# Programmering

* Brug Entity Framework
* Brug Taghelpers
* Valider input client side og server side
* Tilføj:
* \* View components/Partial view
* \* Authorization
* \* Authentication

# 

# Systemudvikling

Opgave: Beskriv jeres systemudviklingsmetode. Hvilke teknikker benytter I i de forskellige fokusområder? Hvordan sikrer i sporbarhed i metoden?

**Systemudvikling Metode**

* Vi anvender DSDM, i form af prototyper og en iterativ fremgangsmåde.

**Opstart og projektstyring:**

* TEKNIK: Vi vil på bedst mulig måde sikre os et godt projektforløb, for at mindske risikoen for at projektet går til grunde og sikre et godt arbejdsudbytte.
  + Værktøj: SCRUM
  + Værktøj: Kanban tavle.
  + Værktøj: Gantt skema.

**Kravindsamling:**

* TEKNIK: Vi vil sikre os at vi laver det produkt som kunden ønsker, for at mindske misforståelser under udviklingsfasen.
  + Værktøj: Lytte og evt. interviewe kunden. Her kan fx de fem W’er bruges (Who, what, when, where, why)
* TENIK: Vi vil sikre os, at vi har dækket krav over forskellige interessenter, for at det endelige system er fordelagtigt for flest mulige aktører.
  + Værktøj: Kategoriser kravene således de mest prominent interessenters krav er dækket. Her kan man inddele kategorierne ift FURPS(+).
* TEKNIK: Vi vil danne et overblik over kunden og dens virksomhed, for at sikre vi laver et brugbart produkt til forretningen og dens krav.
  + Værktøj: FML
  + Værktøj: Business case
    1. Nu når vi har FML, vil vi gerne benytte den til at se hvilke segmenter af virksomheden der mangler optimering.

**High Level Design:**

* TEKNIK: Vi vil lave beskrivelser af brugssituationer for at finde ud af hvad systemet skal kunne gøre
  + Værktøj: Use case
* TEKNIK: Vi vil visualisere stamdata i form af objekter/klasser, som indgår i virksomhedens brugssituationer.
  + Værktøj: Objektmodel
  + Værktøj: Domænemodel
* TEKNIK: Vi vil ud fra projektets brugssituationer, data og krav bestemme kodens overordnede arkitektur, for at sikre læsbar kode, nem vedligeholdelse og skalerbarhed i fremtiden.
  + Værktøj: SOLID
  + Værktøj: GRASP
  + Værktøj: CRUD
  + Værktøj: CL-Struktur “Ordet-rundt” til at finde arkitekturen (MVC, MvvM, osv.)
* TEKNIK: Vi vil danne os et overblik over hvilke komponenter der er i spil i hele systemet, så vi fra start ved hvilke kompetencer i udviklingsteamet der er behov for.
  + Værktøj: Rich Picture, FML

**Low Level Design:**

* TEKNIK: Vi vil undersøge de systemoperationer som indgår i projektets brugssituationer.
  + Værktøj: SSD
* TEKNIK**:** Vi vil med indblik på projektets systemoperationer undersøge, hvilke betingelser der skal eksistere før operationen eksekveres og hvad det medføre.
  + Værktøj: OC
  + Værktøj: SSD
  + Værktøj: SD
  + Værktøj: Objekt Modellen

**Programmering:**

* TEKNIK: Vi vil sikre os den bedst mulige kode fra start, så vi undgår at lave en

refaktorering af en masse kode senere.

* + Værktøj: Best Practice
* TEKNIK: Vi vil have fokus på at fremhæve kodens intention, for at sikre at koden kan videreudvikles og læses af andre udviklere, samt gøre det let for os selv at vende tilbage til koden og forstå den.
  + Værktøj: Bruge passende variable, klasse og metode navne, samt skrive kommentarer til koden, således det er let at sætte sig ind i den.

**Test:**

* TEKNIK: Vi skal sikre at de værdier, som bliver vist i programmet er korrekte, hvis ikke kan det skabe uforudsigelige situationer og forkerte data.
  + Værktøj: Unitest
  + Værktøj: System testing
* TEKNIK: Vi vil sikre at systemet har et brugervenligt design, så vi mindsker læringstiden for at kunne benytte systemet og øge brugernes fortrolighed til programmet.
  + Værktøj: Usability Test

**Udrulning:**

* TEKNIK: Vi skal sikre at virksomhedens ansatte er i stand til at kunne bruge hjemmesiden, fordi?
  + Værktøj: Support / Vejledning
* TEKNIK: Vi skal sikre at systemet kommer mest hensigtsmæssigt ud i verden, fordi?
  + Værktøj: Implementeringsstrategi

**Vedligeholde:**

* TEKNIK: Vi skal sikre at systemet kan vedligeholdes og benyttes fremadrettet, fordi det bliver en byrde for brugeren hvis systemet ikke fungere efter en operativ opdatering.
  + Værktøj: Vedligeholdelsesplan
* TEKNIK: Vi skal sikre at systemet opdateres med jævne mellemrum og ikke for ofte, fordi det kan blive en byrde for brugeren at skulle opdatere systemet ofte.
  + Værktøj: Vedligeholdelsesplan

Man skal lave store opdateringer

Ikke for små opdateringer hele tiden, vente til det kan komme med i en større patch

**SPORBARHED:**

First Step: Teamet skal danne sig et overblik over hvilke krav systemet og kunden skal have, for at vide at det system de skal lave er det kunden ønsker.

* Krav -> Skitseret i Domænet

Second Step: Teamet begynder efterfølgende at danne skitser for domænet og får forståelse inden for det system de skal bygge.

* Skitseret Domæne -> Teoretisk prototype

Third Step:Dernæst skal teamet så småt i gang med at bygge en teoretisk prototype vha. artefakter, finde datatype osv.

* Teoretisk prototype -> Praktisk prototype

Fourth Step: Efterfølgende kan teamet tage den teoretiske prototype og omdanne det til en logisk prototype. Det gør de i programmeringen og kigger altid tilbage på hvad de gjorde i tredje step så de sikrer sporbarhed i koden og artefakter.

* Logisk Prototype -> Udrulning

Fifth Step: Teamet tester prototypen igennem og når denne fase er færdig kan de tage prototypen med til udrulningsfasen.

* Udrulning - > Næste iteration

Sixth Step:Er teamet tilfreds med prototypen sammenkobler de prototypen med den iteration af prototypen som lå før den iteration de har været i gang med. De starter nu på en ny iteration af prototypen.

Teknologi