# Projektgennemførelse

## Udviklingsmodel

Projektet er gennemført med en modificeret udgave af *ASE*-modellen(REFERENCE) som den primære udviklingsmodel. ASE-modellen er som udgangspunkt en vandfaldsmodel. Man arbejder sig systematisk frem med at lave opgaveformulering, kravspecifikation og systemarkitektur, for så at designe og implementere modulerne i hardware og software hver for sig og til sidst laver man en samlet accepttest.

Figur : ASE process

I dette projekt ville man gerne undgå en ren vandfaldsmodel. Derfor blev elementer fra *Verification* and *Validation modellen* også kaldet V-modellen indført. Alle trin i ASE-modellen blevet parret med et tilsvarende afsluttende trin. Se figur XX.



Figur : Verification and Validation model

Kravspecificationen blev udarbejdet sammen med accepttesten, systemarkitekturen sammen integrationstestene og til designfasen/implementeringen laves modultestene til alle enhederne.

V-modellen ligger op til at hvert trin i modellen afsluttes før der påbegyndes et nyt. Dette valgte man undervejs at ændre i projektgennemførelsen da forudsætningen for at dette kan lade sig gøre kræver at man har et overblik over hvilke teknologier der skal bruges i projektet og hvordan disse kan implementeres. V-modellen er derfor blevet brugt for at sammenholde de forskellige trin, men ved siden af, har projektet haft iterærer processer til hvert enkelt delsystem.

Typisk har disse processer bestået af teknologiundersøgelser og forsøg med at udvikle fungerende funktionelle udgaver af delsystemerne.(REFERENCE til teknologi undersøgelserne). Iterationerne er blevet styret ved hjælp af SCRUMsprint og i de senere sprint lykkes det også at have visse delsystemer færdige til at vise kunden/vejlederen.

## Projektstyring

### Scrum

Fra starten af semesteret blev der besluttet at projektet skulle styres så vidt muligt vha. Scrum som er en agil udviklingsmetode, hvilket var en af grundene for at undgå en ren vandfaldsudviklingsmodel. Scrum har de fordele:

* At man ikke behøver at kende alle krav i begyndelsen af projekt før man kan begynder at udvikle.
* Man deler projektet op i iterationer kaldet sprint så projektet bliver lettere at overskue.
* Hver iteration ender ud i et produkt man kan demonstrer for kunden/produktejeren.
* Hver dag (eller hver anden dag for denne gruppe) mødes man til stå op møde og får overblik over opgaver og eventuelle problemer.
* Det er nemt og hurtigt at reallokerer resurser til der hvor der er mest brug for dem.

Det skal nævnes at Scrum er et softwareudviklingsværktøj men at vi i denne også valgte at bruge det til hardwareudvikling.

Nøglebegreberne som vi har brugt fra Scrum er:

* Stå Op møde
* Denne mødeform består i at man, i denne gruppe, hver anden dag har mødtes stående og stillet hvert medlem tre sprøgsmål om hvad man lavede sidst, hvad skal man lave som det næste og om man har eventuelle problemer med at udføre sine opgaver.
* Roller
* Som Scrum foreskriver har gruppen haft 8 medlemmer hvor et par af medlemmerne har delt Scrummasterrollen. Scrummasterrollen har således været fordelt med at et gruppemedlem har fungeret som primær kontaktperson med produktowner/vejleder og en anden har fungeret som ro og ordenholder.
* Produkt Backlog
* Backloggen består af alle opgaver som skal laves før projektet er færdig. I dette projekt er den blevet udfyldt løbende i fællesskab. Backloggens opgaver skal som udgangspunkt være delt op i Storys når det laves. Når et Sprint planlægges ligges opgaverne over på taskboarded.
* Taskboard og Sprint Backlog
* Taskboardet har fungeret som en liste over opgaver med prioritering, organisator og kontrakt med hvad Scrumgruppen har valgt at forpligtige sig til af opgaver.

Kort beskrivelse af hvert Scrum Sprint

* Sprint 1
* Længde: En uge.
* Formål: At lave kravspecifikationen, accepttesten og samt organiseringen af projektet.
* Sprint 2
* Længde: Tre uger.
* Formål: At lave teknologiundersøgelser af forbindelser, sensorer, MIDI, GUI samt at afslutte kravspecifikationen og accepttesten og påbegynde systemarkitekturen.
* Sprint 3
* Længde: To uger
* Formål: At lave de første implementeringer af delsystemer samt forsætte med at lave systemarkitekturen.
* Sprint 4
* Længde: Tre uger.
* Formål: At få lavet en stor del af hardwareimplementeringen færdig, få overblik over Rock delen af system, forsæt med systemarkitekturen og forsætte med implementeringen af delsystemerne.
* Sprint 5
* Længde: To uger
* Formål: At få overblik over Body delen af systemet, videreudvikle delsystemer samt gå ned i den dybere del af software opbygningen på Rock delen af system.
* Sprint 6
* Længde: En uge
* Formål: At få systemet til at generer lyd fra sensor til Body til Rock til lydafgiver.
* Sprint 7
* Længde: To uger
* Formål: At skrive produktdokumentation og projektrapport færdig, ???, ???

### Tidsplan

Den overordnede tidsplan(REFERENCE) der strækker sig igennem hele projektet er blevet tilegnet vha. Microsoft Projekt. Tidsplanen er blevet brugt til at holde overblikket over tidsrammen for projekt og er blevet opdateret efter hvert gruppe/vejledermøde.

### Mødestruktur

Gruppe- og vejledermøder er blevet styret ved hjælp af en mødeindkaldelse efterfulgt af et møde med dagsorden, dirigient og referent. De administrative roller er blevet fastlagt vha. en turnusordning(REFERENCE TIL TURNUSORDNING), hvor at de forskellige roller som mødeindkalder, referent og dirigent skifter fra møde til møde. Dette er gjort for at sikre at alle gruppemedlemmer får et indblik i det administrative arbejde. For at sikre konsensus i dokumenter, er der udarbejdet skabeloner til mødeindkaldelser og referater. Møderne er blevet afholdt efter behov, med udgangspunkt i et møde ved et sprints begyndelse og afslutning. Referatet fra forrige møde er blevet gennemgået og godkendt ved hvert møde.

## Dokumentorganisering og log

Gruppen har benyttet Git til organisering af dokumenter, kode og genereldokumentation for projektet. Git består af et repository som kan rumme alle disse data. Det har den fordel at alt hvad sendes til repository’et skal have en medfølgende beskrivelse og bliver stemplet med både dato/tidspunkt og præcise informationer om hvilke data bliver sendt med. Git’en har derfor også fungeret som gruppens fælles log for projektet. Gits funktion med at merge dokumenter er kun blevet brugt til dels.

(REFERENCE TIL GITloggen)