

Softwareprojekt Dokumentation

Workshop - hvad, hvordan og hvorfor

	Dokument	Beskrivelse
1	Opgave_1_Dokumentation	Åbent spørgsmål om dokumentation. Første del baseret på en mobil-app implementeret med App Inventor. Anden del er Scratch eksempel; beskrivelsen af kode.
2	Opgave_2_Elevarbejde	Elevarbejde; robotstyring dokumenteret med flowdiagrammer og kildekode.
3	Opgave_3_Elevarbejde	Elevarbejde; program til fysiksimulering, dokumenteret med tekst, screenshots og et flowdiagram.
4	1dpong_mda	Udarbejdet MDA analyse for spillet 1D Pong. Beskriver hardware ramme for kontekst, software ramme og abstraktion, opstilling, spilleregler, interaktion og virkemåde på adskillige plan.
5	1dpong_codedoc	Udarbejdet kildekode dokumentation for spillet 1D Pong. Tager udgangspunkt i underliggende algoritmer og funktionalitet for input, styring og timing. Beskriver under 10% af kildekoden.
6	eks-Minieksamen2017-MotivationsApp	(delaflevering) Mobil-app (gamification) til at motivere mere motion. Ser på udviklingens rammer i en teknologianalyse, grafisk design (farvernes virkning), flow- og UI-skitser m.m., rører ved produktionsramme.
7	eks-Eksamen2017-Dorania	Mobil-app (spil) til <i>at motivere børn til at skabe fysiske relationer med hinanden</i> ". Bruger DPU metoden til ide og form. Beskriver koncept, spildesign, grafisk design, MDA og kildekode, m.m..

Workshop plan:

Tid	Hvad	Indhold	Dok.
5 min	Introduktion	Dokumentation skal ses som <i>helhed</i> . Software en <i>del</i> af et teknologiprojekt. Hvordan fordeler man fokus i projektet, således at både bredde og dybde vises.	
12 min	Gruppearbejde	Hvilke krav skal man stille til dokumentationen ud fra (1)? Hvilken kontekst bør beskrivelserne indgå eller er kommenteret kode nok? Er elev-eksemplerne (2, 3) beskrivende nok? Hvad er det gode og hvad er problemerne med eksemplerne?	1, 2, 3
6 min	Opsamling		
12 min	Gruppearbejde	Hvordan beskriver MDA-analysen (4) de teknologiske aspekter ift. konceptuelt design, løsningsmodellering og interaktionsdesign? Hvad er fordelene og ulemperne ved, at udarbejde detaljerede kildekodebeskrivelser (5)?	4, 5
6 min	Opsamling		
12 min	Gruppearbejde	Hvad indtryk giver (6, 7) ift. beskrivelsen af software-produktet som helhed? Hvilke gode ting og mangler identificeres? Er beskrivelserne både bredt nok og dybt nok?	6, 7
6 min	Opsamling		
15 min	Diskussion	Hvor ligger fokus og balancen, når man laver et softwareprojekt ift. teknologi som fag. Hvilke elementer skal være med og hvilke elementer er optioner. Inddragelse af andre fag; hvor meget fokus ligger i det tekniske versus det funktionelle.	

Gruppearbejdet tager udgangspunkt i dokumenterne med spørgsmålene som guide. Man bør ikke blot tage programmet som fokus, men *software* som helhed. Efterfølgende sider giver et overblik over udtryk i bredden af softwareudviklingen og teknologi.

Software teknologiprojekt og rapport fokuspunkter

Generelt:

- Det er ikke afgørende, *hvad* resultatet er, men, *hvordan man er kommet til* resultatet
- Eleverne bør løbende tage stilling til, hvordan programmet, i helheden, løser den samfundsmæssige problemstilling
- Eleverne skal, på overordnet plan, tage stilling til og dokumentere, hvad værktøjer de bruger
- Eleverne bør løbende tage stilling til (den agile¹) udviklingsproces(sen), samt dokumentere og evaluere processens implementering og forløb

Alm. teknologi proces:

- Problemstilling
- Problemtræ
- Problemformulering
- Hv-/arbejds-spørgsmål
- Problemanalyse
- Kravspecifikation
- Idegenerering
- Idesortering

Rammen for beskrivelse af softwarens løsning og virkemåde:

- Konceptuelt design
 - skitsering af funktionelle rammer; nødvendige elementer og funktionalitet
 - hvad teori(er) ligger bag løsningen (eller/og "beregningerne")
 - beskrivelse af "data" man arbejder med/på
 - hvordan oversættes og anvendes teori til en applikation
 - program flow og tilstande (hvad sker der efter hinanden)
 - grafisk indtryk og intention (skitser og moodboard)
 - (spil) at beskrive spillets udgangspunkt, mål og funktionalitet
 - (spil) MDA-analyse
- Løsningsmodellering
 - indragelse af de nødvendige teorier, der gælder for programmets anvendelsesområde
 - definition af input/output og data-transformationer til applik. anvendelse og formål
 - beskrivelse af kommunikation (filer, netværk, etc.)
 - dataflow og hierarki
 - algoritmiske løsninger (virkemåde) og programflow
 - skitsering af opstilling og layout med begrundelse
 - (spil) level design
 - storyboard(s), sekvensering og tidslinjer (spil: indgår delvist i MDA)
- Interaktionsdesign
 - conceptual model²
 - affordances (brugerens association *hvad kan jeg med det*)
 - signifiers (indikatorer for brugeren *"her gør jeg XYZ"*)
 - (spil) interaktionsdesign indgår normalt i MDA sammenhæng

1 <https://www.agilealliance.org/>

2 A conceptual model is the underlying belief structure held by a person about how something works (Living with Complexity; Donald A. Norman, 2010).

Marked, erhverv og miljø (teknologi A):

- Eleverne skal tage stilling til markedsstrategi og markedsføring, således at software bliver til et komplet produkt i markedets forstand
 - pris på produktet
 - licensering af produktet og dets komponenter (proprietær, Open Source Software³, Fri Software⁴ og Creative Commons⁵)
 - distribution af produktet (boxed, download, SaaS, PaaS, IaaS⁶, etc)
 - service struktur
- Kunne abstrahere og mappe brug af software til omkostninger i infrastrukturen, fra bruger til back-end, og dermed kunne estimere miljøets påvirkning
- Reflektere over rækkevidden af brug af software og hvad konsekvenser det medfører
- Tage stilling til distribution og infrastruktur, i forhold til den valgte markedsstrategi
- Tage stilling til, hvordan produktet videreudvikles og opdateres, også mens det er i drift

3 <https://opensource.org/>

4 <https://www.fsf.org/>

5 <https://creativecommons.org/>

6 SaaS: Software as a Service; PaaS: Platform as a Service; IaaS: Infrastructure as a Service