

Systemdesign med ett användarperspektiv

Tentamen består av sju huvudfrågor som ska besvara på en rimlig detaljnivå. Svaren på frågorna bör inte överstiga sju sidor (+- några rader). Kortare svar går naturligtvis bra, men de måste fortfarande innehålla rätt information.

Generella instruktioner:

Vänligen, läs igenom noggrant innan du börjar!

- Det här är en hemtenta, som du har ca fyra dagar på dig att besvara. Det innebär bland annat att jag inte kan kontrollera hur du bär dig åt för att besvara frågorna. Följaktligen är det därmed helt tillåtet att använda all information på hemsidan, att Googla eller till och med (!) att prata med andra studenter om frågorna. Naturligtvis innebär det förstås också att jag förväntar mig lite högre nivå på svaren.
- Var noga med att skriva ditt namn på tentans förstasida (eller i dokumenthuvudet).
- Jag är inte intresserad av detaljer, eller programkod i svaren. Det beror på att jag vill veta om ni har förstått det mer generella i kursinnehållet. Ni ska däremot motivera era svar så att det framgår att ni vet vad ni skriver om, och inte bara skriver ner mer eller mindre färdiga "citat". Det innebär också att ni:
 1. ska kunna ge svar som är på en mer generell nivå.
 2. ska svara på ett sådant sätt att **jag** blir övertygad om att ni har förstått idéerna bakom svaren.
 3. kan upptäcka att några frågor även kan beröra saker som inte ingått direkt i kursen, men som ni bör kunna resonera kring, med hjälp av det vi gått igenom.
- Svaren skall vara sådana att de är möjliga att förstå för en student som inte har tagit kursen än.
- Betyget på tentamen är inte baserat på någon slags poäng-per-fråga, utan kommer att bedömas som en helhet.
- Det är helt OK att ge svar som kanske avviker från kursens (eller min personliga) syn på ämnet. Men då måste du ge en bra motivering till varför ditt svar skulle vara bättre/mer rätt. Det kan fortfarande ge hög poäng på tentan.
- Du kan svara på engelska eller svenska, vilket som känns enklast.
- Några frågor kan komma att överlappa, och det är i sådana fall helt avsiktligt!

Fråga 1 – Yta och djup:

Ett system består oftast av en *backend* och en *frontend*. Ett sätt att se på denna uppdelning är det som i kursen benämns: Model-View-Control¹ (MVC), eller med liknande begrepp. Grundtanken med detta synsätt är att det är viktigt att skilja på gränssnittets (vyns) representation av datastrukturerna i backend (Model).

- A. Förklara varför det i vissa fall kan vara lätt att blanda ihop metaforen (the view) och systemets funktionella bas (the model). Ge gärna något exempel på hur denna sammanblandning kan ta sig uttryck.
- B. Förklara hur det kan påverka användarens förståelse av systemet.
- C. Förklara vad det kan få för konsekvenser för programmeringen av backend.

Fråga 2 – Model-View-Control:

- A. Beskriv principerna bakom Model-View-Control-paradigmen (MVC)
- B. Vilka för- respektive nackdelar finns det med MVC? Motivera noggrant!
- C. MVC är egentligen en ganska enkel princip att förstå, men med gränssnittens utveckling från kommandospråk till dagens multimediala-multi-interaktiva programvara har det blivit svårare att tillämpa MVC. Förklara varför!
- D. Varför är det så viktigt att ändringar i systemet mer eller mindre uteslutande görs i modellen (backend) och inte direkt i gränssnittet (vyn)?
- E. Går det att genomföra en fullständig separation av gränssnittet (vyn) och backend (modellen)?

Fråga 3 – Den kognitiva människan:

Förklara hur vart och ett av de följande mänskliga egenskaperna påverkar hur **vi** måste resonera för att kunna designa ett användbart och bra system. Beskriv i dina svar också styrkor respektive svagheter med egenskaperna *som vi kan nyttja respektive understödja*:

- A. Minnet,
- B. Synen,
- C. Hur vi löser problem.
- D. Förklara varför det är troligt att du kommer komma ihåg de sakerna som vi tagit upp nu på denna tentamen bättre än kunskap som du kan ha råkat på under t.ex. en föreläsning.

Fråga 4 – Den dumma användaren:

- A. Finns det något sätt att på ett mer "neutralt" sätt förklara hur en användare kan utföra något som andra tycker är "urdumt" eller till och med korkat?
- B. Hur ska/kan vi som systemutvecklare/programmerare minska risken för att någon ska göra fel i våra program?

¹ Observera att "Model-view-control" förekommer i flera olika varianter, men huvudtanken är att skilja implementationen av användarens modell av systemet från den mer formalistiska modellen som framförallt bygger på datastrukturer, som t.ex. arrayer, listor eller hash-tabeller.

Fråga 5 – Metaforer:

- A. Beskriv egenskaperna hos en bra metafor.
- B. Finns det någon risk med att använda en metafor?
- C. Drag-and-Drop är en metafor som kom i samband med införandet av direkt manipulering i grafiska gränssnitt. Beskriv på vilket/vilka sätt just denna metafor kan skapa problem för en användare.
- D. På vilket sätt kan den vara problematisk även för en programmerare?

Fråga 6 – Programmeraren och Användaren:

- A. Vad beror det på att det kan uppstå missförstånd när man ska överföra en arbetsuppgift till ett stödjande system? Ge tre exempel och förklara.
- B. Man brukar framhålla vikten av att använda konstanter (variabler med ett fixt värde, t.ex. " $\pi = 3.1415926$ ") istället för att använda numeriska värden direkt i koden. Förklara varför det också är viktigt att använda ett "dictionary" för att lagra alla strängar som ska visas för användaren. Motivera utifrån samspelet mellan programmerare och användare.
- C. Vilken betydelse har MVC för möjligheten att konstruera program som är användbara (usable)?
- D. Hur bör man tänka när det gäller att implementera en mekanism för UNDO-REDO? Vad finns det för viktiga ställningstaganden man måste göra?

Fråga 7. – Systemutveckling med ett användarperspektiv

Beskriv hur du som projektledare skulle genomföra ett mindre systemutvecklingsprojekt i en projektgrupp med ca fem-sex personer. Din beskrivning ska utgå ifrån att projektet har ett starkt användarperspektiv. Hur planerar du projektet? Vilka moment ser du som viktigast och varför? Vad ser du som de största problemen i ett sådant projekt? Är det något som är extra viktigt att infoga i de vanliga systemutvecklingsmodellerna?

Lycka till!

/Lars