NTNU
Det skapende universitet

Institutt for datateknikk og informasjonsvitenskap (IDI)

Eksamensoppgave i TDT4140 - Programvareutvikling

Faglig kontakt under eksamen:
Professor Pekka Abrahamsson tlf.: 405 65 642
Faglærer Carl-Fredrik Sørensen kan kontaktes på tlf.: 951 19 690

Eksamensdato: 14.08.2015
Eksamenstid (fra-til): 09:00 – 13.00
Hjelpemiddelkode/Tillatte hjelpemidler: Kode C: Pensumbok: Ian Sommerville, Software Engineering 9. Bestemt, enkel kalkulator tillatt.

Annen informasjon: Eksamen utviklet av Faglærer Carl-Fredrik Sørensen og kontrollert av John Krogstie

Målform/språk: bokmål
Antall sider: 11
Antall sider vedlegg: 1

Kontrollert av:

_____

Dato Sign

Dokumenter og begrunn eventuelle forutsetninger eller antakelser.

Oppgave 1: Flervalgsoppgave (25%)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på den ene selv). Du kan få nytt ark av eksamensvaktene dersom du trenger dette. Kun ett svar er helt riktig. For hvert spørsmål gir korrekt avkryssing 1,25 poeng. Feil avkryssing eller mer enn ett kryss gir –0,625 poeng. Blankt svar gir 0 poeng. Du får ikke mindre enn 0 poeng totalt på denne oppgaven. Der det er spesielle uttrykk står den engelske oversettelsen i parentes.

1) **Hva er den fundamentale årsaken til at programvare ikke kan bli betraktet å være et resultat av vanlig ingeniørvitenskap?**
   a) Programvare er designet av mennesker og derfor mangelfull og full av feil
   b) Programvareutvikling (software engineering) er i motsetning til andre ingeniørdisipliner, en kunst – og ikke vitenskap
   c) Disiplinen er relativt ny sammenlignet med f.eks. brobygging, som er en aktivitet som har blitt praktisert i tusenvis av år
   d) Kompleksiteten av systemer og deres interaksjoner øker raskere enn mennesker er i stand til å forstå.

2) **DIFI har definert sju arkitekturprinsipp. Hvilket er IKKE ett av disse?**
   a) Interoperabilitet
   b) Testbarhet
   c) Åpenhet
   d) Sikkerhet

3) **En enkel måte å se spiralmodellen på, er en fossefallsmodell der hver fase etterfølges av**
   a) Risikoanalyse
   b) Kodefrys
   c) Testing
   d) Bygging og retting

4) **Relasjonen mellom en subklasse og en baseklasse kalles for**
   a) Arv
   b) Assosiasjon
   c) Aggregering
   d) Instansiering

5) **Et design blir sagt å være et godt design dersom komponentene er**
   a) Strengt koblet og har lav kohesjon
   b) Løst koblet og har høy kohesjon
   c) Løst koblet og har lav kohesjon
   d) Strengt koblet og har høy kohesjon

6) **Problemer med å benytte antall kodelinjer (LOC) til å beregne størrelsen til et produkt inkluderer:**
   a) Skriving av kildekode er bare en del av utviklingsarbeidet
   b) LOC vil være avvikende mellom språk, og er for noen språk ikke målbar
   c) Den endelige størrelsen (kLOC) kan bare bli bestemt når produktet er levert
   d) Alle er riktig

7) Når man skal velge en utviklingsprosess, så vil man vurdere
   a) Programmeringsspråk, tidsplan, konkurransesituasjon
   b) Ekspertisen til utviklingsgruppen, problemkarakteristikker, brukerforventninger
   c) Systemkontekst, brukerpopulasjon, plattformer
   d) Organisasjonsstruktur, brukeroppgaver, ytelseskriterier

8) Konfigureringsaktiviteter i forhold til programvare, inkluderer å
   a) Identifisere endringer
   b) Kontrollere endringer
   c) Rapportere endringer til interessenter
   d) Alle er riktig

9) Under planlegging av et programvareprosjekt, vil man
   a) Finne måter å produsere resultater ved hjelp av begrensede ressurser
   b) Tilpasse tidsplanen for å ta hensyn til feil
   c) Overestimere budsjettet (f.eks. gange med PI)
   d) Strukturere teamet for å forhindre administrativ innblanding

10) I kravanalysen utforsker man
   a) Systemytelse, testplanlegging, organisasjonsstruktur
   b) Språk, plattformer, konkurransesituasjon
   c) Systemkontekst, brukerpopulasjon, brukeroppgaver
   d) Verifisering og validering, formelle metoder, nøyaktighet

11) Hvordan må en prosjektleder agere for å minimalisere risikoen for alvorlige programvarefeil/feilslått prosjekt?
   a) Be om et større budsjett
   b) Benytte små utviklingsteam
   c) Måle fremdrift
   d) Utvide prosjektrammen i forhold til personell

12) Black box testing blir også kalt
   a) Spesifikasjonsbasert testing
   b) Verifikasjon
   c) Strukturell testing
   d) Enhetstesting

13) Hvilken av de følgende er en vanlig metode i framlokking av krav?
   a) Transaksjonsanalyse
   b) Risikovurdering
   c) Implementering av systemet
   d) Observasjon

14) Bruk av en strukturell modell for å representere et arkitekturdesign har hvilken av de påfølgende karakteristikker?
   a) En strukturell modell tillater å identifisere repeterbare arkitekturdesign-rammeverk som man møter på i lignende applikasjoner
   b) En strukturell modell adresserer oppførselsmessige aspekter som indikerer hvordan et system endrer seg som en funksjon av eksterne hendelser

c) En strukturell modell representerer arkitekturen som en organisert samling av programvaremoduler og komponenter
d) En strukturell modell fokuserer på forretnings- eller tekniske prosesser som et system må tilpasse seg til.

15) Hvorfor er feil i krav generelt blant de dyreste feilene å rette?
a) Fordi retting av feil i krav kan medføre en omfattende redesign av systemet
b) Fordi retting av feil i krav medfører tilpasninger av systemer for å håndtere endringer i kjøremiljøet
c) Fordi behovet for å rette feil i krav vanligvis forekommer på grunn av organisatoriske eller forretningsmessige endringer
d) Fordi retting av feil i krav involverer å beholde funksjonalitet i system uendret mens man forbedrer den interne strukturen og generelle ytelsen

16) Et "configuration control board (CCB)" behandler hvilken av de følgende?
a) Valg av verktøy
b) Anmodninger om endringer i programvare
c) Revisjon av programvarekonfigurasjon
d) Release-dokumentasjon av programvare

17) Identifisere og håndtere risikoer er en viktig del av effektiv styring av arbeidet med programvareutvikling. Risikoer er dokumentert:
a) I en konsis forklaring om hva som gikk galt og når det skjedde i livssyklusen til prosjektet.
b) Som klart definerte aktiviteter i prosjektplanen
c) I en konsis forklaring som inkluderer kontekst, betingelser og konsekvenser av at en risiko opptrer
d) Som klart definerte kostnader i prosjektbudsjettet

18) Hva er hovedforskjellen mellom «Unified Process» (UP) og andre prosessmodeller?
a) UP er mer faseorientert enn de andre modellene
b) UP har flere faser enn de andre modellene
c) UP har som den eneste utviklingsprosessen, deployment som en egen arbeidsflyt
d) UP er testdrevet i motsetning til de andre prosessmodellene

19) Prosessmodeller er beskrevet som smidige fordi de
a) Eliminerer behovet for tungvint dokumentasjon
b) Ikke kaster bort verdifull utviklingstid på planleggingsaktiviteter
c) I stor grad benytter prototyper
d) Legger vekt på manøvrerbarhet og tilpasningsevne

20) I kontekst av objekt-orientert programvareutvikling, inneholder en komponent:
a) Attributter og operasjoner
b) Instanser av hver klasse
c) Roller for hver aktør (utstyr eller bruker)
d) Et sett av samarbeidende klasser

INNLEVERING

Svarskjema: Oppgave 1

Oppgavenr	A	B	C	D
1.1				X
1.2		X		
1.3	X			
1.4	X			
1.5		X		
1.6				X
1.7		X		
1.8				X
1.9	X			
1.10			X	
1.11			X	
1.12	X			
1.13				X
1.14			X	
1.15	X			
1.16		X		
1.17			X	
1.18			X	
1.19				X
1.20				X

Oppgave 2 – Programvareutvikling (55%)

Det skal utvikles et digitalt system for mobil arbeidsstøtte for industriarbeidere som er tenkt å øke samarbeid og produktivitet, øke tryggheten på arbeidsplassen, og optimalisere bruk av arbeidskraft og materialer i en stor industribedrift (f.eks. et skipsverft eller en oljeplattform). Alle håndverkere er utstyrt med mobile enheter som inneholder sensorer og programvare som kan beregne posisjon på personene i industrilokalet. Den mobile enheten inneholder også andre sensorer og interaksjonsgrensesnitt som talestyring, kamera/video og alarmer. Hver arbeider har en definert arbeidsprosess og arbeidsmål gitt gjennom et sentralt støttesystem som planlegger arbeidsoppgaver og overvåker fremdrift. Den mobile løsningen skal kunne kommunisere med det sentrale systemet for å motta oppgaver, rapportere fremdrift og materialbruk, samt alarmere ledelse og arbeidere dersom det oppstår noen hendelser som medfører fare eller forsinkelser. Det sentrale systemet består av et logistikksystem for materialer, planleggings- og oppfølgingssystem av arbeidsoppgaver inkludert alle avhengigheter mellom oppgaver og posisjon og materialbehov ved arbeidssted. Hver arbeidsoppgave har spesifisert innhold (hva, hvem, hvor, når), instrukser (både tekstlig og multimediainnhold) og antatt ressursbruk (både personell og materialer). Arbeidsoppgaver har flere tilstander: <planlagt>, <startet>, <stoppet>, <avsluttet>. Noen arbeidsoppgaver krever samarbeid mellom flere personer. Anta at alle sensorer har veldefinerte grensesnitt. Den mobile klienten kan kommunisere med sensorer i omgivelsene. Sensorene kan gi viktig informasjon til arbeideren og arbeidsflytsystemet knyttet til hvorvidt oppgaven er i konflikt med andre oppgaver i omgivelsene. Systemet kan dermed hjelpe til med å skape en forhandlingssituasjon der aktiviteter kan prioriteres basert på en verdi dersom aktivitetene ikke kan foregå samtidig (f.eks. sveising samtidig med at man behandler brannfarlig material i nærheten).

a) (5%) Spesifiser og analyser interessenter og aktører av løsningen.

Interessenter kan både være innenfor og utenfor bedriften. Dette kan være grupper som ledere, spesifikke grupper hos en bedrift, organisasjoner, myndigheter, kontrollorgan. For å få full score på oppgaven, så bør man både angi interessenter, men også analysere hvilke interesser denne har i forhold til løsningen. Aktører for løsningen er de roller som skal benytte løsningen. Disse bør være mest mulig spesifikk for å få en god kobling mellom aktør og de use case som skal tilbys/tilfredsstilles.

Ledelse: Prioritere og samkjøre oppgaver for effektiv og sikker gjennomføring
Ansatte: Bedre sikkerhet, bedre effektivitet siden arbeidsoppgaver som blir påbegynt er koordinert slikt at de kan avsluttes eller stoppes på mer optimale tidspunkt. Oppgavelisten vil inneholde de til enhver tid viktigste oppgavene som kan utføres.
Fagforening: Ivareta rettigheter og arbeidsmiljø for arbeidere
HMS-ansvarlig: Sikre et trygt arbeidsmiljø
Logistikkansvarlig, lageransvarlig: Har ansvar for at utstyr og deler er tilgjengelig til rett tid og rett sted. Kan få rapport om defekt utstyr fra arbeidere og må da skaffe til veie riktig utstyr. Dersom dette ikke er mulig, kommuniseres dette til planleggingssystemet som prøver å finne andre oppgaver til arbeidere som venter på deler/utstyr.
Samfunn: Mindre arbeidsulykker, dvs. lavere helsekostnader og uføreproblematikk

Myndigheter: Lovgivning

Aktører: Aktivitetsplanlegger, aktivitetsutfører

b) (5%) Beskriv et applikasjonsscenario med en gruppe arbeidere som både alene og i samarbeid utfører arbeidsoppgaver som kan ha nytte av løsningen.

I denne oppgaven forventes det en beskrivelse av systemet i bruk. Scenariet kan være enkelt bestående av to aktører som kommuniserer med hverandre gjennom den mobile løsningen, eller som kommuniserer med det sentrale oppgavesystemet.

c) (5%) Lag et UML Use case-diagram for den mobile arbeidsflytløsningen. Inkluder koblinger mellom komponenter/tjenester i det sentrale systemet.

Aktører: Antar at aktørene kun har bruksoppgaver knyttet til den mobile delen av systemer. Her kan det også spesifiseres aktører som ikke er mennesker, f.eks. Arbeidsflytsystem, logistikk/lagersystem, Sensor og sensorsystem.

Arbeider:

- Motta oppgave
- Rapportere fremdrift
- Rapportere materialbruk
- Bestille og kvittere ut materialer ifht planlagt oppgave
- Varsle om fare
- Bli varslet om fare
- Innlede samarbeid
- Varsle avvik

Leder:

- Spesifisere oppgave
- Motta varsling om tilstander som truer fremdrift
- Re-planlegge

System:

- Overvåke arbeidsmiljøet, sjekke om oppgaver er i konflikt, om potensielle oppgaver kan medføre økt fare på arbeidsplass (f.eks. sveising mens man tester gassproduksjon på oljeplattform).
- Alarmere arbeidere og ledere
- Forhandle prioritet på arbeidsoppgaver basert på plan, inkludert prioriteter og kostnader på oppgave, antall aktører involvert.

d) (5%) Lag et tekstlig use case av det mest komplekse use caset. Grunngi hvorfor du valgte dette use caset. Dersom det baserer seg på extends/use relasjoner med andre use case, lag også tekstlige use case for disse. Bruk malen i vedlegg A.

Use case beskrivelsen må inneholde en stegvis beskrivelse av use caset, der bruker og system interagerer for å oppnå det funksjonelle målet. Use caset må være grunngitt for topp score

e) (5%) Lag et sekvensdiagram for ditt valgte use case basert på use case-beskrivelsen i 2d).

I denne oppgaven bør det komme frem viktige klasser/grensesnitt som viser strømmen av interaksjon for å oppnå det funksjonelle målet. Bør ta utgangspunkt i «description» spesifisert i oppgave 2

f) (5%) Definer og grunngi de viktigste ikke-funksjonelle kvalitetskrav/arkitekturkrav til løsningen.

Kravene bør være i henhold til f.eks. ISO 9126. Kravene må være klart definert:

Availability: Systemet skal ha en høy oppetid: f.eks. 99.999%

Performance: Vanlige funksjoner skal ha en responstid på mindre enn 1 sekund

Usability: Systemet skal kunne benyttes utendørs, systemet skal kunne ha talestyring, kontraster må kunne endres ift f.eks. sollys.

Security: Systemet må hindre tilgang på personinformasjon, og arbeidsplaner og regler må ikke eksponeres for uautoriserte brukere.

Safety: Systemet må gi alarm ved farlige situasjoner på arbeidsplassen til alle aktører dersom dette oppdages av systemet.

g) (5%) Lag en arkitekturskisse for systemet. Hvilke hovedkomponenter og koblinger/grensesnitt må tilbys?

Komponentene bør navngis med ordentlige navn, server og klient blir for generelle. Skissen kan vise hva som eksponeres i klient og hva som skal gjøres på en server. Man kan også spesifisere spesifikke mellomvare-komponenter dersom dette passer til løsningen.

h) (5%) Hvilken eller hvilke arkitekturmønstre (patterns) bør et slikt system benytte? Beskriv hvordan og hvorfor et valgt mønster bør benyttes.

Mønsteret bør reflektere skissen i oppgave 2g). Her bør man beskrive hvorfor de valgte mønstrene passer godt til løsningen.

i) (5%) Definer de viktigste klassene og lag et UML klassediagram for arbeidsstøttesystemet.

Klassediagrammet kan være generisk, dvs. ikke innholde mye attributter eller metoder. Klassene bør være de mest sentrale i systemet. Det forventes at assosiasjoner og relasjoner spesifiserer, gjerne med granulariteter.

Klasser: Person, oppgave, alarm, plan, rapport, avvik, prosjekt, ressurser, lokasjon, tid

j) (10%) Spesifiser en testplan for systemet. Hvilke tester bør inngå for å sikre at systemet har tilstrekkelig kvalitet og tilfredsstiller kravene? Hvilke use case er vanskeligst å teste?

Testplanen bør inneholde en beskrivelse av hvilke testtyper som skal utføres: enhetstester, integrasjonstester, systemtester, akseptansetester, brukbarhetstester. For hver testfase, kan det spesifiseres testmetoder, f.eks. black-box, white-box o.l.

De vanskeligste use casene å teste er hvor man må ha mange samtidige aktører i systemet og der det skjer en koordinering i runtime.

Oppgave 3: Risikostyring og prosjektplanlegging (20%)

a) (5%) Beskriv risikoer og strategier for å håndtere disse i **utviklingen** av systemet beskrevet i oppgave 2. Dokumenter hver strategi med passende eksempler og diskuter potensielle problemer som er assosiert med hver strategi.

I denne oppgaven forventes det at risikoer blir beskrevet ifht utviklingen av systemet, dvs. hvilke risikoer som er knyttet til personell, utstyr, økonomi, forankring, utviklingsmetodikk, brukere, prosjektgjennomføring etc. Noen risikoer er standard: sykdom, frafall, dårlig forankring blant ledelse og ansatte (brukere), kravforståelse, testing. I denne oppgaven kan det også være knyttet risiko til utstyr og infrastruktur siden systemet skal kunne fungere i relativt krevende fysiske arbeidsmiljø. Risikoene må sannsynliggjøres og det bør vises til konsekvens av denne.

Strategiene kan være preventive (unngå at risiko inntreffer), skadebegrensende, overvåkende (oppdage og fjerne risiko)

b) (15%)
   i. Definer en grov prosjektplan for utvikling av systemet beskrevet i oppgave 2. I prosjektplanen må det defineres hovedaktiviteter, milepæler og avhengigheter.

   Her forventes det en prosjektplan som viser hovedaktiviteter og milepæler. Aktivitetene vil ofte være knyttet til utviklingsprosessen, og kan med fordel utnytte informasjon fra kravdelen i oppgave 2, for å spesifisere viktige utviklingsaktiviteter. Utviklingsprosess vil ha stor innvirkning ifht milepælsstyring. Inkrementell eller smidig utvikling vil ha iterasjoner, dvs. milepæler knyttet til funksjonelle eller ikke-funksjonelle mål.

   ii. Definer en WBS (work breakdown structure) for utvikling av systemet

Denne er en forfining av planen fra oppgave b) i. hvor man tar for seg deler av systemet og hvordan disse kan spesifiseres som egne utviklingsmoduler.

iii. Skisser hvordan prosjektet eventuelt kan deles inn i delprosjekter.

Basert på plan og WBS, så kan man vurdere hvordan delene henger sammen, og det kan være hensiktsmessig å vurdere om det kan være arbeidsoppgaver som kan defineres som et eget delprosjekt. Delprosjektene bør ha en viss størrelse for at det skal ha noe hensikt å gjøre dette, spesielt siden det medfører større kostnad knyttet til ledelse og administrasjon. Delprosjekter kan være hensiktsmessig dersom det er identifisert områder som er avgrenset eller som anses å ha høyere risiko enn andre. Delprosjekter er også fornuftig dersom man velger å «out-source» noe av utviklingen til andre aktører.

iv. Argumenter for hvilken utviklingsprosess som vil være mest hensiktsmessig.

I denne oppgaven forventes det å ha innsikt i styrker og svakheter med forskjellige utviklingsprosesser. Ved visse type systemer som har høyere krav til lovoppnåelse og som ikke krever utstrakt brukerkontakt, så kan en vannfallsmodell fungere bra. Ved brukertunge systemer, så kan det være fornuftig å utvikle systemet iterativt og inkrementelt, både for å få tilbakemelding på brukervennlighet og se på potensialet for prosessforbedringer hos kunden. Det vil imidlertid være krevende å få tilstrekkelig kundekontakt/brukerkontakt når brukerne ikke har et forhold til bruk av digitale hjelpemidler.

v. Beregn kostnader relatert til utvikling og test av løsningen, se bort i fra utstyrskostnader, anta en fast timepris på NOK 800.

I denne oppgaven vil man sette opp budsjett basert på de andre deloppgavene. Her er det «lett» å ta for lett på kostnadene knyttet til utvikling. Dersom en velger en smidig metode, så vil utviklingen kreve innsats fra andre enn utviklingsteamet. Testing kan være en krevende oppgave, så det bør settes av tid til testing og feilretting (både korrektiv og hvor man må utvikle nye funksjoner siden man lærer nye ting underveis i utviklingen).

Vedlegg A – Mal for tekstlige use case

Use case name: < >
Goal: < >
Primary Actor: < >
Stakeholders and Interests: < >
Precondition: < >
Success End Condition: < >
Failed End Condition: < >
Trigger: < >
Description: < >
Extensions: < >