

## Uppgift 1: Kravspecifikation för programvara

Läs nedanstående text om kravspecifikation för programvara. Svara på flervalfrågorna 1.1 – 1.3 på basen av texten. Svara dessutom på essäfrågan 1.4 både på basen av texten och dina egna kunskaper.

Med programvaruproduktion avser man verksamhet, som riktar in sig på att producera datorprogram. Datorprogrammen skapas för något ändamål: programvaran är gjord för att användas. I denna process är företaget eller organisationen som beställer programvaran en kund och de personer som använder programvaran kallas slutanvändare. I början av processen måste man utreda vad programvaran ska göra: vad är dess mål, hur fungerar den och hur kunden vill använda den. Det centrala i programvaruproduktionsprocessen är alltså att definiera kundens krav på programvaran som ska konstrueras. Programvarans krav kan uppdelas i två klasser: (1) *funktionella krav* berör de funktioner som programvaran erbjuder, d.v.s. allt det man kan göra med programvaran, och (2) *icke-funktionella krav* berör hela programvarans kvalitet (såsom användbarhet och datasäkerhet) och de begränsningar som ställs på programvaran av dess driftsmiljö (till exempel ”programvaran ska implementeras i Windows-omgivning”).

Att fastställa, dokumentera och hantera kraven kallas *kravspecifikation*. I kravspecifikationen skapar man ett dokument, d.v.s. en skriftlig beskrivning, om programvaruprojektets målsättningar och krav. Inom den traditionella tekniska planeringen, t.ex. vid ett brobygge, gör man först upp noggranna planer och börjar bygga först efter att planerna blivit klara. Med den här traditionella metoden kan man också planera datorprogram. Inom programvaruproduktionen går man ändå ofta tillväga iterativt: man planerar, förverkligar, testar att första versionen av programvaran fungerar, man planerar mera och noggrannare, förverkligar lite till, testar osv.; detta upprepas många gånger. Kravspecifikationen borde åtminstone påbörjas före den noggrannare planeringen och förverkligandet av programvaran.

Karaktären hos den programvaran som ska utvecklas, organisationen som utvecklar det och processen som används vid programvaru-utvecklingen påverkar hur kraven definieras. I vart fall bör slutanvändarna och representanter för kunden vara aktivt med i processen. Kravspecifikationen uppdelas ofta i ett antal arbetsskeden:

- kartläggning av kraven
- analys av kraven
- dokumentation av kraven

- validering av kraven
- hantering av kraven.

Oftast överlappar dessa arbetsskeden varandra och kravspecifikationen framskrider stegvis medan den blir allt mer detaljerad: Först kartläggs, analyseras och dokumenteras en del av kraven. Processen försätter, tills de önskade kraven har dokumenterats med nödvändig noggrannhet.

### Kartläggning av kraven

I början av kravspecifikationen är det skäl att klargöra vem som är programvarans intressenter, alltså de som direkt eller indirekt har att göra med programvaran. Intressenterna är till exempel de som är tänkta att vara programvarans slutanvändare, beslutsfattande representanter för företaget som beställt programvaran, samt representanter för organisationer vilka är ansvariga för andra datasystem som är kopplade till programvaran som ska förverkligas. När de olika intressenterna har kartlagts, kan man sedan på mycket olika sätt få fram kraven; man kan till exempel intervjua representanter för intressenterna eller arrangera brainstorming mellan kunden och programvarans utvecklingsteam. Ofta funderar man på vilka olika *användarroller* programvaran har och hittar på typiska användarscenarier för de olika användarrollerna.

### Analys, dokumentering, validering och hantering av kraven

Förutom att samla in kraven, måste de *analyseras*. Är kraven sinsemellan utan motsättningar och omfattar de tillräckligt väl de möjliga användarscenarierna? Är det möjligt och ekonomiskt rimligt att förverkliga kraven? Kan kravet verifieras, d.v.s. kan man fastställa om den färdiga programvaran uppfyller kravet eller inte? Till exempel kravet ”programvaran är lätt att använda” är inte bra, för att det är svårt att verifiera om något är lätt att använda. Krav på användbarhet kan dock specificeras, så ett de går att verifiera.

De kartlagda kraven *dokumenteras* i en eller annan form. Före programmeringen inleds behöver programvaru-utvecklaren en specifikation, alltså en beskrivning på hur programvaran eller en del av den bör fungera. Även för testning behövs en beskrivning på hur man vill att programvaran som ska testas bör fungera.

Under *valideringen* av kraven försäkrar man sig om, att de samlade och dokumenterade kraven motsvarar kundens uppfattning och beskriver sådana system som kunden anser sig behöva.

## Urvalsprov i datavetenskap, våren 2021

---

Speciellt om kraven förändras mitt under programvaruutvecklingsprocessen behövs *kravhantering*. Med hantering avses här till exempel formulering av nya krav som kunden kommer på och ändring av redan noterade krav.

### Funktionella krav

Som tidigare nämnts, delas kraven in i *funktionella* och *icke-funktionella* krav. Funktionella krav beskriver vad man kan göra med programvaran, d.v.s. vilka funktioner den har. Till exempel en nätbutiks funktionella krav kunde bland andra vara följande:

- en kund kan registrera sig som användare av nätbutiken
- en registrerad kund kan lägga en vara i inköpskorgen
- i samband med en lyckad kreditkortsbetalning får kunden bekräftelse per e-post att köpet lyckats
- en kund som loggat in i systemet ser sin egen köphistorik
- administratören kan lägga till nya produkter till butikens utbud
- administratören kan ändra prisuppgifterna för produkterna i systemet.

Oberoende av hur de funktionella kraven dokumenteras är det ganska vanligt att uttrycka kraven i en form, där ett enskilt användarscenario för programvaran beskrivs för någon viss användarroll. Exempelvis ”administratören kan ändra prisuppgifterna för produkterna i systemet” beskriver en funktion hos programvaran för de användare som har rollen ”administratör”.

### Icke-funktionella krav

Den andra klassen av krav, icke-funktionella krav, fördelar sig på två delområden: kvalitetskrav och begränsningar på driftsmiljön.

*Kvalitetskrav* är faktorer som styr och begränsar hela systemets funktionalitet, som till exempel

- användbarhet: hur lätt och smidigt användaren kan utföra det hen vill och hurudan programvarans användarupplevelse är
- datasäkerhet: vem har åtkomst till systemet och den data som behandlas där

## Urvalsprov i datavetenskap, våren 2021

---

- prestanda: hur snabbt reagerar systemet på användarens åtgärder
- skalbarhet: förblir programvaran tillräckligt snabb då antalet användare eller mängden data som behandlas ökar
- stabilitet: hur reagerar programvaran på felsituationer och återhämtar sig från dem.

Programvarans användare kan inte uppfatta alla kvalitetskrav. Sådana krav är till exempel

- expanderbarhet: går det lätt att utöka programvaran med nya funktioner i framtiden
- testbarhet: kan programvarans felfrihet säkerställas, även i samband med vidareutveckling.

Driftsmiljöns begränsningar är bland annat

- teknologi för förverkligande: med vilka programmeringsspråk och bibliotek förverkligas programvaran, vilka databaser används
- användaromgivning: är det meningen att programvaran ska användas till exempel med telefon eller bordsdator, eller med båda
- anslutning till andra system: används till exempel någon utomstående tjänsts användar-ID för inloggning eller data som erbjuds av öppna gränssnitt
- anpassning till lagar och standarder: exempel på sådana är de krav som ställs av EU:s dataskyddsförordning

I motsats till funktionella krav, som ofta beskriver programvarans enskilda funktioner (till exempel att produkten kan läggas i inköpskorpen), berör icke-funktionella krav oftast hela programvaran och inverkar på hur programvarans struktur, d.v.s. arkitektur, ska planeras. Till exempel om vi vill konstruera en nätbutik som är skalbar till miljontals användare, bör den konstrueras på ett annat sätt än en nätbutik som kan ha maximalt tjugo användare samtidigt. Om kvalitetskraven ändras radikalt under programvarans utvecklingsarbete, kan ändringarna vara svåra att göra och kräva stora korrigeringar i hela programvarans konstruktionsprinciper.

### Ett bra krav och dess dokumentation

Ett bra krav bör vara

- lämpligt: kravet beskriver verkliga behov som berör systemet
- förståeligt: kravets innehåll och representation ska vara förståeligt för
- både användaren och utvecklaren
- exakt: kravet bör presenteras exakt och entydigt
- verifierbart: det bör i praktiken gå att bekräfta, om ett krav är uppfyllt.

Hur mycket dokumentation som behövs för ett krav beror på karaktären hos programvaran som ska förverkligas: programvaran för patientmonitorn på sjukhusets intensivavdelning förutsätter ett annat angreppssätt än nätbutikens programvara. På många sätt och vis går programvaruplanering ut på att göra upp specifikationer, börjande från en definition på grov nivå och slutligen avslutande med programvarans formulering med ett programmeringsspråk. Specifikationen på grov nivå kan till exempel representeras som en fritt formulerad text och olika grafer. När man bearbetar specifikationen lägger man till flera detaljer, som gör att den kommer närmare maskinellt behandlingsbara representationer som används vid förverkligande av programvaran.

*Svara på flervalsfrågorna 1.1 – 1.3 på basen av texten. Svara dessutom på den essäliknande frågan 1.4 både på basen av texten och dina egna kunskaper.*

### Frågor i skede 1

**Fråga 1.1 (0-5 poäng)** Nedan finns en lista på krav för ett universitets kursanmälningssystem. Ange för vart och ett, om det är funktionellt eller icke-funktionellt. Markera också om det är exakt, samt om det är förståeligt. Välj alla rätta alternativ.

**Poängsättning:** För varje krav ges 1/2 poäng, om man har valt det rätta alternativet av funktionellt/icke-funktionellt krav; och 1/2 poäng, om man valt alla rätta och inga felaktiga alternativ beträffande exakthet och förståelighet; annars får man 0 poäng för dem.

## Urvalsprov i datavetenskap, våren 2021

---

	funktionellt	icke-funktionellt	exakt	förståeligt
En studerande kommer inte åt att se andra studerandes personuppgifter.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bakgrundsfärgen är vit för de texter som systemet visar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kursens anmälningsuppgifter visas nästan alltid inom 0,5 sekunder för läraren.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Användaren kan välja, om hen vill använda systemet på finska, svenska eller engelska.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Anmälningsystemet innehåller alla funktioner som studerandena behöver.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Fråga 1.2 (0–5 poäng)** Företaget X håller på att utveckla en nätbutik som säljer elektronik. Man utvecklar en ny sökfunktion till nätbutiken. Med funktionen kan användarna söka produkter på basen av olika sökord och andra sökkriterier (till exempel pris, tillgänglighet, ursprungsland osv.). Nedan har fem egenskaper som bör uppfyllas för sökningen listats. Välj för varje egenskap till vilka av följande kvalitetskrav de hör:

- användbarhet
- datasäkerhet
- prestanda
- skalbarhet
- stabilitet
- expanderbarhet.

## Urvalsprov i datavetenskap, våren 2021

---

**Poängsättning:** För varje krav ger varje rätt vald ”hör till” eller ”hör ej till” 1/6 poäng och fel vald  $-1/6$  poäng; valet ”svarar inte” ger 0 poäng. För varje egenskap är minimipoängtalet 0 poäng.

1. Systemets förverkligande ska göra det möjligt att senare lägga till nya typs sökvillkor.

	hör till	hör ej till	svarar inte
användbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
datasäkerhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prestanda	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
skalbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stabilitet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
expanderbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Om databasen som används i sökningen inte fungerar, bör sökfunktionen styra användaren till andra sidor på webbplatsen.

	hör till	hör ej till	svarar inte
användbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
datasäkerhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prestanda	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
skalbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stabilitet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
expanderbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Sökegenskapen bör vara förverkligad senast 15.6.

	hör till	hör ej till	svarar inte
användbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
datasäkerhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prestanda	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
skalbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stabilitet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
expanderbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4. Den registrerade användaren ser vilka sökord hen själv har använt tidigare, men inte dem andra använt.

## Urvalsprov i datavetenskap, våren 2021

---

	hör till	hör ej till	svarar inte
användbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
datasäkerhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prestanda	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
skalbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stabilitet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
expanderbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5. Produktsökningen ska returnera en lista på produkter som uppfyller sökvillkoren inom 2 sekunder oberoende av antalet användare.

	hör till	hör ej till	svarar inte
användbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
datasäkerhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prestanda	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
skalbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stabilitet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
expanderbarhet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Frågor i skede 2

**Fråga 1.3 (0–4 poäng)** Nedan finns en mängd påståenden som hör till den tidigare textens innehåll. Betrakta dem med tanke på texten och ange för varje påstående om det är sant eller falskt.

**Poängsättning:** För varje påstående ges  $1/2$  poäng för rätt val och  $-1/2$  poäng för felaktigt val; för alternativet ”svarar inte” ges 0 poäng.



## Urvalsprov i datavetenskap, våren 2021

---

	sant	falskt	svarar inte
Kraven bör under hela kravspecifikationen beskrivas med samma exakt definierade representationsform.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kraven bör inte längre ändras, när programvaran har börjat förverkligas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Att ta i beaktande lagstadgade krav är en del av kravhanteringen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Till valideringen av kraven hör att försäkra sig om att kraven kan förverkligas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Den som beställer programvaran bör leverera en kravspecifikation, som beskriver programvarans alla önskade egenskaper.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Även de icke-funktionella kraven bör kunna verifieras.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Under kravspecifikationen klargörs i vilken driftsmiljö det är meningen att programvaran som utvecklas ska fungera.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Att intervjua programvarans framtida användare är ett sätt att få reda på kraven.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

*Svara på fråga 1.4 både på basen av texten och dina egna kunskaper.*

**Fråga 1.4 (0–6 poäng)** Föreställ dig en situation med en restaurang som införskaffar en programvara för att stöda beställning och leverans av matportioner. Med programvaran kan kunden beställa en matportion till önskad adress och matbudet levererar den dit. Vi granskar tre användarroller: kund,

## Urvalsprov i datavetenskap, våren 2021

---

matbud och kock. Ange ett funktionellt krav ur varje användarrolls synvinkel för den programvara som ska utvecklas. Ange dessutom tre icke-funktionella krav på programvaran. De olika kraven ska tydligt avvika från varandra. Ange varje krav med ungefär en mening. Kraven bör följa de egenskaper för bra krav som presenteras i texten.

**Poängsättning:** För varje krav kan man få 1, 1/2 eller 0 poäng enligt hur bra det givna kravet motsvarar uppdraget och följer egenskaperna på bra krav.