# Sprint 2 (uge 36-37)

## **Programmering:**

Kobl sitet op til aspnet core mvc, med routing

Sørg for at udskille fælles html i shared layout

Byg en model for data og vis den i et razor view

Lav en custom route til en underside på sitet

## **Systemudvikling**

Beskriv usikkerheder og kompleksiteter i projektet – Hvordan har I fundet frem til dem? Hvordan påvirker de jeres procesmodel? Reflekter over hvordan dette påvirker jeres projektstyringsaktiviteter**.**

**Teknik:**

* Vi vil nedskrive kompleksiteter og usikkerheder, for at skabe et varieret overblik over projektet, samt forbedre os på dets fremtidige forløb.

Værktøj:

* + Vi bruger passende CL-strukturer, såsom “tænk-par-del” og “Ordet-rundt”, for at finde frem til kompleksiteter og usikkerheder, først individuelt (computational thinking) og senere fremført og diskuteret i teamet.

**Kompleksitet**

* Vi er allesammen på et forskelligt niveau i forhold til programmering og systemudvikling.
* Vi har ikke en PO og sparrer med.
  + Arbejder vi lineært (da vi skal følge undervisningsforløbet) eller cirkulært?
* Vi har ikke særlig meget erfaring indenfor projekter, så den lineære processmodel kan være svær at forholde sig til, da vi højst sandsynligt får problemer med nogle tasks. Det gør at vi rigtig gerne vil vende tilbage til nogle tasks.
* Vi er ikke selv herrer over vores sprints, da de er prædefineret til et læringsforløb.

**Usikkerhed**

* Systemet skal laves som website og det har vi ikke prøvet før.
* Vi ved ikke hvad vi skal lave i næste uge.
* Vi skal lave et system ud fra et perspektiv vi ikke har arbejdet med før.

**Procesmodel**

* Vi forestiller os at vores projektmodel er inspireret af den cirkulær-inkrementelle procesmodel. Eftersom vi for hvert sprint laver en del af det samlede produkt. Dermed vil vi først have salgbart produkt til sidst i projektet.
  + Ved at undersøge usikkerheder og kompleksiter, får vi en bedre forståelse for projektets forløb og dermed de fremtidige udfordringer der kan opstå. Dette er med til at styrke vores cirkulær-inkrementelle procesmodel, da det mindsker antallet af udfordringer/ overraskelser og dermed antallet af gange vi ikke når at færdiggøre vores sprints.

**Refleksion:**

* Lad os antage at vores projektstyring foregår lineært, dette kan medfører at der opstår problemer i forhold til ens projekt styringsaktiviteter. Lad os antage at vi ikke når de tasks, som er på det pågældende sprint er det umuligt for os at tage dem videre til næste sprint. Dette har altid været muligt da vi fra 1 til 2 semester har arbejdet med cirkulære procesmodeller og vil derfor blive en udfordring hvis vi skulle arbejde med en lineær proces model..

## **Teknologi:**

Opdater listen fra Iteration 1.

Tilføj eller uddyb forklaringen af protokoller der bliver anvendt via jeres løsning (direkte eller indirekte).

* (OPGAVEBESKRIVELSE SPRINT 1): I skal i alle iterationer overveje hvilken teknologisk kontekst jeres løsning, og de enkelte produkter I udarbejder, skal fungere i. Selvom I ikke selv direkte implementerer eller kontrollerer de netværksmæssige værktøjer og teknologier, skal I have en overordnet forståelse af, hvad der er i spil ”under motorhjelmen” samt hvad I skulle producere, hvis disse værktøjer og teknologier ikke var til rådighed (”hvad nu, hvis I skulle gøre det samme, men befandt jer i år 1999?”).

Udarbejde derfor en liste over emner/begreber fra faget Teknologi II, som anvendes (i en eller anden forstand) i jeres løsning. Giv hvert emne en kort, beskrivende kommentar.

### **Sprint 1:**

TOPOLOGI:

* Mesh Topologi (Multipoint connection)

NETVÆRK:

* WAN
* Domæne (rettigheder)
* Server (Webhotel/ hosting)
* Database

### **Sprint 2:**

**Vi forestiller os følgende scenarie efter ARP-protokollen er taget i brug:**

* Vi har en server Host (vores hjemmeside, simply.com)
  + Enhed: Server (simply) ?
  + Server host’en vil have en default gateway (router), som server host’en kender vha. af et ARP table, hvori routerens IP-adresse og MAC-adresse er kendt.
  + Server host’en vil være koblet til en switch i sit lokale netværk (LAN),
    - Switchen vil have en tilhørende “MAC Adresse Table”, med server host’ens MAC-adresse og switch-port den er tilkoblet. Derudover vil tabellen indeholde Routerens MAC-adresse og switch-port (OSI L2).
  + Switchen er koblet til en router, for at kommunikere til clients hostnes netværk (IP-adresse).
    - Routeren vil have en tilhørende “Router ARP Table”, med server host’ens IP-adresse og tilhørende MAC-adresse (OSI L3).
    - Routeren vil derudover have en tilhørende “Routing Table”, med server host’ens IP-adresse(+ submask), samt dens tilhørende Router interface og method, som vi antager er Directly Connect. Samtidigt vil tabellen indeholde client host’nes IP-adresse(+ submask), samt dens tilhørende Router interface og method, som vi antager er Dynamic Routing (OSI L3).
* Vi har en til mange client Hosts (brugerne af hjemmesiden)
  + Enhed: Stationær computer, bærbar computer
  + En client host vil have en default gateway (router), ligsom server host’en
  + En client host vil være koblet til en switch i sit lokale netværk (LAN), ligesom det er beskrevet ovenover i forbindelsen med server host’en.
* Switchen er koblet til en router, for at kommunikere til server host’ens netværk (IP-adresse).
  + Routeren vil have en tilhørende “Router ARP Table”, med client host’ens IP-adresse og tilhørende MAC-adresse.
  + Routeren vil derudover have en tilhørende “Routing Table”, med client host’ens IP-adresse(+ submask), samt dens tilhørende Router interface og method, som vi antager er Directly Connect. Samtidigt vil tabellen indeholde server host’ens IP-adresse(+ submask), samt dens tilhørende Router interface og method, som vi antager er Dynamic Routing.

**Hvad sker der når clienten skriver vores url i deres browser:**

1. Først skriver clienten url’en for at connecte til serveren, hjemmesiden er hosted på.
2. Browseren leder efter ip addressen af domænet i DNS(domain name server).
3. browseren starter en tcp connection med serveren.
4. browseren sender en http request til serveren.
5. serveren tager sig af http requesten og sender en http response tilbage til clienten.
6. browseren viser nu html indholdet på siden.
7. clienten er beskyttet af protokollen https. Det er en sikkerhedsprotokol der beskytter på 3 primære områder. Disse områder er kryptering, godkendelse og dataintegritet.