

# Vaatimusmäärittely ja Scrum-backlogit

# Vaatimusmäärittely

- *Vaatimusmäärittely* on ohjelmistotuontoprosessin vaihe, jossa määritellään *asiakkaan vaatimukset* toteutettavalle ohjelmistolle
- Vaatimusmäärittelyn aikana vaatimukset selvitetään ja *dokumentoidaan*, jotta kehitystiimi voi aloittaa teknisen suunnittelun ja toteutuksen
- Ketterässä ohjelmistokehityksessä vaatimusmäärittelyä tehdään iteratiivisesti vähän kerrassaan
- Asiakas (Scrumissa tuoteomistaja) *priorisoi* vaatimukset
- Kuhunkin iteraatioon valitaan toteutettavaksi ne vaatimukset, jotka tuovat asiakkaalle *mahdollisimman paljon liiketoiminnallista arvoa*

# User story

- Ketterän vaatimusmäärittelyn tärkein työväline on *user story*
  - User storyt kuvaat loppukäyttäjän kannalta *arvoa tuottavia toiminnallisuuksia*
  - User storyt kirjoitetaan *asiakkaan ymmärtävällä kielellä*, eli ne eivät saa sisältää turhan teknistä kieltä
- 
- ✗ Käyttäjänä voin lähettää HTTP-kutsun rekisteröitymisestä vastaavaan REST API-endpointtiin, joka lisää käyttäjätunnuksen ja salasanan tietokantatauluun "users"
  - ✓ Käyttäjänä voin rekisteröityä järjestelmään käyttäjätunnuksella ja salasanalla

# Hyvän user storyn kriteerit

- Erään suositun määritelmään mukaan hyvä user story toteuttaa *INVEST*-kriteerit:
  - **I**ndependent: user story on riippumaton muista user storyista
  - **N**egotiable: user storyn kuvauksessa on joustoa toteutuksen suhteen
  - **V**aluable: user story tuottaa arvoa
  - **E**stimable: user storyn toteutukseen kuluva aika on määriteltävissä
  - **S**mall: user story on riittävän pieni, toteutus vie enintään yhden sprintin
  - **T**estable: user storyn toteutuksen toimivuus on todennettavissa

# Hyvän user storyn kriteerit

Yleisin rike INVEST-kriteereitä kohtaan on se, että *user story on liian suuri*:

✗ Käyttäjänä voin rekisteröityä järjestelmään käyttäjätunnuksella, salasanalla, profiilikuvalla ja profiilikuvauksella

Tässä tilanteessa user story kannattaa *pilkkoa useaksi user storyksi*:

✓ Käyttäjänä voin rekisteröityä järjestelmään käyttäjätunnuksella ja salasanalla

✓ Käyttäjänä voin rekisteröitymisen yhteydessä antaa profiilikuvan

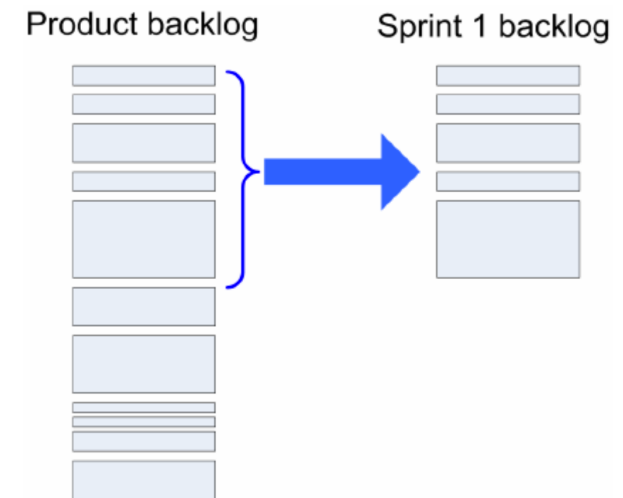
✓ Käyttäjänä voin rekisteröitymisen yhteydessä antaa profiilikuvauksen

# Product backlog

- Scrumissa *product backlog* on tuoteomistajan priorisoima lista user storyja
- Product backlogin kärjessä eli korkeimmilla prioriteetilla olevat user storyt valitaan toteutettavaksi seuraavan sprintin aikana
- Product backlogin user storyja lisätään, muokataan ja priorisoidaan jatkuvasti sprinttien edetessä

# Sprint backlog

- Product backlogilta valitaan sprintin aikana toteutettavat user storyt sprintin tehtävänä toimivaan *sprint backlogiin*
- Kehitystiimi valitsee product backlogin kärjestä user storyt, jotka kokevat pystyvänsä toteuttaa sprintin aikana
- Kehitystiimi pilkkoo user storyt teknisen tason *taskeiksi*
- Taskit lisätään sprint backlogille



# User storyn pilkkominen taskeiksi

Esimerkiksi seuraavan user storyn:

Käyttäjänä voin rekisteröityä järjestelmään käyttäjätunnuksella ja salasanalla

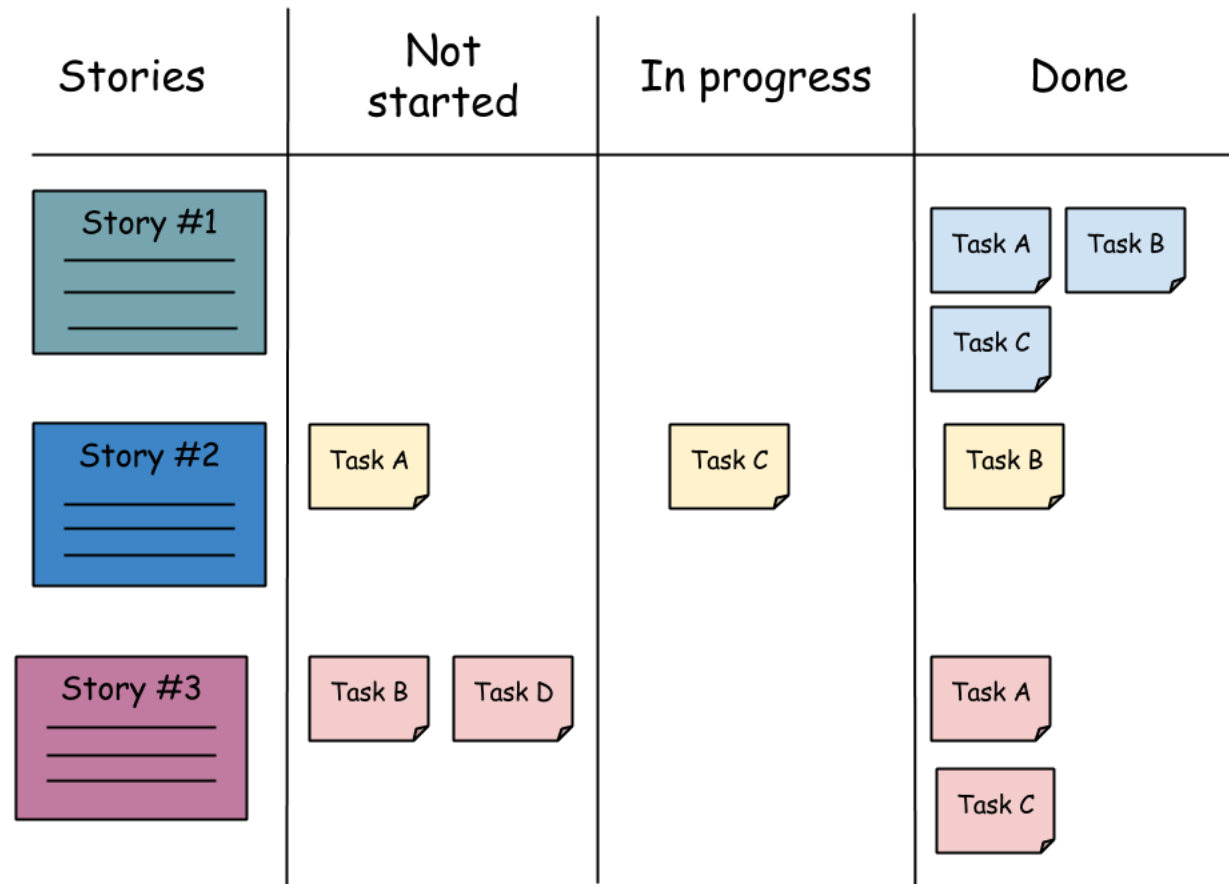
Voisi jakaa esimerkiksi seuraaviin teknisiin taskeihin:

- Backendiin User JPA entity -luokka ja UserRepository JPA repository -luokka
- Backendiin UserRestController controller -luokka ja POST /api/users REST API endpoint -metodi, joka tallentaa käyttäjän tietokantaan
- Frontendiin React-komponentti rekisteröitymislomakkeelle, jossa on kentät käyttäjätunnukseksi ja salasanalle



# Sprint backlog

- Sprint backlog organisoidaan usein taulukkomaiseksi *taskboardiksi*, jossa on yksi rivi kutakin sprinttiin valittua user storya kohti
- User storyyn liittyvät taskit kulkevat vasemmalta oikealle niiden tilaa kuvaavien sarakkeiden, kuten "not started", "in progress", "done" kautta



# Backlogien käyttö sprintin suunnittelussa

- Sprintin suunnittelun aikana Scrum-tiimi päivittää product backlogia tulevaa sprinttiä varten:
  - Lisää tarvittaessa uusia user storyja, tai muokkaa olemassaolevia
  - Priorisoi user storyja
- Kun product backlogin on ajantasalla, valitsee kehittäjätiimi sopivan määrän user storyja seuraava sprinttiä varten
- Valitut user storyt pilkotaan kehittäjätiimin toimesta teknisiksi taskeiksi ja taskit lisätään sprint backlogille
- Kehittäjätiimi sopii sprintin työnjaosta: kuka ottaa minkäkin taskin vastuulleen

# Backlogien käyttö sprintin aikana

- Sprintin aikana kehitystiimi päivittää jatkuvasti sprint backlogilla olevien taskien tilaa
  - Esimerkiksi kun jokin taski valmistuu, tulee se siirtää "done"-tilaan
- Sprint backlogin tarkoitus on siis peilata reaaliajassa sprintin edistymistä ja edistää prosessin *läpinäkyvyyttä*
- Sprint backlogia tarkkaillaan jatkuvasti ja siitä keskustellaan kehitystiimin tapaamisissa

# Backlogien hallintaan soveltuvia alustoja

- Lähitöissä backlogit ovat tyypillisesti valkotalulle muodostettuja taskboardeja, joissa taskit ja user storyt ovat post-it -lappuja
- Backlogien hallintaan verkossa löytyy useita palveluita, kuten *GitHub projects*, *Jira* ja *Trello*
- Näistä etenkin GitHub projects on saavuttanut suurta suosiota, koska se linkittyy luontevasti GitHubin versionhallintapalveluihin
- Päättäkää tiimin kesken omaan työskentelyynne parhaiten sopiva alusta. Voitte myös halutessanne kokeilla useampaa alustaa kurssin aikana