

DA336A Systemutveckling och projekt I

Projektguide VT18

Innehåll

| | |
|-------------------------------------|----|
| Process..... | 3 |
| Arbetsfördelning..... | 3 |
| Handledning | 3 |
| Teknisk handledning | 3 |
| Självgranskning | 4 |
| Handledningsmöten..... | 4 |
| Retrospektmöten..... | 5 |
| Presentation | 5 |
| Granskning | 6 |
| Testning | 6 |
| Planering..... | 6 |
| Arbetstid och tidrapportering | 7 |
| Övergripande struktur | 7 |
| Utställningsdag..... | 9 |
| Affischer..... | 9 |
| Verktyg..... | 9 |
| Versionshantering..... | 9 |
| Projektverktyg | 9 |
| Artefakter | 10 |
| Socialt kontrakt | 10 |
| Kod..... | 10 |
| Dokumentation av projektet | 10 |
| Individuell fördjupning | 15 |
| Reflektionsrapport..... | 15 |
| Referenser..... | 16 |

Process

Arbetsfördelning

Alla gruppmedlemmar måste bidra med att i någon utsträckning skriva programkod. Ingen i gruppen kan enbart syssla med grafiskt relaterad design, skriftlig dokumentation som ej är kod (exempelvis skriva kravdokument) eller andra sysslor i projektet som inte innebär att skriva kod (exempelvis testning). Att enbart skriva HTML och CSS räknas inte som tillräckligt för att uppfylla den här punkten. För webbaserade projekt så måste alla deltagare bidra genom att skriva annan kod än enbart deklarativ kod (som HTML och CSS). Detta kan innebära att skriva kod med exempelvis JavaScript eller Python.

Alla gruppmedlemmar måste också bidra till att producera den dokumentation som krävs av projektet. Ingen gruppmedlem kan enbart producera kod, syssla med design eller liknande.

Inför varje vecka bör planeringen innebära att alla gruppmedlemmar sysslar med någon del som innebär att producera kod och någon del som innebär att producera dokumentation.

Detta innebär att alla uppgifter inte endast kan läggas på den som är starkast på detta i gruppen. Vissa kommer att vara snabbare på att göra vissa saker men de kan inte göra alla dessa saker. Så vissa arbetsuppgifter kommer att utföras av personer som tar lite längre tid på sig med dessa än vad någon annan i gruppen skulle gjort. Se detta som en möjlighet att hjälpa varandra att lära och utvecklas. Det är resan genom projektet som är det viktigaste ur utbildningssynpunkt i det här fallet. Slutresultatet spelar roll men det viktigaste är att alla i gruppen får en chans att utvecklas och öva sina färdigheter. Detta kan inte ske för alla i gruppen om någon blir utestängd från arbetsuppgifter för att hen utför dessa långsammare än övriga eller behöver mer hjälp.

Handledning

Varje grupp har en grupphandledare som träffar gruppen regelbundet. Grupphandledarens uppgift är att stötta gruppen och hjälpa gruppen att hitta strategier för att ta sig över hinder som uppstår i arbetet. Grupphandledaren för också anteckningar över sina möten med gruppen och gruppens aktivitet och dessa är del av det underlag som används för examination. Så all handledning är att betrakta som en del av en löpande examination.

Det är viktigt att gruppen inte döljer problem som finns för handledaren utan vågar vara öppna med hur gruppen fungerar och eventuella problem som uppstår. Om gruppen inte är öppen med hur saker fungerar för handledaren så riskerar vi att examination sker på missvisande underlag. En risk med detta är också att en grupp inte presterar så väl som de borde kunna och medlemmar i gruppen riskerar att få komplettera projektet efter slutinlämningen.

Teknisk handledning

Med tekniska handledning avses hjälp med problem med kod, dokumentation, verktyg eller liknande. Denna handledning ges inte enbart av gruppens grupphandledare utan alla lärare som deltar i kursen ger handledning inom de områden de kan bäst. Gruppen kan bli uppmanad av sin grupphandledare att stämma av med någon annan handledare för hjälp och återkoppling. För att uppmuntra grupperna att söka stöd så finns det vissa speciella handledningstillfällen för vissa områden. Dessa tillfällen är markerade som ”drop-in hjälp” på Kronox-schemat.

Planerat tillgängliga lärare på Kronox-schemat:

- Anton Tibblin TSANTI ger extra stöd i webbrelaterade frågor.
- Rolf Axelsson TSROAX ger extra stöd i Java- och Android-relaterade frågor.
- Kristina Allder TSKRAL ger extra stöd i frågor som rör dokumentation och testning.
- Olle Lindeberg TSOLLI och assistenten Andreas Göransson ger extra stöd i frågor som rör C# och Monogame

Självgranskning

På Kronox-schemat finns vissa tillfällen som är angivna som självgranskning av någon dokumenttyp. Dessa granskningar leds av Kristina och innebär att vi går igenom det angivna dokumentet och talar om de delar som då bör finnas där. Detta införs som ett nytt sätt att ge stöd i år utifrån att många grupper missade samma eller liknande saker i sina dokument tidigare. Grupperna skickar en till två personer till självgranskningen som där följer med och kontrollerar att man inte missat någon del av det dokument som granskas. Gruppen tar själv med sig sin senaste version av det aktuella dokumentet antingen som en digital variant eller som utskrifter. Ingen inlämning görs på förhand och frågor besvaras muntligt under relaterade punkter på granskningen.

Handledningsmöten

Handledningsmöten sker varje vecka med gruppens handledare i form av handledning för gruppen separat eller i form av retrospektmöten (RS-möten, se mer information i nästa avsnitt) med andra grupper. Om det finns orsak att träffa en grupp utan övriga grupper närvarande en RS-vecka så kan extra handledningsmöten komma att bokas.

Handledningsmötena (och RS-möten) är att betrakta som obligatoriska. Har man förhinder så meddelas detta snarast till handledaren och gruppen. Att man missar enstaka handledningsmöten ger inte direkt problem men om man uteblir från flera handledningsmöten så är det stor risk att man inte blir godkänd på projektet.

Handledningsmöten har inte en exakt förbestämd tid på schemat utan bokas av handledaren och gruppen i samförstånd. Det finns grova tider bokade på Kronox-schemat men hur respektive handledare fördelar sina grupper under denna tid varierar. Handledarna informerar om hur de avser göra vid kick-off-tillfället måndagen 5/3.

Under de veckovisa handledningsmötena ska en person i gruppen (vem kan skifta från vecka till vecka) föra protokoll för vad som avhandlas på handledningsmötet och vad som eventuellt beslutas i någon fråga. Protokollen läggs upp i mapp för detta på kursplatsen. Protokollen ska som minst innehålla följande punkter:

- Vilka som är närvarande
- Tidrapport (information om arbetad tid tas ut via projekthanteringsverktyg)
 - Summering av tidrapport för
 - Varje person
 - Gruppen som helhet
 - Summeringen av tidrapporten ska antal timmar och procentsats av total timbudget för personer och gruppen som helhet.

- Kommentar om man ligger i fas med planerat arbetad tid eller om man ligger före/efter denna.
- Planering
 - För varje person i gruppen ska det finnas en kort beskrivning av:
 - Vad personen jobbat med sedan senaste mötet.
 - Vad personen är planerad att jobba med den kommande veckan.
 - Kommentar om man ligger i fas med den gjorda planeringen eller om man behövt göra några förändringar.

Fler punkter kan tillkomma beroende på vad som tas upp på mötet. De veckor som det är RS-möten så lämnar gruppen in ett protokoll som man skriver vid ett eget gruppmöte (utan närvarande grupphandledare).

Retrospektmöten

Under kursens gång kommer fyra olika retrospektmöten att hållas. Vid dessa så träffas 3-4 grupper gemensamt och demonstrerar sitt resultat och diskuterar sina projekt. Vilka grupper som håller retrospektmöten ihop avgörs av handledare. Samma grupperingar behålls genom hela projektet så att de olika grupperna följer varandra. Vid retrospektmötet så presenterar grupperna sitt arbete och alla grupper ger återkoppling på arbetet utifrån granskning som genomförts. Om man inte kan närvara på ett RS-möte så kommer en skriftlig komplettering att krävas.

På Kronox-schemat visas deadlines för att göra materiel tillgängligt för granskning för andra grupper inför retrospektmöten (även översikt nedan). Senast denna tid ska gruppen lägga upp en kopia av allt sitt materiel i en mapp på kursplatsen (där kommer att finnas specifikt avsedda mappar för detta). Grupperna kan då gå in och hämta materiel man ska granska på denna plats.

Presentation

Varje grupp ska vid retrospektmötet kunna demonstrera den senaste versionen av sitt projekt samt kunna visa upp övriga artefakter. Gruppen genomför sin demonstration av produkt och berättar om nuläget i projektet. Följande aspekter ska tas upp:

- Vad har fungerat bra.
- Vad har fungerat mindre bra.
- Vilka problem har man haft och hur löste man dessa. (Känsliga problem av personlig karaktär kan undvikas vid dessa möten.)
- Vilka utmaningar ser man inför den kommande sprinten.

Presentationen ska stödjas av något bildspel/slides för de olika punkterna.

Det sista retrospektet kommer att få lite mer detaljerade instruktioner då detta adresserar hela projektet och inte endast den senaste sprinten. Det sker inte heller någon granskning av andra gruppers materiel inför RS4.

Granskning

Ett par arbetsdagar innan varje retrospektmöte så ska varje grupp överlämna en kopia av sitt materiel till de övriga grupperna för granskning (se ovan). De mottagande grupperna ska läsa igenom dokument och kod för att kunna ge återkoppling på materialet till den överlämnande gruppen. Återkoppling ska ske på ett bra och konstruktivt sätt. Se Eklund (2010 kap. 3) för bra riktlinjer om att ge feedback/återkoppling.

Planera för att varje person i projektgruppen lägger 1-2 timmar på att granska de övriga gruppernas arbete inför ett retrospektmöte. Alla i projektgruppen behöver inte granska allt som lämnas över från de övriga grupperna. Det är lämpligt att göra någon form av indelning så att en person granskar en grupp, en annan person en annan grupp eller att en person granskar alla gruppernas kravdokument, en person koden, en person designdokument och så vidare.

Inför varje retrospekt så kommer en specifik lista över vad som specifikt bör granskas att publiceras på kursplatsen. Observera att tiden mellan deadline och retrospektmötet kan vara kort och planera in detta arbete i förhand.

Testning

Under den tredje sprinten ska grupperna genomföra testning åt varandra. Detta innebär att grupperna under en dedikerad testdag låter en annan grupp genomföra tester för kravbaserad testning enligt de testfall man skrivit. Den testande gruppen lämnar en testrapport för testerna som genomfördes. Bytet av testgrupper sker inom retrospektgrupperna och handledaren för grupperna hjälper till att sätta upp ett schema för vem som testar åt vem.

Då det kan vara svårt att få ingång annans kod utan erfarenhet av denna tidigare så kan testerna behöva göras på någon annans dator. Därför så görs detta på en dedikerad dag i vecka 18 (mer information kommer via kursplats och Kronox-schema).

Utöver denna testning så förväntas grupperna själva genomföra kravbaserad testning av krav och även regressionstestning av detta.

Planering

Planeringen sker enligt agila principer där det först görs en övergripande plan som bryts ner allt efter som behovet uppstår. Den första planeringen som görs är inget som slaviskt måste följas och genomföras. Om behov uppstår för att ändra den övergripande planen så görs detta och en ny version av projektplanen skapas.

Arbetet ska planeras för att utveckla produkten iterativt och inkrementellt. Den iterativa aspekten innebär att det kommer att ske förfining av krav, detaljdesign, kodskrivande, testning, med mera vid flera tillfällen under projektet. Allt detta sker i någon omfattning i varje iteration.

Den inkrementella aspekten innebär att planeringen ska utföras så att gruppen först utvecklar en första version av produkten som är det minsta möjliga körbara som går att göra. Sedan utvecklas nya versioner som fyller på allt mer funktionalitet. Nya versioner bör släppas med en till två veckors mellanrum. Varje version ska innebära att kraven för den planerade versionen förfinas, arbetet detaljplaneras och arbetsuppgifter skapas och ansvariga utses, detaljdesign

genomförs, implementering av kod sker och den nya versionen testas och testningen dokumenteras. Inom arbetet med en version kan olika arbetsmoment upprepas fler gånger om behov uppstår.

Arbets tid och tidrapportering

Gruppen har en budget på 220 timmar per person. Denna tid ska man planera att nyttja. Detta innebär att varje person bör planeras för drygt 18 timmar i veckan i 12 veckor.

Arbetstiden ska timrapporteras via projekthanteringsverktyg och redovisas vid handledningsmöten. Allt arbete som utförs i direkt koppling till projektet ska rapporteras och räknas in i budgeten. Detta innebär att möten med gruppen, handledningsmöten, att läsa in ny kunskap som behövs, testa ett nytt verktyg och liknande ingår i arbetstiden för projektet. Även den tid som läggs på den individuella fördjupningen räknas in i projektarbetet. Att läsa på inför en eventuell omtenta räknas däremot inte som projektarbete.

Tidrapportering ska ske genom att tiden loggas i projekthanteringsverktyget på lämplig issue. Detta innebär att allt arbete som sker i projektet behöver ha issues i projekthanteringsverktyget. Även saker som handledningsmöten behöver alltså läggas upp som arbete och flera personer rapporterar tid till detta issue. Tidrapportering bör ske dagligen och hållas uppdaterad löpande. Det är varje persons ansvar att rapportera den tid man arbetat.

All rapporterad tid i så kallade work items till ett issue ska loggas med en typ (typer för detta definieras där man slår på tidrapportering i projektet). Som minst ska följande typer av arbete definieras:

- Möte (handledningsmöten, RS-möten, andra möten som är avstämningar på olika sätt)
- Planering (tid som innebär arbete med att planera projektet, vad som ska göras en viss vecka eller liknande)
- Kravhantering (allt arbete som innebär att ta fram krav, analysera och dokumentera och på andra sätt arbeta med själva kravens utformning)
- Koddesign (att arbeta med hur koden ska struktureras, gäller även struktur på exempelvis databaser)
- Grafisk design (att rita bilder, bestämma hur grafiska gränssnitt ska se ut)
- Kodning (att skriva kod enligt en plan för vad som ska göras, mindre korrigeringar av fel som inte är upptäckta i regelrätta tester räknas in här)
- Testning (all typ av formell testning och utformande av testfall)
- Omarbetning kod (tid som läggs på att korrigera fel som upptäckts i formella tester eller granskningar)
- Omarbetning dokument (tid som läggs på att korrigera fel som upptäckts i formella tester eller granskningar)
- Granskning (tid som läggs på att ta fram riktlinjer och tid för själva granskningen, även tid som hör till olika former av analyser som görs i individuella fördjupningar)

Övergripande struktur

En viss övergripande struktur är fastslagen och måste följas. Denna består av en indelning i fyra olika sprintar om 3 veckor vardera. Varje sprint avslutas med ett retrospektmöte. Inför varje

sprint finns vissa gemensamma krav på vad som ska vara uppnått. Observera att detta är minimikrav. Projektgrupperna kommer att behöva komplettera dessa med egen planering.

- Sprint 1 v10-12. Till deadline för retrospektmöte 1 ska följande som minst ha uppnåtts:
 - Projektplan ska finnas i en första färdig version.
 - Kravdokument ska finnas i en första färdig version. Alla krav är dock kanske inte definierade i detalj ännu. Kravdokumentet kommer sannolikt att förändras men en första uppsättning funktionella och icke-funktionella krav ska finnas.
 - Det ska kunna genomföras en demo av produkten med fungerande kod. Inga krav behöver vara färdigimplementerade och testade formellt men det ska finnas exekverbar kod eller en statisk webbsida som minst.
- Sprint 2 v13-15. Till deadline för retrospektmöte 2 ska följande som minst ha uppnåtts:
 - Det ska finnas en version av produkten med en delmängd av de specificerade kraven ur kravdokumentet implementerade.
 - Det ska finnas en första version av verifiering- och validerings-dokumentet.
 - Det ska finnas ett designdokument i en första version.
 - Det ska ha genomförts kravbaserade tester enligt verifiering- och validerings-dokumentet och dessa ska ha dokumenterats i minst en testrapport.
 - Minst en kodgranskning och en dokumentgranskning ska ha genomförts (delar av individuell fördjupning).
- Sprint 3 v16-v18. Till deadline för retrospektmöte 3 ska följande som minst ha uppnåtts.
 - Det finns en första version av produkten som är en helhet.
 - Alla krav prioriterade som ”ska”/”must” (Eklund 2010 kap. 10) är implementerade och testade enligt verifiering- och validerings-dokument.
 - Tester ska ha genomförts för de implementerade kraven enligt testfall i verifiering- och validerings-dokument och dessa tester ska ha dokumenterats.
 - Tester ska ha körts av annan grupp och denna grupp ska ha lämnat testrapport på testerna som kördes
 - Användbarhetsanalyser, användbarhetstester och white-box-testning (delar av individuell fördjupning) ska ha genomförts minst en gång om någon gör dessa.
- Sprint 4 v19-22. Till deadline för retrospektmöte 4 ska slutinlämning av projektet ske. Under sprint 4 ska projektet förfinas utifrån den version som fanns efter sprint 3. Som minst så ska följande ha uppnåtts:
 - Problem som noterats tidigare ska ha åtgärdats och vissa av de krav som prioriterats som ”bör”/”should” ska ha implementerats.
 - Regressionstester ska ha genomförts för förändringar som skett och ny implementation ska ha testats minst en gång.
 - Som minst ska en andra omgång av granskningar och tester som faller under individuell fördjupning ha genomförts. För analyser så ska en kompletterande rapport ha gjort som adresserar hur problem som pekats ut i den första analysen åtgärdats.

Deadlines för inlämning av materiel inför retrospektmöten:

- Retrospektmöte 1: fredag 23 mars 12.00
- Retrospektmöte 2: torsdag 12 april 12.00
- Retrospektmöte 3: torsdag 3 maj 12.00
- Retrospektmöte 4 (slutinlämning av projekt): söndag 26 maj 23.55, dock ingen granskning av andra gruppers materiel för detta RS

Utställningsdag

Torsdagen 24 maj är det planerat att alla projektgrupper visar upp sina projekt på en utställning i Niagara. Grupperna visar då sina senaste stabila produkter. Mer information om utställningsdag publiceras på kursplatsen när denna närmar sig. Räkna dock med att detta är en heldag där ni behöver närvara och hjälpa till med att ställa i ordning och plocka undan utställningen utöver att närvara i er monter på själva utställningstiden.

Affischer

De flesta grupper har helt digitala produkter som kan vara svåra att få att synas på håll- Därför så görs affischer för projekten av grupperna. För att stötta detta så hålls en föreläsning om att skapa affischer måndagen 16/4 13:15-15.00. Denna föreläsning hålls av Jens Sjöberg som undervisar i mediaproduktion.

Under vecka 17 hålls också workshoppar där grupperna får hjälp av Jens med att skapa sin affisch till utställningen. Vilka grupper som har sin workshop när publiceras på kursplatsen.

Affischerna trycks upp av Mau:s tryckeri och därför så måste dessa lämnas in i god tid till utställningsdagen. Deadline för att lämna in affischer är söndag 6 maj 23.55. Riktlinjer för vad som gäller för dessa publiceras via kursplatsen. Ansvarig lärare för att ordna med tryckning är Zahra Ghaffari så frågor om affischerna kan ställas till henne.

Verktyg

Versionshantering

I projektet ska versionshantering med Git användas för kod. Andra artefakter kan versionshanteras om gruppen så önskar med det är inget krav.

Det erbjuds en labb i att använda Git som gruppen förväntas gå på. Närvaro är inte obligatorisk men man bör då gå igenom labbmaterielet på egen hand för att säkerställa att man förstått vad som krävs.

Man bör förbereda sig inför den labb – se instruktioner på kursplats.

Projektverktyg

YouTrack från JetBrains kommer att användas som projektverktyg på kursen. Här ska gruppen registrera arbetsuppgifter och tidrapportera dessa. Tänk på att även möten och liknande bör hanteras som arbetsuppgifter. Mer information finns i tycket Arbetstid och tidrapportering ovan.

Det erbjuds en labb med hjälp att komma igång med YouTrack men gruppen behöver lägga tid på att lära sig använda verktyget och gå igenom olika tutorials utöver denna tid. Hjälp med YouTrack kan också fås på drop-in-hjälp för dokumentation.

Man bör ha förberett sig inför labben genom att läsa på om YouTrack via labbmaterielet.

Artefakter

I mallarna för de olika artefakterna så ges mer detaljerade beskrivningar av innehåll.

Socialt kontrakt

Det sociala kontraktet fastslår riktlinjer för hur projektgruppen ska förhålla sig till varandra och projektet. Detta dokument är något gruppen kan använda som ett verktyg i konflikter som uppstår och är ett sätt att arbeta förebyggande för att minska risken för konflikter.

Riktlinjerna bör vara av utformade så att man försöker förebygga problem och bidra till att skapa ett positivt och konstruktivt arbetsklimat. Riktlinjerna ska inte vara av enbart bestraffande natur som säger vad man gör om något bryter mot en regel.

Kod

Kod-filer ska organiseras logiskt i en struktur som underlättar att förstå hur delar av koden relaterar till varandra.

Kod ska stödas av dokumentationskommentarer som underlättar förståelsen av kodens syfte samt anger vem som är ansvarig för denna del av koden (innebär inte att denna person nödvändigtvis skriver all relaterad kod).

Det ska gå att identifiera vem som skrivit och modifierat vilken kod. För kod så ska Git användas för att stödja versionshantering.

Koden ska inkluderas i det som lämnas över till de andra grupperna inför retrospektmötena. För att underlätta för de andra grupperna att få ingång er kod så ska det finnas en README-fil som ger instruktioner för att starta programvaran. Det kan behöva anges saker som vilka externa bibliotek som behövs, vilken fil som main-metoden ligger i, om det måste finnas en viss katalogstruktur för att sökvägar ska stämma eller liknande.

Dokumentation av projektet

All dokumentation ska ha ett enhetligt utseende gällande försättsblad, teckensnitt, struktur, med mera. Det ska synas att de dokument som skapats för ett projekt tillhör samma projekt. Det är upp till gruppen att själva utforma detta i samråd med handledare. Där finns vissa mallar men dessa behöver inte följas strikt i fråga om layout. Mallarna är fokuserade på innehåll. De krav som presenteras nedan är minimikrav på innehållet i dokumenten. Minimikraven på innehåll ska följas och det går bra att utöka innehållet efter diskussion med handledare. Är där någon del av minimi-innehållet som är problematiskt att uppfylla för något projekt så kan undantag göras men endast efter att handledare och kursansvarig gett sitt godkännande. Om handledare anser det vara nödvändigt, så kan handledaren utöka kraven på ett dokument för en grupp.

Dessa riktlinjer gäller för alla dokument, undantaget socialt kontrakt och protokoll från handledningsmöten:

- Det ska finnas ett försättsblad som anger
 - projektnamn
 - gruppnummer
 - dokumentnamn
 - versionsnummer
 - datum för versionen
- Dokumenthistorik som anger vem som skrivit vad i dokumentet eller vem som gjort vilken ändring. Denna ska föras löpande för att dokumentera vem som skrivit/ändrat vad. Det ska framgå av dokumenthistoriken
 - vad som lagts till/ändrats/tagits bort
 - vem som gjort det
 - när det gjorts
- Innehållsförteckning
- Syfte med dokumentet, här beskrivs även dokumentets omfattning.
- Ordlista
- Referenser (kanske inte behövs för alla dokument)
- Sidnummer på alla sidor (försättsblad undantaget)
- Sidhuvud med (försättsblad undantaget)
 - projektnamn
 - gruppnummer
 - dokumentnamn
 - versionsnummer

All dokumentation är levande dokumentation vilket innebär att nya versioner skapas när något ändras. En ny version ska dokumenteras i dokumenthistoriken. Ändringar som ger upphov till nya versioner kan exempelvis vara förfining av krav inför en ny version av produkten eller uppdatering av riskanalys på grund av ändrade omständigheter i projektet. Dokumenthistoriken behöver inte ta med varje mindre ändring enskilt utan ett antal ändringar kan sammanfattas. Exempelvis så dokumenteras inte varje liten ändring av stavfel eller liknande men en längre genomläsning kan ge upphov till en kommentar i dokumenthistoriken så som ”Korrekturläsning, åtgärdat mindre mängd stavfel och felaktigt layout.”

Projektplan

Projektplanen ska som minst innehålla (utöver de allmänna kraven):

- Översikt av projektet som beskriver
 - syfte med projektet
 - omfattning av projektet
 - mål för projektet
- En kortfattad övergripande produktbeskrivning.
- En beskrivning av målgruppen för produkten.
- En beskrivning av den utvecklingsprocess som projektet avser följa (vissa saker är bestämd enligt information om process i projekten).
- Beskrivning av projektets bemanning.
- Övergripande ansvarsområden för gruppmedlemmarna. Övergripande ansvar kan vara att ta ansvar för exempelvis ett dokument (exempel kravdokument), en övergripande syssla (exempel testning), ett tekniskt område (exempel design med CSS), eller något annat avgränsat område som gruppen kan identifiera.
- En grov planering som visar övergripande aktiviteter för varje vecka och olika milstolpar.
- Varje aktivitet i grovplanen ska ha en uppskattad tid.
- Varje aktivitet och milstolpe i planeringen ska ha en kort beskrivning. Ett fåtal meningar är tillräckligt.
- Milstolpar innebär att något ska levereras (exempelvis att en första release av kravspecifikationen ska vara klar eller att en viss release av produkten ska vara klar). Varje leverabel som ingår i en milstolpe ska listas. En milstolpe kan innehålla flera saker som ska vara avklarade för att milstolpen ska uppnås.
- Grovplaneringen ska visas som ett enkelt Gantt-schema (se exempel i Eklund (2010 kap. 12.9) eller Tsui (2014 Figure 13.5)).
- Riskanalys och diagram som anger sannolikhet och konsekvens för varje risk. För varje risk skrivs en kortfattad beskrivning samt en kortfattad handlingsplan om risken inträffar och en plan för att minska risken.

Det kan vara fördelaktigt att ta hjälp av tekniker som Work Breakdown Structure, WBS, och verktyg för att bryta ner projektet i hanterbara delar. Gantt-schemat kan också skapas med något verktyg (möjlighet finns i YouTrack) och sedan inkluderas i dokumentet. Det är dock fullt tillräckligt att skapa ett Gantt-schema för hand med ett kalkyl-ark (som Excell).

De aktiviteter som finns i projektplanen ska ha en motsvarighet i projektverktyget YouTrack där de sedan förfinas i en mer detaljerad nedbrytning. I YouTrack bör aktiviteter brytas ner till en nivå där de kan klaras av under en arbetsdag för en person.

Tänk på att ta med den tid som krävs för planering, möten, kommunikation och liknande sysslor i planeringen. Planeringen ska utgå från en budget om 220 timmar per gruppmedlem.

Krav- och analys-dokumentation

Som minst ska det finnas ett kravdokument med följande innehåll:

- En mer utförlig produktbeskrivning än den beskrivning som finns i projektplanen.
- En mer utförlig analys av målgruppen än den som finns i projektplanen.
- Beskrivning av intressenter för projektet.
- Kravlista med
 - Funktionella krav för produkten i naturligt språk.
 - Dessa ska struktureras i logiska grupper.
 - Koncepten användarkrav (user requirements) och systemkrav (system requirements) ska användas (Somerville 2010, kap. 4)
 - De funktionella kraven ska prioriteras enligt MoSCoW-modellen (Eklund 2010 kap. 10).
 - Kvalitativa krav (icke-funktionella krav)
 - Dessa ska grupperas i logiska grupper beroende på typ av krav.
 - Som minst ska begränsningar som finns dokumenterade. Begränsningar kan exempelvis vara vilka programspråk som används, vilka operativsystem produkten fungerar med, tekniker som måste användas för att skapa produkten.
 - Varje krav ska förses med en unik identifierare. Identifierare på formatet ett bokstavsprefix i formen av en mnemonik och ett löpnummer rekommenderas.
 - Krav ska vara testbart formulerade.

Kravdokumentet kan utökas med andra delar så som andra former att dokumentera krav eller annat innehåll som är relevant för projektet. Annan typ av dokumentation kan behövas för att beskriva aspekter som är svåra att fånga i direkta krav. Detta kan exempelvis vara begrepp som spelbarhet. Detta görs i samråd med handledare.

Designdokumentation

Designdokumentationen ska som minst innehålla:

- Ett system-diagram som visas produktens övergripande arkitektur (exempelvis klient, server, databas, GUI, fysikmotor och liknande) och vilken kommunikation som sker mellan dessa delar. För system som inte har någon kommunikation utan kör helt lokalt kan systemdiagrammet ersättas med ett diagram av typ motsvarande ett klassdiagram.
- System-diagrammet ska kompletteras med en text-beskrivning som beskriver de olika delarna och den kommunikation som sker.
- Ett användningsfalls-diagram för produkten (use case-diagram).
- Scenarion eller användningsfalls-beskrivningar
- Skisser på användargränssnitt som visar typiska drag för denna.
- Mer detaljerade diagram över systemets uppbyggnad, exempelvis klassdiagram eller fildiagram. Den detaljerade dokumentationen kan delas upp i flera delar.
- ER-diagram för eventuella databaser.

Designdokumentationen kan utökas och bör utformas utifrån varje projekts specifika natur. Detta sker i samråd med handledare.

Verifiering- och validerings-dokument

Den minsta omfattning som ska finnas för denna typ av dokumentation är ett verifiering- och validerings-dokument med tillhörande testrapporter. Testrapporter inkluderas i verifiering- och validerings-dokumentet. Verifiering- och validerings-dokumentet ska som minst innehålla:

- Beskrivning av testprocess för projektet. Som minst ska kravbaserad systemtestning utföras med utgångspunkt i kravdokumentet. Beskrivningen ska ange hur testning genomförs, när testning sker i förhållande till utvecklingsprocessen och hur denna dokumenteras. Det ska också beskrivas hur eventuella problem som testningen identifierar ska hanteras.
- Mer detaljerad beskrivning av testmetoder som används. Som minst ska kravbaserad systemtestning beskrivas och hur detta tillämpas i det här projektet.
- Om inte alla krav kan testas så ska det beskrivas hur krav har prioriterats för test.
- Testfall som skapas utifrån valda testmetoder (som minst kravbaserad systemtestning).
- Varje testfall ska som minst innehålla:
 - namn på testfall
 - id för testfallet
 - förberedelser som behövs för att genomföra testet
 - beskrivning av teststeg som ska utföras i själva testet
 - beskrivning av vilket resultat man förväntar sig av testet
- Spårningsmatris(er) som anger vilka testfall som testar vilka krav.
- Testrapporter ska skrivas för alla tester som utförs enligt testprocessen. Testrapporter ska som minst innehålla:
 - Datum för testtillfället
 - Vem som utfört testerna
 - På vilken kodversion testerna utförts
 - Vilka testfall som genomförts vid testtillfället
 - Resultat av genomförandet av testfallen

Andra typer av verifiering och validering kan genomföras (exempelvis granskningar, andra typer av testmetoder användas) beroende på vad som är lämpligt för just detta projekt. De individuella fördjupningarna innebär exempelvis viss utökning av verifiering och validering. Avstämning av utökning sker i samråd med handledare.

Individuell fördjupning

Den individuella fördjupningen innebär att varje student väljer någon uppgift som hen tar speciellt ansvar för i projektet. Dessa fördjupningar innebär att tillämpa olika metoder inom Software Engineering på ett sådant sätt att de skapar mervärde för projektet. Resultatet av de individuella fördjupningarna införlivas med projektets övriga artefakter (i verifiering och validerings-dokumentet) eller dokumenteras i egna artefakter (samma generella krav ställs på dessa artefakter som övriga obligatoriska artefakter). För varje persons ansvarsområden i projektplanen så ska det anges vilken av de möjliga individuella fördjupningarna (och vilken variant) som personen ansvarar för.

Den individuella fördjupningen innebär att någon tar ansvar för en aktivitet men det är kanske inte så att denna aktivitet kan genomföras av den här personen ensam. Andra gruppmedlemmar kan behöva bidra i olika gruppaktiviteter som exempelvis granskningar av artefakter. Gruppmedlemmar i den egna gruppen ska bidra till andras individuella fördjupning om denna är av en sådan natur att aktiviteten på något sätt utförs i grupp. Vissa individuella fördjupningar kräver samarbete med en annan projektgrupp men det är inte tvingande att ställa upp på en sådan förfrågan från en annan grupp. Om en student väljer en individuell fördjupning som kräver medverkan av en annan projektgrupp men inte kan hitta någon villig grupp så måste hen kanske byta val av individuell fördjupning eller en annan testgrupp hittas.

Möjliga individuella fördjupningar:

- Dokumentgranskning, ska utföras så någon måste ta denna fördjupning, denna involverar flera gruppmedlemmar (max en person)
- Kodgranskning, ska utföras så någon måste ta denna fördjupning, denna involverar flera gruppmedlemmar (max en person)
- White-box-testning, kan göras i max två varianter (max två personer)
- Användbarhetstestning, kan utföras i max två varianter (max två personer), kräver en partnergrupp som utför testerna
- Användbarhetsanalys, kan utföras i max två varianter (max två personer)

För detaljer runt de olika typerna av individuell fördjupning se instruktioner i respektive dokumentet. Observera att det finns krav i översikten över sprintarna som relaterar till möjliga individuella fördjupningar.

Reflektionsrapport

I slutet av projektet så skrivs en individuell reflektionsrapport. Denna delas inte med andra studenter. Reflektionsrapporten innebär en examination av att varje student kan reflektera kring betydelsen av samarbete och ledarskap i en projektgrupp. Reflektionsrapporten används även för att kontrollera att handledarens uppfattning av gruppens arbete stämmer med gruppmedlemmarnas egna arbete. Detaljer för reflektionsrapporten publiceras senare på kursplatsen.

För att inte behöva sitta och försöka komma i håg vad som hände i projektet så vecka 23 innan reflektionsrapporten ska lämnas in så rekommenderas det att varje student för en egen dagbok över sina tankar om projektet och för händelser som kan användas för att exemplifiera saker i reflektionsrapporten.

Referenser

Eklund, S. (2010). Arbeta i projekt – individen, gruppen, ledaren, Studentlitteratur

Sommerville, I. (2010). Software Engineering, 9:th ed, Pearson Higher Education

Tsui F., Karam O., Bernal B. (2013/2014). Essentials of Software Engineering 3rd Edition, Jones & Bartlett Learning