**Sprinttien aikataulu**

**Aikataulu**

Kehitystyö tehdään sprinteissä käyttäen Scrum-mallia. Kokonaisuudessaan tehdään 7 sprinttiä. Sprintin päätteeksi tehdään katselmointi/retropektiivi/suunnitelma seuraavasti sprintistä. Dokumentaatio talletataan versionhallintaan (esim. erillinen kansio projektidokumenteille)

**30.10 - 12.11 . Sprint zero:**

Esitellään verkkokaupan aihe, alustava etusivu (voi olla kovakoodattu, koodit jaettu ja tallennettu Githubiin), tiimin jäsenet ja roolit sekä alustava tuotteen kehitysjono (backlog) Hacknplanissä. Hacknplan ja Github tulee olla jaettuna tiimin jäsenten kesken sekä opettajille Administrator-oikeuksilla. Lisäksi on hyvä hahmotella jo suunnitelmaa UI:stä (Wireframe/Mockup)

Sprint 1: 13.11 - 19.11

Sprint 2: 20.11 - 26.11

Sprint 3: 27.11 - 3.12

Sprint 4: 4.12 - 10.12

Sprint 5: 11.12 - 17.12

18.12 Projektin palautus

**Arvioitavat osa-alueet**

Kaikki tekevät ryhmästään vertaisarvion. Opettaja arvioi työn ja käyttää arviossa apuna vertaisarvioita. Näin ollen kaikilla ei ryhmässä ole välttämättä sama arvosana, jos esim. osallisuudessa selkeitä eroja.

**Arvioitavat osa-alueet:**

* Sivuston toteutus
* Projektinhallinta kuten backlog, Kanban -ylläpito (hacnplan)
* Tehtävien jako ja ryhmän yhteydenpito
* Dokumentaatio kuten sprinttien palaverimuistiot ja esim. UI-suunnitelma  
  -Asianmukainen versionhallinta
* Vertaisarviot (pakolliset)  
  -Työtuntien arviot ja kirjaamiset  
  -Esittelyvideo

**Katselmus, Retro ja suunnitelma**

Jokaisen (paitsi viimeisen) sprintin päättyessä pidetään sprintin katselmointi, retrospektiivi sekä seuraavan sprintin suunnittelupalaveri. Alla palaverissa läpi käytäviä asioita, joista kirjataan oleelliset asiat ylös:

* Todetaan paikallaolijat
* Saavutettiinko sprintille asetetut tavoitteet?
* Tehtiinkö kaikki sprintin käyttäjätarinat/tehtävät?
* Tapahtuiko sprintissä jotain poikkeamia?
* Miten tiimi toimii? Mikä toimii hyvin, missä on haasteita?
* Miten yksilöt ja tiimi voisi parantaa toimintaansa?
* Mikä on seuraavan sprintin tavoitteet?
* Mitkä käyttäjätarinat/tehtävät toteutetaan seuraavassa sprintissä?
* Onko tekijöitä, jotka vaikuttavat työn tekemiseen (poissaoloja, vapaapäiviä)?

Muistakaa palauttaa palaverin memo versionhallintaan!

Chatgpt:n vastaus:

Sprintin katselmus (Sprint Review) ja retrospektiivi (Sprint Retrospective) ovat kaksi tärkeää tapahtumaa Scrum-kehyskäytännössä, mutta niillä on erilaiset tarkoitukset ja painopisteet.

**Sprintin katselmus (Sprint Review):**

* Tarkoitus: Sprintin katselmus on tarkoitettu siihen, että kehitystiimi esittelee valmiin inkrementin (toimivan osuuden tuotteesta) sidosryhmille ja asiakkaille.
* Kohdeyleisö: Katselmuksen yleisö koostuu sidosryhmistä, kuten asiakkaista, tuotepäälliköistä ja muista kiinnostuneista osapuolista.
* Sisältö: Katselmuksessa käydään läpi toteutettu toiminnallisuus, keskustellaan saavutetuista tuloksista ja mahdollisesti muutetaan tuotteen tulevaa suunnitelmaa tai prioriteetteja.

**Retrospektiivi (Sprint Retrospective):**

* + Tarkoitus: Retrospektiivi keskittyy tiimin oman työskentelyn parantamiseen. Tavoitteena on tarkastella menneen sprintin tapahtumia, tunnistaa, mitä meni hyvin ja mitä voitaisiin parantaa, ja suunnitella parannustoimenpiteitä tulevaisuutta varten.
  + Kohdeyleisö: Retrospektiivi on suunnattu tiimille itselleen. Tiimi keskustelee avoimesti ja rehellisesti omasta työskentelystään ja pyrkii yhdessä löytämään tapoja parantaa tehokkuutta ja tiimityötä.
  + Sisältö: Retrospektiivissä voidaan käyttää erilaisia menetelmiä, kuten Start-Stop-Continue, Glad-Sad-Mad tai muut, arvioimaan tiimin suorituskykyä ja löytämään parannusmahdollisuuksia. Tulokset voivat johtaa konkreettisiin toimenpiteisiin seuraavalle sprintille.

Yhteenvetona, sprintin katselmus on suunnattu tuotteen sidosryhmille ja keskittyy valmiin inkrementin esittelyyn ja keskusteluun siitä, kun taas retrospektiivi on suunnattu tiimille itselleen ja pyrkii parantamaan tiimin työskentelytapoja tulevaisuutta varten. Molemmat tapahtumat ovat keskeisiä jatkuvaan parantamiseen Scrum-kehyskäytännössä.

**Roolit**

Opiskelijoilla voi olla tiimissä erilaisia rooleja. Roolit voivat olla koko projektin ajan pysyviä tai rooleja voidaan vaihtaa, mikäli tiimi näin haluaa. Mahdollisia rooleja ovat:

**Scrum Master**

* Scrum-prosessin toteuttaminen: Scrum Master varmistaa, että tiimi noudattaa Scrum-prosessia ja sen periaatteita. He auttavat tiimiä ymmärtämään Scrum-roolit, tapahtumat ja artefaktit.
* Esteiden poistaminen: Scrum Masterin tehtävänä on tunnistaa ja poistaa esteitä, jotka voivat haitata tiimin etenemistä. Tämä voi sisältää ongelmia, jotka liittyvät esimerkiksi resursseihin, viestintään tai ulkoisiin riippuvuuksiin.
* Valmentaminen: Scrum Master toimii valmentajana tiimille, auttaen heitä ymmärtämään ja omaksumaan ketterät periaatteet. Tavoitteena on parantaa tiimin itseorganisoitumista ja lisätä heidän kykyään ratkaista ongelmia.
  + Sprint-tapahtumien fasilitointi: Scrum Master helpottaa Sprint-suunnittelua, päivittäistä Scrumia, Sprint-arviointia ja Sprint-arviointia varmistaen, että tapahtumat pidetään ja että niissä käsitellään asianmukaisia asioita.
* Yhteistyö sidosryhmien kanssa: Scrum Master toimii linkkinä tiimin ja sidosryhmien välillä. He auttavat kommunikoimaan tiimin edistymisestä ja pitävät sidosryhmät tietoisina projektin tilasta.
* Jatkuva parantaminen: Scrum Master kannustaa tiimiä jatkuvaan parantamiseen. He auttavat tiimiä reflektoimaan omasta työskentelystään ja etsimään tapoja tehdä asiat entistä tehokkaammin.
* On tärkeää huomata, että Scrum Master ei ole esimies tiimille eikä myöskään ole vastuussa projektin tuloksista. Heidän tehtävänsä on enemmänkin tukea tiimiä omaksumaan ketterät periaatteet ja helpottaa heidän työskentelyään mahdollisimman tehokkaasti.

**Kehittäjä (front-end, backen-end tai molemmat)**

**Graafinen suunnittelija**

**Tuotteen omistaja eli product owner**

* Product Backlogin ylläpito: Product Owner vastaa tuotteen kehityslistaan, jota kutsutaan Product Backlogiksi. He määrittelevät, priorisoivat ja päivittävät backlogia niin, että se heijastaa parhaiten asiakkaan tarpeita ja yrityksen tavoitteita.
  + Vaatimusten määrittely: Product Owner selventää tiimille asiakkaan vaatimuksia ja odotuksia. He työskentelevät yhdessä asiakkaan kanssa ja keräävät palautetta, jotta tiimi voi ymmärtää paremmin, mitä tuote vaatii.
  + Priorisointi ja sprint suunnittelu: Product Owner on vastuussa backlogin priorisoinnista ja sprintin suunnittelusta. He valitsevat tärkeimmät ja kiireellisimmät ominaisuudet ja määrittelevät niiden tavoitteet.
* Viestintä tiimin ja sidosryhmien välillä: Product Owner välittää tietoa tiimin ja sidosryhmien (kuten asiakkaiden tai liiketoiminnan edustajien) välillä. Tämä varmistaa, että kaikki osapuolet ymmärtävät tuotekehityksen suunnan ja etenemisen.
* Inkrementin hyväksyntä: Product Ownerilla on valtuudet hyväksyä tai hylätä tuotteen kehityksen tuloksia. He varmistavat, että tiimin kehittämä tuote vastaa asiakkaan odotuksia ja vaatimuksia.
* Muutosten hallinta: Jos asiakkaan tai sidosryhmien tarpeet muuttuvat projektin aikana, Product Owner vastaa näiden muutosten hallinnasta ja integroinnista kehitysprosessiin.
* Yhteenvetona voidaan sanoa, että Product Ownerilla on keskeinen rooli tuotteen menestyksen varmistamisessa koko kehitysprosessin ajan. He ovat vastuussa siitä, että tiimi keskittyy oikeisiin asioihin ja että kehitettävä tuote tuottaa arvoa asiakkaalle ja organisaatiolle.

Roolien infot chatgpt :P

Edellä mainittuja rooleja ei ole pakko täyttää mutta joka tapauksessa olisi hyvä, että ryhmästä valitaan yksi vastuullinen, joka huolehtii sprintti-kokouksien järjestämisestä.

**Kansiorakennetta**

my-app/  
├── public/  
│ └── index.html # Staattinen HTML-pohja, johon React-sovellus kiinnittyy.  
├── src/  
│ ├── components/ # Kaikki React-komponentit.  
│ │ ├── common/ # Yleiset, uudelleenkäytettävät komponentit.  
│ │ │ ├── NavBar.js # Navigointipalkin komponentti.  
│ │ │ ├── Footer.js # Alatunnisteen komponentti.  
│ │ │ └── ... # Muut yleiset komponentit.  
│ │ ├── ProductList.js # Tuoteluettelon komponentti.  
│ │ ├── ProductCard.js # Yksittäisen tuotteen kortti.  
│ │ └── ... # Muut sovelluskohtaiset komponentit.  
│ ├── pages/ # Sivukohtaiset komponentit, kuten näkymät tai layoutit.  
│ │ ├── Home.js # Etusivun komponentti.  
│ │ ├── About.js # "Tietoa meistä" -sivun komponentti.  
│ │ └── ... # Muut sivut.  
│ ├── styles/ # CSS-tyylitiedostot.  
│ │ ├── common.css # Yleiset tyylit.  
│ │ ├── NavBar.css # NavBar-komponentin tyylit.  
│ │ ├── Footer.css # Footer-komponentin tyylit.  
│ │ ├── ProductList.css # ProductList-komponentin tyylit.  
│ │ └── ... # Muut tyylitiedostot.  
│ ├── utils/ # Apufunktiot ja utility-skriptit.  
│ ├── hooks/ # Custom React-hookit.  
│ ├── context/ # React Context API:n määritelmät.  
│ ├── assets/ # Staattiset tiedostot, kuten kuvat ja ikonit.  
│ │ ├── images/ # Kuvatiedostot.  
│ │ ├── icons/ # Ikonitiedostot.  
│ │ └── ... # Muut staattiset resurssit.  
│ ├── App.js # Sovelluksen juurikomponentti.  
│ ├── index.js # Sovelluksen pääsyöttökohta.  
│ └── ... # Muut mahdolliset JavaScript-tiedostot.  
├── package.json # NPM-paketin määrittelytiedosto.  
├── server/ # serverin hilppeet  
└──...