

02161 Software Engineering 1

Obligatorisk Opgave

Version 3, February 24, 2013

1 Tasks

I afsnit 2 findes projektbeskrivelsen, der vedrører udvikling af et planlægningsværktøj, til brug i forbindelse med styring og planlægning af softwareudviklingsprojekter.

Kravene til besvarelser af denne obligatoriske opgave findes i følgende.

1.1 Formalia

- Opgaven skal løses i grupper på enten 2, 3 eller 4 studerende.
- Rapporten skal skrives på dansk eller engelsk.
- Rapporten og kildekoden for denne obligatoriske opgave skal afleveres senest **kl 17:00, Mandag den 13 maj** ved brug af opgave modulen på CampusNet.
- Desuden skal **rapporten og program uploades ved brug af opgave modulen på campusnet.**
- Program skal afleveres som Eclipse eller NetBeans projekt som indeholder kildekoden, testene, det kørende program og en instruktion hvordan man kører programmet og testene.
- Rapporten skal have en forside med titel, gruppe nummer, dato og studienummer og navn for alle gruppens medlemmer.
- Hvert afsnit eller use case, diagram o.s.v. skal have navnet på fra personen, som har lavet afsnittet, use case, diagrammet o.s.v. Der kan være kun et navn på hvert afsnit, use case, diagram, o.s.v.
- Hver klasse og method skal have navnet på fra personen, som har skrevet klasse/metode. Der kan være kun et navn på hvert klasse/metode.
- *Diskussion af den obligatoriske opgave er tilladt også med andre gruppens medlemmer og kursusdeltager fra tidligere år, men ingen form for anvendelse af andres program eller rapportdele er tilladt. Hver besvarelse skal være selvstændig. **Afskrift uden kildeangivelse betragtes som snyd.***

- Senest **tirsdag den 2. april** skal hver projektgrupp uploade et planlægningsdokument med en tidsplan for projektet ved brug af CampusNets opgavemodul. Dette skal vedlægges i et bilag i rapporten.
- **Mandag den 13 maj** giver hver gruppe en kort demo af deres program i E-databaren. Demoen skal (som minimum) vise testresultaterne fra afsnit 4 i rapporten.
- Yderligere information vil blive oplyst via kursets hjemmeside.

1.2 Opgaven

Projektbeskrivelsen i afsnit 2 er udgangspunktet for udviklingen af et system til brug i forbindelse med projektplanlægning for softwareudviklingsprojekter, herunder registrering af arbejdstidsforbrug på de forskellige projekter.

På baggrund af dette oplæg skal der udarbejdes

1. kravspecifikation
2. programdesign og
3. program,

saledes at de væsentlige begreber fra oplægget kan spores i det udviklede program.

Det er ikke alle aspekter, der omtales i oplægget, der skal medtages i det system der specificeres og udvikles. En del af denne øvelse består i at afgrænse opgaven, saledes at det er muligt at udvikle et sammenhængende produkt (rapport og tilhørende program) af høj kvalitet indenfor den afsatte tid. Men det er også vigtigt at der tages stilling til, hvordan systemet kan udvides med de aspekter, man så bort fra i forbindelse med afgrænsningen.

Programmets applikationslag skal udvikles test-dreven. Til hver funktion i applikationslag skal der findes en eller fler svarende JUnit tests som sikrer at funktionen gøre det den skal (liegsom vi har gjort i øvelsener.) Brugergrænssfladen behøves ikke udvikles test-dreven. Ved projekt demonstration skal i vise testene.

Ved bedømmelse af rapporten vil der blive lagt vægt på

- fuldstændighed, dvs. at de valgte funktioner bliver komplet behandlet, og
- sporbarhed, herunder konsistens igennem alle abstraktionsniveauer.

Det er bedre at sikre disse kvalitetsegenskaber igennem en bevidst afgrænsning af problemet, end at brede sig for meget ud over problemstillingen.

1.3 Afgrænsning

Kunden prioriterer projekt planlægning før registrering af arbejdstidsforbrug. Til sidst kommer rapporter til brug ved projektmøder.

Angående programmet, sa er det ikke et krav, at data kan gemmes på en fil eller en database. Det er heller ikke et krav, at systemet kan benyttes via en grafisk brugergrænseflade.

1.4 Rapporten

Rapporten skal indeholde følgende afsnit og bilag.

1. *Indledning*: Dette afsnit skal indeholde en kort indledning og opsummering af besvarelsen.
2. *Kravspecifikation*: Afsnittet har følgende underafsnit:
 - (a) *Væsentlige begreber*: Dette afsnit skal indeholde en opremsning af de væsentlige begreber fra oplægget. Hvert begreb gives en kort definition på nogle få linier.
 - (b) *Use cases*: Dette afsnit skal give en beskrivelse af de væsentlige operationer i form af et (eller flere) use case diagrammer og (deetaliert) use cases med hoved- og alternative/fejl scenarier. Der skal være mindst 4 use cases for tomandsgrupper, 6 for tremandsgrupper og 8 for firmandsgrupper. Benyt samme skabelon som brugt i forbindelse med forelæsninger og øvelser. Udvalget af use caserne skal ske efter vigtigheden til brugeren. Desuden skal use caserne (hoved- og alternative/fejl scenarier) implementeres test-dreven.

Afsnittet afsluttes med en kort diskussion, hvor, for eksempel, uklarheder i oplægget diskuteres, og hvor der gøres rede for valg og afgrænsninger.

3. *Programdesign*: Afsnittet har følgende underafsnit:
 - (a) *Klassediagram*: Dette klassediagram skal være bindeledet mellem kravspecifikation og program. Klasserne skal indeholde væsentlige attributter og operationer med typer, således at det fremgår hvordan de væsentlige begreber modelleres og samtidigt skal diagrammet give et overblik over kildekoden.
 - (b) *Sekvensdiagrammer*: Dette afsnit indeholder sekvensdiagrammer for hovedscenarierne af nogle af de use cases som blev valgt i afsnit 2.b, således at læseren får et overblik over hvordan programmet vil udføre disse scenarier. Der skal være 2 sekvensdiagrammer for tomandsgrupper, 3 for tremandsgrupper og 4 for firmandsgrupper.

Afsnittet afsluttes med en kort diskussion hvor der, for eksempel, gøres rede for de valg der er truffet. Afsnittet kan også omtale valg af algoritmer og datastrukturer.

4. *Systematiske test*: Dette afsnit skal indeholde planer for systematiske tests til alle use case scenarier nævnt i afsnit *use cases* ved brug af systematiske test metoden for black box test (cf. slides uge 4). I det mindste dog alle scenarier til 4 use cases for tomandsgrupper, 6 for tremandsgrupper og 8 for firmandsgrupper. Benyt samme skabelon som brugt i forbindelse med forelæsninger. De planlagte systematiske test skal implementeres og udføres ved brug af JUnit og skal afleveres som en del af Eclipse projektet. Desuden skal testene demonstreres i programdemoen på Mandag den 13 maj. Desuden skal rapporten have et diagram som viser code coverage af de enkelte Java filer efter alle testene er kørt, sammen med en forklaring af code coverages resultater.

5. *Konklusion*: Der skal gives en kort status og vurdering af det afleverede produkt. Desuden skal der være et kort afsnit, der reflekterer omkring projektets forløb, og sammenligner det med det oprindelige planlægningsdokument.
6. *Bilag 1*. Oprindeligt planlægningsdokument

I skal uploade Eclipse projekt (kørende program, kilde kode, alle test) som ZIP fil (RAR fil er ikke tilladt) og rapporten som PDF ved brug af opgavemodulen. I skal bruge navne `project_xx.zip` og `rapport_xx.pdf` til Eclipse projects fil og rapporten, hvor xx er jeres projektgruppenummer skrevet med to cifre (fx 01, 02, ..., 10, ...). Eclipse projektet skal have et readmefil (`readme.txt`), som indeholder, hvordan man kører programmet. Programmet skal køre ud fra de filer der er i Eclipse projektet og på en komputer som er ikke tilsuttet til internet. Det er fx. ikke tilad at bruge en ekstern database.

2 Timeregistrering og projektstyring

Softwarehuset A/S bruger i dag et ældre system til timeregistrering og projektstyring. Systemet er på flere måder tungt at arbejde med. Specielt klager mange medarbejdere over besværet med at registrere hvor mange timer man har arbejdet på forskellige aktiviteter. Projektlederne klager over at det er svært at bemande et projekt fordi man ikke kan se hvem der har tid hvornår.

Softwarehuset A/S har ca. 50 udviklingsmedarbejdere og 30 igangværende projekter. Mange projekter varer blot et par måneder, men nogle varer flere år. Hvert projekt har en projektleder (valgt blandt udviklingsmedarbejderne). En af projektlederens opgaver er at opdele projektet i aktiviteter, f.eks. kravspecifikation, projektledelse, analyse, design, programmering, test med mere.

Et projekt har typisk 30 aktiviteter, men nogle har næsten 100. Et projekt oprettes, når der skal arbejdes på en opgave. Det kan være en mulig kunde, som ønsker en udvikling, eller det kan være et "internt" projekt, der går ud på at lave noget for Softwarehuset A/S selv. På oprettelsestidspunktet er information om aktiviteter og starttidspunkt ikke fuldstændige, men dog kendt i et vist omfang. Projektlederen er ikke nødvendigvis udpeget fra starten.

Projektlederen skal gradvis opdele projektet i aktiviteter. Aktiviteterne skal kunne planlægges længe inden arbejdet på dem begynder, men planerne skal kunne ændres hyppigt. For hver aktivitet skal projektlederen kunne anføre forventet antal arbejdstimer til løsning af opgaven (budgetteret tid). Endvidere skal der kunne anføres start- og sluttid på aktiviteten. Der ønskes en opløsning på ugenummer niveau. Projektlederne skal kunne bruge systemet til at bemande projekter i god tid. Der kan være behov for, at mere end en medarbejder deltager i at udføre en bestemt aktivitet. Systemet skal tillade, at en medarbejder kan søge assistance hos en kollega og at denne så kan registrere den forbrugte tid på aktiviteten, selv han ikke direkte er tilordnet den af projektlederen. Softwarehuset A/S ønsker et system der kan hjælpe projektlederne i forbindelse med bemanning af projekter. Det skal således give overblik over, hvilke medarbejdere, der burde være ledige på det tidspunkt, hvor aktiviteten skal udføres.

En medarbejder arbejder typisk på mindre end 10 aktiviteter ad gangen og varigheden af den enkelte aktivitet er typisk 2-3 uger. Systemet skal tillade enkelte medarbejdere at arbejde på 20 aktiviteter i løbet af en uge. Der skal være faste aktiviteter for registrering af ting som ferie, sygdom, kurser med videre, som ikke kan pålignes det enkelte projekt.

Medarbejderne skal dagligt registrere hvor meget tid de har brugt på de forskellige aktiviteter (1/2 times nøjagtighed er tilfredsstillende). Det skal være enkelt at foretage denne registrering, så medarbejderen ikke føler det er en byrde. Medarbejderen skal enkelt kunne se, om han/hun har registreret alle timer man har arbejdet i dag. Det skal være muligt at rette i allerede registrerede data ligesom det skal være muligt at registrere fremtidige aktiviteter, f.eks. ferie.

Projektlederen skal kunne bruge systemet til opfølgning. Det betyder, at projektlederen let skal kunne se, hvordan timeforbruget udvikler sig per aktivitet og for hele projektet. Der skal kunne laves rapporter til brug ved Softwarehuset A/s' ugentlige projektmøder. Ved disse møder skal projektlederen endvidere se det forventede restarbejde på projektet.

2.1 Lidt data

Medarbejdere identificeres med initialer på indtil 4 bogstaver. Projektnumre skal tildeles af systemet og have formen årstal og et løbenummer, f.eks. 030901. Desuden skal projektet kunne have et navn.

Systemet er til internt brug, så det er ikke nødvendigt med adgangskontroller.

Med tak til Michael R. Hansen og Søren Lauesen for lån af ideen til system og for store dele af beskrivelsen.