**Titelblad**

Resume:

Redegørelse for opgavens overordnede problem og hvilke metoder/resultater I brugte/fik til at undersøge problemet og stille krav til produktet. Hvilket produkt I fremstillede, hvordan det løser problemet og opfylder kravene.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Navn på gruppemedlemmer:

Skolen navn:

Klasse og årgang:

Vejleders navn:

Antal sider jfr. indholdsfortegnelsen:

Antal anslag uden mellemrum:

Perioden i hvilken projektet har været gennemført:

Jeg bekræfter herved med min underskrift, at opgavebesvarelsen er udarbejdet af mig. Jeg har ikke

anvendt tidligere bedømt arbejde uden henvisning hertil, og opgavebesvarelsen er udfærdiget uden

anvendelse at uretmæssig hjælp og uden brug af hjælpemidler, der ikke har været tilladt under

prøven.

Dato:\_\_\_\_\_\_\_\_\_\_\_\_ Underskrift: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dato:\_\_\_\_\_\_\_\_\_\_\_\_ Underskrift: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dato:\_\_\_\_\_\_\_\_\_\_\_\_ Underskrift: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dato:\_\_\_\_\_\_\_\_\_\_\_\_ Underskrift: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dato:\_\_\_\_\_\_\_\_\_\_\_\_ Underskrift: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DDU/Teknikfags-rapport – Skabelon.**

Nærværende dokument er bygget op, ud fra vejledningen 5P’er fra UVM - 2021. Man må gerne bruge den som skelet/skabelon, eller I kan lave jeres egen struktur, men hav rigtigt gode argumenter (internt i gruppen) hvis der indgår punkter/afsnit i skabelonen, I vælger at skippe. Alt hvad der er skrevet med rødt er ”Meta-tekst” (forklarende tekst). Det slettes og erstattes med det endelige indhold. Husk det er jer der er afsender af jeres rapport! Den røde tekst er ikke spørgsmål I skal besvare, men idéer og gode råd til indhold.

Double Diamond modellen er en anden måde at strukturere rapporten på. Herunder en skitse af hvordan 5P kobler sig til Double Diamond:

Et billede, der indeholder tekst, skærmbillede, brev, design

Automatisk genereret beskrivelse

Checkliste – Rigtig gode begreber at have med uanset hvad I laver:

|  |  |
| --- | --- |
| De 5’Per (Problemidentifikation, problemanalyse…) |  |
| Double diamond |  |
| Scrum (eller ligende) |  |
| Tidsplan (hack/plan, iterationer i scrum eller anden) |  |
| Målgruppeanalyse (minerva, gallup etc) |  |
| Personaer |  |
| Idegenerering (brainstorm, rollespil, de Bonos tænkehatte, mindmap etc) |  |
| Indledende kravspecifikation (kravmatrix) |  |
| Detaljeret kravspecifikation ( nice to have, need to have) |  |
| Strukturdiagram |  |
| Skitser |  |
| Wireframes |  |
| Paper prototype/marvel prototype |  |
| High fidelity/low fidelity |  |
| Mechanics dynamics aestetics (hvis I laver spil) |  |
| Programmering |  |
| Flowdiagrammer/pseudocode/klassediagrammer |  |
| Brugertest (kvalitative og kvantitative tests) |  |
| IT-sikkerhed |  |
| Brugervejledning |  |
| Rapport |  |
| Minimum viable product |  |
| Overvejelser om roller i forløbet (Scrums, Belbins etc) |  |
| Interaktionsdesign (usability, gestaltlove, Nielsens 10 principper) |  |
| IT-sikkerhed (kryptering etc) |  |
| GDPR? |  |
| Ophavsret |  |

**FORSIDE**

Evt. illustration som hænger sammen med emne (husk kilde på billede / illustration)

**Indholdsfortegnelse**

Forord 2

Indledning 3

Problemidentifikation 3

Problemanalyse 3

Problemformulering 3

Produktprincip 4

Produktudformning 4

Produktionsforberedelse 4

Realisering 4

Test 5

Konklusion 5

Vurdering 5

Kildeliste 5

Bilag 5

Tidsplan 5

Logbog 5

Diverse illustrationer og andet, som er for store til at være i selv rapporten 5

# Forord

Rammer for værkstedbrug, tidsramme, tak til personer, der har hjulpet jer.

Arbejdsfordeling, evt. belbins grupperoller hvis anvendt, evt. gruppekontrakt.

**(Eksempel på) gruppekontrakt:**

1. Vi indvilger i at bruge den tid vi har til rådighed i skolen

2. Hvis vores mål ikke er blevet nået i den tid der er til rådighed på skolen, aftaler vi at mødes og arbejde i fritiden.

3. Vi snakker sammen om tingene, (laver aftaler og overholder dem). Hvis nogen er utilfredse med noget, så siger man det.

4. Vi vil lave en fuldt ud gennemført opgave, og stræbe efter at få en god karakter ved fælles indsats.

Underskrifter ….

# Indledning

Hvorfor er emnet interessant? (gør det spændende for læseren, giv den nødvendige baggrundsviden, der kan lede op til problemanalysen) Hvad er det I vil undersøge? Hvordan vil I undersøge dette?

Hvad bidrager jeres arbejde til?

Hvordan har vi opbygget rapporten (metakommuniker strukturen for læseren)

# Problemidentifikation

Opstil en relevant teknisk problemstilling, hvis den ikke allerede er givet. Afgræns og afgør hvad der er vigtigt at undersøge i den senere problemanalyse, for at kunne lave den tekniske løsning. Research og dokumenter, at der er et reelt problem, der formodes at kunne afhjælpes med en teknisk løsning (tænk et ”problem”, som noget der kan have én eller flere tekniske løsninger). Når problemet er identificeret, konkretiseret og afgrænset opstilles problemformuleringen.

**Problemformulering.** i Teknikfag, arbejder I med et teknisk problem, det må gerne skinne igennem her. F.eks. *”I hvilken grad kan problemet med ’A’, afhjælpes med en løsning der anvender ’Teknologien B’”. Eller ”Hvordan kan man ved hjælp af A og B designe et system der kan C”* Det er vigtigt at problemformuleringen ikke kan besvares med Ja eller Nej og det er på den anden side også vigtigt, at den er konkret nok, til at den kan besvares/konkluderes på, i slutningen af projektet.

Hvis man hopper for hurtigt hen over problemidentifikationen risikerer man at:

* lave en løsning til et fiktivt problem
* spilde tid på at analysere på irrelevante ting
* Mangle fokus og at løsningen stikker i flere retninger / er fragmenteret / ukonkret

Mere konkret kan I blandt andet arbejde med følgende aspekter/faser:

1. **Problemforståelse**: Start med at forstå problemet grundigt. Hvad er det specifikke tekniske problem, du ønsker at løse? Hvilke aspekter af det påvirker løsningen?
2. **Stakeholder-analyse**: Identificer de interessenter, der er involveret i problemet. Dette kan omfatte brugere, udviklere, ledere, kunder osv. Stakeholdernes behov og perspektiver kan påvirke faktorerne.
3. **Miljømæssige faktorer**: Overvej det tekniske miljø, hvor problemet opstår. Er der specifikke hardware-, software- eller netværksbegrænsninger? Er der sikkerhedsmæssige overvejelser?
4. **Tidsbegrænsninger**: Er der en tidsramme for at løse problemet? Er der deadlines eller tidskritiske faktorer?
5. **Økonomiske faktorer**: Hvad er budgettet for løsningen? Er der økonomiske begrænsninger eller muligheder for investering?
6. **Tekniske afhængigheder**: Identificer eventuelle afhængigheder af andre systemer, teknologier eller komponenter.
7. **Risikofaktorer**: Overvej potentielle risici og usikkerheder i forbindelse med problemet.

Husk, at faktorerne kan variere afhængigt af den specifikke problemstilling, så det er vigtigt at tilpasse denne liste til dit konkrete projekt

Afsnittet skal gerne munde ud i **Relevant teknisk problemstilling**: Formulering af en teknisk udfordring, der er relevant for det givne projekt.

# Problemanalyse

Hovedformålet med problemanalysen er at finde ud af hvilke parametre der påvirker problemet, så I ultimativt set, kan lave en teknisk løsning, hvor I har nogle ”knapper at dreje på”, som indvirker på netop de parametre, der påvirker problemet.

Research og Indsaml på baggrund af problemidentifikationen, viden til hvordan I bedst muligt løser det tekniske problem. Brug rigeligt med relevante pålidelige kilder, statistik, empiri og hold egne meninger og mavefornemmelser tilbage. Det kan nogen gange være nødvendigt at eksperimentere og lave forsøg i denne fase, for at blive klogere (indsamle empiri).

**Tydeliggør afsenderen. Tag ikke ansvar for ekspertviden!** Selvom I bruger ekspertkilder, er der ingen garanti for at det er den endegyldige sandhed I har fat i, så sørg for at læsse ansvaret over på ekspertens egne skuldre, så I ikke kan blive klantret, men i stedet bare har refereret videre, hvad eksperten har sagt. Brug gerne citater, så det er tydeligt ”hvem der taler”.

Den værste bivirkning ved ikke at have lavet en ordentlig problemanalyse er: I udvikler noget teknologi, som muligvis virker, men ikke har nogen indvirkning på problemet I forsøger at løse.

Gode ting at have med:

1. **Kravspecifikation**: Identifikation og dokumentation af de nødvendige krav til løsningen. Dette inkluderer funktionelle og ikke-funktionelle krav.
2. **Brugerbehovsanalyse**: Forståelse af, hvad brugerne ønsker og har behov for. Dette indebærer ofte interviews, observationer og brugerundersøgelser.
3. **Systemkontekst**: Definering af det større system, som det aktuelle problem er en del af. Dette hjælper med at afgrænse problemet og identificere relevante aktører og komponenter.
4. **Fejlanalyse**: Undersøgelse af tidligere fejl eller problemer relateret til lignende systemer eller teknologier. Dette kan hjælpe med at undgå gentagne fejl.
5. **Risikoanalyse**: Identifikation af potentielle risici og usikkerheder i forbindelse med løsningen. Dette kan omfatte tekniske, tidsmæssige eller økonomiske risici.

# Produktprincip

I har nu lavet en god problemanalyse (!), og ved derfor præcis hvilke parametre der er relevante og påvirker problemet. Hvordan kan man dernæst konstruere noget teknologi, som kan ”dreje på nogle af de knapper”?

Idegenerer!

Brug de tekniske metoder I kender fra teknologifaget; brainstorm, associationsteknikker, kravmatrix etc). Bearbejd de forskellige løsningsforslag (diskutér dem, afprøv evt. nogle af idéerne hvis relevant og muligt). Vurder herunder også gennemførbarhed af løsningerne. Tegn gerne skitser!

Vigtige aspekter

1. **Brugervenlighed**: Produktet skal være let at bruge og intuitivt for brugerne. Dette omfatter aspekter som navigation, layout og tilgængelighed.
2. **Skalérbarhed**: Produktet bør kunne håndtere forskellige mængder data og brugere uden at miste ydeevne eller funktionalitet.
3. **Sikkerhed**: Beskyttelse af data og brugere er afgørende. Produktet skal implementere passende sikkerhedsforanstaltninger.
4. **Vedligeholdelse**: Produktet skal være nemt at vedligeholde og opdatere. Dette inkluderer kodekvalitet og dokumentation.
5. **Fleksibilitet**: Produktet skal kunne tilpasses forskellige behov og ændringer over tid.

# Produktudformning

Her konkretiseres løsningskonceptet.

Præsenter løsningsforslaget via tegninger, tekst, teknisk dokumentaton, relevant argumentation (tjek evt. tilbage gennem faserne, at der er en rød tråd, dvs. f.eks. at argumentationen besvarer og knytter sig direkte til de opstillede krav). Brug gerne empiri/målinger/beregninger der underbygger argumentationen. Beskriv hvilke tests, der kunne, kan og skal udføres for at validere, i hvilken grad systemet opfylder kravene.

1. **Brugervenlighed**: Produktet skal være let at bruge og intuitivt for brugerne. Dette omfatter aspekter som navigation, layout og tilgængelighed.
2. **Skalérbarhed**: Produktet bør kunne håndtere forskellige mængder data og brugere uden at miste ydeevne eller funktionalitet.
3. **Sikkerhed**: Beskyttelse af data og brugere er afgørende. Produktet skal implementere passende sikkerhedsforanstaltninger.
4. **Vedligeholdelse**: Produktet skal være nemt at vedligeholde og opdatere. Dette inkluderer kodekvalitet og dokumentation.
5. **Fleksibilitet**: Produktet skal kunne tilpasses forskellige behov og ændringer over tid.

# Produktionsforberedelse

Værktøjer, skaf komponenter, produktionsplan (Hvad skal gøres, hvem skal gøre det og hvornår?). Det er i denne fase, I har mulighed for at lægge en realistisk tidsplan der gør at I mener, I kan nå det afgrænsede mål. En tidsplan er det aller vigtigste projektstyringsværktøj, til at få udnyttet den arbejdskraft der er tilgængelig i gruppen, designe pipelines (dvs. Hvordan man gør ting i den smarteste rækkefølge, uden at bruge for meget ventetid) og nå sine deadlines. En tidsplan er gætterier, så man ændrer og opdaterer den løbende, som man bliver klogere!

Der laves desuden en BOM (**B**ill **O**f **M**aterials) aka. Stykliste, med priser. Det er vigtigt at I gør jer klar, om det er en BOM for prototypen (når man køber komponenter stykvis), eller en lille produktion (når man køber komponenter i 100 eller 1000-vis). Der er typisk MEGET stor forskel på priserne afhængigt af om man gør det ene eller det andet.

Opstil den endelige kravspecifikation her! Sørg for at Need to haves er simple hårde krav I er ret sikre på at kunne nå og noget som er til at teste på. Nice to haves kan godt være mere løse, fordi I jo ikke lover noget.

Eksempler på krav der er svære at teste på:

* Det skal være sjovt!
* Det skal ikke være for stort og heller ikke for tungt
* Det skal være æstetisk flot og ikke for dyrt

Refleksion: Hvem skal synes det er sjovt? Hvordan måler man humor? Hvad er stort og hvad er småt, tungt og let? Er en elefant på 1 ton let? Er en mus på 2kg er tung? Hvad med med æstetik og pris? Er en Boing727 til 800.000,- dyr, eller hvad en kuglepen til 450,- ?. Det er ikke fordi man ikke må have bløde krav, nogen gange er det relevant, men tænk over hvordan I vil teste når I opstiller kravene. Bløde krav kræver tit langt mere omfattende test og argumentation, end hårde krav.

Eksempler på krav der er nemme at teste på:

* Systemet skal kunne vise temperaturen med en afvigelse på +/- 2%
* Systemet skal kunne udføre test og dataopsamling uden indblanding fra operatør
* System skal kunne fungere i en temperatur på -10 til +80 grader

Vigtige overvejelser:

1. **Prototyping**: Oprettelse af en prototype eller en prøveversion af produktet for at teste funktionalitet, brugervenlighed og ydeevne.
2. **Materialer og leverandører**: Identifikation af de nødvendige materialer og leverandører til produktionen. Dette inkluderer at vælge de rigtige komponenter og sikre, at de er tilgængelige.
3. **Produktionsplanlægning**: Udvikling af en detaljeret plan for produktionsprocessen, herunder tidsplan, ressourcer og arbejdsgange.
4. **Kvalitetskontrol**: Implementering af procedurer og tests for at sikre, at produktet opfylder de nødvendige kvalitetsstandarder.
5. **Logistik og distribution**: Planlægning af, hvordan produktet skal distribueres til kunderne. Dette omfatter lagerstyring, forsendelse og levering.
6. **Miljømæssige overvejelser**: Sikring af, at produktionsprocessen overholder miljømæssige standarder og bæredygtighedskrav.

# Realisering

Udvikling af produktet. Husk det er en prototype / eller et mockup.

skru hellere ned for æstetikken og prioriter teknologien på det fysiske produkt, hvis der er tidspres. Hvilke værktøjer anvendes, billeder etc.

Vigtige begreber:

1. **Implementering**: Dette indebærer at skrive den faktiske kode eller udvikle den hardware, der udgør produktet. Det kan omfatte programmering, konstruktion af kredsløb, integration af komponenter osv.
2. **Test og validering**: Produktet testes grundigt for at sikre, at det fungerer som forventet. Dette inkluderer enhedstests, integrationstests og systemtests.
3. **Dokumentation**: Oprettelse af teknisk dokumentation, brugervejledninger og andre relevante materialer. Dette hjælper med at forstå og vedligeholde produktet.
4. **Produktionsforberedelse**: Forberedelse af produktionsprocessen, herunder valg af produktionsmetoder, materialer og leverandører.
5. **Fejlfinding og optimering**: Identifikation og løsning af eventuelle problemer eller flaskehalse i produktet. Optimering af ydeevne og effektivitet.

# Test

Afprøvning og test af produkt. Hold det op mod de tidligere angivede krav og vurder om / og hvor godt det opfylder disse. Beskriv hvordan I vil teste og argumenter for, at disse testmetoder er valide.

Test 1: Emnet testes ved at sænke det ned på 10m dybde, hvor det opbevares i en time.

Test 2: xxx

|  |  |  |  |
| --- | --- | --- | --- |
| **Krav** | Test x | Udfald | Vurdering |
| Skal være vandtæt ned til 10 meter | Udført | Emnet har ikke optaget vand efter manuel inspektion | Ok |
| Skal kunne… |  |  |  |
| Må ikke.. |  |  |  |
|  |  |  |  |

***Den dårlige test:*** *“Vi har testet systemet og vi synes, at det fungerede rigtigt godt!”*

*Hvad er der galt? Der er ingen reel information og der indgår desuden en påstand! Må vi se testen? Efter hvis mening? Hvor godt er rigtigt godt og hvilke kritterier er det vurderet ud fra?*

***En bedre test:*** *“I følgende skema ses systemet testet, i et tidsinterval på 10 min. og et temperaturinterval på 20-40o. Testen viser at. osv..”.*

# Konklusion

Konkluder på forløb og produkt. Husk at der aldrig må fremkomme noget nyt i en konklusion, det er kun en sammenfatning. Man kan sige, at hvis man læser indledningen og går ned og læser konklusionen, har man speed-læst rapporten.

# Vurdering

Vurdering af arbejdsproces. Forslag til forbedringer, hvad kunne I have gjort bedre/anderledes / kan gøre anderledes næste gang.

# Kildeliste

Bog, forfatter, optryk, årstal

Internet kilde, dato, enten direkte link, eller specifikation af præcis hvor informationen står

# Bilag

## -Tidsplan

Brug f.eks. excel. Indsæt alle afleveringer + ting I ved/eller tror I skal have med. Der skal være datoer og personer (hvem af jer der har ansvaret for at arbejdet bliver gjort til tiden). Husk at en tidsplan er et dynamisk projektstyringsværktøj, som man løbende opdaterer efterhånden som projektet skrider frem og man bliver klogere.

## -Logbog

## Gerne i punktform.

## Indhold: Dato, hvem er til stede i dag? Hvad er planen, hvor langt nåede I? Logbogen er desuden et smart værktøj til, når I støder på et problem IGEN, som I kæmpede 2 timer med at løse, forrige uge og ikke kan huske hvordan I fiksede. Eller for at sikre, at man ikke kommer er til at annullere vigtige designbeslutninger, bare fordi man ikke ved/eller har glemt hvilket grundlag de er taget på.

## -Diverse illustrationer og andet, som er for store til at være i selv rapporten

---

**Generelle råd til rapportskrivning:**

**Brug Tabeller, tydelige afsnit/underafsnit/underoverskrifter**. Detkan give indtryk af systematisk arbejdsgang i en teknisk rapport (uanset om en kaotisk arbejdsmetode er anvendt)

**- Vitaliser formuleringerne, indlejr. Informationer, kig kritisk på hvad I skriver og spørg jer selv hvad afsnittet bidrager med. Bliver læseren bliver klogere, eller ville det være det samme, hvis man ikke havde læst siden?**

Overvej sproget. Den måde I formulerer jer på påvirker ubevidst helhedsindtrykket hos læseren. Få den i gruppen der er bedst til at formulere og til grammatik, til at læse igennem inden I afleverer. Hvis man vil, kan man kan give sproget et mere formelt / videnskabeligt udtryk, ved at undgå ”jeg” og ”vi”. Det er ikke noget krav og det er smag og behag om I vil og kan lide det! Men det kommer tættere på det akademiske sprog der anvendes i videnskabelige artikler og mange universiteter. Her er nogle eksempler:

***Org****: “Efter vi havde besluttet hvad vi ville arbejde videre med, aftalte vi at interwieve nogle elever fra 1g ” ->*

**Eks1**: *“Efter beslutninger omkring arbejdsområdet var taget, foretog to personer fra gruppen et interview med tre udvalgte elever fra 1g”.*

**Eks2***: “Efter beslutningsprocessen vedrørende det videre forløb var klarlagt, blev et kvalitativt interview, med tre 1g elever igangsat”.*

**- Modeller uden tekst er No-Go!** (Beskriv ALTID “hvad vi kigger på”. At skrive under en model, at den er selvforklarende hjælper ikke!

(det kan I værste fald opfattes arrogance eller ugidelighed. At skrive ingenting, kan i bedste fald ligne en forglemmelse).