Hisspresentation av programdesign

Projektplan: Kommunikation i teknisk utbildning, 2014-2015

Ulf Asklund, Datavetenskap

1. Mål

- Inom ramen för kursen förbättra studenternas skriftliga och muntliga förmåga att förklara vilka designbeslut de tagit och varför?
- Ytterligare (jämfört med tidigare kurser) fördjupa studenternas kunskap i designmönster och när de ska användas.

2. Metod

Utöka en inlämningsuppgift med moment att skriftligt och muntligt beskriva vald designlösning och motivera den. De två coachande möten som redan finns utökas med muntlig respons på de nya momenten. En föreläsning kommer att kompletteras med teori och goda exempel av beskrivningar och motiveringar.

3. Bakgrund

Kurserna ”Objektorienterad modellering och design” (EDA061) och ”Objektorienterad modellering och diskreta strukturer” (EDAF10) omfattar båda kunskapen att strukturera programkod på ett sådant sätt att den blir lättare att förstå och underhålla. Syftet med att strukturera koden bra är tvåfald:

1. koden i sig är strukturerad på ett sådant sätt att man tekniskt enklare kan modifiera den enligt nya krav och förutsättningar, t ex genom att göra det möjligt att *lägga till* kod på *ett* ställe snarare än att *ändra* befintlig kod på *många* tillsynes oberoende ställen.
2. den är enklare att förstå och diskutera programmerare emellan, gärna genom att använda större befintliga lösningsblock i diskussionen som alla inblandade redan känner till och direkt förstår användningen av. Detta att jämföra med att behöva förklara varje enskild kodrad.

De två viktigaste ingredienserna för att strukturera bra är:

- principer och mönster, vilket handlar om att utforma och implementera större objektorienterade program genom att tillämpa designprinciper och använda designmönster.

- UML (Unified Modeling Language) - en notation vi använder för att beskriva och diskutera olika designalternativ.

Det finns ofta många olika sätt att konstruera ett program som alla uppfyller kraven och det är inte alltid självklart vilken lösning som är bäst. Det har ofta olika för- och nackdelar och vilken som ska väljas beror på hur man prioriterar dessa. För att träna studenterna i att analysera lösningar och att själv kunna värdera val de måste göra i sin design har vi både övningar och inlämningsuppgifter där studenterna själva ska presentera och motivera sina lösningar. Övningarna diskuteras i studentgrupper ledda av en lärare, medan inlämningsuppgifterna presenteras för en lärare som ger omedelbar respons på deras lösning.

4. Nuvarande moment för inlämningsuppgift 2

1. Studera de tillhandahållna paketen. Paketet expr skall bara användas och inte modifieras. I klassen TestExpr finns ett exempel på hur man kan använda paketet. Paketet view innehåller ett skelett till användargränssnittet. Här skall man både modifiera och lägga till klasser och det är viktigt för designarbetet att veta vad som finns. Det går att starta programmet via klassen XL, men funktionaliteten är mycket begränsad.
2. Gå igenom de 10 punkterna på sidan 3 i uppgiften och gör en design med klassdiagram. Senast 24 timmar före det redovisningstillfället skall ni skicka in klassdiagrammen för alla paket utom expr elektroniskt. Bifoga också hela eclipseprojektet som en zip-fil.
3. Efter det första redovisningsmötet reviderar gruppen sin design och implementerar den. Det färdiga projektet inlämnas elektroniskt senast 24 timmar före det andra designmötet.

Studenterna använder sig av UML för att beskriva sina klassdiagram, vilket ju är det kommunikationsformat vi lär ut i kursen. Även om UML-graferna är korrekta händer det ganska ofta att gruppen har svårt att förklara vad de har gjort. Ännu svårare är det att förklara varför deras lösning ser ut som den gör. Inte sällan har de glömt vad de gjort (det var ju några dagar sedan) och handledaren får visa dem deras kod och ställa väldigt specifika frågor. En sådan presentation av en design hade inte fungerat om inte mottagaren redan visste hur den borde se ut…, och är m a o helt utan värde i ”verkligheten”. Observera att lösningen fortfarande kan vara korrekt och att det resulterande programmet hade varit till belåtenhet för en ev. kund.

5. Förslag till förändring

- Uppgiften utökas med ett krav på beskrivning av designen i skriftlig form. Språket

ska utnyttja den terminologi kursen lärt dem och beskriva lösningen på en hög abstraktionsnivå i termer av mönster och principer snarare än koddetaljer, den ska vara ett komplement till de befintliga UML-diagrammen.

- Gruppen ska presentera sin lösning för handledaren och motivera sina val. Studenterna ska leda presentationen och handledaren ger respons inte enbart på själva designen utan även på det sätt den presenteras.

6. Utvärdering

- Alla inblandade handledare samlas innan kursen och diskuterar igenom de nya momenten så att alla förstår vad som ska göras och varför.
- Efter kursen ska alla handledare göra en kort analys om hur de tycker att det gick, dvs deras uppfattning om på vilket sätt de nya momenten förändrade uppgiften och vad studenterna lärde sig.

7. Funderingar och frågor

Den skriftliga beskrivning av designen som studenterna ska göra är inte populärvetenskaplig utan snarare en klassisk specifikation/dokumentation. Kommer den då ge de positiva resultat som en populärvetenskaplig text gör enligt Pelger och Santesson i ”Retorik för naturvetare”? Jag tror det. Skriften ska hållas på en abstraktionsnivå av mönster och enbart vara det komplement som behövs för att man snabbt ska förstå ”tänket” bakom lösningen och de mest fundamentala delarna. Det skulle kunna jämföras med en ”elevator pitch” av deras lösning - något som ska intressera mottagaren att ta reda på mer detaljer om deras lösning. Att behöva förstå vad som är viktigast att beskriva och att göra detta kort, med rätt terminologi och med rätt motivering av resp. använt mönster kräver en djup förståelse av de centrala delarna i kursen. Detta i motsats till att lägga fram alla detaljer på en gång och låta mottagaren försöka förstå och ställa frågor.

Blir det fel balans i kursen om mer tid läggs på hur dokumentation och presentation görs, snarare än hur bra själva lösningen är? Det är ju ändå lösningen som är resultatet. Enligt de ”nya” agila utvecklingmetoder som används för att programvara ska koden vara självförklarande och man ska undvika onödig dokumentation då den ju aldrig kommer kunna hållas uppdaterad vid alla förändringar som görs. Är den föreslagna kursändringen då ett ”steg bakåt” och inte alls i linje med vad kursen faktiskt ska lära ut?

8. Tidplan 2014

- juni - augusti: utveckla de nya momenten och väva in det i kursen

- september-oktober: genomföra kursen med de nya momenten.
- oktober-november: analys och utvärdering av de nya momenten

Preliminär plan för införande av kommunikationsmoment i EITF05 och EIT060

Martin Hell och Paul Stankovski

1 Utförande

Vi inriktar införandet av kommunikationsmoment på våra två kurser

- EITF05 Webbsäkerhet (LP1, C3 och D4, 80 studenter),
- EIT060 Datasäkerhet (LP3, C2 och D3, 120 studenter).

I båda kurserna har studenterna projektarbeten som redovisas medelst inlämning av skriftlig rapport, samt datorpresentation för projektledare.

Vår grundplan är att införa ett kommunikationsmoment, där studenterna presenterar sina projektarbeten för sina kursare och för projekthandledaren, ungefär som en mini-konferens.

Efter samtal med studentrepresentater har det framgått att lärarens återkoppling på den skriftliga rapporten värderas högt.

2 Nuvarande projektutformning av EITF05

Projektet består i att utforma en liten e-handelssajt, där särskild hänsyn skall tas till webbsäkerhetsaspekter. Projektet utförs i grupper om 1-2 personer.

Varje grupp skriver en kort skriftlig rapport på ungefär 3 sidor och presenterar sitt program för projekthandledaren. Redovisningstiden är cirka 20 minuter per grupp, och grupperna får återkoppling av läraren på både rapport och program.

3 Föreslagna förändringar för EITF05

Vi lägger till ett efterfrågat moment, att implementera en eller två av de webbaserade attackerna som presenteras i föreläsningarna, samt att visa hur man skyddar sig. Rapportstorleken ökas till ungefär 5 sidor, där dessa attacker och motsvarande skydd ska beskrivas tekniskt.

Vi bildar kluster om 3 grupper. Varje grupp ska granska de två övriga klustergruppernas rapporter och producera ett ifyllt granskningsformulär till varje rapport. Granskningsformuläret har vi ännu inte utformat, men fokus kommer att ligga på tekniskt innehåll.

Varje grupp ska också på ett A4 redovisa hur de inkommande granskningskommentarerna har behandlats – hur de har förbättrat rapporten.

Beträffande rapporten kommer läraren bara att ge återkoppling på skrivteknik och rapportstruktur. Teknisk feedback ges genom granskningsprodecuren.

Varje grupp ska också presentera sina implementerade attacker inför sina kursare i konferensformat. Målsättningen är att gruppernas presentationener på kort tid ska visa vad de har åstadkommit. Presentationerna ska

- vara korta (ska ge tidspress, 7 minuter?),
- tydligt visa att de har implementerat attack och skydd,
- tydligt visa att de har förstått attack och skydd,
- tydligt visa att alla gruppmedlemmar är insatta.

Efter varje presentation får den presenterande återkoppling från kursare och lärare. Förväntat tempo: 3 grupper per time (45 minuter), inräknat efterdiskussion.

För att kompensera studenternas merarbete ökas gruppstorleken till 3-4 personer.

4 Kontinuitetsadaption av EIT060

En uppföljning är tänkt till EIT060 Datasäkerhet, med motsvarande och fohoppningsvis förbättrade åtgärder. Antalet studenter är dock större här (högre risk), så vi väntar med en detaljplan tills vi har utvärderat utfallet av EITF05.