

# Ohjelmistotuotanto

Matti Luukkainen ja ohjaajat Jami Kousa, Tero Tapio, Mauri Karlin

syksy 2019

Luento 4

4.11.2019

# nopea kertaus eilseltä

- ▶ User story
  - ▶ “määritelmä”:
  - ▶ description
  - ▶ conversations
  - ▶ acceptance criteria
- ▶ INVEST
- ▶ Estimointi
  - ▶ Miksi? Miten? Kuka?
- ▶ Product Backlog
  - ▶ Kuka vastuussa?
  - ▶ Miten saadaan projektin alussa muodostettua?

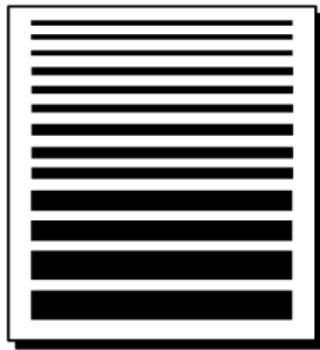
## Hyvä product backlog on DEEP

- ▶ Mike Cohn lanseerasi lyhenteen DEEP kuvaamaan hyvän backlogin ominaisuuksia
  - ▶ Detailed appropriately
  - ▶ Estimated
  - ▶ Emergent
  - ▶ Prioritized

## Hyvä product backlog on DEEP

- ▶ Mike Cohn lanseerasi lyhenteen DEEP kuvaamaan hyvän backlogin ominaisuuksia
  - ▶ Detailed appropriately
  - ▶ Estimated
  - ▶ Emergent
  - ▶ Prioritized
- ▶ *Detailed appropriately* eli sopivan detaljoitu
  - ▶ ylempänä tarkkoja
  - ▶ alempana suurpiirteisempiä

High  
Priority



Fine-grained, detailed items ready  
to be worked on in the next sprint

Low  
Priority

Large, coarse-grained items

- ▶ *Emergent* kuvaaa backlogin muuttuvaa luonnetta:
  - ▶ uusia storyja tulee
  - ▶ vanhoja poistetaan, uudelleenpriorisoidaan ja uudelleenestimoidaan, muokataan ja pilkotaan

- ▶ *Emergent* kuvaaa backlogin muuttuvaa luonnetta:
  - ▶ uusia storyja tulee
  - ▶ vanhoja poistetaan, uudelleenpriorisoidaan ja uudelleenestimoidaan, muokataan ja pilkotaan
- ▶ Muuttuvan luonteen takia backlogia tulee hoitaa (backlog grooming/refinement) projektin edetessä
  - ▶ pääasiallinen vastuu on product ownerilla
  - ▶ backlogin hoitamiseen osallistuu koko ohjelmistotuotantoiimi
  - ▶ Scrum suosittlee että noin 10% sprintin työajasta käytetään backlog groomingiin

## “Ready” story ja epiikki

- ▶ Hyvä story on siis INVEST (independent, negotiable, valuable, estimable, small, testable)
- ▶ DEEP taas taas sanoo, että backlogin pitää olla sopivan detaljoitu

## “Ready” story ja epiikki

- ▶ Hyvä story on siis INVEST (independent, negotiable, valuable, estimable, small, testable)
- ▶ DEEP taas taas sanoo, että backlogin pitää olla sopivan detaljoitu
- ▶ INVEST päteekin vain backlogin korkeamman prioriteetin storyihin
- ▶ Joskus sanotaan että story on *ready*, kun se on valmiina toteutettavaksi (hyvin tunnettu ja INVEST)

## “Ready” story ja epiikki

- ▶ Hyvä story on siis INVEST (independent, negotiable, valuable, estimable, small, testable)
- ▶ DEEP taas taas sanoo, että backlogin pitää olla sopivan detaljoitu
- ▶ INVEST päteekin vain backlogin korkeamman prioriteetin storyihin
- ▶ Joskus sanotaan että story on *ready*, kun se on valmiina toteutettavaksi (hyvin tunnettu ja INVEST)
- ▶ Alemman prioriteetin stoorit voivat olla *epiikkejä* (epic): scope ei tiedossa, ei mielekästä estimoida

## Velositeetti

- ▶ Estimoinnin yksi tarkoitus on mahdollistaa koko projektin viemän aikamäärän summittainen arvointi

## Velositeetti

- ▶ Estimoinnin yksi tarkoitus on mahdollistaa koko projektin viemän aikamäärän summittainen arvointi
- ▶ Estimoinnin yksikkönä on abstrakti käsite *story point*, miten sen avulla voidaan arvioida projektin kestoa?

## Velositeetti

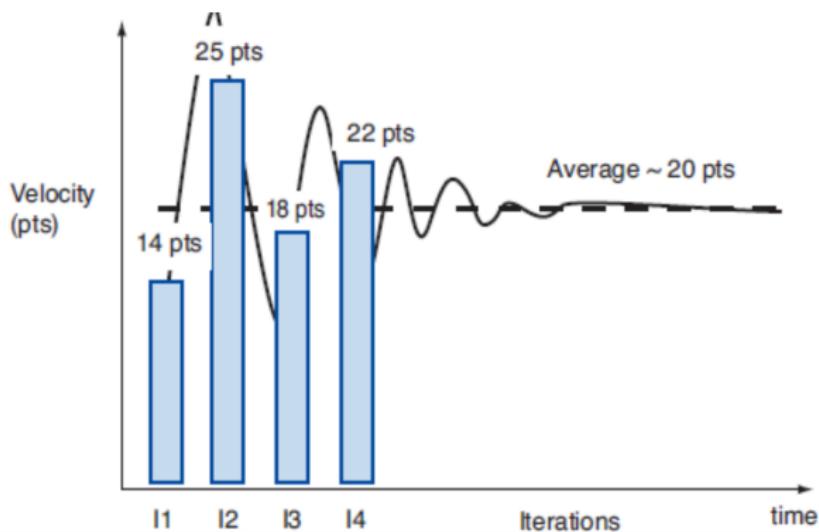
- ▶ Estimoinnin yksi tarkoitus on mahdollistaa koko projektin viemän aikamäärän summittainen arvointi
- ▶ Estimoinnin yksikkönä on abstrakti käsite *story point*, miten sen avulla voidaan arvioida projektin kestoa?
- ▶ Kehitystiimin *velositeetti* (engl velocity) tarjoaa osittaisen ratkaisun tähän
- ▶ Velositeetilla tarkoitetaan *tiimin keskimäärin yhdessä sprintissä toteuttamien story pointtien määrää*

## Velositeetti

- ▶ Estimoinnin yksi tarkoitus on mahdollistaa koko projektin viemän aikamäärän summittainen arvointi
- ▶ Estimoinnin yksikkönä on abstrakti käsite *story point*, miten sen avulla voidaan arvioida projektin kestoa?
- ▶ Kehitystiimin *velositeetti* (engl velocity) tarjoaa osittaisen ratkaisun tähän
- ▶ Velositeetilla tarkoitetaan *tiimin keskimäärin yhdessä sprintissä toteuttamien story pointtien määrää*
- ▶ Jos velositeetti on selvillä ja projektissa toteutettavaksi tarkoitettut storyt on estimoitu, lasketaan projektin kestoarvio seuraavasti

$(User\ storyjen\ estimaattien\ summa)\ / \ velositeetti\ * \ sprintin\ pituus$

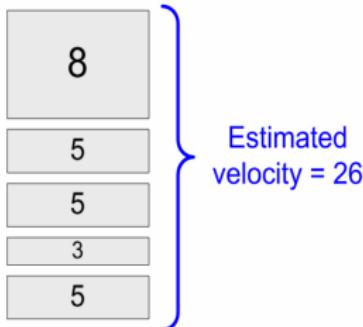
- ▶ Projektin alkaessa velositeetti ei ole selvillä, ellei kyseessä ole jo yhdessä työskennellyt tiimi
- ▶ Velositeetti vaihtelee alussa melko paljon
  - ▶ Estimointi aluksi vaikeampaa varsinkin jos sovellusalue ja käytetyt teknologiat eivät ole täysin tuttuja



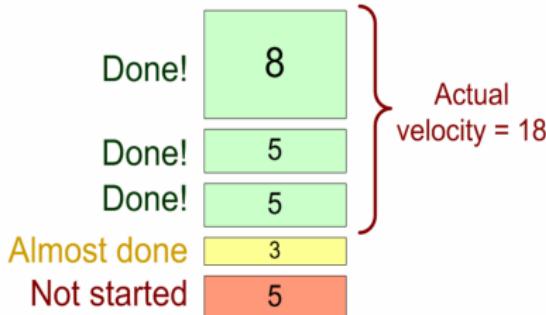
- ▶ Velositeetti ja siihen perustuva projektin keston arvio alkaakin tarkentumaan pikkuhiljaa

- ▶ Ketterissä menetelmissä on oleellista kuvata mahdollisimman realistisesti projektin etenemistä
- ▶ Velositeettiin lasketaan mukaan ainoastaan definition of done mukaisesti toteutetut storyt
  - ▶ "lähes valmiiksi" tehtyä työtä ei siis katsota ollenkaan tehdyksi työksi

Beginning of sprint

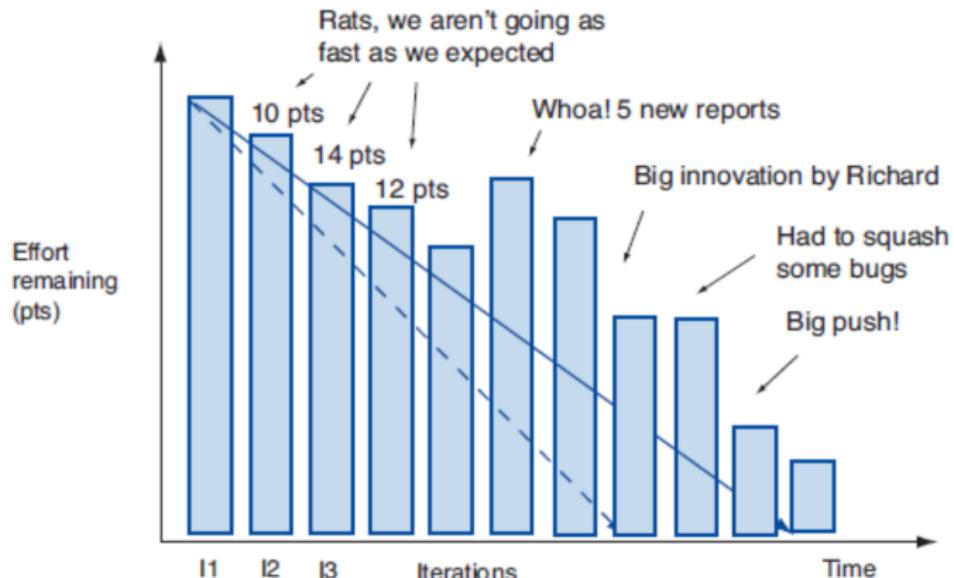


End of sprint



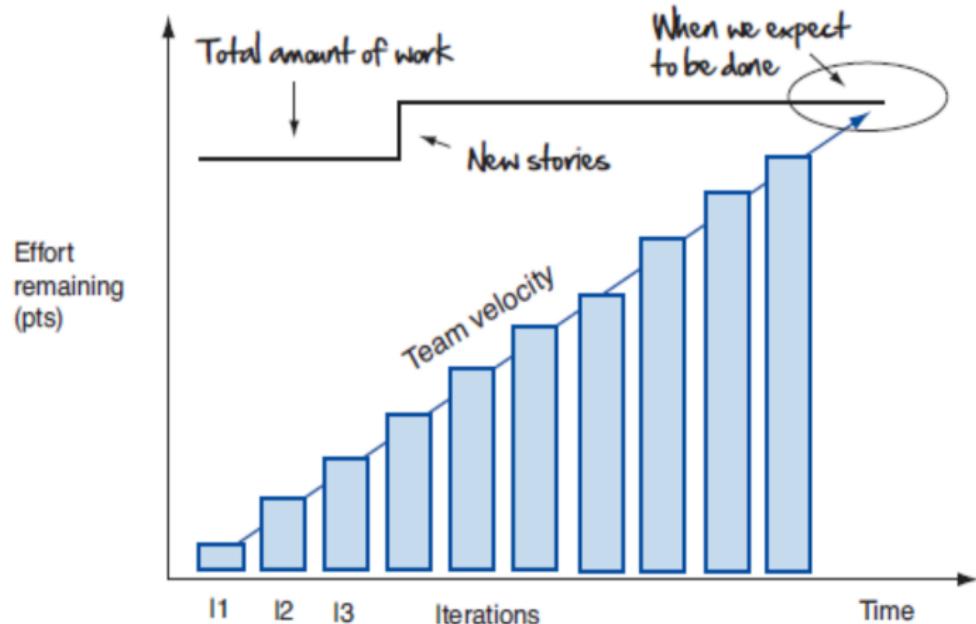
# Burndown

- ▶ Projektin etenemistä kuvataan yleensä release burndown -kaavion avulla



# Burnup

- ▶ Tuo selkeämmin esiin kesken projektin etenemisen tapahtuvan työmääärän kasvun



## Kannattaako estimointi - #NoEstimates

- ▶ Storyjen viemän työmääärän arvioimiseen kaksi motivaatiota
  - ▶ auttaa asiakasta priorisoinnissa
  - ▶ mahdollistaa koko projektin tai tiettyjä osien viemän ajan ja täten myös kustannuksen arvointi

## Kannattaako estimointi - #NoEstimates

- ▶ Storyjen viemän työmääärän arvioimiseen kaksi motivaatiota
  - ▶ auttaa asiakasta priorisoinnissa
  - ▶ mahdollistaa koko projektin tai tiettyjä osien viemän ajan ja täten myös kustannuksen arvointi
- ▶ Story point -pohjainen suhteellinen estimointi on saavuttanut vankan aseman
  - ▶ Scrum guide mainitsee että backlogin vaatimukset estimoitujia samoin kuten monet parhaat käytänteet kuten DEEP

## Kannattaako estimointi - #NoEstimates

- ▶ Storyjen viemän työmääärän arvioimiseen kaksi motivaatiota
  - ▶ auttaa asiakasta priorisoinnissa
  - ▶ mahdollistaa koko projektin tai tiettyjä osien viemän ajan ja täten myös kustannuksen arvointi
- ▶ Story point -pohjainen suhteellinen estimointi on saavuttanut vankan aseman
  - ▶ Scrum guide mainitsee että backlogin vaatimukset estimoituja
  - ▶ samoin kuten monet parhaat käytänteet kuten DEEP
- ▶ #NoEstimates-liike ruvennut kyseenalaistamaan story point -perustaista estimointitapaa
- ▶ pitää siitä saavutettuja hyötyjä liian vähäisinä verrattuna käytettyyn aikaan ja vaivaan

## Kannattaako estimointi - #NoEstimates

- ▶ Storyjen viemän työmääärän arvioimiseen kaksi motivaatiota
  - ▶ auttaa asiakasta priorisoinnissa
  - ▶ mahdollistaa koko projektin tai tiettyjä osien viemän ajan ja täten myös kustannuksen arvointi
- ▶ Story point -pohjainen suhteellinen estimointi on saavuttanut vankan aseman
  - ▶ Scrum guide mainitsee että backlogin vaatimukset estimoituja
  - ▶ samoin kuten monet parhaat käytänteet kuten DEEP
- ▶ #NoEstimates-liike ruvennut kyseenalaistamaan story point -perustaista estimointitapaa
- ▶ pitää siitä saavutettuja hyötyjä liian vähäisinä verrattuna käytettyyn aikaan ja vaivaan
- ▶ Yksinkertainen vaihtoehto: *arvioidaan velositeetti laskemalla kussakin sprintissä valmistuneiden storyjen lukumäärää*
- ▶ Monien kokemuksen mukaan toimii varsin hyvin, erityisesti jos storyt riittävän tasakokoisia

# Sprintin suunnittelu

## Sprintin suunnittelu

- ▶ Kertauksena viime viikolta: Scrum määrittelee pidettäväksi ennen jokaista sprinttiä suunnittelupalaverin
- ▶ Palaverin ensimmäinen tavoite on selvittää *mitä* sprintin aikana tehdään

## Sprintin suunnittelu

- ▶ Kertauksena viime viikolta: Scrum määrittelee pidettäväksi ennen jokaista sprinttiä suunnittelupalaverin
- ▶ Palaverin ensimmäinen tavoite on selvittää *mitä* sprintin aikana tehdään
- ▶ Lähtökohtana on sopivassa tilassa oleva eli DEEP product backlog
- ▶ Product owner esittelee backlogin kärjessä olevat vaatimukset
- ▶ Tiimin on tarkoitus olla riittävällä tasolla selville mitä vaatimuksilla tarkoitetaan

## Sprintin suunnittelu

- ▶ Kertauksena viime viikolta: Scrum määrittelee pidettäväksi ennen jokaista sprinttiä suunnittelupalaverin
- ▶ Palaverin ensimmäinen tavoite on selvittää *mitä* sprintin aikana tehdään
- ▶ Lähtökohtana on sopivassa tilassa oleva eli DEEP product backlog
- ▶ Product owner esittelee backlogin kärjessä olevat vaatimukset
- ▶ Tiimin on tarkoitus olla riittävällä tasolla selville mitä vaatimuksilla tarkoitetaan
- ▶ Tiimi valitsee tehtäväksi niin monta backlogin storyistaa kuin se arvioi kykenevänsä sprintin aikana toteuttamaan definition of doneen määrittelemällä laatutasolla

## Sprintin tavoite

- ▶ Suunnittelun yhteydessä määritellään *sprintin tavoite* (sprint goal)
- ▶ Lyhyt, yhden tai kahden lauseen kuvausta siitä, mitä tiimi on aikeissa sprintin aikana tehdä

## Sprintin tavoite

- ▶ Suunnittelun yhteydessä määritellään *sprintin tavoite* (sprint goal)
- ▶ Lyhyt, yhden tai kahden lauseen kuvausta siitä, mitä tiimi on aikeissa sprintin aikana tehdä
- ▶ K. Schwaber ensimmäisen sprintin tavoitte *demonstrate a key piece of user functionality on the selected technology*
- ▶ Esimeriksi verkkokauppa kehitettäessä sprinttien tavoitteita voisivat olla:
  - ▶ Ostoskorin perustoiminnallisuus: tuotteiden lisäys ja poisto
  - ▶ Ostosten maksaminen ja toimitustavan valinta

## Sprintin tavoite

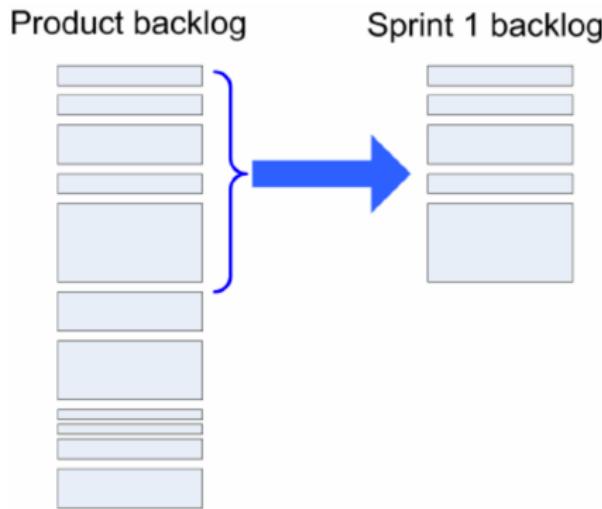
- ▶ Suunnittelun yhteydessä määritellään *sprintin tavoite* (sprint goal)
- ▶ Lyhyt, yhden tai kahden lauseen kuvausta siitä, mitä tiimi on aikeissa sprintin aikana tehdä
- ▶ K. Schwaber ensimmäisen sprintin tavoitte *demonstrate a key piece of user functionality on the selected technology*
- ▶ Esimeriksi verkkokauppa kehitettäessä sprinttien tavoitteita voisivat olla:
  - ▶ Ostoskorin perustoiminnallisuus: tuotteiden lisäys ja poisto
  - ▶ Ostopalveluiden maksaminen ja toimitustavan valinta
- ▶ Tavoitteen lyhyt kuvaus parempi monille ohjelmiston sidosryhmille, joita ei kiinnosta seurata tapahtumia yksittäisten storyjen tarkkuudella

## Sprinttiin valittavat storyt

- ▶ Sprintin tavoitteen asettamisen lisäksi tulee valita backlogista sprintin aikana toteutettavat storyt
- ▶ Kehitystiimi päättää kuinka monta storya sprinttiin otetaan

## Sprintiin valittavat storyt

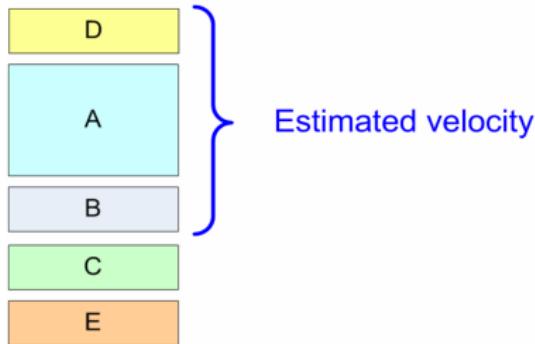
- ▶ Sprintin tavoitteen asettamisen lisäksi tulee valita backlogista sprintin aikana toteutettavat storyt
- ▶ Kehitystiimi päätää kuinka monta storya sprinttiin otetaan
- ▶ Jos velositeetti on selvillä, on valinta periaatteessa helppo



- ▶ Jos velositettia ei tiedossa, käytetään harkintaa #

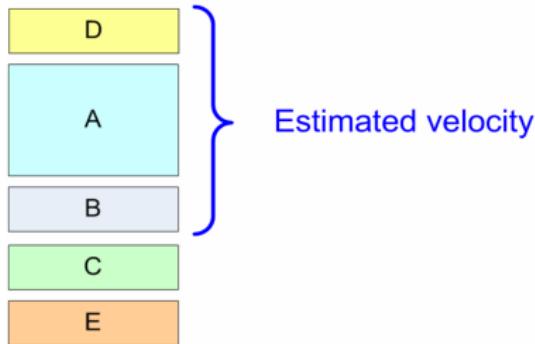
► Uudelleenpriorisoidaan

Option 1



- ▶ Uudelleenpriorisoidaan

Option 1



- ▶ Entä jos myös A halutaan mukaan?

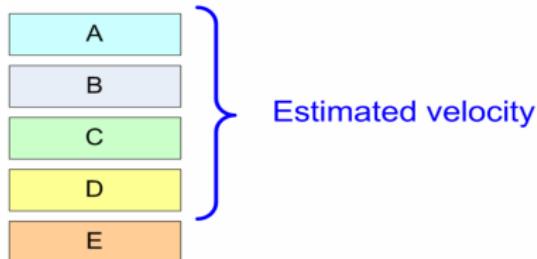
- ▶ Pienennetään A:n kuvaamaa toiminnallisuutta

### Option 2



- ▶ Pienennetään A:n kuvaamaa toiminnallisuutta

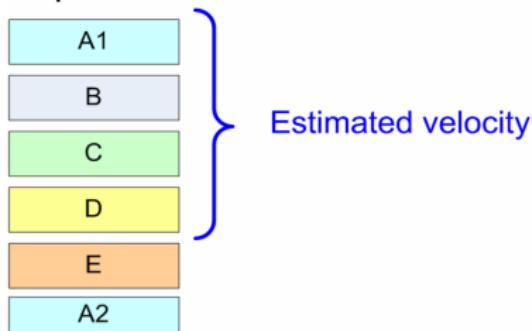
### Option 2



- ▶ Entä jos A ei saa pienentyää

- ▶ Jaetaan A kahteen osaan: tärkeämpään ja vähemmän tärkeään

### Option 3



## Miten sprintin tavoitteeseen päästään?

- ▶ Sprintin suunnittelun yhteydessä sprinttiin valituille User storyille tehdään karkean tason suunnittelu
- ▶ Mietitään mitä teknisen tason tehtäviä (task) on toteutettava, jotta user story saadaan valmiaksi

## Miten sprintin tavoitteeseen päästään?

- ▶ Sprintin suunnittelun yhteydessä sprinttiin valituille User storyille tehdään karkean tason suunnittelu
- ▶ Mietitään mitä teknisen tason tehtäviä (task) on toteutettava, jotta user story saadaan valmiiksi
- ▶ Suunnitellaan komponentteja ja rajapintoja karkealla tasolla
- ▶ Huomioidaan user storyn aiheuttamat muutokset olemassa olevaan osaan sovelluksesta

# Miten sprintin tavoitteeseen päästään?

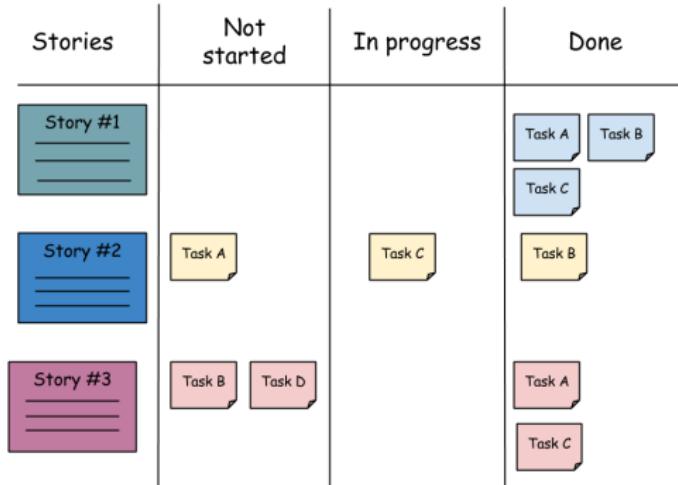
- ▶ Sprintin suunnittelun yhteydessä sprinttiin valituille User storyille tehdään karkean tason suunnitelu
- ▶ Mietitään mitä teknisen tason tehtäviä (task) on toteutettava, jotta user story saadaan valmiiksi
- ▶ Suunnitellaan komponentteja ja rajapintoja karkealla tasolla
- ▶ Huomioidaan user storyn aiheuttamat muutokset olemassa olevaan osaan sovelluksesta
- ▶ Kaikkia storyyn liittyviä taskeja ei sprintin suunnittelun aikana löydetä
- ▶ Uusia taskeja generoidaan tarvittaessa sprintin edetessä

## Storyn jako taskeihin, esimerkki

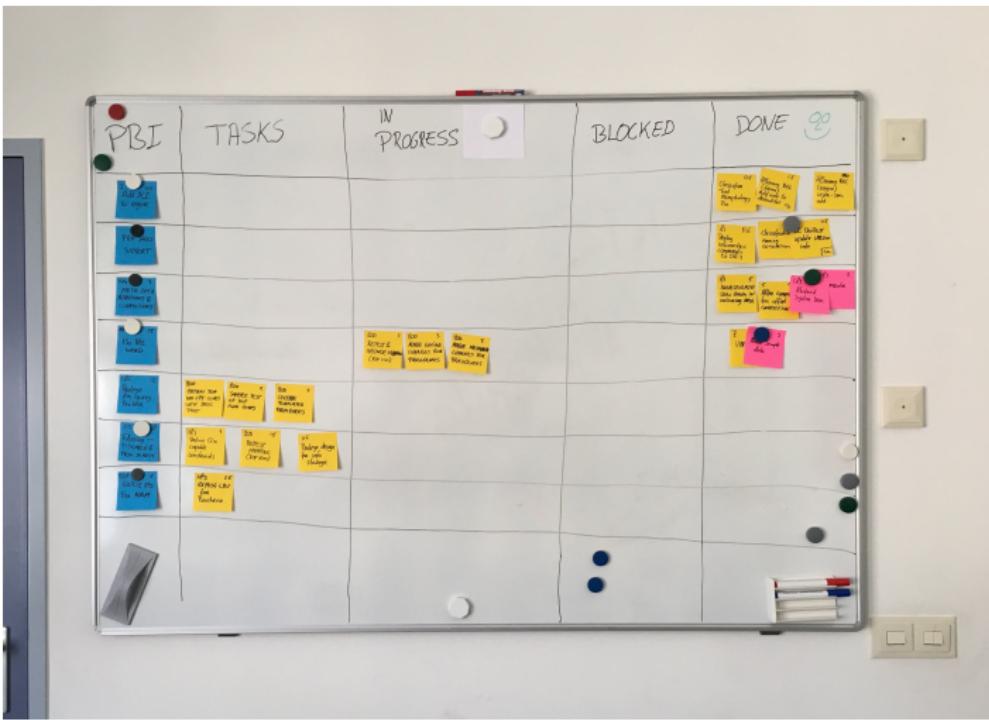
- ▶ Esimerkiksi *tuotteen lisääminen ostoskoriiin*, voitaisiin pilkkoaa seuraaviin taskeihin:
  - ▶ tarvitaan sessio, joka muistaa asiakkaan
  - ▶ domain-olio ostoskorin ja ostoksen esittämiseen
  - ▶ laajennus tietokanttaskeemaan
  - ▶ html-näkymää päivitettyvä tarvittavilla painikkeilla
  - ▶ kontrolleri painikkeiden käsittelyyn
  - ▶ yksikkötestit kontrollerille ja domain-olioille
  - ▶ hyväksymätestien automatisointi

# Sprint backlog

- ▶ Sprint backlog koostuu sprinttiin valituista storeista ja niihin liittyvistä tehtävistä eli taskeista
- ▶ Sprint backlog usein organisoitu taskboardiksi



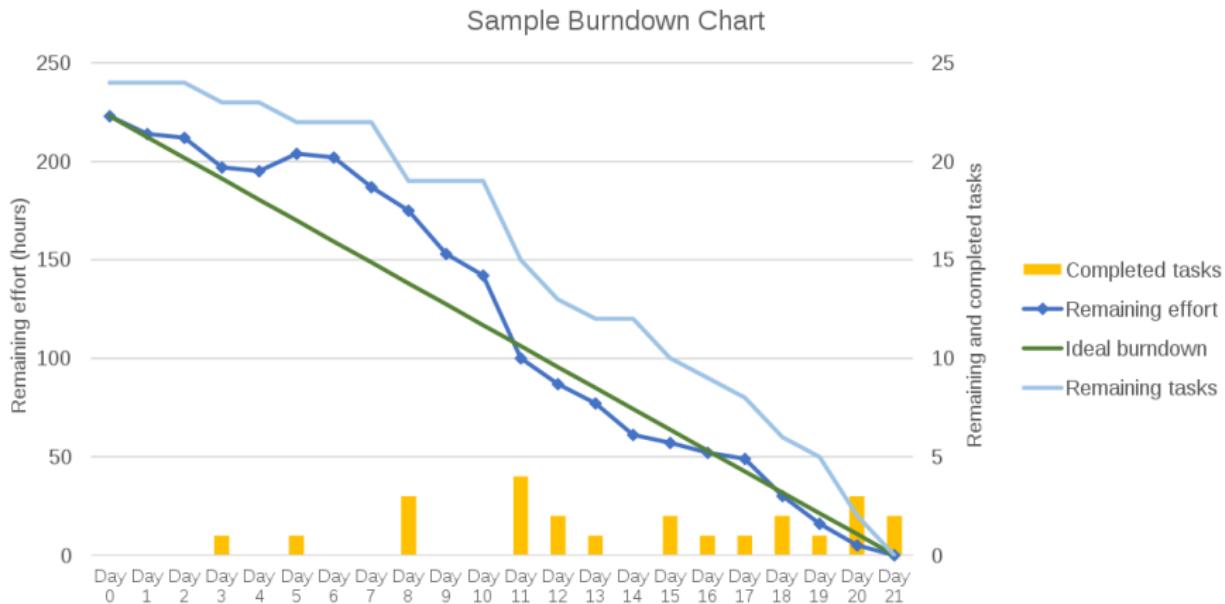
- ▶ Taskit niiden valmistumisastetta kuvaavassa sarakkeessa



- ▶ Taskeilla voi olla muitakin statuksia

# Taskien työmäääräarviot ja sprintin burndown

- ▶ Sprintissä arvioidaan päivittäin jäljellä olevaksi arvioitua työmäärää
- ▶ Usein arviot tunteina

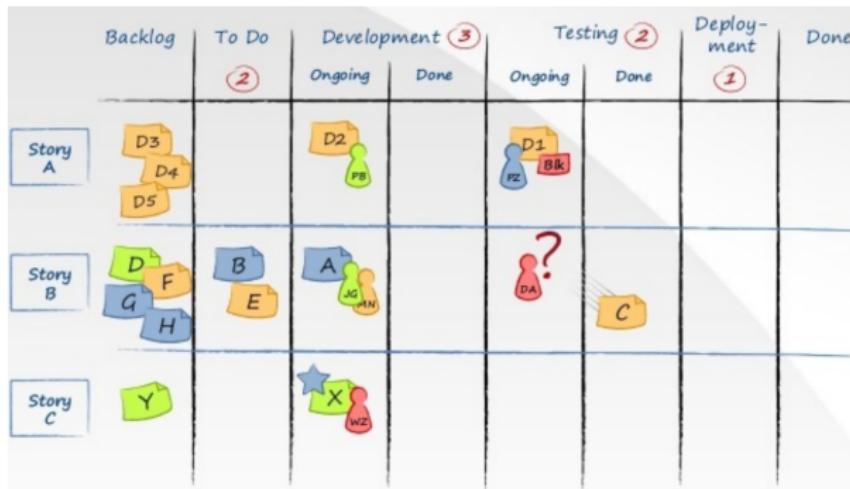


## WIP-rajoitteet

- ▶ Yhtä aikaa työn alla olevien taskien suuri määrä voi koitua ongelmaksi
- ▶ Riski sille, että sprintin päätyttyä paljon osittain valmiita storyja kasvaa

# WIP-rajoitteet

- ▶ Yhtä aikaa työn alla olevien taskien suuri määrä voi koitua ongelmaksi
- ▶ Riski sille, että sprintin päätyttyä paljon osittain valmiita storyja kasvaa
- ▶ Ratkaisu: *work in progress eli WIP* -rajoitteet



# Kanban ja Lean

- ▶ WIP-rajoitusten idea on peräisin *Kanban*-menetelmästä, joka on eräs keskeisimmistä *Lean*-ajattelun työkaluista

# Kanban ja Lean

- ▶ WIP-rajoitusten idea on peräisin *Kanban*-menetelmästä, joka on eräs keskeisimmistä *Lean*-ajattelun työkaluista
- ▶ Lean-ajattelun taustalla on idea *hukan* eli asiakkaalle arvoa tuottamattomien asioiden eliminoimisessa
- ▶ Lean-ajattelu on peräisin jo kymmeniä vuosia vanhasta Toyota Production Systemistä

# Kanban ja Lean

- ▶ WIP-rajoitusten idea on peräisin *Kanban*-menetelmästä, joka on eräs keskeisimmistä *Lean*-ajattelun työkaluista
- ▶ Lean-ajattelun taustalla on idea *hukan* eli asiakkaalle arvoa tuottamattomien asioiden eliminoimisessa
- ▶ Lean-ajattelu on peräisin jo kymmeniä vuosia vanhasta Toyota Production Systemistä
- ▶ Turhuksia: osittain tehty työ, välivarastointi ja turha odottaminen
  - ▶ Työn alla olevat storyt jotka eivät ole vielä DoD-tasolla valmiina edustavat hukkaa
  - ▶ testaamista odottavien toiminnallisuksien katsotaan olevan välivarastoituna
  - ▶ samoin testatut mutta tuotantoon viemistä vielä odottavat toiminnallisuudet

# Kanban ja Lean

- ▶ WIP-rajoitusten idea on peräisin *Kanban*-menetelmästä, joka on eräs keskeisimmistä *Lean*-ajattelun työkaluista
- ▶ Lean-ajattelun taustalla on idea *hukan* eli asiakkaalle arvoa tuottamattomien asioiden eliminoimisessa
- ▶ Lean-ajattelu on peräisin jo kymmeniä vuosia vanhasta Toyota Production Systemistä
- ▶ Turhuksia: osittain tehty työ, välivarastointi ja turha odottaminen
  - ▶ Työn alla olevat storyt jotka eivät ole vielä DoD-tasolla valmiina edustavat hukkaa
  - ▶ testaamista odottavien toiminnallisuksien katsotaan olevan välivarastoituna
  - ▶ samoin testatut mutta tuotantoon viemistä vielä odottavat toiminnallisuudet
- ▶ Toiminnallisuudet tuovat arvoa vasta kun ne saadaan käyttöön, sitä ennen ne sitovat turhaan kustannuksia ja aiheuttavat harmia

## WIP-rajoitteiden soveltaminen

- ▶ WIP-rajoitteita voidaan soveltaa Scrumin yhteydessä monella tapaa

## WIP-rajoitteiden soveltaminen

- ▶ WIP-rajoitteita voidaan soveltaa Scrumin yhteydessä monella tapaa
- ▶ Järkevästi on rajoittaa sprintin aikana yhtäaikaa työn alla olevien storyjen määrää mahdollisimman pieneksi
- ▶ On myös tavallista rajoittaa eri työvaiheessa, esim. toteutuksen olevien taskien määrää
- ▶ tai yksittäisellä sovelluskehittäjän kerrallaan työn alla olevien töiden määrää

## WIP-rajoitteiden soveltaminen

- ▶ WIP-rajoitteita voidaan soveltaa Scrumin yhteydessä monella tapaa
- ▶ Järkevästi on rajoittaa sprintin aikana yhtäaikaa työn alla olevien storyjen määrää mahdollisimman pieneksi
- ▶ On myös tavallista rajoittaa eri työvaiheessa, esim. toteutuksen olevien taskien määrää
- ▶ tai yksittäisellä sovelluskehittäjän kerrallaan työn alla olevien töiden määrää
- ▶ WIP-rajoitteita säädetään usein retrospektiivien yhteydessä jos kehitystyössä havaitaan ongelmia

## Kannattaako sprintin sisäinen estiomointi

- ▶ On hieman kiistanalaista kannattaako storyihin liittyvien taskien työmääriä estimoida. Syksyllä 2019 julkaistu kirja A Scrum book, ei suosittele taskien tasolla pidettävää työmääräarviota, vaan kehottaa seuraamaan sprinttienkin aikana ainoastaan sitä kuinka monen story pointin verran user storyja on saatu valmiiksi.

## Kannattaako sprintin sisäinen estiomointi

- ▶ On hieman kiistanalaista kannattaako storyihin liittyvien taskien työmääriä estimoida. Syksyllä 2019 julkaistu kirja A Scrum book, ei suosittele taskien tasolla pidettävää työmääräarviota, vaan kehottaa seuraamaan sprinttienkin aikana ainoastaan sitä kuinka monen story pointin verran user storyja on saatu valmiiksi.
- ▶ Neuvo onkin varsin viisas, on nimittäin mahdollista, että tiimi saa sprintissä valmiiksi lähes kaikki taskit, saamatta kuitenkaan yhtäkään storya täysin valmiiksi, eli burn down voi näyttää melko hyvältä mutta asiakkaan saama arvo on lopulta nolla.

## Kannattaako sprintin sisäinen estiomointi

- ▶ On hieman kiistanalaista kannattaako storyihin liittyvien taskien työmääriä estimoida. Syksyllä 2019 julkaistu kirja A Scrum book, ei suosittele taskien tasolla pidettävää työmääräarviota, vaan kehottaa seuraamaan sprinttienkin aikana ainoastaan sitä kuinka monen story pointin verran user storyja on saatu valmiiksi.
- ▶ Neuvo onkin varsin viisas, on nimittäin mahdollista, että tiimi saa sprintissä valmiaksi lähes kaikki taskit, saamatta kuitenkaan yhtäkään storya täysin valmiaksi, eli burn down voi näyttää melko hyvältä mutta asiakkaan saama arvo on lopulta nolla.
- ▶ Fyysisä taskboardia käytettäessä sprintin työmäärän estimoinnin ja burndown-käyrän piirtämisen sijaan erittäin simppeli tapa sprintin etenemisen seurantaan on laskea, tai jopa ainoastaan visuaalisesti katsoa taskboardilta, mikä on jo valmiiden ja vielä valmistumattomien sprinttiin kuuluvien taskien lukumäärä.

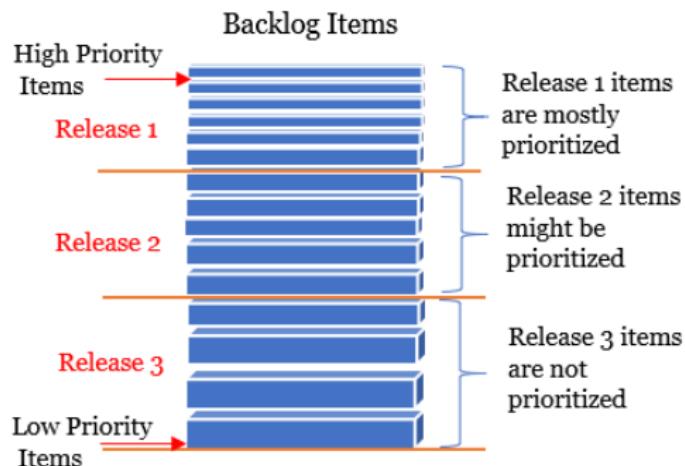


## Julkaisun suunnittelu ja projektin roadmap

- ▶ Useamman sprintin yli ulottuvaa suunnittelua kutsutaan *julkaisun suunnitteluksi*

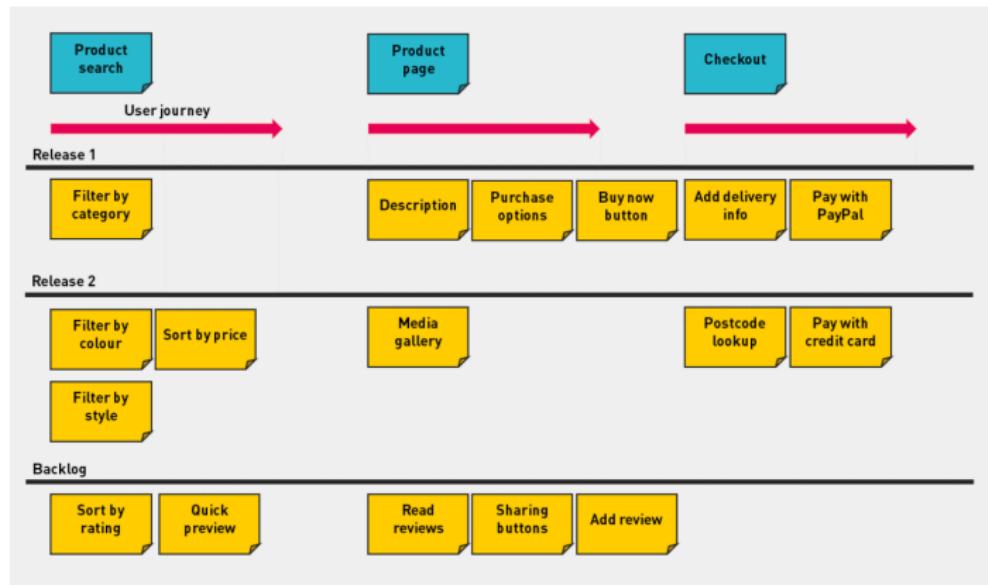
# Julkaisun suunnittelu ja projektin roadmap

- ▶ Useamman sprintin yli ulottuva suunnittelua kutsutaan *julkaisun suunnitteluki*
- ▶ Jakaa backlogin isompiin teemallisiin kokonaisuuksiin
  - ▶ milestone/release 1: verkkokaupan perustoiminnallisuus
  - ▶ milestone/release 2: tuotteiden arvostelu ja arvosteluihin perustuva suosittelu
  - ▶ milestone/release 3: käyttäjien ostoshistoriaan liittyvä tuotteiden suosittelu



# User story mapping

- ▶ Kokonaisuus voi olla vaikeaa hahmottaa backlogin tarjoamasta lineaarisesta näkymästä



- ▶ Rivilistät sprinttejä/julkaisuja