

Rapporten

Kurset evalueres primært på baggrund af den rapport hver enkelt gruppe afleverer ved kursets afslutning. Rapporten skal dokumentere det projektarbejde gruppen har udført.

De generelle retningslinier for projektarbejdet og for den afsluttende rapport kan findes i [Sha00] Robin Sharp and Jens Thyge Kristensen, *Software Development Projects*, Department of Information Technology, DTU, 2000 - se **Notat om SW-projekter**.

Denne notes kapitel 5 omhandler dokumentation af softwareprojekter og præsenterer i afsnit 5.2 en generelt anvendelig rapportstruktur.

I dette kursus forgængerkursus, 02120 Informatikfagpakkeprojekt, har fem årgange af studerende taget udgangspunkt i denne note når de skulle dokumentere deres løsninger på de mere end 25 forskellige projektopgaver der blev stillet i den periode. Grupperne har så tilpasset deres rapporter til opgaverne hvor det var nødvendigt.

Nedenfor vil vi først videregive tidligere erfaringer og dernæst kommentere rapportstrukturen fra afsnit 5.2 i den ovenfor refererede note.

Erfaringer

Læsning af tidligere rapporter har givet anledning til nedenstående råd/retningslinier, som man bør være opmærksom under udarbejdelsen af den endelige rapport (og delafleveringer i al almindelighed):

Rapportens læsere

Det er vigtigt, at I gør jer klart, hvem der skal have et udbytte af jeres rapport, og at I skriver rapporten til disse læsere.

Udgangspunkt: læseren er en medstuderende med samme generelle baggrund som jer selv, dvs. Informatikfagpakkestuderende på 4. semester, som dog ikke kender den konkrete opgave, som I løser.

Hele rapporten skal kunne læses af en sådan medstuderende. Til enkelte afsnit kan der være yderligere krav. F.eks. skal en kravspecifikation og en brugervejledning også kunne læses af opgavestilleren/kunden.

Tænk på følgende situation: En medstuderende skal lave en ændring og en udvidelse til jeres system. Jeres rapport skal være skrevet på en sådan måde, at denne studerende hurtigt kan få information og overblik, blandt andet om:

- Hvad er det for et problem I løser?
- Hovedkomponenterne i softwareløsningen
- Væsentlige datastrukturer og algoritmer for implementeringen
- Hvilke softwarekomponenter/datastrukturer/algoritmer skal revideres?
- Hvor er den tilsvarende kode?

Formkrav

- Der skal være en forside indeholdende: navne på forfatterne (gerne studienumre), gruppenummer, dato, titel.
- Rapporten skal være forsynet med jeres underskrifter.
- Rapporten skal bestå af en række nummererede afsnit (evt. kapitler) og der skal være en indholdsfortegnelse. Siderne i rapporten skal nummereres.
- **Af hensyn til den individuelle bedømmelse skal der for hvert kapitel/afsnit angives, HVEM der har produceret indholdet. Der må kun stå ét navn.**
Har I delt arbejdet mellem jer (hvad der jo er naturligt i gruppearbejde), må I altså forsøge at fordele "godbidderne" ligeligt mellem jer (f.eks. ved at indsætte ekstra afsnit, så alle kan få deres navn med). Klager over dette krav rettes til Undervisningsministeriet!
- Rapporten skal indeholde et afsnit med referencer til relevant materiale (bøger, artikler, rapporter, redskaber, som jeres arbejde på en eller anden måde bygger på).
- Rapportens sider skal være fysisk sammenhængende, f.eks. hæftede. Lad være med at aflevere et løsbladesystem!
- Tilstræb en præcis begrebsmæssig sprogform. Undgå slang som "putte ind", "spytte ud", "poppe frem", "klikke sig videre", osv.

Kommentarer til rapportstrukturen

Vi vil tage udgangspunkt i den refererede notes afsnit 5.2, foreslå kapiteloverskrifter og give supplerende kommentarer til indholdet. Husk at læse/slå op i den refererede notes kapitel 2, hvor emnerne kravspecifikation (med analyse), modellering og design, implementering og test perspektiveres og beskrives.

Grupperne er velkomne til at inddrage elementer lært i kursus 02161 Software Engineering 1 i deres rapport.

1. Indledning

Ingen kommentarer

2. Opgavebeskrivelse og -analyse

Ud over det i noten nævnte er det nærliggende at analysere opgaven i et omfang så der er lagt op til den efterfølgende kravspecifikation.

3. Kravspecifikation

Kravene bør prioriteres, så man kan se, hvad der er vigtigst at få implementeret hvis man kommer i tidnød senere, og ikke kan nå at lave alt. Kravene skal nummereres, se de let kan refereres i forbindelse med test. Det bør også klart fremgå "hvilket krav der kommer af hvad", f.eks ved at liste brugsscenarier og under hvert scenarie angive de afledte krav. Kravene bør være testbare ellers kan man jo ikke godtgøre at det færdige produkt opfylder kravspecifikationen. Hvordan fejlhåndteringen skal være kommer også naturligt ind her.

4. Overordnet design

Moduler og deres samspil beskrives (husk nødvendig argumentation for de indførte moduler). Anvend efter behov klassediagrammer, sekvensdiagrammer eller beskrivelse af hændelsesforløb. Det kan være svært at afgøre hvad der skal stå i dette afsnit og hvad der gemmes til næste afsnit. Hvis vi tænker på Java-notationen kan man angive interfaces og abstrakte klasser.

5. Detaildesign og implementering

I den refererede note er det valgt at beskrive samspillet mellem modulerne i dette afsnit. Her anføres for modulerne metodesignaturer (og/eller prototyper). Men det er vigtigt at holde sig for øje at detaljeret programtekst placeres i appendix.

6. Test

Overordnet beskrivelse af testfilosofien, gennemførelsen af tests og deres resultater. Detaljer skal i appendix.

7. Konklusion**8. Litteraturreferencer**

Ud over de klassiske referencer til trykte formater kan der være behov for referencer, hvor man angiver forfatter(e), titel og `http-reference(r)`.

9. Ordliste (placeres enten her eller eventuelt allerforrest)

Forklaring på de benyttede fagtermer, som ikke skønnes umiddelbart forståelige for læserne.

10. Appendices

Supplerende materiale, som minimum reviewreferat(er) og timeregnskaber.

Selv om det kræver lidt papir bør al programlistning som hovedregel findes udskrevet på papir. Helt trivial Java-kode kan udelades, men husk afsnitsoverskrifterne. Alle filer bør også foreligge på en vedlagt CD-rom eller efter forudgående aftale være uploadet til CampusNet.

Første udgave: 4. maj 2006