon erillisten tiimien toimesta
  - ▶ ...

## Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Motion** työntekijöiden tarpeeton liikkuminen

## Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Motion** työntekijöiden tarpeeton liikkuminen
- ▶ Ohjelmistotuotannossa *task switching*
  - ▶ työskentely yhtäaikaa monessa projektissa
  - ▶ liian monen storyn tekeminen yhtä aikaa

## Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Motion** työntekijöiden tarpeeton liikkuminen
- ▶ Ohjelmistotuotannossa *task switching*
  - ▶ työskentely yhtäaikaa monessa projektissa
  - ▶ liian monen storyn tekeminen yhtä aikaa
- ▶ **Waiting** turha odotus

# Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Motion** työntekijöiden tarpeeton liikkuminen
- ▶ Ohjelmistotuotannossa *task switching*
  - ▶ työskentely yhtäaikaa monessa projektissa
  - ▶ liian monen storyn tekeminen yhtä aikaa
- ▶ **Waiting** turha odotus
- ▶ Ohjelmistotuotannossa
  - ▶ aika, joka joudutaan odottamaan että yrityksen johto hyväksyy vaatimusmäärittelyn
  - ▶ ... testaajat ehtivät testaamaan ohjelman uuden version
  - ▶ ... ylläpito vie sovelluksen uuden version tuotantoon
  - ▶ ... joku mergeää pull requestin

# Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Defects viat**
  - ▶ Koodissa on väkisinkin vikoja jossain tuotannon vaiheessa
  - ▶ Testaaminen ja vikojen havaitseminen vasta tuotannon loppuvaiheessa erittäin epäoptimaalista

# Lean-hukka ohjelmistotuotannon näkökulmasta

- ▶ **Defects viat**
  - ▶ Koodissa on väkisinkin vikoja jossain tuotannon vaiheessa
  - ▶ Testaaminen ja vikojen havaitseminen vasta tuotannon loppuvaiheessa erittäin epäoptimaalista
- ▶ Myöhemmin on ehdotettu alkuperäisten lisäksi uusia, mm.
  - ▶ *Under-realizing people's potential and varied skill, insight, ideas, suggestion*

## Mura–ja muri–tyyppinen hukka

- ▶ Seitsemän *muda*-tyyppisen hukan lisäksi olemassa kaksi muuta hukkatyypia: *mura* ja *muri*

## Mura–ja muri–tyyppinen hukka

- ▶ Seitsemän *muda*-tyyppisen hukan lisäksi olemassa kaksi muuta hukkatyypia: *mura* ja *muri*
- ▶ **Mura:** epäsäännöllisyyttä tai epäyhdenmukaisuutta työtavoissa tai työstettävässä tuotteessa
  - ▶ Esim. toteutettavien user storyjen suuresti vaihteleva koko

## Mura–ja muri–tyyppinen hukka

- ▶ Seitsemän *muda*-tyyppisen hukan lisäksi olemassa kaksi muuta hukkatyypia: *mura* ja *muri*
- ▶ **Mura:** epäsäännöllisyyttä tai epäyhdenmukaisuutta työtavoissa tai työstettävässä tuotteessa
  - ▶ Esim. toteutettavien user storyjen suuresti vaihteleva koko
- ▶ **Muri:** ylikuormitus tai mahdottomat vaatimukset
  - ▶ koneiden käyttäminen 100% kapasiteetilla ilman säännöllistä huoltoa
  - ▶ henkilöstön ylityöllistäminen

## Mura–ja muri–tyyppinen hukka

- ▶ Seitsemän *muda*-tyyppisen hukan lisäksi olemassa kaksi muuta hukkatyypia: *mura* ja *muri*
- ▶ **Mura:** epäsäännöllisyyttä tai epäyhdenmukaisuutta työtavoissa tai työstettävässä tuotteessa
  - ▶ Esim. toteutettavien user storyjen suuresti vaihteleva koko
- ▶ **Muri:** ylikuormitus tai mahdottomat vaatimukset
  - ▶ koneiden käyttäminen 100% kapasiteetilla ilman säännöllistä huoltoa
  - ▶ henkilöstön ylityöllistäminen
- ▶ Seurauksena saattaa olla *muda*-tyyppinen hukka, esimerkiksi välivarastointin tarve tai viat
  - ▶ Heikentää arvon läpivirtausta, eli aikaa, miten kauan user storyltä kestää backlogille asettamisesta valmiiksi tuotteen osaksi

## Jaktuva parantaminen: kaizen

## Jatkuva parantaminen: kaizen

- ▶ **Kaizen** eli jatkuva parantaminen: *optimoidaan toimintaa eliminoimalla asiakkaalle arvoa tuottamatonta hukkaa*

## Jatkuva parantaminen: kaizen

- ▶ **Kaizen** eli jatkuva parantaminen: *optimoidaan toimintaa eliminoimalla asiakkaalle arvoa tuottamatonta hukkaa*
- ▶ Kaikkia työntekijöitä koskeva toimintafilosofia:
  - ▶ As a mindset, it suggests “My work is to do my work **and to improve my work**”
  - ▶ “continuously improve for its own sake”

## Jatkuva parantaminen: kaizen

- ▶ **Kaizen** eli jatkuva parantaminen: *optimoidaan toimintaa eliminoimalla asiakkaalle arvoa tuottamatonta hukkaa*
- ▶ Kaikkia työntekijöitä koskeva toimintafilosofia:
  - ▶ As a mindset, it suggests “My work is to do my work **and to improve my work**”
  - ▶ “continuously improve for its own sake”
- ▶ Kaizen on myös konkreettinen tapa toimia
  - ▶ valitaan toimintatapa/prosesi ja tehdään siitä uusi **toiminnan standardi**
  - ▶ noudatetaan uutta toimintatapaa, kunnes se tunnetaan täysin
  - ▶ mietitään parannuksia, ja luodaan uusi toiminnan standardi
  - ▶ toistetaan ikuisesti...

## Jatkuva parantaminen: kaizen

- ▶ **Kaizen** eli jatkuva parantaminen: *optimoidaan toimintaa eliminoimalla asiakkaalle arvoa tuottamatonta hukkaa*
- ▶ Kaikkia työntekijöitä koskeva toimintafilosofia:
  - ▶ As a mindset, it suggests “My work is to do my work **and to improve my work**”
  - ▶ “continuously improve for its own sake”
- ▶ Kaizen on myös konkreettinen tapa toimia
  - ▶ valitaan toimintatapa/prosesi ja tehdään siitä uusi **toiminnan standardi**
  - ▶ noudatetaan uutta toimintatapaa, kunnes se tunnetaan täysin
  - ▶ mietitään parannuksia, ja luodaan uusi toiminnan standardi
  - ▶ toistetaan ikuisesti...
- ▶ Kaizeniin liittyvä syklinen parannusprosessiin saattavat liittyä tasaisin väliajoin pidettävät tilaisuudet “kaizen event”:it

TAUKO 10 min

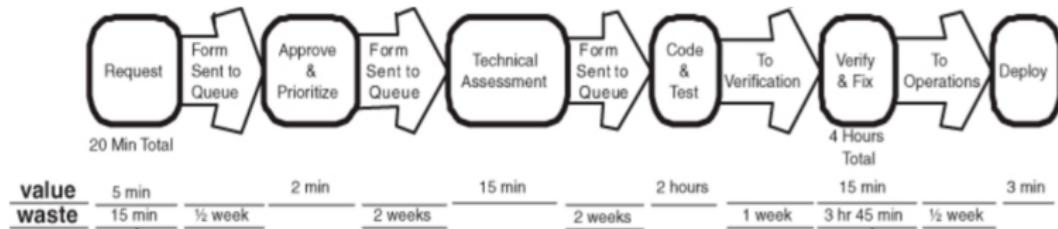
# Lean-työkalu: value stream mapping

## Lean-työkalu: value stream mapping

- ▶ Tekniikka/työkalu hukan kartoittamiseen
  - ▶ kuvaaa tuotteen kulkua eri työvaiheiden läpi
  - ▶ visualisoii arvoa tuottavat työvaiheet suhteessa elinkaareen

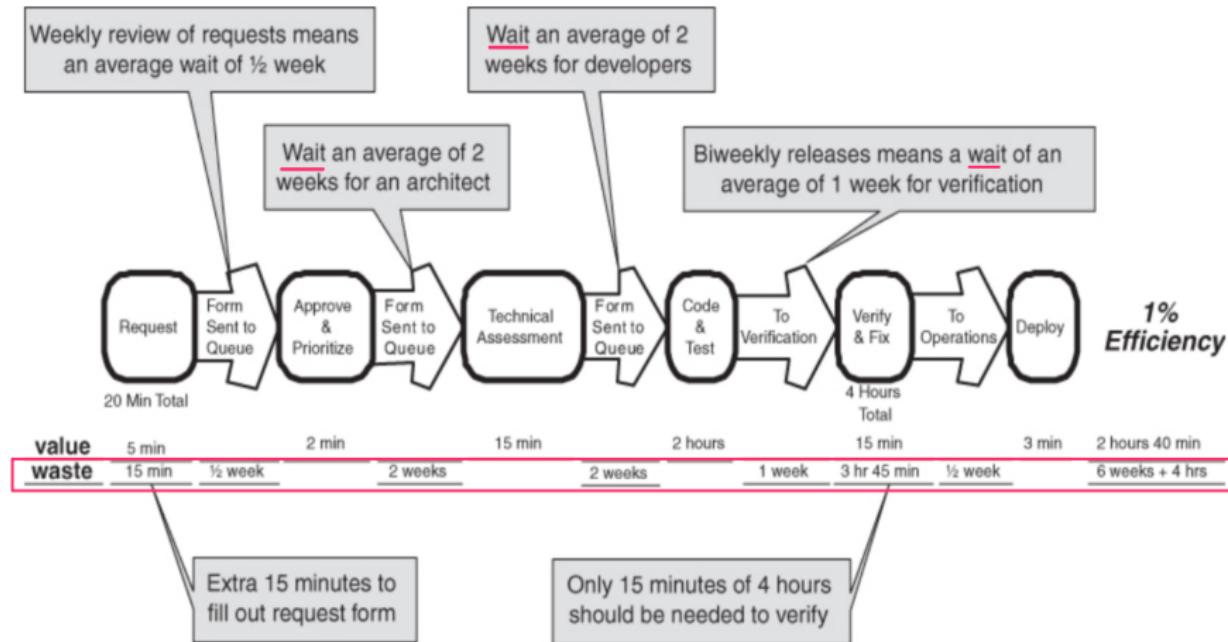
# Lean-työkalu: value stream mapping

- ▶ Tekniikka/työkalu hukan kartoittamiseen
  - ▶ kuvailee tuotteen kulku eri työvaiheiden läpi
  - ▶ visualisoibat arvoa tuottavat työvaiheet suhteessa elinkaareen



# Lean-työkalu: value stream mapping

- Tekniikka/työkalu hukan kartoittamiseen
  - kuvailee tuotteen kulku eri työvaiheiden läpi
  - visualisoibat arvoa tuottavat työvaiheet suhteessa elinkaareen



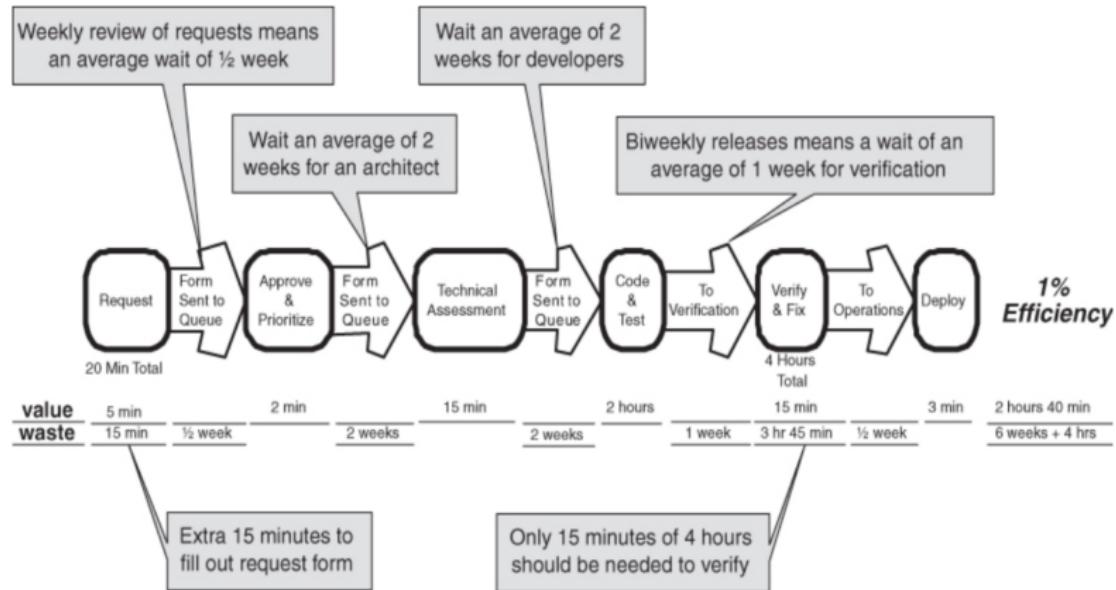
## Lean-työkalu: perimmäisen syyn analyysi

- ▶ Jos tuotantoprosessista on hukkaa, tulee se eliminoida

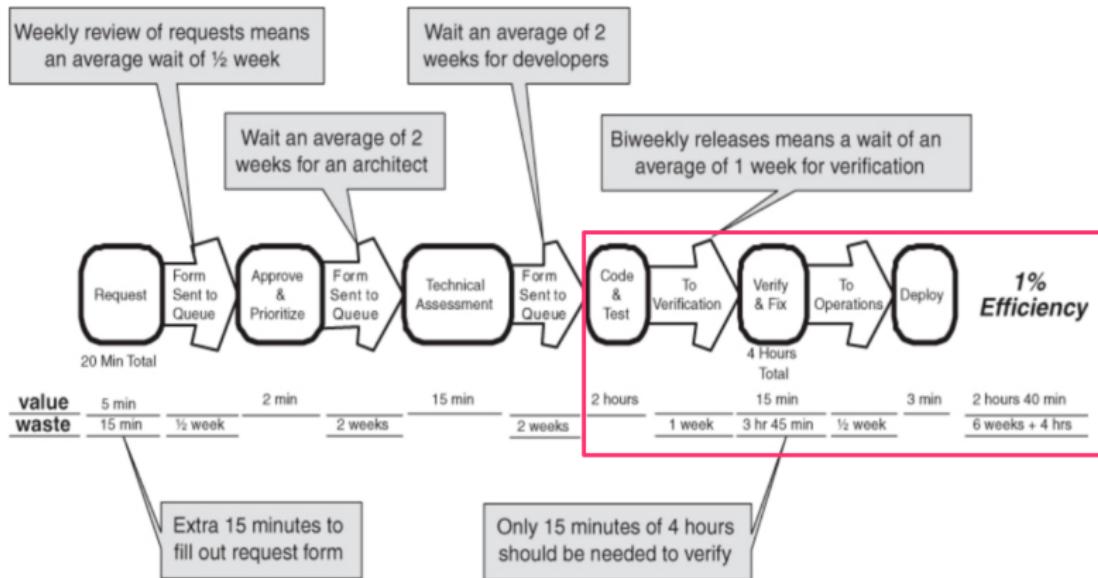
## Lean-työkalu: perimmäisen syyn analyysi

- ▶ Jos tuotantoprosessista on hukkaa, tulee se eliminoida
- ▶ Kaizenissa tarkoitus tehdä *perimmäisen syyn analyysi* (root cause analysis) ja korjata hukan taustalla olevat ongelmat

# Perimmäisen syyn analyysi

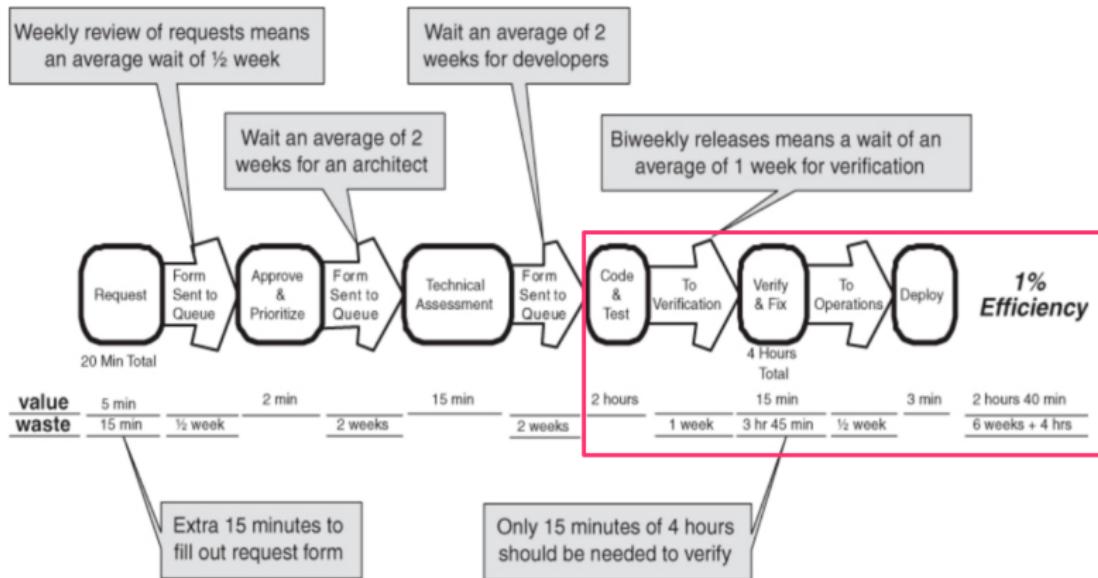


# Perimmäisen syyn analyysi: five whys



- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen

# Perimmäisen syyn analyysi: five whys



- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen
- ▶ **Miksi?**

## Perimmäisen syyn analyysi: five whys

- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen. **Miksi?**

## Perimmäisen syyn analyysi: five whys

- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen. **Miksi?**
- ▶ QA-osaston on vielä varmistettava, että koodi toimii staging-ympäristössä. **Miksi?**

## Perimmäisen syyn analyysi: five whys

- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen. **Miksi?**
- ▶ QA-osaston on vielä varmistettava, että koodi toimii staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla ei ole aikaa testata koodia itse staging-ympäristössä. **Miksi?**

## Perimmäisen syyn analyysi: five whys

- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen. **Miksi?**
- ▶ QA-osaston on vielä varmistettava, että koodi toimii staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla ei ole aikaa testata koodia itse staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla on kiire sprintin tavoitteena olevien user storyjen tekemisessä. **Miksi?**

## Perimmäisen syyn analyysi: five whys

- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen. **Miksi?**
- ▶ QA-osaston on vielä varmistettava, että koodi toimii staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla ei ole aikaa testata koodia itse staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla on kiire sprintin tavoitteena olevien user storyjen tekemisessä. **Miksi?**
- ▶ Edellisten sprinttien aikana tehtyjen storyjen bugikorjaukset vievät yllättävän paljon aikaa. **Miksi?**

## Perimmäisen syyn analyysi: five whys

- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen. **Miksi?**
- ▶ QA-osaston on vielä varmistettava, että koodi toimii staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla ei ole aikaa testata koodia itse staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla on kiire sprintin tavoitteena olevien user storyjen tekemisessä. **Miksi?**
- ▶ Edellisten sprinttien aikana tehtyjen storyjen bugikorjaukset vievät yllättävän paljon aikaa. **Miksi?**
- ▶ Laadunhallintaa ei ehditä koskaan tekemään kunnolla siinä sprintissä missä storyt toteutetaan. **Miksi?**

## Perimmäisen syyn analyysi: five whys

- ▶ Koodin valmistumisesta menee 1.5 viikkoa sen tuotantoon saamiseen. **Miksi?**
- ▶ QA-osaston on vielä varmistettava, että koodi toimii staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla ei ole aikaa testata koodia itse staging-ympäristössä. **Miksi?**
- ▶ Ohjelmoijilla on kiire sprintin tavoitteena olevien user storyjen tekemisessä. **Miksi?**
- ▶ Edellisten sprinttien aikana tehtyjen storyjen bugikorjaukset vievät yllättävän paljon aikaa. **Miksi?**
- ▶ Laadunhallintaa ei ehditä koskaan tekemään kunnolla siinä sprintissä missä storyt toteutetaan. **Miksi?**
- ▶ Sprintteihin otetaan aina liian monta user storya

# Leanin periaatteita

## Leanin periaatteita: pull-systeemi

## Leanin periaatteita: pull-systeemi

- ▶ Leanissa on siis tarkoitus optimoida aikaa, mikä kestää tuotteen suunnittelusta sen asiakkaalle toimittamiseen
  - ▶ Arvo pyritää saamaan *virtaamaan* (flow) asiakkaalle ilman turhia viiveitä ja työvaiheita

## Leanin periaatteita: pull-systeemi

- ▶ Leanissa on siis tarkoitus optimoida aikaa, mikä kestää tuotteen suunnittelusta sen asiakkaalle toimittamiseen
  - ▶ Arvo pyritää saamaan *virtaamaan* (flow) asiakkaalle ilman turhia viiveitä ja työvaiheita
- ▶ Leanin mekanismi virtauksen optimointiin on *pull-systeemi*
  - ▶ tehdään tuotteita, tai tuotteiden tarvitsema komponentteja ainoastaan tilauksen saapuessa
  - ▶ tapa toteuttaa JIT (just in time) -tuotanto

## Leanin periaatteita: pull-systeemi

- ▶ Leanissa on siis tarkoitus optimoida aikaa, mikä kestää tuotteen suunnittelusta sen asiakkaalle toimittamiseen
  - ▶ Arvo pyritää saamaan *virtaamaan* (flow) asiakkaalle ilman turhia viiveitä ja työvaiheita
- ▶ Leanin mekanismi virtauksen optimointiin on *pull-systeemi*
  - ▶ tehdään tuotteita, tai tuotteiden tarvitsemaa komponenttia ainoastaan tilauksen saapuessa
  - ▶ tapa toteuttaa JIT (just in time) -tuotanto
- ▶ Vastakohta on *push-systeemi*, missä tuotteita ja komponentteja tehdään etukäteen varastoon
  - ▶ pizzeria: pull-systeemi
  - ▶ Unicafe: push-systeemi

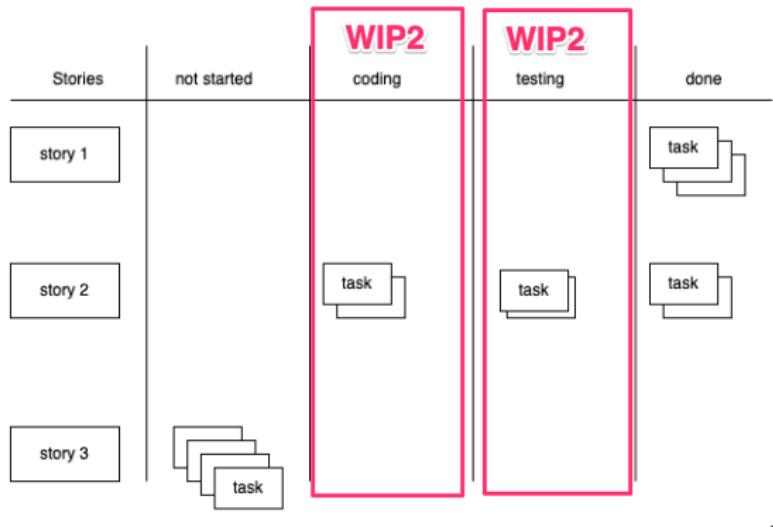
## Leanin periaatteita: pull-systeemi

- ▶ Leanissa on siis tarkoitus optimoida aikaa, mikä kestää tuotteen suunnittelusta sen asiakkaalle toimittamiseen
  - ▶ Arvo pyritää saamaan *virtaamaan* (flow) asiakkaalle ilman turhia viiveitä ja työvaiheita
- ▶ Leanin mekanismi virtauksen optimointiin on *pull-systeemi*
  - ▶ tehdään tuotteita, tai tuotteiden tarvitsemaa komponenttia ainoastaan tilauksen saapuessa
  - ▶ tapa toteuttaa JIT (just in time) -tuotanto
- ▶ Vastakohta on *push-systeemi*, missä tuotteita ja komponentteja tehdään etukäteen varastoon
  - ▶ pizzeria: pull-systeemi
  - ▶ Unicafe: push-systeemi
- ▶ Pull-systeemi toteutetaan usein *kanbanin* avulla
  - ▶ visuaalisen ohjaus, jonka avulla työntekijöiden on helppo tietää miten seuraavaksi tulee toimittaa

# Kanban ohjelmistokehityksessä

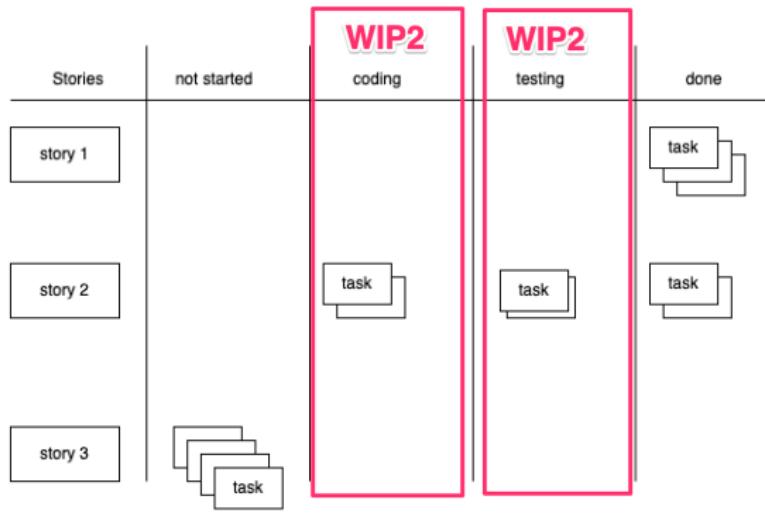
# Kanban ohjelmistokehityksessä

- ▶ Story/task kulkee eri työvaiheiden kautta
- ▶ Nopea läpivirtaus saadaan aikaan rajoittamalla tietyissä työvaiheissa kesken olevan työn määrää *WIP-rajoitteilla*



# Kanban ohjelmistokehityksessä

- ▶ Story/task kulkee eri työvaiheiden kautta
- ▶ Nopea läpivirtaus saadaan aikaan rajoittamalla tietyissä työvaiheissa kesken olevan työn määrää *WIP-rajoitteilla*



- ▶ Voi paljastaa pullonkaulat työprosessissa: mikä “sarake” ruuhkautuu

# Leanin periaatteita

## Leanin periaatteita

- ▶ Jotta arvo virtaisi tasaisesti, on hyvä jos työvaiheiden kestoon ei liity liikaa varianssia: *level the work*
  - ▶ pyrkimys suunilleen tasakokoisiin user storyihin

## Leanin periaatteita

- ▶ Jotta arvo virtaisi tasaisesti, on hyvä jos työvaiheiden kestoon ei liity liikaa varianssia: *level the work*
  - ▶ pyrkimys suunilleen tasakokoisiin user storyihin
- ▶ Yksi varianssin aiheuttaja ovat viat. Periaatteena:
  - ▶ Stop and fix
  - ▶ Build quality in

## Leanin periaatteita

- ▶ Jotta arvo virtaisi tasaisesti, on hyvä jos työvaiheiden kestoon ei liity liikaa varianssia: *level the work*
  - ▶ pyrkimys suunilleen tasakokoisiin user storyihin
- ▶ Yksi varianssin aiheuttaja ovat viat. Periaatteena:
  - ▶ Stop and fix
  - ▶ Build quality in
- ▶ *Stop and fix*: kuka tahansa velvollinen pysäyttämään tuotantolinjan vian havaitessaan
  - ▶ vian perimmäinen syy tulee selvittää mahdollisimman nopeasti ja pyrkiä eliminoimaan vian mahdollisuus tulevaisuudessa
  - ▶ laadun tulee olla sisäänrakennettua *build quality in*

## Leanin periaatteita

- ▶ Jotta arvo virtaisi tasaisesti, on hyvä jos työvaiheiden kestoon ei liity liikaa varianssia: *level the work*
  - ▶ pyrkimys suunilleen tasakokoisiin user storyihin
- ▶ Yksi varianssin aiheuttaja ovat viat. Periaatteena:
  - ▶ Stop and fix
  - ▶ Build quality in
- ▶ *Stop and fix*: kuka tahansa velvollinen pysäyttämään tuotantolinjan vian havaitessaan
  - ▶ vian perimmäinen syy tulee selvittää mahdollisimman nopeasti ja pyrkiä eliminoimaan vian mahdollisuus tulevaisuudessa
  - ▶ laadun tulee olla sisäänrakennettua *build quality in*
- ▶ Jatkuva integraatio ja automaattinen testaus *stop and fix* - ja *build quality in* -periaatteiden ilmentymänä

## Leanin periaatteita

- ▶ *Decide as late as possible*
  - ▶ Pull-systeemeissä tehdään tuotantopääökset vasta tarvittaessa
  - ▶ esim. miten paljon tuotetta ja sen tarvitsemia komponentteja tulee valmistaa

## Leanin periaatteita

- ▶ *Decide as late as possible*
  - ▶ Pull-systeemeissä tehdään tuotantopäätökset vasta tarvittaessa
  - ▶ esim. miten paljon tuotetta ja sen tarvitsemia komponentteja tulee valmistaa
- ▶ *Commit at the last responsible moment*, päätöksiä viivytetään, mutta ei kuitenkaan liian kauaa

## Leanin periaatteita

- ▶ *Decide as late as possible*
  - ▶ Pull-systeemeissä tehdään tuotantopäättöökset vasta tarvittaessa
  - ▶ esim. miten paljon tuotetta ja sen tarvitsemia komponentteja tulee valmistaa
- ▶ *Commit at the last responsible moment*, päätöksiä viivytetään, mutta ei kuitenkaan liian kauaa
- ▶ Kun päätös tehdään myöhään sen tueksi on käytettäväissä maksimaalinen määrä tietoa

## Leanin periaatteita

- ▶ *Decide as late as possible*
  - ▶ Pull-systeemeissä tehdään tuotantopäättöökset vasta tarvittaessa
  - ▶ esim. miten paljon tuotetta ja sen tarvitsemia komponentteja tulee valmistaa
- ▶ *Commit at the last responsible moment*, päätöksiä viivytetään, mutta ei kuitenkaan liian kauaa
- ▶ Kun päätös tehdään myöhään sen tueksi on käytettäväissä maksimaalinen määrä tietoa
- ▶ Kun päätökset on tehty, toimitaan mahdollisimman nopeasti: *implement rapidly tai deliver as fast as possible*
  - ▶ näin arvo saadaan virtaamaan asiakkaalle ilman turhia viiveitä

## Leanin periaatteita

- ▶ *Decide as late as possible*
  - ▶ Pull-systeemeissä tehdään tuotantopäättöökset vasta tarvittaessa
  - ▶ esim. miten paljon tuotetta ja sen tarvitsemia komponentteja tulee valmistaa
- ▶ *Commit at the last responsible moment*, päätöksiä viivytetään, mutta ei kuitenkaan liian kauaa
- ▶ Kun päätös tehdään myöhään sen tueksi on käytettäväissä maksimaalinen määrä tietoa
- ▶ Kun päätökset on tehty, toimitaan mahdollisimman nopeasti: *implement rapidly tai deliver as fast as possible*
  - ▶ näin arvo saadaan virtaamaan asiakkaalle ilman turhia viiveitä
- ▶ Mitä nopeammin arvo virtaa, sitä enemmän päätöksiä mahdollista viivyttää ja tehdä ne paremman tiedon valossa

# Arvon virtaaminen ketterässä ohjelmistotuotannossa

## Arvon virtaaminen ketterässä ohjelmistotuotannossa

- ▶ Vaatimuksia hallitaan product backlogilla, joka on parhaassa tapauksessa DEEP
  - ▶ *detailed aproproately, emergent, estimated, prioritized*
  - ▶ tarkkoja vaatimuksia ei määritellä spekulatiivisesti vaan *at the last responsible moment*

## Arvon virtaaminen ketterässä ohjelmistotuotannossa

- ▶ Vaatimuksia hallitaan product backlogilla, joka on parhaassa tapauksessa DEEP
  - ▶ *detailed aproproately, emergent, estimated, prioritized*
  - ▶ tarkkoja vaatimuksia ei määritellä spekulatiivisesti vaan *at the last responsible moment*
- ▶ Kun product owner valitsee storyn seuraavaan sprinttiin: määritellään hyväksymäkriteerit ja suunnitellaan story
  - ▶ *deliver as fast as possible* tehdään valmiiksi sprintin aikana

# Arvon virtaaminen ketterässä ohjelmistotuotannossa

- ▶ Vaatimuksia hallitaan product backlogilla, joka on parhaassa tapauksessa DEEP
  - ▶ *detailed aproproately, emergent, estimated, prioritized*
  - ▶ tarkkoja vaatimuksia ei määritellä spekulatiivisesti vaan *at the last responsible moment*
- ▶ Kun product owner valitsee storyn seuraavaan sprinttiin: määritellään hyväksymäkriteerit ja suunnitellaan story
  - ▶ *deliver as fast as possible* tehdään valmiiksi sprintin aikana
- ▶ *Scrum on pull-systeemi:*
  - ▶ jokaiseen sprinttiin otetaan asiakkaan edustajan viime hetkellä viimeistelemät tilaukset
  - ▶ arvo eli valmiit toiminnallisuudet virtaa asiakkaalle sprinttien määrittelemässä rytmässä

Arvon virtaaminen ohjelmistotuotannossa *kiihtyy*

## Arvon virtaaminen ohjelmistotuotannossa *kiihtyy*

- ▶ Scrum vie uusia ominaisuuksia tuotantoon sprinteittäin

## Arvon virtaaminen ohjelmistotuotannossa *kiihtyy*

- ▶ Scrum vie uusia ominaisuuksia tuotantoon sprinteittäin
- ▶ Viime aikojen trendinä on ollut tihentää sykliä
  - ▶ *jatkuva toimittaminen* (CD) voi tarkoittaa sitä, että jopa jokainen commit johtaa tuotantoonvientiin

## Arvon virtaaminen ohjelmistotuotannossa *kiihtyy*

- ▶ Scrum vie uusia ominaisuuksia tuotantoon sprinteittäin
- ▶ Viime aikojen trendinä on ollut tihentää sykliä
  - ▶ *jatkuva toimittaminen* (CD) voi tarkoittaa sitä, että jopa jokainen commit johtaa tuotantoonvientiin
- ▶ Scrum rajoittaa kesken olevan työn määrää siten, että sprinttiin otetaan vaan tiimin velositeetin verran user storyjä

## Arvon virtaaminen ohjelmistotuotannossa *kiihyy*

- ▶ Scrum vie uusia ominaisuuksia tuotantoon sprinteittäin
- ▶ Viime aikojen trendinä on ollut tihentää sykliä
  - ▶ *jatkuva toimittaminen* (CD) voi tarkoittaa sitä, että jopa jokainen commit johtaa tuotantoonvientiin
- ▶ Scrum rajoittaa kesken olevan työn määrää siten, että sprinttiin otetaan vaan tiimin velositeetin verran user storyjä
- ▶ Kaikissa konteksteissa, esim. jatkuva käyttöönottoa sovellettaessa aikarajoitettu sprintti ei ole mielekäs

## Arvon virtaaminen ohjelmistotuotannossa *kiihyy*

- ▶ Scrum vie uusia ominaisuuksia tuotantoon sprinteittäin
- ▶ Viime aikojen trendinä on ollut tihentää sykliä
  - ▶ *jatkuva toimittaminen* (CD) voi tarkoittaa sitä, että jopa jokainen commit johtaa tuotantoonvientiin
- ▶ Scrum rajoittaa kesken olevan työn määrää siten, että sprinttiin otetaan vaan tiimin velositeetin verran user storyjä
- ▶ Kaikissa konteksteissa, esim. jatkuva käyttöönottoa sovellettaessa aikarajoitettu sprintti ei ole mielekäs
- ▶ Osin siirrytty puhtaampaan pull-systeemiin, storyja toteutetaan yksi ( tai muutama) kerrallaan mahdollisimman nopeasti
  - ▶ kun tuotantokapasiteettia vapautuu, valitsee product owner tärkeimmän storyn
  - ▶ story määritellään, suunnitellaan ja sitten toteutetaan välittömästi alusta loppuun

## Johtaminen leanissa

- ▶ Johtamiskulttuurissa keskiössä on lean-ajattelun opettajina, mentoreina ja työn valmenajana toimivat johtajat/managerit
  - ▶ *grow leaders*: johtajat kasvatetaan Toyotan arvoihin
  - ▶ *my manager can do my job better than me*: johtajat firman sisällä eri työtehtävien kautta vastuuseen nousseita

## Johtaminen leanissa

- ▶ Johtamiskulttuurissa keskiössä on lean-ajattelun opettajina, mentoreina ja työn valmenajana toimivat johtajat/managerit
  - ▶ *grow leaders*: johtajat kasvatetaan Toyotan arvoihin
  - ▶ *my manager can do my job better than me*: johtajat firman sisällä eri työtehtävien kautta vastuuseen nousseita
- ▶ Tärkeää johtamisen periaate on *go see* (*genchi genbutsu*)
  - ▶ asiat tulee “nähdä asiat oman silmin” eikä pelkästään istua työpöydän ääressä raportteja lukemassa

## Johtaminen leanissa

- ▶ Johtamiskulttuurissa keskiössä on lean-ajattelun opettajina, mentoreina ja työn valmenajana toimivat johtajat/managerit
  - ▶ *grow leaders*: johtajat kasvatetaan Toyotan arvoihin
  - ▶ *my manager can do my job better than me*: johtajat firman sisällä eri työtehtävien kautta vastuuseen nousseita
- ▶ Tärkeää johtamisen periaate on *go see* (*genchi genbutsu*)
  - ▶ asiat tulee “nähdä asiat oman silmin” eikä pelkästään istua työpöydän ääressä raportteja lukemassa
- ▶ Johtajien oletetaan johtavat etulinjassa (*gembra*) eli siellä missä työ tosiasiallisesti tehdään

## Johtaminen leanissa

- ▶ Johtamiskulttuurissa keskiössä on lean-ajattelun opettajina, mentoreina ja työn valmenajana toimivat johtajat/managerit
  - ▶ *grow leaders*: johtajat kasvatetaan Toyotan arvoihin
  - ▶ *my manager can do my job better than me*: johtajat firman sisällä eri työtehtävien kautta vastuuseen nousseita
- ▶ Tärkeää johtamisen periaate on *go see* (*genchi genbutsu*)
  - ▶ asiat tulee “nähdä asiat oman silmin” eikä pelkästään istua työpöydän ääressä raportteja lukemassa
- ▶ Johtajien oletetaan johtavat etulinjassa (*gembra*) eli siellä missä työ tosiasiallisesti tehdään
- ▶ Scrum masterin rooli on osin leanin ideaalien mukainen

# Lean-tuotekehityen periaatteita

## Lean-tuotekehityen periaatteita

- ▶ Sovellettaessa leania tuotannon optimoinnin sijaan *tuotekehitykseen* nousee esiin uusia periaatteita

## Lean-tuotekehityen periaatteita

- ▶ Sovellettaessa leania tuotannon optimoinnin sijaan *tuotekehitykseen* nousee esiin uusia periaatteita
- ▶ *Out-learn the competitors, through generating more useful knowledge and using and remembering it effectively*

## Lean-tuotekehityen periaatteita

- ▶ Sovellettaessa leania tuotannon optimoinnin sijaan *tuotekehitykseen* nousee esiin uusia periaatteita
- ▶ *Out-learn the competitors, through generating more useful knowledge and using and remembering it effectively*
- ▶ Fokukseen nousee *oppimisen kiihdyttäminen* (amplify learning)
  - ▶ *high-value information*: pyritään arvokkaaseen tietoon
  - ▶ *focus on uncertain things*: kiinnittämällä huomio epävarmoihinasioihin

## Lean-tuotekehityen periaatteita

- ▶ Sovellettaessa leania tuotannon optimoinnin sijaan *tuotekehitykseen* nousee esiin uusia periaatteita
- ▶ *Out-learn the competitors, through generating more useful knowledge and using and remembering it effectively*
- ▶ Fokukseen nousee *oppimisen kiihdyttäminen* (amplify learning)
  - ▶ *high-value information*: pyritään arvokkaaseen tietoon
  - ▶ *focus on uncertain things*: kiinnittämällä huomio epävarmoihin asioihin
- ▶ Epävarmat ja suuren teknisen riskin sisältävät ideat tulee toteuttaa nopeasti, viivyttelyllä voi olla korkea hinta
  - ▶ *cost of delay*

## Lean-tuotekehityen periaatteita

- ▶ Eräs mekanismi oppimisen nopeuttamiseen on *set based concurrent development*
  - ▶ jos tarkoituksena on kehittää esim. uusi moottorin jäähdytysjärjestelmä, aletaan yhtä aika kehittämään useita vaihtoehtoisia ratkaisuja eri tiimien toimesta
  - ▶ tasaisin väliajoin kehitettäviä ratkaisuja ja osa niistä karsitaan
  - ▶ lopulta parhaaksi osoittautuva valitaan loppitulooteeseen

## Lean-tuotekehityen periaatteita

- ▶ Eräs mekanismi oppimisen nopeuttamiseen on *set based concurrent development*
  - ▶ jos tarkoituksesta on kehittää esim. uusi moottorin jäähdytysjärjestelmä, aletaan yhtä aika kehittämään useita vaihtoehtoisia ratkaisuja eri tiimien toimesta
  - ▶ tasaisin väliajoin kehitettäviä ratkaisuja ja osa niistä karsitaan
  - ▶ lopulta parhaaksi osoittautuva valitaan loppituotteeseen
- ▶ Poikkeaa radikaalisti iteratiivisesta kehityksestä
  - ▶ Ohjelmistotuotannossa harvinainen

## Lean-tuotekehityen periaatteita

- ▶ Eräs mekanismi oppimisen nopeuttamiseen on *set based concurrent development*
  - ▶ jos tarkoituksena on kehittää esim. uusi moottorin jäähdytysjärjestelmä, aletaan yhtä aika kehittämään useita vaihtoehtoisia ratkaisuja eri tiimien toimesta
  - ▶ tasaisin väliajoin kehitettäviä ratkaisuja ja osa niistä karsitaan
  - ▶ lopulta parhaaksi osoittautuva valitaan loppuduotteeseen
- ▶ Poikkeaa radikaalisti iteratiivisesta kehityksestä
  - ▶ Ohjelmistotuotannossa harvinainen
- ▶ Toyotalla tuotekehitystä johtaa *chief technical engineer*
  - ▶ sekä tekninen että liiketoiminnallinen vastuu
  - ▶ etulinjassa toimiva johtaja, tuntee tarkasti käytännön työn, mutta on erittäin lähellä asiakasta

Leanin soveltaminen

## Leanin soveltaminen

- ▶ 90-luvulta alkaen lean on herättänyt maailmalla suurta kiinnostusta ja sitä on sovellettu laajalti

## Leanin soveltaminen

- ▶ 90-luvulta alkaen lean on herättänyt maailmalla suurta kiinnostusta ja sitä on sovellettu laajalti
- ▶ Scrumissa paljon vaikutteita leanista
  - ▶ viimeaikoina agile vienyt arvoketjun optimoimista user storyjen läpimenoaikeja minimoimalla Scrumin periaatteita pidemmälle
  - ▶ nykyään puhutaan paljon *leanista ohjelmistokehityksestä*

## Leanin soveltaminen

- ▶ 90-luvulta alkaen lean on herättänyt maailmalla suurta kiinnostusta ja sitä on sovellettu laajalti
- ▶ Scrumissa paljon vaikutteita leanista
  - ▶ viimeaikoina agile vienyt arvoketjun optimoimista user storyjen läpimenoaikeja minimoimalla Scrumin periaatteita pidemmälle
  - ▶ nykyään puhutaan paljon *leanista ohjelmistokehityksestä*
- ▶ Agilessa ja leanissa sama fundamentaali periaate: *toimintojen jatkuva kehittäminen*
  - ▶ rajaveto leanin ja ketterän välillä ei olekaan ollenkaan selvä

## Leanin soveltaminen

- ▶ 90-luvulta alkaen lean on herättänyt maailmalla suurta kiinnostusta ja sitä on sovellettu laajalti
- ▶ Scrumissa paljon vaikutteita leanista
  - ▶ viimeaikoina agile vienyt arvoketjun optimoimista user storyjen läpimenoaikoja minimoimalla Scrumin periaatteita pidemmälle
  - ▶ nykyään puhutaan paljon *leanista ohjelmistokehityksestä*
- ▶ Agilessa ja leanissa sama fundamentaali periaate: *toimintojen jatkuva kehittäminen*
  - ▶ rajaveto leanin ja ketterän välillä ei olekaan ollenkaan selvä
- ▶ **Ketteryyden ydin** on läpinäkyvyyden mahdollistava toimintojen parantamiseen keskittyvä inspect-and-adapt-sykli
- ▶ Käytännössä tämä on täsmälleen sama idea kuin leanin kaizen

## Leanin soveltamisen vaikeus

- ▶ Leanin soveltamisessa on kohdattu myös paljon ongelmia

## Leanin soveltamisen vaikeus

- ▶ Leanin soveltamisessa on kohdattu myös paljon ongelmia
- ▶ Lean on ajattelumalli, joka on kehitetty Toyotan tarpeisiin, ja jalostunut vuosikymmenten kuluessa
  - ▶ epäselvää miten Toyotan käytänteet siirretään eri aloille

## Leanin soveltamisen vaikeus

- ▶ Leanin soveltamisessa on kohdattu myös paljon ongelmia
- ▶ Lean on ajattelumalli, joka on kehitetty Toyotan tarpeisiin, ja jalostunut vuosikymmenten kuluessa
  - ▶ epäselvää miten Toyotan käytänteet siirretään eri aloille
- ▶ Usein Lean tulkitaan mekanistisesti, keskittyen työkaluihin
  - ▶ esim. kanban ja value stream mapping

## Leanin soveltamisen vaikeus

- ▶ Leanin soveltamisessa on kohdattu myös paljon ongelmia
- ▶ Lean on ajattelumalli, joka on kehitetty Toyotan tarpeisiin, ja jalostunut vuosikymmenten kuluessa
  - ▶ epäselvää miten Toyotan käytänteet siirretään eri aloille
- ▶ Usein Lean tulkitaan mekanistisesti, keskittyen työkaluihin
  - ▶ esim. kanban ja value stream mapping
- ▶ Unohtaen periaatteet: *jatkuva parantaminen ja ihmisten kunnioittaminen*
  - ▶ saadaan ehkä aikaan hetkellisiä parannuksia tuotantoketjussa
  - ▶ parannukset eivät ole kauaskantoisia jos ne eivät vaikuta koko organisaation ajatteluun ja toimintatapoihin