

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 *käyttäjätarina* (user story)
- Käyttäjätarinat kuvaat loppukäyttäjän kannalta *arvoa tuottavia toiminnallisuuksia*
- Käyttäjätarinat kirjoitetaan *asiakkaan ymmärtävällä kielellä*, eli ne eivät saa sisältää turhan teknistä kieltä

✗ Käyttäjänä voin lähettää HTTP-kutsun rekisteröitymisestä vastaavaan API-endpointtiin, joka lisää käyttäjätunnuksen ja salasanan tietokantatauluun "users"

✓ Käyttäjänä voin rekisteröityä järjestelmään käyttäjätunnuksella ja salasanalla

Hyvän käyttäjätarinan kriteerit

- Erään suositun määritelmään mukaan hyvä käyttäjätarina toteuttaa *INVEST*-kriteerit:
 - **Independent**: käyttäjätarina on riippumaton muista käyttäjätarinoista
 - **Negotiable**: käyttäjätarinan kuvauksessa on joustoa toteutuksen suhteen
 - **Valuable**: käyttäjätarina tuottaa arvoa
 - **Estimable**: käyttäjätarinan toteutukseen kuluva aika on määriteltävissä
 - **Small**: käyttäjätarina on riittävän pieni
 - **Testable**: käyttäjätarinan toteutuksen toimivuus on todennettavissa

Hyvän käyttäjätarinan kriteerit

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

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

Tässä tilanteessa käyttäjätarina kannattaa *pilkkoa useaksi käyttäjätarinaksi*:

✓ Käyttäjänä voin rekisteröityä järjestelmään käyttäjätunnuksella ja salasanalla

✓ Käyttäjänä voin rekisteröitymisen yhteydessä antaa profiilikuvan

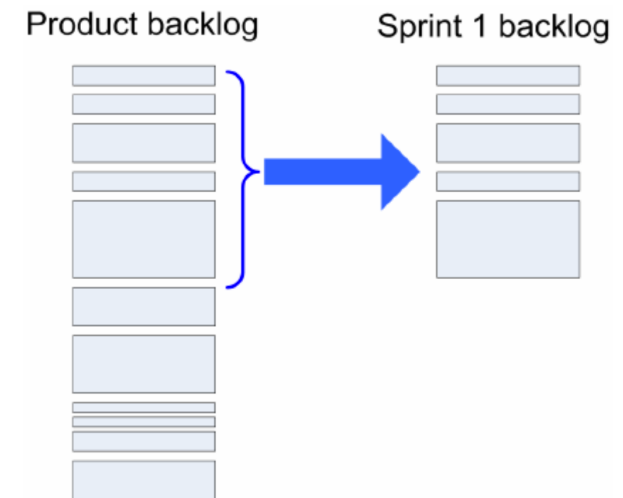
✓ Käyttäjänä voin rekisteröitymisen yhteydessä antaa profiilikuvauksen

Product backlog

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

Sprint backlog

- Product backlogilta valitaan sprintin aikana toteutettavat käyttäjätarinat sprintin tehtävälistana toimivaan *sprint backlogiin*
- Kehitystiimi valitsee product backlogin kärjestä käyttäjätarinat, jotka kokevat pystyvänsä toteuttaa sprintin aikana
- Kehitystiimi pilkkoo käyttäjätarinat teknisen tason *tehtäviksi* (task)
- Tehtävät lisätään sprint backlogille



Käyttäjätarinan pilkkominen tehtäviksi

Esimerkiksi tämän käyttäjätarinan:

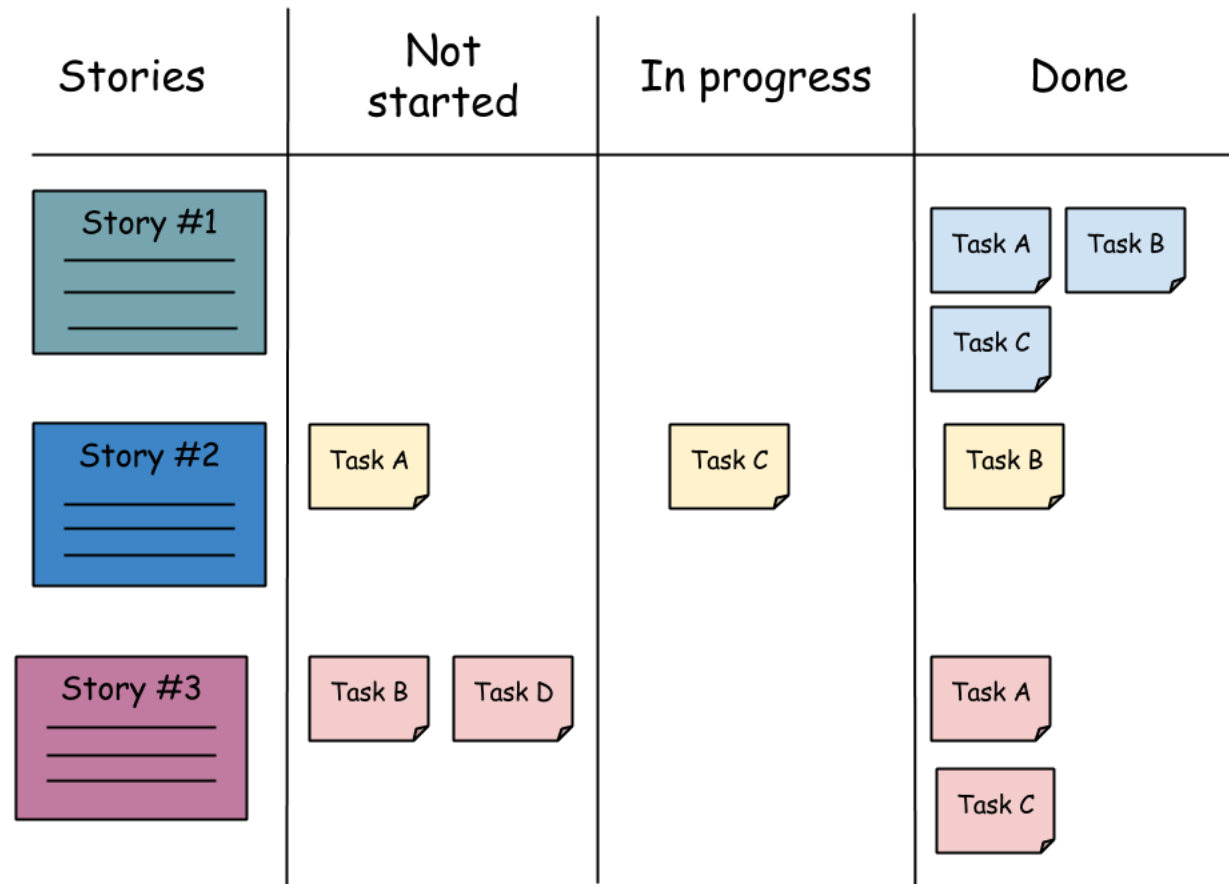
Käyttäjänä voin rekisteröityä järjestelmään käyttäjätunnuksella ja salasanalla

Voisi jakaa esimerkiksi näihin teknisiin tehtäviin:

- Kirjautumislomake, jossa on tekstikentät käyttäjätunnukselle ja salasanalle
- Tietokantaulu "users", jossa sarakkeet "username" ja "password"
- HTTP API-endpoint /api/register rekisteröitymislomakkeen lähettämiseksi

Sprint backlog

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



Sprintin suunnittelu

- Sprintin suunnittelu -tapaamisen aikana Scrum-tiimi päivittää product backlogia tulevaa sprinttiä varten:
 - Lisää tarvittaessa uusia käyttäjätarinoita, tai muokkaa olemassaolevia
 - Priorisoi käyttäjätarinoita
- Kun product backlogin on ajantasalla, valitsee kehittäjätiimi sopivan määrän käyttäjätarinoita seuraava sprinttiä varten
- Valitut käyttäjätarinat pilkotaan kehittäjätiimin toimesta teknisiksi tehtäviksi ja tehtävät lisätään sprint backlogille
- Kehittäjätiimi sopii sprintin työnjaosta: kuka ottaa minkäkin tehtävän vastuulleen

Sprintin aikana

- Sprintin aikana kehitystiimi päivittää jatkuvasti sprint backlogilla olevien tehtävien tilaa
 - Esimerkiksi kun jokin tehtävä valmistuu, tulee se siirtää "done"-tilaan
- Sprint backlogin tarkoitus on siis peilata reaaliajassa sprintin edistymistä
- Sprint backlogia kannattaa tarkkailla aktiivisesti, etenkin kehittäjätiimin tapaamisissa