



Handelshögskolan  
VID GÖTEBORGS UNIVERSITET  
Institutionen för informatik  
2003-05-30

# Dokumentation som undervisningsmoment i utbildningar inom systemutveckling

## Abstrakt

Problemområdet behandlade vilken betydelse undervisningen inom systemutvecklingsutbildningar hade för universitetsstudenters syn på dokumentationsarbete. Uppsatsen hade sin utgångspunkt hos AstraZeneca. Syftet var att utifrån intervjuer med lärare, studenter och utifrån genomgång av kurslitteratur beskriva undervisning gällande dokumentationsarbete samt att analysera undervisningens betydelse för studenternas syn på detsamma. Undersökta utbildningar var Systemvetarprogrammet vid institutionen för Informatik samt Datateknik- och Elektroteknikprogrammet på Chalmers i Göteborg. Vi tolkade att utbildningens betydelse för studenternas syn på dokumentation b.l.a. var att studenterna förmedlades en bild av dokumentation som viktig. Studenterna hade således en syn på dokumentation som viktig teoretiskt sett. Gästföreläsare och övningsuppgifter med förankring i näringslivet och som berörde näringslivets dokumentationsförfarande, var av betydelse för studenternas syn på dokumentation då det kunde höja motivationen hos studenterna. Även feedback hade betydelse för studenternas syn på dokumentationsarbete.

Nyckelord: Dokumentation, dokumentationsarbete, systemutveckling, utbildning och undervisningsmoment

Författare: Ann-Charlott Larsson & Magnus Persson  
Handledare: Mathias Klang  
Examinator: Kjell Engberg  
Magisteruppsats, 20 poäng

# Innehållsförteckning

<b>1 INLEDNING</b>	<b>1</b>
1.1 BAKGRUND	1
1.2 PROBLEMDISKUSSION	4
1.3 FRÅGESTÄLLNING	6
1.4 SYFTE	7
1.5 PROBLEMAVGRÄNSNING	7
1.6 DEFINITION AV DOKUMENTATION	8
<b>2 TEORETISK REFERENSRAM</b>	<b>12</b>
2.1 SYN PÅ DOKUMENTATION OCH DESS ROLL I SYSTEMUTVECKLINGSPROCESSEN	12
2.1.1 DOKUMENTATIONEN TILLHÖR SYSTEMET	12
2.1.2 DOKUMENTATIONENS BETYDELSE FÖR KVALITET	13
2.1.3 DOKUMENTATIONENS SYFTE OCH DESS ROLL I SYSTEMUTVECKLINGSPROCESSEN	13
2.2 HELHETSSYN OCH SYSTEMUTVECKLINGSPROCESSEN	15
2.2.1 DOKUMENTATIONSARBETE OCH MOTIVATION	15
2.2.2 HELHETSSYN, KOMMUNIKATION OCH REFLEKTION VID KOMPLEX PROBLEMLÖSNING	17
2.3 LÄRANDE OCH UTBILDNING	19
2.3.1 DISKUSSION KRING YRKESVERKSAMMAS KUNSKAPER OCH FÖRMÅGOR	19
2.3.2 UTBILDNINGSSINNEHÅLL, UTFORMNING OCH FEEDBACK	20
<b>3 METOD</b>	<b>24</b>
3.1 VAL AV ANGREPPSSÄTT OCH METOD	24
3.2 UNDERSÖKNINGSOMRÅDE	25
3.2.1 ASTRAZENECA	26
3.2.2 INFORMELL FÖRSTUDIE PÅ ASTRAZENECA	26
3.3 LITTERATURGRANSKNING	26
3.4 DET EMPIRISKA MATERIALET	27
3.4.1 BESKRIVNING AV SYSTEMVETARPROGAMMET	29
3.4.2 BESKRIVNING AV CHALMERSUTBILDNINGARNA	29
3.5 DATAINSAMLING OCH URVAL	30
3.5.1 TILLVÄGAGÅNGSSÄTT VID DATAINSAMLING	30
3.5.2 URVAL AV KURSLITTERATUR	31
3.5.3 URVAL AV INTERVJUPERSONER	31
3.6 ANALYS OCH TOLKNING	33
3.7 TROVÄRDIGHET	34
3.8 METODDISKUSSION OCH SJÄLVKRITIK	34

<b>4 RESULTAT</b>	<b>36</b>
<b>4.1 SYN PÅ DOKUMENTATION OCH DESS ROLL I SYSTEMUTVECKLINGSPROCESSEN</b>	<b>36</b>
4.1.1 VAD INTERVJUPERSONERNA MENADE MED BEGREPPET DOKUMENTATION	36
4.1.2 INTERVJUPERSONERNAS SYN PÅ DOKUMENTATION I SAMBAND MED SYSTEMUTVECKLING	37
4.1.3 LÄRARNAS ERFARENHETER FRÅN NÄRINGS LIV OCH UNIVERSITET	40
<b>4.2 HELHETSSYN OCH SYSTEMUTVECKLINGSPROCESSEN I UTBILDNINGARNA</b>	<b>40</b>
4.2.1 LÄRARNAS SYN PÅ UTBILDNINGENS FOKUS OCH INNEHÅLL	41
4.2.2 STUDENTERNAS SYN PÅ UTBILDNINGENS DELAR	42
<b>4.3 LÄRANDE OCH UTBILDNING</b>	<b>44</b>
4.3.1 VAD SOM LÄRS UT OCH VAD STUDENTERNA TYCKER DE HAR LÄRT SIG	44
4.3.2 UPPFATTNINGAR KRING OLIKA UNDERVISNINGSMOMENT	46
4.3.3 UPPFATTNINGAR KRING FEEDBACK	49
4.3.4 UPPFATTNINGAR KRING KURSLITTERATUREN	51
<b>4.4 RESULTAT FRÅN GENOMGÅNG AV KURSLITTERATUR</b>	<b>52</b>
<b>5 ANALYS OCH DISKUSSION</b>	<b>67</b>
<b>5.1 DISKUSSION KRING SYN PÅ DOKUMENTATION OCH DESS ROLL</b>	<b>67</b>
<b>5.2 DISKUSSION OM HELHETSSYN PÅ SYSTEMUTVECKLINGSPROCESSEN</b>	<b>70</b>
<b>5.3 DISKUSSION OM LÄRANDE OCH UTBILDNING</b>	<b>72</b>
<b>5.4 DISKUSSION KRING KURSLITTERATUREN</b>	<b>76</b>
<b>5.5 SAMMANFATTNING AV ANALYS OCH DISKUSSION</b>	<b>77</b>
<b>5.6 SJÄLVREFLEKTIONER SAMT FÖRSLAG TILL FORTSATT FORSKNING</b>	<b>79</b>
<b>6 SLUTSATS</b>	<b>81</b>
<b>REFERENSER</b>	<b>83</b>
<b>BILAGA 1. MALL FÖR INTERVJUUNDERLAG TILL LÄRARE</b>	<b>86</b>
<b>BILAGA 2. MALL FÖR INTERVJUUNDERLAG TILL STUDENTER</b>	<b>87</b>
<b>BILAGA 3. LITTERATURLISTA ÖVER UNDERSÖKTA BÖCKER</b>	<b>88</b>

## Figur och tabellförteckning

<b>Figur 3:1</b>	Vår syn på angreppssätt på metoden. Omarbetad efter Mason (1989).....	24
<b>Figur 3:2</b>	Undersökningens arbetsgång.....	25
<b>Figur 3:3</b>	Empiriskt material med avseende på undersökt område och urval.....	28
<b>Tabell 3:1</b>	Antal tillfrågade och antal intervjuade.....	32
<b>Tabell 3:2</b>	Bortfallsanalys av studenterna på Elektroteknikprogrammet.....	32
<b>Tabell 4:1a</b>	Gemensamma böcker Systemvetarprogrammet.....	54
<b>Tabell 4:1b</b>	Forts. gemensamma böcker Systemvetarprogrammet.....	55
<b>Tabell 4:2</b>	Böcker för Systemvetarprogrammet, inriktning Business Informatics.....	56
<b>Tabell 4:3</b>	Böcker för Systemvetarprogrammet, inriktning IT Management.....	57
<b>Tabell 4:4a</b>	Forts. böcker för Systemvetarprogrammet, inriktningning Systems, Design and Engineering.....	58
<b>Tabell 4:4b</b>	Forts. böcker för Systemvetarprogrammet, inriktningning Systems, Design and Engineering.....	59
<b>Tabell 4:4c</b>	Forts. böcker för Systemvetarprogrammet, inriktningning Systems, Design and Engineering.....	60
<b>Tabell 4:4d</b>	Forts. böcker för Systemvetarprogrammet, inriktningning Systems, Design and Engineering.....	61
<b>Tabell 4:5</b>	Bok i kursen Software Engineering, Systemvetarprogrammet.....	62
<b>Tabell 4:6a</b>	Forts. böcker som läses/kan läsas på D- och/eller E-programmet.....	63
<b>Tabell 4:6b</b>	Forts. böcker som läses/kan läsas på D- och/eller E-programmet.....	64
<b>Tabell 4:6c</b>	Forts. böcker som läses/kan läsas på D- och/eller E-programmet.....	65
<b>Tabell 4:6d</b>	Forts. böcker som läses/kan läsas på D- och/eller E-programmet.....	66

## 1 Inledning

*Den här uppsatsen behandlar vilken betydelse undervisningen har inom systemutvecklingsutbildningar för universitetsstudenters syn på dokumentationsarbete. Undersökningen genomförs i samarbete med AstraZeneca som upplever ett behov att underlätta så långt det är möjligt vid dokumentationsarbetet i sin verksamhet. Företaget har därför ett intresse av att undersöka hur utbildningar som härrör till systemutveckling hanterar ämnet dokumentationsarbete under utbildningarna samt vilken betydelse detta får på studenters syn på dokumentation. Anledningen är att företaget bättre vill kunna planera sitt framtida dokumentationsarbete och sin internutbildning utifrån den kunskapsnivå studenterna har som AstraZeneca rekryterar.*

*Detta kapitel inleds vidare med en bakgrundspresentation för att beskriva hur tidigare forskning ser på dokumentation och därefter följer en problemdiskussion som mynnar ut i det valda problemområdet. Sist följer uppsatsens syfte, avgränsningar och en definition av det centrala begreppet dokumentation.*

### 1.1 Bakgrund

Procaccino et al. (2002) menar att trots att många undersökningar genomförts på området varför systemutvecklingsprojekt misslyckas, så händer detta fortfarande i många mjukvaruprojekt. Författarna hänvisar till olika undersökningar vars siffror pekar på att av systemutvecklingsprojekt misslyckas mellan 20 % - 85 %, på ett eller annat sätt.

Ett datorsystem kan anses vara misslyckat på grund av stora förseningar, överskridna kostnadsbudgetar samt att det inte upplevs som ett drifts- eller funktionsdugligt system (Poulymenakou & Holmes, 1996). Procaccino et al. (2002) menar att misslyckandena kan ha att göra med projektledningen, dålig kravspecifikation, dålig tidsuppskattning samt att det kan bero på brister i utvecklingsprocessen. Även Poulymenakou och Holmes (1996) nämner olika förklaringar till varför datorsystem misslyckas. En förklaring är att trots att människor är en av de största komponenterna i utvecklingsprocessen av ett system så kommer den mänskliga faktorn i form av arbetssätt, organisation och kultur på arbetsplatsen ofta i andra hand efter ett tekniskt fokus. En förklaring till olika problem vid utvecklandet eller vid förvaltandet av mjukvaran är Viscontis (1994) förklaring som menar att huvudorsakerna till detta är beroende på dålig dokumentation, ouppdaterad dokumentation eller att dokumentation saknas.

Sommerville (2001) menar i likhet med Poulymenakou och Holmes (1996) att den mänskliga faktorn är av betydelse i systemutvecklingsprocessen. Det finns ett starkt samband mellan kvaliteten på den utvecklade mjukvaruprodukten och kvaliteten på processen som använts för att skapa produkten, menar Sommerville (2001) och framhäver människans betydelse i denna process. Mjukvarukvaliteten är inte beroende av en mekanisk tillverkningsprocess utan är beroende av en designprocess där individuella mänskliga överväganden är signifikanta. I vissa fall är de involverade människorna mer betydelsefulla än processen som används, menar Sommerville (2001), vilket bidrar till att den mänskliga faktorn i dokumentationsarbetet är av intresse för AstraZeneca att

undersöka. Mjukvarans produktkvalitet påverkas främst av processkvalitet, kostnader, tidsschema, den mänskliga kompetensens kvalitet och utvecklingsteknologi. För stora projekt är systemutvecklingsprocessen den mest bestämmande faktorn då teammedlemmar spenderar stor del av tiden med att kommunicera och förstå andra delar av systemet. Kommunikationen är den dominerande faktorn som styr deras produktivitet (Sommerville, 2001). Bunse et al. (1998) menar i likhet med Sommerville (2001) att mjukvarans kvalitet beror på hur systemutvecklingsprocessen fungerat och tar även med dokumentationen som en faktor som underlättar arbetet och påverkar resultatet. Processdokument och modeller underlättar nämligen kommunikationen mellan utvecklare under processens gång och hjälper personer med varierande specialistroller i ett systemutvecklingsprojekt genom att utgöra en bas utifrån vilken en gemensam förståelse kan skapas. Mathiassen et al. (2001) menar i likhet med detta att i en objektorienterad utvecklingsprocess är dokumentationens roll att bidra till att skapa sammanhang under hela systemutvecklingsprocessen. Dokumentationens syfte är att fungera som verktyg för dokumentation av systemkrav, design och dokumentation av överenskommelser. Dokumentationens syfte bör även vara att samla ihop och strukturera delresultat samt att fungera som styrverktyg över arbetets framskridande.

De senaste åren har enligt Sommerville (2001) intresset för processförbättring ökat inom software engineering<sup>1</sup> då detta kan leda till förbättrad produktkvalitet och/eller reducera kostnader och utvecklingstiden. Att sträva efter att förbättra arbetsprocesser är något som aktualiseras i denna uppsats då detta kan sägas intressera AstraZeneca gällande dokumentationsarbetet. Mjukvaruprocesser är dock komplexa och det är inte möjligt att optimera alla processer samtidigt. Krävs exempelvis en snabb utvecklingsprocess kan det vara nödvändigt att reducera processens visibility. Att göra en process visible innebär att skapa dokument vid regelbundna intervall för att reflektera över vad som producerats, vilket således kan vara ett moment som tas bort när snabbheten i leveransen prioriteras. Anledningen till det är att dokumentproduktion kan sakta ner processen och ryms kanske därför inte inom det tidsintervall som står till förfogande. Kvalitetsgranskning som kan ses som ett annat sätt att se på leverans, (jämfört med att prioritera snabbhet i leveransen) handlar om processgranskning och produktkomponentgranskning där granskning av skapade dokument även ingår. Det är en metod för validering av kvaliteten i processen eller produkten. Granskningens syfte är att genomföra en teknisk analys av dokumentationen för att finna eventuella brister i matchningen mellan specifikationen och designen, koden och dokumentationen (Sommerville, 2001). Denna typ av längre process med noggrann kontroll är en del av AstraZenecas dokumentationshantering.

Sommerville (2001) representerar ett synsätt som menar att likställandet av termen mjukvara med datorsystem är en alltför restriktiv syn. Mjukvaran är inte enbart programmen utan även all dokumentation och konfigurationsdata som behövs för att dessa program ska kunna drivas korrekt. Mjukvarusystem består oftast av ett antal separata program, konfigurationsfiler, systemdokumentation och användardokumentation. En viktig aspekt hos mjukvaran är att det ska möjliggöra underhåll, d.v.s.

---

<sup>1</sup> Ett flertal ord i uppsatsen har vi valt att behålla i sin engelska form då vi främst inte vill riskera att göra en felaktig översättning. Vissa ord anser vi har en specifik innebörd bland de särskilda grupper av människor där orden främst används och kan därför inte med fördel översättas.

mjukvaran ska vara skriven på ett sådant sätt att den kan förändras och möta kundens förändrade behov. Humphrey (2000) ger uttryck för en liknande syn genom att mena att dokumentation är en essentiell del av varje mjukvaruprodukt och är i många fall även viktigare än själva programkoden. Humphrey (1998) menar att källkoden som sådan inte är tillräckligt tydlig för att användas som enda dokumentationsmaterial. Om källkoden används som enda dokumentation och den då är mycket lite kommenterad är den dessvärre otillgänglig för någon annan än designers och implementörer (Humphrey, 1998). Polak (2002) menar att dokumentation är lika viktig som kod då dokumentation bidrar till att systemet är konsistent och underlättar underhåll.

Dokumentation i ett systemutvecklingsprojekt är också viktigt, menar Sommerville (2001) eftersom dokument är det enda uttalade, konkreta sättet att representera mjukvaran och mjukvaruprocessen på. Dokumentation kan sålunda ses vara en modell för att representera och förklara mjukvaran eller systemet, vilket liknar Dawsons (2003) tankar kring modeller som syftar till att representera abstrakta artefakter. Modeller kan appliceras på fenomen som människor har svårt att förstå och förklara. Men i försöket att avbilda ett stort system genom dokumentation kan överblickbarheten gå förlorad p.g.a. att dokumentationen kan bli för komplex. Detta kan förklaras med något som inom datorsimulering kallas Bonini's paradox och som innebär att modellen blir lika svår och komplex att förstå som det fenomen den skapats i syfte att förenkla och förklara.

Mathiassen et al. (2001) skriver att ett bra dokument kännetecknas av klarhet och elegans. Att beakta dessa begrepp är ett sätt att skapa överblickbarhet och relevans i dokumentationen. Elegans handlar om relevans, d.v.s. att dokumentet ska vara ändamålsenligt och lösa det problem det avser att lösa. Klarheten å andra sidan handlar om att dokumenten ska vara lätta att läsa och denna klarhet kan ökas t.ex. genom att samma term används för samma fenomen. Mathiassen et al. (2001) beskriver standarddokument när det handlar om objektorienterad analys och design. Författarna menar att för att uppnå denna klarhet och elegans kan dokumentation göras till standardiserade dokument som syftar till att skapa ett enhetligt dokumentationsinnehåll. Detta kan sedan fungera som gemensam referensram för inblandade utvecklare. Även Sommerville (2001) talar om standardiserade dokument vars syfte är att ha en konsistent struktur, ett konsistent utseende samt konsistent kvalitet. Dessa dokument ska vara lättare att läsa och förstå än dokumentation som inte är standardiserad. AstraZeneca använder sig av ett standardiserat dokumentförfarande, med liknande syfte som Sommerville (2001) och Mathiassen et al. (2001) nämner. Mathiassen et al. (2001) beskriver mallar som bör användas och som schematiskt definierar dokumenten för att underlätta orientering i dessa, samt för att klargöra det som beskrivits. Mallarna för standarddokumenten kan användas som ett slags checklistor för vilka dokument som måste finnas i slutet på designen.

Över ett systems livstid kommer systemet att behöva ändras i takt med förändrade krav eller i samband med att ny hårdvara introduceras. Dokumentation som är skapad och som uppdateras och förändras på ett ändamålsenligt sätt kan vara en stor hjälp vid underhåll av ett system. I underhållsfasen är det därför viktigt att dokumenten är tydliga och läsbara, menar Sommerville (1996), vilket standardiserad dokumentation kan bidra till.

Detta nämner även Humphrey (1998) i ett avsnitt om mjukvarukvalitet där dokumentationen beskrivs som en del av produkten när den levereras och att den därför måste vara kontrollerad så att den kan förstås. En annan aspekt som nämns i samband med mjukvarukvalitet är kostnader i samband med olika typer av fel. Diagnostisering av ett problem kostar, liksom att göra nödvändiga reparationer samt för att komma tillbaka i drift. Att förebygga problem kostar också som t.ex. kostnader i samband med att identifiera källorna till felaktigheterna samt åtgärder för att förebygga dessa. Att ändra dokumentationen behövs för att spegla och reflektera det som rättats till i systemet. Målet bör vara att ta bort felaktigheter från krav, design och från koden så snart dessa upptäcks, vilket reducerar risken för dyrbara omarbeten.

Angående den tid som kan spenderas i systemutvecklingsprojekt på dokumentation menar Brown (2002) i likhet med Booch, som refereras i Brown (2002), att företag inte bör dokumentera i onödan. Företag bör inte spendera mer än mellan 5 % och 10 % av den totala utvecklingsinsatsen på all form av dokumentation. Detaljerad dokumentation som Booch (refererad i Brown, 2002) beskriver som high-ceremony, är endast nödvändigt då människors liv är beroende av mjukvaran, t.ex. kontrollsystem för flygplan och självguidande fordon samt övervaknings mjukvara för patienter, faller inom denna ram. Det senare understryks även av Pradhan (1996) som i sin bok om feltoleranta system menar att tillförlitlighet (som dokumentationen är en del av) är kritiskt viktig i situationer där ett datorhaveri kan få katastrofala resultat som t.ex. i sjukhusmiljö med patientövervakande system.

## **1.2 Problemdiskussion**

Studien som presenteras i denna uppsats har sin utgångspunkt i AstraZenecas läkemedelsindustri inom Research and Development (RD) i Mölndal. AstraZenecas produktutveckling, produktion och försäljning av läkemedel sker i skilda enheter, där i sin tur produktutvecklingen är uppdelad på olika så kallade "siter" efter bestämda forsknings- eller sjukdomsområden. Varje forsknings-site (RD) är i sin tur huvudsakligen uppdelad i en del med laborativ forskning (Discovery) och en del med klinisk forskning (Development). För varje laborativ- respektive klinisk forskningsdel finns en IT-funktion, vilka kallas Discovery IS och Development IS (AstraZeneca, personlig kommunikation, 6 maj, 2003).

Inom forskningssiten RD i Mölndal bedrivs medicinsk forskning inom mage/tarm och hjärt/kärlsjukdomar. Forskningen styrs och kontrolleras delvis med hjälp av mjukvara. Kraven på processkontroll av läkemedelsframtagningar är höga från olika myndigheters sida ur laglig, etisk och hälsorisksynpunkt. Kraven på processkontroll omfattar även systemutvecklingsprocessen och eftersom datasystem används skall systemen vara dokumenterade och testade. Dokumentationens syfte är bl.a. att utgöra ett bevis som med hög säkerhet visar att systemen fungerar och kommer att fungera konsekvent och korrekt enligt gällande specifikationer och kvalitetskrav. Myndigheterna utövar sin kontroll dels genom studium av de produktbeskrivande dokument AstraZeneca lämnar in, dels genom att myndigheten kan genomföra inspektioner på företagets anläggningar. Varje lands myndighet, där AstraZeneca begärt att få sälja sina mediciner, kan genomföra en sådan inspektion. Gruppen Quality Management (QM) arbetar med IT-kvalitetssäkring inom



AstraZeneca för att bl.a. ligga i fas med externa kontroller, vilket även innefattar kvalitetskontroller av dokumentationen (AstraZeneca, personlig kommunikation, 6 maj, 2003).

Två avdelningar inom AstraZenecas RD utgör således utgångspunkten för problemområdet, Development IS samt Discovery IS, där frågor väckts kring dokumentationsförfarandet genom funderingar hos medarbetare på företaget. En tidigare magisteruppsats (Alexandersson & Johansson, 2002) som skrivits på Göteborgs Universitet genomförde en undersökning som belyste vissa aspekter av dokumentationsarbetet på företaget. Några aspekter som framkom av undersökningen var att inblandade personer när de arbetade med dokumentation hade svårt att uttrycka sig när de skulle dokumentera. Det upplevdes som att det kunde vara svårt att finna en lämplig språklig nivå i dokumentationen då det fanns en osäkerhet kring läsarens mottagande av innehållet. Det upplevdes även kring feedback att den var otydlig, inkonsekvent och inte omedelbar då det tog lång tid innan feedback gavs. I uppsatsen framkom även att dokumentationsarbetet av vissa upplevdes som mindre motiverande och hämmande av kreativiteten. Undersökningen antydde även att en lägre förståelse för dokumentationens roll i systemutvecklingsarbetet kunde bero på bristande kunskaper inom dokumentationsarbete (Alexandersson & Johansson, 2002).

Sammantaget, utifrån ovannämnda anledningar, skapades ett intresse från AstraZenecas sida för innehållet i systemutvecklingsinriktade universitetsutbildningar varifrån delar av AstraZenecas personal rekryteras. AstraZeneca upplever ett behov att underlätta så långt det är möjligt vid dokumentationsarbetet för involverade personer och för sin verksamhet. Företaget har därför ett intresse av att undersöka hur utbildningar som härrör till systemutveckling hanterar dokumentationsarbete under utbildningarna för att företaget bättre ska kunna planera sitt framtida dokumentationsarbete och internutbildning samt för att i förlängningen kunna förbättra sin arbetsprocess kring dokumentation. Vår undersökning utformas som en fortsättning utifrån Alexandersson och Johanssons (2002) framkomna resultat från intervjuer med systemutvecklare på AstraZeneca. I vår uppsats ämnar vi fokusera på vad studenter lär sig om dokumentation då studenterna i viss mån utgör underlaget utifrån vilket AstraZeneca sedan rekryterar. Systemutvecklingsutbildningar samt studenternas syn på dokumentationsarbete är centralt i vår undersökning.

Vår undersökning kan knytas till tidigare forskning kring utbildning inom systemutveckling och software engineering på så sätt att vi i likhet med nedan referenser diskuterar utbildningarnas innehåll, utformning och industrirelevans. Enligt Wohlin och Regnell (1999) är t.ex. det viktiga i utbildningarna inom software engineering på universitetet att dessa har en industrirelevans på så sätt att utbildningarna förbereder studenterna för att hantera storskalig systemutveckling. Studenterna måste förberedas genom att få undervisning inom kravspecifiering, processförbättring, testning av mjukvara och kvalitetsförsäkring för att nämna några av de behov de uttrycker att industrin har. Humphrey (2000) skriver att den ökade komplexiteten i systemutvecklingsprocessen och kravet från industrin på bättre kvalificerade och bättre förberedda studenter borde innebära att läroplaner inom systemutveckling måste ge kunskap och erfarenhet som är relaterad till tillämpningen av ämnesområdet.

En undersökning utförd av Lethbridge (2000) uttrycker uppdaterade behov från industriellt håll jämfört med Wohlin och Regnell (1999). Lethbridge (2000) slår fast i denna undersökning att universitet inom utbildningar för informationsteknologi, kan förbättra sina kurserbjudanden genom att lägga till utbud som berör software design, arkitektur, användargränssnitt, projektstyrning, presentationsteknik och tekniskt skrivande. Ett flertal s.k. ”mjuka färdigheter” hamnade högt i undersökningen, som förmågan att ge bra presentationer, förmågan till teknisk skrivning samt ämnen som etik och professionalism. Dessa ämnen borde utifrån industrins behov därför utökas inom utbildningar, menade Lethbridge (2000).

Inom tidigare forskning har det således föreslagits förändringar inom utbildningar av software engineers samt andra utbildningar inom informationsteknologi. Det kan tyda på att det har funnits och finns en diskussion om vad som lärs ut på universiteten samt på vilket sätt det lärs ut. Ovannämnda forskare pekar på behov av kunskap inom ”mjukare färdigheter” som processförbättring och kvalitetsförsäkring, vilket vi menar att dokumentation kan ses vara en del av. Budskapet från forskarna (Humphrey, 2000; Lethbridge, 2000; Wohlin & Regnell, 1999) är även att utbildningar bör präglas av ett samarbete mellan industri och akademi, vilket även AstraZeneca efterlyser och vill undersöka.

Vårt problemområde kommer utifrån ovannämnda resonemang beröra utbildningar inom systemutveckling och kring hur dokumentationsarbete hanteras, samt vilken syn på dokumentation detta kan ge studenterna. Studenternas syn på dokumentation är av betydelse för AstraZeneca i bemärkelsen vad nyutexaminerade studenter i form av kunskap och uppfattningar kan föra med sig in i företaget. Problemområdet har en industriell relevans för AstraZenecas del, men kan även vara av intresse för andra företag som, liksom AstraZeneca, lägger stor vikt vid dokumentation under systemutvecklingsprocessen.

Vi tror att utbildningen kan säga något om den syn på och den förståelse studenterna har av dokumentation som arbetsuppgift. Vi är medvetna om att det kan finnas flera faktorer som påverkar studenternas syn, men vi ville undersöka vilken syn på dokumentation som grundläggs i utbildningen och vilken betydelse detta kan ha för synen på dokumentation som arbetsuppgift. Studenternas syn på och förståelse av dokumentation som arbetsuppgift kan således säga något om studenternas kunskaps- och motivationsnivå angående uppgiften, vilket kan få betydelse för internutbildningar samt dokumentationsarbetet som arbetsprocess med avseende på effektivitet och utformning. Därför är synen på dokumentationsarbete intressant för AstraZeneca.

### **1.3 Frågeställning**

*Vilken betydelse har undervisningen inom systemutvecklingsutbildningar för studenternas syn på dokumentation?*

## 1.4 Syfte

Syftet är att utifrån intervjuer med lärare och studenter samt utifrån genomgång av kurslitteratur beskriva systemutvecklingsundervisning gällande dokumentationsarbete. Syftet är även att analysera undervisningens betydelse för studenternas syn på dokumentationsarbete. Vi önskar likaså bidra med ett informationsunderlag till AstraZeneca och andra företag som rekryterar nyutexaminerade studenter från Göteborg. Denna information syftar till att bidra till bättre planering av framtida dokumentationsarbete och internutbildningar.

## 1.5 Problemavgränsning

Vår undersökning innebär genomförande av intervjuer med lärare och studenter samt strukturerad genomgång av kurslitteratur. De utbildningar vi undersökt är Systemvetarprogrammet inom institutionen för Informatik i Göteborg och två av Chalmers civilingenjörsprogram, Datateknikprogrammet med inriktning mot systemutveckling samt Elektroteknikprogrammet. Elektroteknikprogrammets inriktning som vi undersökt kallas Dator- och programsystem. Av de på AstraZeneca som arbetar som systemutvecklare är den största delen utexaminerade från Systemvetarprogrammet vid Göteborgs universitet och från Chalmers Datateknik- och Elektroteknikprogram. Vårt urval av studenter, lärare och kurslitteratur har vi därför valt från dessa utbildningar i Göteborgsområdet för att detta ger en bild av undervisningen vid de utbildningar där AstraZeneca till störst del rekryterar sin personal. Andra universitetsutbildningar i landet finns också representerade hos AstraZeneca, men inte i lika stor omfattning.

Vi väljer att genomföra undersökningen samt besvara frågeställningen med hjälp av tre teman. Dessa tre teman återkommer i vår teoretiska referensram, resultatet samt i analys och diskussion och utgörs av:

Syn på dokumentation och dess roll i systemutvecklingsprocessen  
Helhetssyn och systemutvecklingsprocessen  
Lärande och utbildning

Det första temat kallas *Syn på dokumentation och dess roll i systemutvecklingsprocessen*. Synen på dokumentation och dess roll i utvecklingsprocessen tror vi uttrycker vilken betydelse denna uppgift anses ha för utvecklandet av ett system eller ett program. Den syn lärarna uttrycker menar vi påverkar utbildningen liksom den kurslitteratur kan göra som valts ut som material. Den syn studenterna har på dokumentation och den syn de anser att universitetet förmedlar tror vi har betydelse för studenternas uppfattning kring dokumentationsarbete. Studenternas syn är av betydelse att undersöka för att synen på dokumentation som arbetsuppgift kan hänga samman med studenternas kunskapsnivå inom området. Kunskaps- och motivationsnivån kan få betydelse för internutbildningar samt dokumentationsarbetet som arbetsprocess för AstraZenecas del eller för andra företag vars dokumentationsarbete liknar AstraZenecas i t.ex. omfattning. Temat avser att belysa vad författare, forskare, lärare, studenter och kurslitteratur menar med dokumentation, dess syfte samt dess roll i systemutvecklingsprocessen.

Temat *Helhetssyn och systemutvecklingsprocessen* har uppstått ur våra reflektioner kring tidigare forskning. Även AstraZenecas uppmärksammade upplevelser av dokumentationsarbetet som mindre motiverande och hämmande av kreativiteten (Alexandersson och Johanssons, 2002) har bidragit till detta temas uppkomst. Med temat *Helhetssyn och systemutvecklingsprocessen* ville vi undersöka vilken betydelse i utbildningen en eventuell närvaro/frånvaro av en helhetsbild av systemutvecklingsprocessen kan få på studentens syn på dokumentationsarbete. En helhetssyn på systemutvecklingsprocessen är intressant att undersöka eftersom vi tror att en helhetssyn kan ha betydelse för förståelsen för dokumentation som arbetsuppgift och för dess roll i systemutvecklingsprocessen. Avsaknad av en helhetsbild på systemutvecklingsprocessen i utbildningen tror vi kan minska förståelsen för dokumentation som arbetsuppgift, vilket i sin tur kan påverka det som upplevs som mindre motiverande med uppgiften.

Temat *Lärande och utbildning* är ett tema som vi valt utifrån forskares teoretiska funderingar kring utbildningsfrågor. Tidigare forskning ger i referensramen tillhörande detta tema uttryck för tankar kring utbildningsfrågor som t.ex. utbildningars innehåll, utformning, syfte och mål. Den teoretiska referensramen inleds med en redogörelse för olika forskares syn på relationen industriellt behov och akademiskt utbud i utbildningshänseende. Den teoretiska referensramen tar upp resultat från undersökningar som inte enbart speglar svenska förhållanden, men som däremot kan spegla det aktuella läget inom branschen. Detta läge överensstämmer i viss mån med AstraZenecas upplevelser; nämligen en upplevd diskrepans mellan industriellt behov av duktiga yrkesverksamma inom systemutveckling och vad akademien lär ut inom detta område. Detta aktualiserar även ett intresse från AstraZenecas sida för vad som lärs ut inom dokumentationsområdet. Funderingarna från den teoretiska referensramen använder vi sedan för att förstå empirin för att kunna svara på frågan vilken betydelse utbildningen kan ha för studenternas syn på dokumentation med avseende på utbildningarnas innehåll och form.

AstraZenecas dokumentationsförfarande, med avseende på rutiner och processer kring skapandet av olika dokumenttyper, kommer inte att behandlas i denna uppsats då vi inte anser det nödvändigt att närmare förklara deras tillvägagångssätt. Uppsatsen har visserligen sin utgångspunkt i AstraZenecas dokumentationsförfarande och upplevda funderingar kring detsamma, men det är inte inom företaget i sig undersökningen utförts eftersom scenen för frågeställningen är utanför företaget. Vi har således inte för avsikt att på något sätt utvärdera AstraZenecas nuvarande dokumentationshantering utan ämnar ge företaget en bild av de utbildningar varifrån företaget i nuläget gör viss rekrytering till sin personalstyrka.

## **1.6 Definition av dokumentation**

En entydig definition av dokumentation som begrepp är svårare att finna då dokumentation innebär olika dokumenttyper och hanteringsförfarande för olika företag och organisationer. Begreppet kan därför upplevas som något vagt, olika personer ger begreppet olika innebörd utifrån skilda preferenser och utifrån vilket syfte denna person uppfattar att dokumentationen har. I denna uppsats upplever vi dock att vi bör ha någon form av förhållningssätt och arbetsdefinition och ämnar därför presentera ett par

definitioner som vi förhåller oss till. Dokumentation kan också ha många syften, på olika nivåer i ett företag, som exempelvis i AstraZenecas fall. Syftet med dokumentationen kan även vara föränderligt, vilket också påverkar dess innebörd, som vi ser det. Vi ämnar inte beskriva alla tänkbara syften eller ”grader” av dokumentationshantering som finns beskrivet på olika sätt, utan i denna uppsats kommer vi att hålla oss till ett visst syfte, främst med utgångspunkt från AstraZenecas användande av dokumentation inom systemutveckling. Dokumentationens syfte samt inkluderade dokument som AstraZeneca använder, kommer presenteras nedan utifrån AstraZeneca (personlig kommunikation, 9 maj, 2003).

Nationalencyklopedin (1991) definierar dokumentation inom databehandling som: ”... en skriftlig presentation av ett program, avsedd att underlätta användning, underhåll och förståelse av programmet.” (S. 70)

Detta innefattar kärnan i det vi menar med dokumentation men för att utveckla begreppet för att skapa en djupare förståelse för vad vi menar, hänvisar vi även till Sommerville (2001) och Brookshear (1997). Dessa båda författare beskriver i grunden dokumentation på ett liknande sätt som Nationalencyklopedin (1991). En skillnad är dock att dessa författare mer definierar utifrån vilka dokument som kan användas i ett systemutvecklingsprojekt. Sommerville (2001) och Brookshear (1997) utvidgar således Nationalencyklopedins (1991) definition och delar upp dokumentationen i två huvudgrupper; systemdokumentation och användardokumentation. Med systemdokumentation menar de dokumentation som beskriver hur systemet är uppbyggt, medan användardokumentationen beskriver hur systemet ska användas.

Projektdokumentation är något som AstraZeneca använder sig av, men det anses vara skilt från systemdokumentationen. Detta överensstämmer med Sommersvilles (2001) och Brookshears (1997) beskrivningar av system- och användardokumentation, där projektdokumentationen inte heller anses vara en del av systemdokumentationen eller användardokumentationen. Sommerville (1996) nämner dock att projektdokumentation behövs vid systemutveckling.

En definition som påminner om Sommerville (2001) och Brookshears (1997) definition av möjliga dokument i ett systemutvecklingsprojekt är Encyclopedia of computer science (London, 2000) som delar in dokumentationen i följande fem huvudgrupper:

- **Analytisk dokumentation:** All dokumentation som produceras vid initieringen av ett systemutvecklingsprojekt. De analytiska dokumenten består oftast av kravspecifikation, projektplan och en förstudie för att se om kraven går att genomföra inom de ramar man har till förfogande.
- **Systemdokumentation:** Innefattar all dokumentation som krävs för att kunna programmera, testa och implementera ett system.
- **Programdokumentation:** Är bl.a. själva koden. Innefattar även dokumentation som beskriver programmets struktur.
- **Driftdokumentation:** Den personal som ska sköta om datasystemet behöver dokumentation för dess drift.

- **Användardokumentation:** Innefattar all den dokumentation som användarna behöver för att kunna använda och sköta systemet.

Vår uppfattning av Encyclopedia of computer science (London, 2000) beskrivning av dokumentation som systemdokumentation, programdokumentation, driftdokumentation, användardokumentation samt delar av den analytiska dokumentationen utgör ännu ett perspektiv och därmed en utvidgning till Nationalencyklopedins (1991) kortare beskrivning.

AstraZeneca uttrycker inte en stringent definition av dokumentation som är gemensam för hela företaget. Olika avdelningar världen över arbetar delvis på skilda sätt och syftet med dokumentationen skiljer sig åt mellan olika delar av företaget. Trots att begreppet kanske är svårt att beskriva eftersom det kan innefatta olika hanteringsförfaranden och syften, så har vi funnit det möjligt att definiera begreppet genom beskrivningar av olika dokumenttyper. Dokumenttyper samt förhållningssätt till dokumentationsarbete definieras på de internutbildningar som bl.a. berör dokumentering som AstraZeneca ger. En av dessa internutbildningar som alla anställda som arbetar med informationssystem (IS) på AstraZeneca i Mölndal måste gå, kallas Standard Operating Procedures utbildning (SOP-utbildning).

Inom SOP-utbildningarna berörs de dokument som ska ingå i systemutveckling, från initiering till avveckling vilket skiljer sig beroende på vilka krav som ställs på applikationen som ska utvecklas. Vi tar bara här upp när kraven på dokumentering är som allra högst ställda på systemutvecklingen och/eller driften. Då ger SOP s.k. riktlinjer för dokumentation. De riktlinjer som ges är att det ska finnas en central innehållsförteckning för all dokumentation. Det ska även finnas versionsbeteckning på all dokumentation. Den sista riktlinjen säger att all dokumentation och programkod som tillhör en validerad version ska arkiveras.

Minst följande dokument ska tas fram när kraven på dokumentering är som allra högst ställda på program- och systemutvecklingen och/eller driften på AstraZeneca i Mölndal:

- **Projektspecifikation:** Ska bl.a. innehålla tidsplan, ansvarsfördelning och uppskattad resursförbrukning.
- **Kravspecifikation:** Är en övergripande beskrivning av kraven som beställare ställer på systemet. Den innehåller även de funktioner som systemet skall bestå av. Kravspecifikationen tas fram tillsammans med beställare på ett språk som förstås av densamme.
- **Funktionell specifikation:** Är en mer detaljerad beskrivning av de funktioner som har definierats i kravspecifikationen.
- **Designspecifikation:** Beskriver systemets utformning och arkitektur. Det kan t.ex. vara flödesscheman, tillståndsdigram, klassdiagram samt interna och externa gränssnitt.
- **Kodbeskrivning:** Är oftast kommentarer i koden.
- **Testdokumentation:** Ska bl.a. innehålla testplan, testspecifikationer och testresultat.

- **Valideringsdokumentation:** Innehåller bl.a. en valideringsplan och en valideringsrapport, som är en sammanfattning av valideringsprocessen. Med hjälp av valideringsrapporten fattar systemägaren beslut om driftsättning eller ej.
- **Användarhandledning:** Innefattar all den dokumentation som användarna behöver för att kunna använda och sköta systemet.
- **Driftsdokumentation:** Ska bl.a. innehålla vem/vilka som är driftansvariga, hur förändringar i driftsmiljö ska hanteras och dokumenteras samt rutiner vid datorhaveri.
- **Avvecklingsplan:** Ska bl.a. innehålla orsak till avvecklingen, påverkan av andra system och datum för avveckling.
- **Central innehållsförteckning:** Det är en innehållsförteckning över samtliga dokument.
- **Systemdokumentation:** Är ett samlingsnamn för all dokumentation som har producerats under systemutvecklingen. Den innefattar även all dokumentation som produceras när systemet väl är i drift samt den dokumentation som skapas vid avveckling av systemet.

Syftet med att dokumentera och att producera de ovan nämnda dokumenten är att ha ett dokumenterat bevis som med hög säkerhet visar att systemet fungerar och kommer att fungera korrekt och konsekvent enligt gällande samt förutbestämda specifikationer och kvalitetskrav. Däremot förenklar detta vidare utveckling och förvaltning av systemet. Dokumentationen ska även visa att systemet motsvarar beställarens krav.

AstraZenecas definition enligt ovan kan enligt oss inte till fullo kortas ned till Nationalencyklopedins (1991) kortare beskrivning av dokumentation, även om den senare uttrycker kärnan. När vi i uppsatsen menar dokumentation så menar vi all dokumentation som behövs för att utveckla, förvalta och avveckla ett informationssystem. Detta överensstämmer således i stort sett med Nationalencyklopedins (1991) beskrivning men med utvidgning av AstraZenecas definition och beskrivning av olika dokument.

## 2 Teoretisk referensram

*I detta kapitel presenteras tankar och teorier som är resultat av tidigare forskning som den teoretiska referensram i vilken analys av resultatet senare baseras. Den teoretiska referensramen är indelad i tre teman som alla behandlar dokumentationsarbete utifrån olika aspekter. Dessa teman är Synen på dokumentation och dess roll i systemutvecklingsprocessen, Helhetssyn och systemutvecklingsprocessen samt Lärande och utbildning.*

### **2.1 Syn på dokumentation och dess roll i systemutvecklingsprocessen**

Från Alexandersson och Johanssons (2002) uppsats framkom att synen på dokumentation inte är helt entydig inom AstraZeneca. Undersökningen antydde även att en lägre förståelse för dokumentationens roll i systemutvecklingsarbetet kan bero på bristande kunskaper inom dokumentationsarbetet. Eftersom synen inte var entydig ville vi undersöka synen hos lärare och studenter för att förstå detta utifrån tidigare litteratur inom området. Denna del av den teoretiska referensramen syftar således till att belysa hur tidigare litteratur och forskning har behandlat ämnet.

#### **2.1.1 Dokumentationen tillhör systemet**

Pressman (2001) och Sommerville (2001) menar båda att det finns en utbredd uppfattning om att mjukvara enbart består av ett eller flera program och inte som dessa båda författare menar bestående av program, data och dokumentation. Enligt Pressman (2001) är uppfattningen om mjukvara enbart bestående av programdelar en myt som har sitt ursprung i programmeringskulturen som vuxit fram under programmeringens cirka 50-åriga historia. Inom software engineering är uppfattningen att enorma mängder dokumentation bör produceras, vilket även det framhävs som en myt. Enorm mängder dokumentation kan t.o.m. sakta ner arbetet så att en utvecklingsprocess tar längre tid än nödvändigt. Nickerson (1998) menar i likhet med Pressman (2001) och Sommerville (2001) att dokumentation är en del av mjukvaran. Mjukvaran kan inte anses som klar förrän all dokumentation är färdigställd, menar Nickerson (1998).

Pressman (2001) hävdar att utöver att dokumentationen är en del av mjukvaran är dokumentationen även en del av ett datasystem. Datasystemet i sig består av mjukvara och dess dokumentation, hårdvara, databaser och i sin tur dess dokumentation i form av manualer samt bestående av människor och procedurer för att utföra de uppgifter som ska göras med hjälp av datasystemet. Även Pfleeger (1997) menar att dokumentation är en del av ett system där dokumentationen är en av systemets tillgångar och menar att det kan vara en stor säkerhetsrisk om datasystemets dokumentation är undermålig. Är exempelvis användarmanualerna felaktiga eller oklara och användarna av den anledningen begår allvarliga misstag kan det få ödesdigra konsekvenser. Särskilt om det handlar om datasystem som används i olika situationer som har betydelse för människors liv eller hälsa.



Brown (2002) och Booch som refereras i Brown (2002) menar att vilka dokument och i vilken omfattning dokument ska skrivas beror på vad syftet är med utvecklingen. Det är försvarbart att skriva mycket dokumentation om systemet som byggs kan påverka människors liv och hälsa om något skulle gå fel. Om systemet inte ska användas mer än ett par gånger för att sedan kasseras, behövs inte någon dokumentation överhuvudtaget för att spara tid och ekonomiska resurser. I de flesta fall, utöver de nyss nämnda, bör dokumentationen inte uppta mer än 5 % - 10 % av utvecklingstiden.

### 2.1.2 Dokumentationens betydelse för kvalitet

Flera författare diskuterar vad som spelar in i mjukvarans kvalitet (Pressman, 2001; Sommerville, 2001 & Visconti, 1994). Sommerville (2001) menar exempelvis att mjukvaruprodukter har ett antal attribut som speglar kvaliteten på mjukvaran. Dessa attribut är inte direkt förknippade med vad mjukvaran gör. Det reflekterar snarare mjukvarans beteende när det används, strukturen och organiseringen av programmen samt strukturen på dokumentationen. Kvaliteten har således att göra även med dokumentationens struktur. Även Visconti (1994) har en liknande uppfattning och har i sin doktorsavhandling kommit fram till att hur och vad som dokumenteras är en nyckelfaktor för att uppnå hög kvalitet på mjukvaran. Pressman (2001) menar i likhet med Visconti (1994) att software engineering inte handlar om att skapa dokument utan om att skapa kvalitet. Bättre kvalitet leder till reducerat omarbete, vilket i sin tur leder till snabbare leveranstider. Visconti (1994) menar att en av huvudorsakerna till olika problem vid utvecklandet eller vid förvaltandet av mjukvaran beror på dålig dokumentation, ej uppdaterad dokumentation eller dokumentation som inte kan hittas. Ju bättre kvalitet på dokumentationen, desto bättre kvalitet är det också på mjukvaruprodukten, vilket medför förenklad testning och förvaltning. Även Nickerson (1998) påtalar att det är lättare att rätta till fel eller göra förändringar i ett system om systemets uppbyggnad finns beskrivet i någon form av dokumentation.

### 2.1.3 Dokumentationens syfte och dess roll i systemutvecklingsprocessen

Mathiassen et al. (2001) beskriver syftet och betydelsen av dokumentationen och betraktar den som mycket viktig för systemutvecklarna då den tjänar som gemensam referensram under utvecklingen. Dokumentationen fungerar som arbetsverktyg som samlar ihop och strukturerar delresultat allteftersom de uppnås. Dokumentationen tjänar som styrverktyg och ger ett mått på hur arbetet fortskrider samt dokumenterar överenskommelser om systemkrav och design. Genom att tjäna som en gemensam referensram utgör dokumentationen också en viktig roll i systemutvecklingsprocessen som kommunikationsmedel för de personer som arbetar tillsammans med ett utvecklingsprojekt. Fyra andra funktioner dokumentationen uppfyller och tjänar är:

- **Kommunikativ funktion:** Om de kommunikativa dokumenten inte är av god kvalitet kan lätt missförstånd uppstå bland de inblandade, vilket också försvårar för nyanställda att sätta sig in i projektets mål, struktur m.m.

- **Historisk referens:** Den historiska referensdokumentationen är den dokumentation som visar hur systemet fungerar. Om den informationen är av god kvalitet är det lätt att göra ändringar i systemet när systemet är i drift.
- **Kvalitets- och kvantitetskontroll:** I takt med att systemet byggs så blir fler och fler dokument färdiga. De färdiga dokumenten kan t.ex. användas för att följa projektets framsteg och för att se dess kvalitet.
- **Instruktionsfunktion:** Om de instruktiva dokumenten är av god kvalitet kan de t.ex. användas för framtida framtagande av nya system. Man kan på så sätt exempelvis avgöra vad som kan återanvändas. (London, 2000)

Mathiassen et al. (2001) menar att för att dokumentationen ska betraktas som god bör den vara ändamålsenlig och präglas av klarhet. Det viktiga är hur dokumentationen skrivs: ”... ett alltför detaljerat designdokument kommer inte att inspirera programmerare till kreativitet och effektivitet.” (s. 32) Författarna menar att effektiviteten kan hotas om dokumentationen har dålig kvalitet och ingen använder den varken för styrning eller i efterföljande arbete. Ett sätt att motverka detta är dock att skriva korta, koncisa dokument med hög kvalitet. Humphrey (2000) beskriver på liknande vis hur dokumentation kan skapas för att öka dess läsbarhet. Författaren menar att enkla ord, korta meningar, listor och tabeller kan användas. Långa textstycken bör undvikas då de tar lång tid att skriva och läsa. Ett bra dokument ska kretsa kring läsarens behov och inte kring produktens struktur. Mathiassen et al. (2001) påpekar att dokumentationen bör hålla hög kvalitet för att uppfylla sitt syfte och om den inte gör det kan det leda till fatala fel och misstag som blir dyrbara att rätta till om felet ens går att finna i ett senare skede. Ett exempel som Mathiassen et al. (2001) tar upp angående det viktiga med att dokumentera är betydelsen av att dokumentera testning då det kanske enbart går att utföra testet en enda gång. Om det inte då finns dokumenterat blir det svårt att analysera testet efteråt.

Sommerville (1996) är av samma uppfattning som Mathiassen et al. (2001) och menar att dokumentation som är skapad på ett ändamålsenligt sätt kan vara till stor hjälp under utvecklingsarbetet och i underhållsfasen av ett system. Dokumentationens roll i systemutvecklingsprocessen varierar med utvecklingsmodeller, menar Sommerville (1996). Olika systemutvecklingsmodeller tas upp där dokumentation ges olika mycket utrymme, exempelvis presenteras vattenfallsmodellen där varje fas avslutas med olika dokument som bygger på varandra. Sommerville (2001) tar även upp underhållsfasen som en naturlig fortsättning på ett systems utvecklingsprocess som även den innehåller specifikation, design, implementation och testning. Sommerville (2001) menar att det kan vara kostnadseffektivt att investera kraft under designen och implementeringen på att reducera underhållskostnaderna, vilket dokumentation kan vara en del av. Det är dyrare att lägga till funktionalitet efter leverans på grund av behovet av att förstå det existerande systemet och behov av att analysera systemförändringarnas påverkan. Efter ett utvecklingsprojekt bryts projektgruppen upp och de nya team som skapas för att underhålla systemet får spendera mycket tid på att försöka förstå bakgrunden till de fattade designbesluten. Kontrakt kring systemutvecklingen och kontrakt för underhållet av systemet sluts inte alltid med samma företag eller personer. Det gör i sin tur att de som utvecklar upplever att det inte ingår i deras kontrakt att skriva koden på ett sådant sätt att det är lätt för andra att förstå. En faktor som kan leda till höga underhållskostnader är

således avsaknad helt av dokumentation eller avsaknad av konsistent dokumentation (Sommerville, 2001).

Att underhållsarbetet bör beaktas i utvecklingsprocessen är något Xiaoping (2000) i likhet med Sommerville (1996) anser. Underhållsfasen bör vara i fokus av olika anledningar menar Xiaoping (2000). En anledning är att när det gäller system med lång livslängd så kommer underhållskostnaderna överstiga utvecklingskostnaderna och därför bör design som underlättar underhåll tas i beaktande från start. En annan anledning är att om underhållet och förvaltningen är dålig blir systemet inte heller bra. En hög nivå av förvaltningsbarhet kräver flexibilitet i designen och i implementeringen av systemet.

Sommerville (2001) beskriver vidare olika dokument som har betydelse för underhållsfasen, däribland systemdokumentationen. Denna typ av dokumentation inkluderar alla dokument som beskriver hur systemet är uppbyggt och utgörs av dokument som är resultat av olika arbetsmoment. Systemdokumentationen omfattar dokument från kravspecifikation till dokumentation av acceptanstesterna. Dokument som specifikt kan underlätta för underhåll av systemet är kravspecifikationen och dokument som beskriver systemarkitekturen där varje program i systemet bör ha en sådan beskrivning. Listor med programkällkod bör vara kommenterad, helst bör koden dock vara självdokumenterande utan behov av förklarande kommentarer. Andra dokument är valideringsdokument som beskriver hur varje program validerats och hur valideringsinformationen relateras till kravspecifikationen. En guide kring underhållsarbetet som beskriver hur evolution och förändring har inberäknats i designen bör också finnas, hävdar Sommerville (1996).

## **2.2 Helhetssyn och systemutvecklingsprocessen**

I denna teoretiska redogörelse beskriver Mathiassen et al. (2001) dokumentationsarbetet som en trivial uppgift och Parnas och Madey (1995) beskriver det som en uppgift som inte ses som en del av designarbetet, vilket medtagits då detta i viss mån överensstämmer med de problem AstraZeneca upplever kring dokumentationsarbetet. Därefter behandlas Hackman och Oldhams teorier (refererade i Bakka et al., 1993) om arbetet i förhållande till motivation där arbetstillfredsställelsen bl.a. kan påverkas av känslan av att arbetsuppgiften har betydelse för andra. Detta stycke avser att ge en bild av hur motivation och arbetsuppgift hänger samman, vilket vi menar att dokumentation och motivation också kan göra. Den teoretiska referensramen som behandlar motivation, samt betydelsen av en helhetsförståelse, reflektion och kommunikation som ett verktyg för processförbättring och komplex problemlösning, avser att utgöra det sammantagna material utifrån vilket resultatet kommer att tolkas och analyseras. Tolkningen och analysen kommer att ske kring vikten av helhetssyn i utbildningen för att förstå betydelsen av olika delar i systemutvecklingsprocessen.

### **2.2.1 Dokumentationsarbete och motivation**

Parnas och Madey (1995) menar att dokumentationen inte ses som en del av designaktiviteten, utan ses som en extra arbetsuppgift som måste sammanställas p.g.a. byråkratiska regler. Systemdesigndokumentation kan därför vara inaktuell redan när den

levereras och blir därefter sällan uppdaterad. Dokumentationen kan användas både som designmedium men även som input till analys och testaktiviteter och är därför lika viktig som produkten själv. Om dokumentationen är god kan mjukvaruprodukten ersättas eller utvecklas på ett relativt okomplicerat sätt, men med inadekvat dokumentation kan längden på mjukvarans livstid och värde diskuteras (Parnas & Madey, 1995).

Även Mathiassen et al. (2001) uttrycker dokumentation som betydelsefull och menar att dokumentationen bidrar till att skapa sammanhang under hela systemutvecklingsprocessen. Trots att de flesta är eniga om att dokumentation är viktigt så tas ofta arbetet med dokumentation inte på allvar. Många betraktar dokumentation som något trivialt och tråkigt, vilket är en beskrivning som påminner om Parnas och Madeys (1995) som hävdar att dokumentationsarbete ibland ses som en motbjudande uppgift. När dokumentationen är av sämre kvalitet är fallet ofta att ingen använder den, varken för styrning eller för uppföljning. Ingen vill lägga ner arbete på att producera ett dokument som ingen använder, menar Mathiassen et al. (2001). Det här kan jämföras med Alexandersson och Johanssons (2002) uppsats i vilken det framkom att en del utvecklare inte upplevde dokumentationsarbete som motiverande och en slutsats kring det var att arbetsuppgiftens krav på kunskap översteg den faktiska kunskapen hos den som skulle utföra uppgiften.

Det Alexandersson och Johansson (2002) tar upp om att arbetsuppgifter och motivation kan sättas i förhållande till varandra beskriver även Bakka et al. (1993). Författarna tar i samband med motivation i förhållande till arbete upp Hackman och Oldhams instrument för forskning inom arbetstillfredsställelse, Job Diagnostic Survey (JDS). JDS tillämpades på hundratals arbetstagare inom olika yrkeskategorier och pekade på vilka faktorer som påverkade tillfredsställelsen med arbetet vilket sedan knöts till teorier om motivationstillstånd. Fem aspekter identifierades som påverkade arbetstillfredsställelsen:

1. *Arbetsuppgiftens krav på olika förmågor (skill variety)*  
I vilken grad arbetsuppgiften kräver olika förmågor. Ett arbete som kräver olika färdigheter upplevs ur psykologisk synpunkt som mer meningsfullt.
2. *Arbetsuppgiftens identitet (task identity)*  
I vilken utsträckning arbetet kräver att den anställde gör klart ett helt arbete, från början till slut med synligt resultat påverkar tillfredsställelsen.
3. *Arbetsuppgiftens betydelse (task significant)*  
Arbetet upplevs som viktigt om det har konkret inverkan på andra människors behov och situation.
4. *Befattningens autonomi*  
Tillfredsställelsen påverkas av i hur stor utsträckning arbetet erbjuder olika valmöjligheter.
5. *Feedback i arbetet*  
Tillfredsställelsen påverkas av i vilken omfattning arbetsaktiviteterna som utförs ger direkt eller klar information om resultatet av arbetsinsatsen.

Bakka et al. (1993) skriver att Hackman knyter samman dessa egenskaper arbetet har med motivationstillstånd där innebörden av höga utfall på dessa fem är att den anställde växer med arbetsuppgiften samt att tillfredställelsen är hög och arbetseffektiviteten blir hög. Hackman menar att även andra variabler påverkar tillfredställelsen med arbetet. En sådan variabel är medarbetarens kunskaper och färdigheter inom området. Om kompetensen inte överensstämmer med kraven kan personen i fråga uppleva otillfredsställelse och frustration i arbetet som kan leda till en otillfredsställt utförd arbetsuppgift, vilket personer inom AstraZeneca menade kunde vara fallet. Arbetsammanhanget, som t.ex. befattning och det organisatoriska sammanhanget är en annan variabel som påverkar tillfredställelsen (Bakka et al., 1993). Organisationsammanhanget innefattar en rad förhållanden som t.ex. lönesystem, arbetskamrater, organisationskulturen samt ledarstilar på arbetet.

### **2.2.2 Helhetssyn, kommunikation och reflektion vid komplex problemlösning**

I Alexanderssons och Johanssons (2002) uppsats framkom funderingar kring huruvida förmågan att se helheten i systemutvecklingsprocessen hade betydelse för synen på dokumentationen som uppgift. Funderingar kring detta kan styrkas med Sommervilles (2001) uppfattning att det krävs en förståelse av systemutvecklingsprocessen i sin helhet vid problemlösning inom densamma.

Sommerville (2001) motiverar behovet av en helhetsförståelse med förklaringen att problem som uppstår oftast är beroende av resultatet av beslut som fattats under processens gång. Att tänka ut hur ett program ska fungera och att skriva det är en problemlösningssprocess som ingår i yrket som software engineer som innebär aktiviteter som att specificera, designa, implementera, validera, utveckla och underhålla system som en helhet. Mjukvaruutveckling kräver därför en förståelse av problemet och kräver en problemlösningstrategi som sedan ska översättas till programform. Systemutvecklingsprojekt bör därav inkludera personer som generellt har god problemlösning förmåga och erfarenheter från det aktuella området, snarare än personer med specifika programspråksfärdigheter.

Behovet av det helhetstänkande Sommerville (2001) beskriver, hänger även samman med författarens förklaring av ett systems karaktäristika. Karaktäristiskt för ett system är att egenskaperna i systemet och beteendet hos systemets komponenter är sammanhängande, menar författaren och hänvisar till Checkland (1999) som skriver att dessa s.k. emergent (ungefär uppkommande, framträdande) properties inte kan hänföras till någon enskild specifik del av systemet. Dessa egenskaper framträder enbart när systemet ses som en helhet. Komplexa förhållandena mellan komponenterna i ett system innebär såtillvida att systemet är mer än bara summan av dess enskilda delar. Den framgångsrika funktionaliteten hos varje systemkomponent är beroende av funktionen hos en annan komponent. Systemet kan vara uppbyggt av subsystem med komplicerade interna förhållanden, som inte kan förstås genom att analysera individuella systemkomponenter.

Checkland (1999) nämner vidare att det krävs en medvetenhet om att förändringar på olika nivåer kan uppkomma under analysprocessen av ett problem, vilket kan öka komplexiteten ytterligare. Det handlar således inte enbart om att se systemet som helhet gällande tekniska komponenter, utan även att kunna ta hänsyn till mer mänskliga aspekter och förändringar bland dessa. Förändringar att ta hänsyn till är strukturella förändringar som organisationsförändringar, procedurförändringar i form av förändringar i rapport- och informationssystem samt attitydförändringar som människor utvecklar i den organisation där problemet befinner sig. Därför tvingar problem i den verkliga världen analysmetoder att bli tillvägagångssätt för att organisera diskussioner, debatter och argument framför att lösa problemet utifrån ett hårdare synsätt med en smalare lösning för att kunna fånga komplexiteten av problemet. Ingenjörsskap som yrke tenderar att locka till sig aktionsinriktade människor som värdesätter praktiskt genomförande mer än processen, ett resultat av detta är att de blir otåliga med teoretiserandet. Efter att en design skapats är de inte intresserade av att ägna tid åt att konstatera hur lösningen togs fram. Till följd av bland annat förändringar i mänskliga önskemål kan komplexiteten i ett system vara av mycket hög grad. Då måste ingenjörsskapet ta i beaktande och ha förmåga att förutsäga de framväxande egenskaperna som är resultatet av systemet som helhet och inte av dess delar. Möjlig design av datorsystem kan innefatta steg från att skapa designförslag och prototyper, till testning, träning och utvärdering menar Checkland (1999).

Checkland (1999) grundar systemtänkandet på begreppen emergence, kommunikation och kontroll, där de två senare begreppen är sammankopplade till det första. Kommunikationen och kontrollens betydelse för begreppet emergence är att kommunikationen i öppna system syftar till kontroll över den rådande situationen och uppföljning av den. För system, skapade av människor, innebär detta att processer för kommunikation och kontroll måste utföras för att systemet ska överleva gentemot de tryck systemet utsätts för från systemets omgivning.

Det är viktigt att kommunikationen är god mellan medlemmar av mjukvaruutvecklingsprojekt (Sommerville, 2001), vilket påminner om Checklands (1999) men även om Mathiassen et al. (2001) tankar om dokumentation som gemensam referensram för inblandade programmerare. Dokumentationen stärker banden mellan involverade personer samt förbättrar samordningen av arbetet mellan programmerarna, menar Mathiassen et al. (2001). Sommerville (2001) skriver att gruppmedlemmarna i ett systemutvecklingsprojekt måste utväxla information om statusen på sitt arbete och utbyta information om beslut som fattats samt eventuella nödvändiga förändringar i dessa beslut, varpå kommunikationskanaler fordras. Varje avdelning bör ha kommunikationsverktyg. Normal utvecklingsmiljö är nätanslutna arbetsstationer med installerad mjukvara som CASE-verktyg och dokumentationsverktyg (Sommerville, 2001). Hazzan (2002) hävdar att bättre kommunikation kan bli resultatet av att teammedlemmar i ett systemutvecklingsprojekt tillämpar reflektion kring sitt arbete. Författaren förespråkar reflektion även av anledningen att reflektion kring arbetsprocesser och resultat kan förbättra utförandet vilket är av betydelse eftersom komplexiteten under utveckling av mjukvara ofta är hög. Reflektionen kan leda till bättre förståelse för sina egna tankeprocesser och detta sammantaget med reflektion kring andras tankesätt kan leda till

bättre kommunikation inom teamet. Reflektionsmetodologier kan förbättra förståelsen för processer och för utveckling av komplexa objekt och ju bättre förståelse för utvecklingsprocessen är, desto bättre är förståelsen för metodologierna som styr dessa processer.

## **2.3 Lärande och utbildning**

Kring detta tema behandlas förhållandet mellan industri och akademi samt vad systemutvecklingsutbildningar kan innehålla och i vilka former undervisningen bedrivs eller kan bedrivas. Denna teoretiska referensram kommer senare att användas som tolkningsram för vilken betydelse utbildningen har för studenternas syn på dokumentation med avseende på utbildningens innehåll och form.

### **2.3.1 Diskussion kring yrkesverksammas kunskaper och förmågor**

Alexandersson och Johanssons (2002) uppsats behandlade uppfattningar bland en del av AstraZenecas medarbetare som innebar att dokumentationsarbetet inte upplevdes ingå bland deras arbetsuppgifter och därför kändes omotiverande att utföra. Detta aktualiserar en diskussion om vad som kan tänkas ingå i ett yrke som handlar om att utveckla datorsystem och hur utbildningar kan utformas. Detta kan sedan jämföras med de undersökta utbildningarnas innehåll. Nedan följer även en redogörelse för diskrepansen mellan industrins och akademins uppfattningar avseende utbildningars innehåll, där forskningen eftersöker mer involvering från industrins sida i utbildningarna. Vad som inom forskning diskuteras kring utbildningarnas innehåll kan vara intressant att relatera till de undersökta utbildningarna.

En diskussion kring yrket software engineer tas upp av Sommerville (2001) som menar att software engineering är en disciplin som innefattar alla aspekter kring produktion av mjukvara. Mjukvaruprodukten består av utvecklade program samt tillhörande dokumentation och produkten bör designas för att uppfylla krav på användbarhet, underhåll, tillförlitlighet och effektivitet. Yrkesverksamma inom software engineering har ansvar gentemot yrket och samhället i form av professionalism och de bör inte enbart beakta tekniska förhållanden och ämnen. Organisationer som ACM och IEEE nämns som normgivande inom yrkets professionalism och publicerar yrkeskoder för beteenden och genomförandenormer som yrkesverksamma inom området bör beakta liksom utbildare, ledare, skapare av policys samt studenter. I ACM/IEEE:s yrkeskod kring etik och professionalism tas principer upp som behandlar etiska aspekter som förhållandet mellan kund och utvecklare. Aspekter kring professionalism berör åtaganden att engagera sig i analys, specifikation, design, utveckling, testning och underhåll. I det professionella ingår även att stödja kollegor samt deltagande i livslångt lärande gällande sitt yrke (Sommerville, 2001). Hazzan (2002) menar att studenter inte bara måste vara tekniskt kunniga utan även bör besitta kunskaper inom arkitektur, design samt även bör kunna lyssna, övertyga, vara professionella, ärliga och pålitliga, vilket i stort överensstämmer med ACM/IEEE:s yrkeskod.

Wohlin och Regnell (1999) hänvisar även de till IEEE gällande yrket software engineer som definieras av författarna som ett systematiskt och disciplinerat angreppssätt på

utveckling, drift och underhåll av mjukvara. Utbildning inom software engineering bör därför innehålla, inte bara moment som mjukvarudesign, metoder och programmeringsspråk utan även kravspecificering, processförbättring samt aspekter som berör testning och kvalitet. En av de viktigaste uppgifterna utbildningen inom just software engineering har, är att ge studenterna en god bas för storskalig utveckling i industriella miljöer. För att möjliggöra detta krävs industrirelevans i utbildningen som kan åstadkommas på flera sätt utifrån industrins behov. Samtidigt poängteras en strävan mot att inte lösa tillfälliga problem inom industrin, vilket är en risk med hänsynstagande till industriella anspråk.

Även Humphrey (2000), tar upp vikten av att införa industrirelevans i utbildningar. Den ökade komplexiteten i systemutvecklingsprocessen och kravet från industrin på bättre kvalificerade och bättre förberedda studenter borde innebära att läroplaner inom systemutveckling bör ge kunskap och erfarenhet som är relaterad till tillämpningen av ämnesområdet (Humphrey, 2000). Saiedian (2002) menar i likhet med detta att för att minska det existerande gapet mellan industri och nytutexaminerade studenter inom software engineering, bör industrin involveras mer i utbildningsprocessen.

Även Lee et al. (2002) antyder en diskrepans mellan akademi och industriella behov. Akademiker inom informationssystemområdet och utövare (t.ex. IS-manager, systemprogrammerare, systemanalytiker) är enligt författarna inte eniga om de förmågor och kunskaper en yrkesverksam inom området bör besitta. En student som examineras från en utbildning inom, vad författarna kallar informationssystemutbildningar, bör ha förberetts väl för den ingångsnivå som krävs för att kunna arbeta inom yrket i arbetslivet. Ett område där akademiker och verksamma i arbetslivet inte värderade olika typer av arbetsområden som lika viktiga var implementering, drift och underhåll där akademiker värderade detta som mindre viktigt än yrkesverksamma inom branschen. Akademiker värderade även variabler som kommunikationsmöjligheter och sociala aspekter som mindre viktiga än verksamma inom området och akademikerna värderade de flesta mjukvaruverktyg som mer viktiga än vad utövarna gjorde. Lee et al. (2002) avslutar redogörelsen för sina resultat med att mena att utan att minska det gap de hävdar existerar mellan akademiker och yrkesverksamma, kan utbildningarna inte fortsätta att tillfredställa det industriella behovet, vilket AstraZeneca önskar att utbildningarna gör.

Annan kritik som riktas mot undervisningsformerna från forskningshåll, är kritik gällande att det sätt på vilket datorprogram i utbildningar presenteras för studenter inte överensstämmer med den komplexa verklighet som omger ett systemutvecklingsprojekt, vilket således inte överensstämmer med hur industrin arbetar (Hazzan, 2002).

### **2.3.2 Utbildningsinnehåll, utformning och feedback**

Andersson (2000) refererar till Magers arbete som anser att utbildningens syfte ofta är att påverka olika individers kunskaper, attityder och färdigheter. Genom denna påverkan kan individen få ett förändrat beteende i önskad riktning utifrån tankar som format utbildningen. Även Sjöstrand och Levin et al. (refereras av Andersson, 2000) talar om utbildningens påverkan på individen. De refereras på följande sätt av Andersson (2000):



”I vid mening kan utbildning definieras som en form av systematisk påverkan på individer i viss riktning och under organiserade former.” (s. 76)

Att utbildningens innehåll och utformning syftar till att påverka studenter i en viss riktning syftar ovannämnda avsnitt att visa. Vidare behandlas Wohlin och Regnells (1999) undersökning som berör software engineering men även kan appliceras inom andra områden och som syftar till att visa hur utbildningen kan påverkas för att på ett bättre sätt involvera industrin. Undersökningen syftar till att inspirera utbildningsansvariga att ta alternativa undervisningssätt i beaktande när det handlar om software engineering för att få in industrirelevans. Tillvägagångssätt för att införa industrirelevans i utbildningarna inkluderar att låta industriella behov påverka ämnesval i undervisningen och ett annat sätt är att fundera kring undervisningsformen.

Om det är möjligt med involvering av industrin i utbildningen (t.ex. genom gästföreläsningar och forskning knuten till industrin) är det positivt då det ökar utbildningens trovärdighet när studenterna ser att industrin brottas med samma typ av problem och lösningar som undervisningen förmedlar (Wohlin & Regnell, 1999). En problembaserad utbildningsmetod som liknar realistiska utvecklingsscenarion inspirerade av industrin, är av betydelse för att uppnå en djupare förståelse än genom traditionell föreläsningsform. Författarna menar vidare att software engineering handlar om att arbeta i projektform varav utbildning som bedrivs i projektform är av betydelse för att skapa industriell relevans. Projekten kan vara sammansatta i grupper och subgrupper där en subgrupp agerar rollen som projektledare, en subgrupp ansvarar för att hålla samman systemet, andra grupper är utvecklingsgrupper och testgrupper. Utförandet av projektprocessen bedöms lika viktig som den slutgiltiga produkten. I processen ingår att skapa dokument som kravanalysdokument som blir kravspecifikation och därefter högnivådesign. Därpå arbetar studenterna utifrån ett detaljerat designdokument samt med testning. Därefter levereras produkten som åtföljs av en projektrapport som ska innehålla en rapport om arbetets framskridande, tidsmätning samt de fel som upptäckts under arbetets gång. Den levererade produkten acceptanstestas och studenterna får feedback på sitt utförande (Wohlin & Regnell, 1999).

Studenter kan inte bli lärda vad de behöver kunna utan de kan bara bli guidade i en viss riktning (Hazzan, 2002). Att lära studenter att reflektera via feedback kan ge studenterna en förståelse för mjukvarumetodologier. Författaren argumenterar för tillämpandet av reflektion i systemutvecklingsprocessen som ett led till bättre förståelse för utförda processer och tillvägagångssätt samt bättre kommunikation. Feedbacken bör inte bestå av enbart feedback från de program studenterna bygger, då en sådan form inte säger något om arkitektur eller design. Lärare bör därför erbjuda denna feedback och guidning för att skapa reflekterande studenter (Hazzan, 2002).

Feedback från lärare poängteras även av Humphrey (1998), som en accepterad undervisningsform för att uppmuntra till motivation och fortsatt lärande och förståelse hos studenter. Humphrey (1998) behandlar Personal Software Process (PSP) som syftar till att beskriva hur en utvecklingsprocess kan systematiseras genom processarbete, processmätning samt processförbättring. Humphrey (1998) hävdar att kurser inom detta

område behövs då studenter efter examen upplevt att sådana projektkurser har varit värdefullt förberedande inför arbetslivet. När studenter utsätts för realistiska problem och sedan guidas till effektiva lösningar, får de en bättre uppskattning och bättre uppfattning av värdet av en strukturerad process. Ett vanligt problem är annars att studenter under utbildningstiden koncentrerar sig för mycket på produkten och ignorerar processen. Många oerfarna engineers tror att kravarbetet är slöseri med tid. För dem tycks kodning och testning vara det riktiga utvecklingsarbetet. Planeringen, framtagningen av kraven och designarbetet är dock det som kan göra skillnad mellan lyckade och misslyckade utvecklingsprojekt. En definierad process som är mätbar underlättar en förståelse för det som presterats vilket kan göra processen bättre. Software professionals kommer att bättre förstå vad de gör om de definierar, mäter och spårar sitt arbete vilket är en del av meningen med PSP.

Mellan kvaliteten på den utvecklade mjukvaruprodukten och kvaliteten på software processen som använts för att skapa produkten finns ett starkt samband hävdar Sommerville (2001) i likhet med Humphrey (1998). Mjukvarukvaliteten är beroende av en designprocess där individuella mänskliga beslut och överväganden är av betydelse. I vissa fall är de involverade människorna än mer viktiga än den process som används, menar Sommerville (2001), vilket förstärker vikten av att vara noggrann med utformandet av processen som Humphrey (1998) således menar, något som även aktualiserar ett integrerande av processtänkande i utbildningen.

Sommerville (2001) hävdar att även underhållsfasen påverkas av den tidigare processen. Förändringar i mjukvaran är oundvikligt, det är inte möjligt att producera system som inte behöver förändras eller underhållas. Dessutom är det många organisationer idag som förlitar sig på sina system vilket förhöjer underhållsfasens betydelse. Ett problem som uppstår ifrån det faktum att organisationer fortfarande gör skillnad mellan systemutveckling och underhåll är att underhåll har en dålig image hos software engineers och ses som en andraklassuppgift. Saiedian (2002) instämmer med detta att underhåll och förvaltning ses som en andraklassuppgift, vilket inte överensstämmer med dess verkliga kritiska betydelse. Underhållsarbete är något som på en arbetsplats lämnas över till nyanställda som får inleda sin karriär inom detta område. Vanligtvis hamnar mindre erfarna programmerare som underhållsprogrammerare, medan de mer erfarna arbetar med nyutvecklingsprojekt. Många mindre erfarna programmerare och utvecklare saknar dock den träning och den erfarenhet som krävs. Detta innefattar aktiviteter som att förstå det existerande programmet, att kunna åstadkomma de nödvändiga förändringarna eller tilläggen samt att kunna försäkra att design och analysdokument är uppdaterade så att de reflekterar genomförda förändringar eller tillägg

Trots underhållsfasens betydelse finns det sällan några incitament att spendera mer ekonomiska resurser under systemutvecklingsprocessen för att reducera kostnaderna av systemförändringar i framtiden (Sommerville, 2001). Den enda långsiktiga lösningen på detta problem är att acceptera att system sällan har en definierad livstid utan fortsätter i en eller annan form för en obestämd tidsperiod. En attitydförändring i form av hänsynstagande bör tas till att system förändras och utvecklas över tid i förhållande till nyuppkomna krav och bör designas därefter.

Andra forskare som enas kring underhållsarbetets betydelse och som förespråkar införande av undervisning kring detta, är som nämnts ovan, Saiedian (2002) och Lee et al. (2002). Saiedian (2002) menar kring underhållsarbetets betydelse att en erfaren och kunnig person som arbetar med systemunderhåll är en organisations mest betydelsefulla tillgång för att uppnå kvalitet kring mjukvaran samt att denna person är av strategisk betydelse för förbättring av underhållsarbetet och utvecklingsprocesserna.

Resultatet av Lee et als. (2002) undersökning pekar på att underhållsarbete är något som värderas högre av yrkesverksamma inom informationssystemarbete än av akademiker. En diskrepans mellan industri och akademi uppstår, vilket dock kan överbryggas genom att se över utbildningars innehåll inom området. En undersökning utförd av Lethbridge (2000) visade att undervisning inom områden som med fördel kan utökas är software design, arkitektur, användargränssnitt, projektstyrning, presentationsteknik och tekniskt skrivande. Några av dessa räknades till s.k. mjuka färdigheter och ett par av dessa hamnade högt i undersökningen bland industrins önskemål, som t.ex. förmågan att ge bra presentationer, förmågan till teknisk skrivning samt ämnen som etik och professionalism.

## 3 Metod

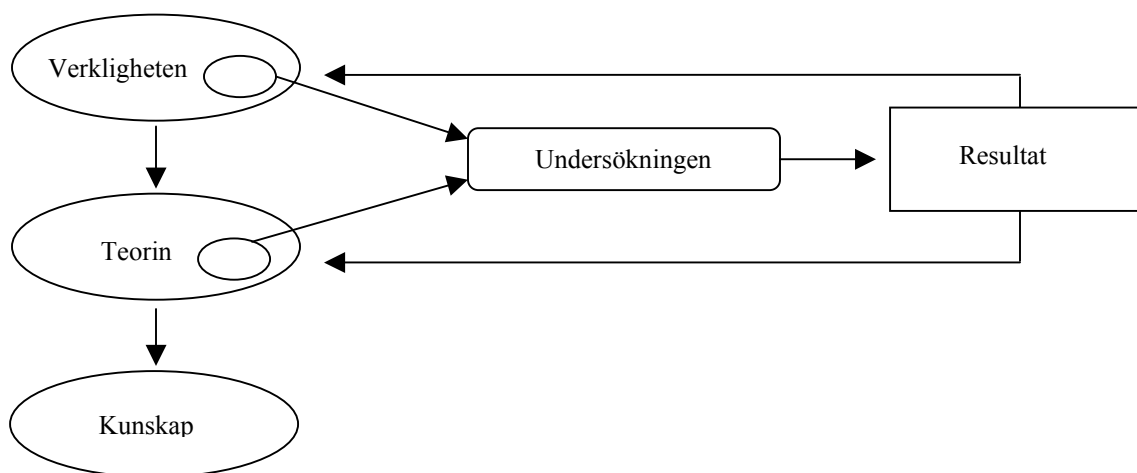
I metodavsnittet följer en redovisning av de metodmässiga val som gjorts under arbetets gång med avseende på angreppssätt, materialinsamling, genomförande av intervjuer, bearbetning av resultat, analys och tolkning. Kapitlet avslutas med begreppen validitet och reliabilitets förhållande till uppsatsen samt en diskussion kring metodkritik.

### 3.1 Val av angreppssätt och metod

Vilken typ av metod, kvalitativ eller kvantitativ, som väljs beror på vad det är för företeelse som ska undersökas och vilket syfte undersökningen har (Kvale, 1996). Faktorer som spelar in i valet av metod är problemställning, undersökningens syfte, egna förutsättningar och resurser, egenskaper hos studieobjekten och forskarens förhållande till datakällorna. Syftet med undersökningen kan vara att utveckla en helhetsförståelse av specifika förhållanden eller att få en representativ översikt över generella förhållanden och utifrån det väljs metod (Halvorsen, 1989).

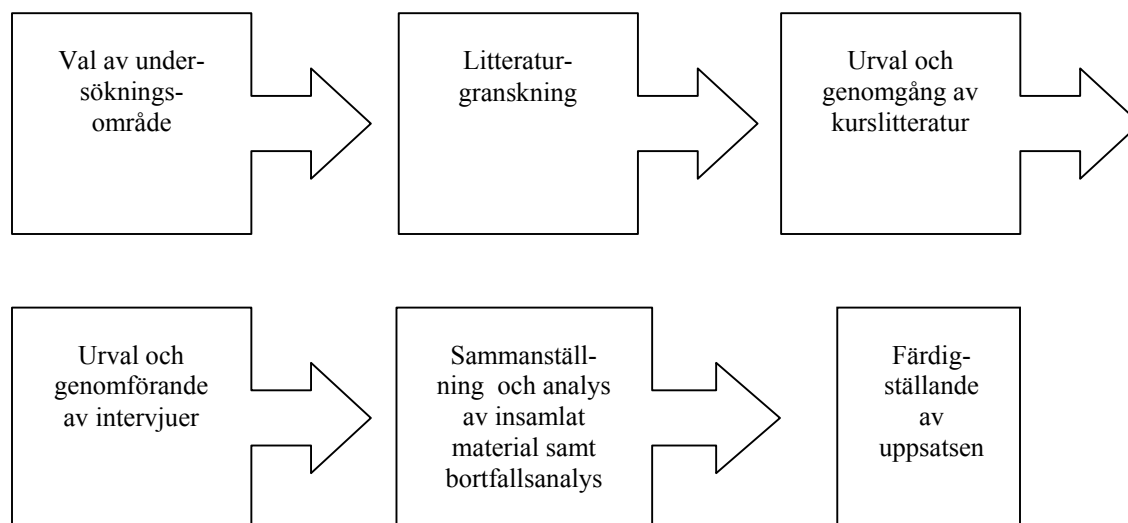
Det problem vi ämnade undersöka upplevde vi som komplext och ett problem som därmed krävde en helhetsförståelse för att kunna beskrivas och förklaras. För att kunna besvara frågeställningen krävdes delvis samtal där möjligheten fanns att ställa följdfrågor för att få ökad förståelse, vilket gjorde att en kvalitativ ansats valdes.

Vår syn på angreppssätt på metoden kan illustreras i figur 3:1. Den mindre ringen i ”Verkligheten” är det valda undersökningsområdet (se kapitel 3.2). Den mindre ringen i ”Teorin” är tidigare forskning som vi tagit del av (se kapitel 2). Utifrån undersökningsområdet utformas ”Undersökningen”, varpå det empiriska materialet (se kapitel 3.4) uppstår utifrån olika metodval. Ur ”Undersökningen” framkommer ett ”Resultat” (se kapitel 4) som återförs genom analys i form av diskussion (se kapitel 5) till ”Verkligheten” och ”Teorin” (se kapitel 2). ”Resultatet” kan sedan utifrån ”Verkligheten” och ”Teorin” tolkas och leda till ny kunskap.



Figur 3:1 Vår syn på angreppssätt på metoden. Omarbetad efter Mason (1989).

Rektangeln ”Undersökningen” från figur 3:1 illustreras mer detaljerat i figur 3:2 i form av en beskrivning av undersökningens arbetsgång. Det bör dock påpekas att det i figuren inte varit ”vattentäta skott” mellan blocken, vilket innebär att ett block inte alltid var helt avslutat innan nästa block påbörjades. Blocken visar dock ordningen på vår huvudsakliga arbetsgång.



Figur 3:2 Undersökningens arbetsgång.

### 3.2 Undersökningsområde

Idén till undersökningsområdet hade AstraZeneca fått från en tidigare uppsats utförd av Alexandersson och Johansson (2002) och som genomfördes på företaget hösten 2002. AstraZeneca upplever ett behov av att underlätta verksamhetens dokumentationsarbete. AstraZeneca var därför intresserade av en undersökning kring vad som lärs ut angående dokumentationsarbete vid universiteten. Genom att få en inblick i detta samt inblick i vilken syn studenter kan ha på dokumentationsarbete, vill AstraZeneca med resultatets hjälp bättre kunna planera sitt framtida dokumentationsarbete och sin internutbildning.

Vi är medvetna om att det existerar en tidsdifferens mellan denna undersöknings utförande och utexamineringen av de systemutvecklare som idag arbetar på AstraZeneca. Detta kan sålunda väcka frågor kring huruvida detta material kan förklara något om dagsläget på AstraZeneca. Denna uppsats avser dock inte att förklara några förhållanden på AstraZeneca av idag. Undersökningen är intressant för AstraZeneca i den aspekten att rekrytering sker från nämnda universitet. Anledningen till intresset är således att företaget är tillräckligt nära utbildningarna p.g.a. rekryteringsunderlaget, för att vara intresserade av utbildningarnas innehåll med avseende på dokumentering.

### 3.2.1 AstraZeneca

AstraZeneca bildades 1999 av de två läkemedelsbolagen Astra AB samt det brittiska läkemedelsbolaget Zeneca Group PLC. AstraZeneca vars huvudkontor ligger i London är ett globalt företag och är ett av de största läkemedelsbolagen i världen. De utvecklar och producerar främst receptbelagda läkemedel. Forskningen bedrivs främst i Sverige, USA och Storbritannien medan produktionen av läkemedel bedrivs i flera länder världen över. År 2003 har AstraZeneca cirka 54 000 anställda varav ungefär 11 000 i Sverige och av dem arbetar runt 1 700 på företagets anläggning i Mölndal. AstraZeneca i Mölndal är en forskningsanläggning där forskningen fokuserar på att ta fram läkemedel inom områdena hjärta/kärl och mage/tarm (AstraZeneca, 26 februari, 2003a & 2003b). Avdelningarna som är av intresse i denna undersökning är Development IS och Discovery IS på AstraZeneca i Mölndal, vilka är IT-funktioner inom avdelningarna Development och Discovery. Dessa två IT-funktioner har flest anställda systemutvecklare bland IT-funktionerna inom AstraZeneca i Mölndal och utvecklar system enbart åt AstraZeneca (AstraZeneca, personlig kommunikation, 6 maj, 2003 och AstraZeneca, 26 februari, 2003).

### 3.2.2 Informell förstudie på AstraZeneca

För att veta vilka utbildningar som var mest relevanta att undersöka tog vi reda på vilken utbildningsbakgrund systemutvecklarna hade på IT-funktionsavdelningarna Development IS och Discovery IS på AstraZeneca. Vi skickade ut ett brev via e-post till samtliga anställda på de båda avdelningarna. Av systemutvecklarna fick vi genom detta förfarande veta vilken utbildning de hade, vilket examensåret var och huvudsaklig utbildningsort. Av e-posten kom det fram att den vanligaste utbildningen var Systemvetarprogrammet på Göteborgs Universitet, därefter kom Datateknikprogrammet och Elektroteknikprogrammet på Chalmers i Göteborg. Förstudien visade att nämnda utbildningar till största delen representerar AstraZenecas normala urval för rekrytering och därför tog vi beslutet att undersöka de tre aktuella programmen.

## 3.3 Litteraturgranskning

Backman (1998) kallar den fas som inleder forskningsprocessen, vilket är en rad aktiviteter, för litteraturgranskningsfasen. Även om vi inte vill kalla vårt arbete en forskningsprocess, snarare en undersökningsprocess, så har vi samma uppfattning som Backman (1998). Författaren menar att innan påbörjandet av undersökningen bör den som genomför undersökningen vara påläst på det aktuella kunskapsområdet och på befintligt material. Givetvis är det inte möjligt att vara påläst på allt befintligt material utan det blir en selektiv process, där subjektiviteten också spelar in, anser vi.

Backman (1998) menar att litteraturgranskningsfasen har flera funktioner. Studiet av dokument och litteratur hjälper till att formulera en meningsfull problemställning och denna fas visar även på sammansättningen av befintligt material och hur begrepp inom området preciserats, definierats och använts tidigare. Det är på dessa sätt vi använt oss av vad vi kallar litteraturgranskning; för att närmare undersöka området, för att skapa en problemformulering, för att använda befintligt material till att skapa vår teoretiska

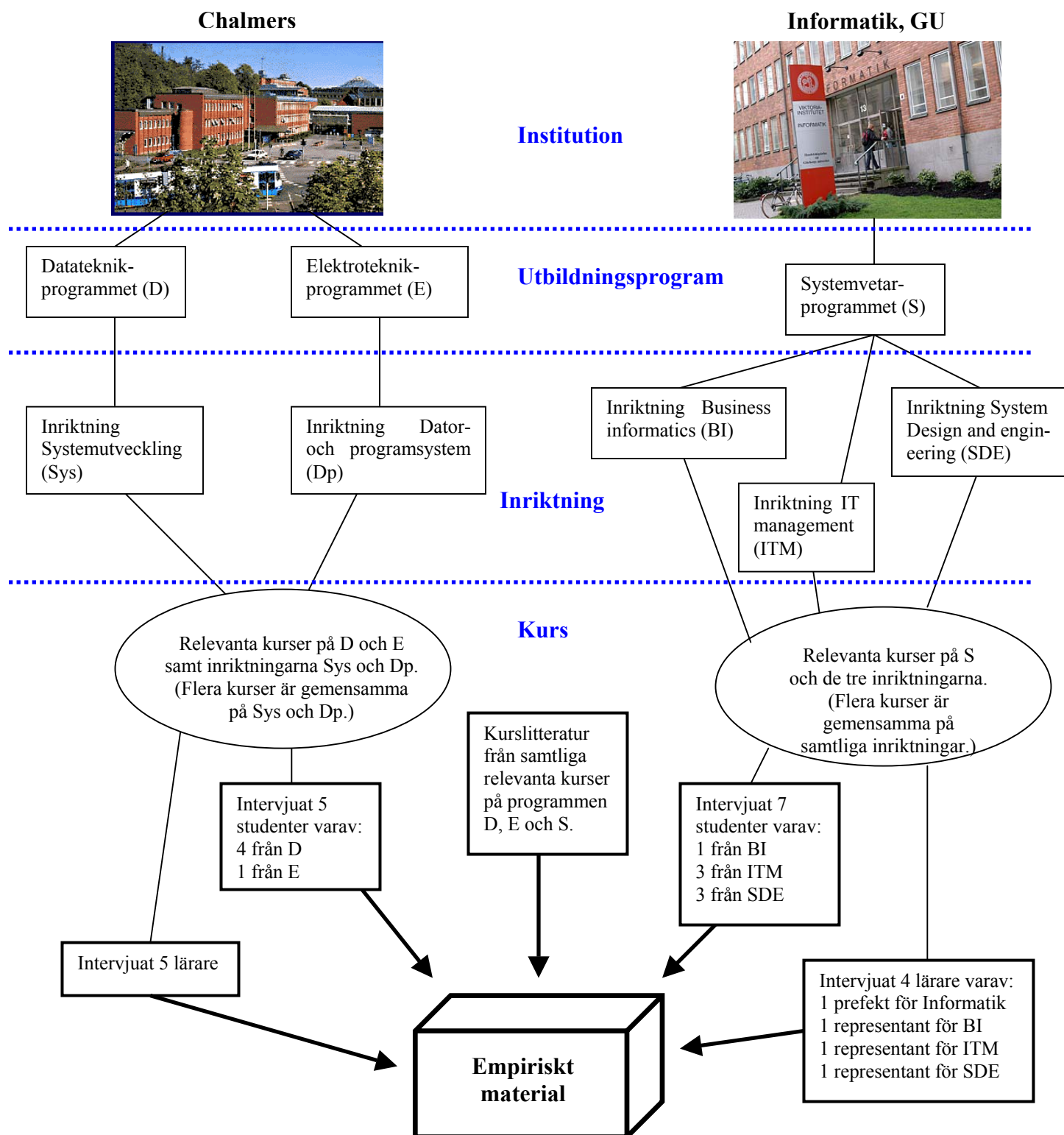
referensram samt material till intervjufrågorna. Vi har också använt oss av tidigare tolkningar av diverse begrepp för att relatera dessa till vårt eget arbete.

Vår litteraturgenomgång har i huvudsak baserats utifrån databaser som Academic Search Elite – Ebsco Host, Avhandlingar vid Göteborgs universitet, Computer Swedens arkiv, ERIC via AskERIC, Digital dissertations, Index to theses, Libris, Mediearkivet, Science Direct – Elsevier, Web of Science och Wiley InterScience.

Sökningarna utfördes vid ett flertal tillfällen där olika sökord sammansattes i olika kombinationer. Exempel på sökord vi använde oss av var; documentation, system engineering, software engineering, system development, education, educational problems och practioners. Referenser från artiklar och litteratur ledde oss vidare i sökandet efter intressant material där vi sedan sorterade våra träffar efter det som verkade vara relevant för uppsatsens innehåll.

### **3.4 Det empiriska materialet**

Vi har utgått från Systemvetarprogrammet, Datateknikprogrammet samt Elektroteknikprogrammet och inom dessa program valt de kurser och inriktningar vi ansåg var relevanta och som riktar sig mot blivande systemutvecklare. På kurserna har vi genomfört intervjuer med lärare samt studenter och undersökt relevant kurslitteratur. Gällande kurslitteraturen har vi utifrån en frågemall gått igenom deras innehåll. Se figur 3:3 för att få en överblick av det empiriska materialet.



Figur 3:3 Empiriskt material med avseende på undersökt område och urval.



### 3.4.1 Beskrivning av Systemvetarprogrammet

Systemvetarprogrammet på institutionen för Informatik vid Göteborgs Universitet består av 160 poäng. Det är en tvärvetenskaplig utbildning som kan kombineras med ämnen som ekonomi, människa, organisation och teknik. Efter avslutad utbildning får studenten en fil. Mag.–examen i Informatik. Systemvetarprogrammet har tre inriktningar där alla inriktningar har med systemutveckling att göra. Följande inriktningar fanns våren 2003:

- **Business informatics** (hette tidigare Teknik och samarbete): Den koncentrerar sig på användningen av IT i affärer. T.ex. e-business, strategi, ledning och beslutsstöd.
- **IT Management** (hette tidigare Strategisk IS/IT-planering): Inriktar sig mot IT i organisationers verksamhet. T.ex. ledning och organisering av IT, knowledge management, IS-arkitekturer, datorstött samarbete och intranät.
- **Systems Design and Engineering** (hette tidigare Programvaruutveckling): Inriktningen har ett s.k. software engineeringperspektiv. T.ex. design och utveckling av stora verksamhetsbaserade system, kvalitetsstyrning och projektstyrning (Göteborgs Universitet, 2003 och Institutionen för Informatik, 2003).

### 3.4.2 Beskrivning av Chalmersutbildningarna

Både Datateknikprogrammet och Elektroteknikprogrammet är civilingenjörutbildningar och innefattar 180 poäng vardera. Inom de båda utbildningarna finns det flera fördjupningsinriktningar som väljs inför år fyra. Det bör dock tilläggas att innan studenterna väljer sina fördjupningsinriktningar finns det kurser som har med systemutveckling att göra som de har möjlighet att välja. Vi har valt att undersöka de kurser och de fördjupningsinriktningar som riktar sig mot systemutveckling. Fördjupningsinriktningarna 2003 var:

- **Systemutveckling** på Datateknikprogrammet
- **Dator- och programsystem** på Elektroteknikprogrammet

De båda inriktningarna har ett liknande innehåll då flera av kurserna är gemensamma. Enligt Chalmers programkatalog får de som läser datateknik bred kompetens inom datateknik, däribland inom program- och hårdvaruområdet. Elektroteknik kombinerar fysik och systemvetenskap (Chalmers, 2003). Några av de mest populära kurserna på både Datateknik- och Elektroteknikprogrammet är Individuell programvaruutveckling och processförbättring (IPOP), Objektorienterad systemutveckling (OO) samt Systemutveckling i team (SUIT).

På Datateknikprogrammet har vi således undersökt inriktningen Systemutveckling, på Elektroteknikprogrammet har vi undersökt inriktningen Dator- och programsystem. Vi har på Systemvetarprogrammet undersökt alla tre inriktningarna. Vid de tillfällen vi ansett kurserna relevanta för blivande systemutvecklare har vi på alla tre programmen även undersökt de kurser som alla studenter inom programmet, oavsett inriktning kan välja att läsa.

### **3.5 Datainsamling och urval**

Intervjuer och observationer är som datainsamlingsverktyg delar av den kvalitativa metoden (Easterby-Smith et al, 2002). Intervjun är en interaktion mellan människor som innehåller ett samspel av referensramar och sociala roller. Intervjuer kan vara mer eller mindre strukturerade. I den ostrukturerade intervjun håller intervjuaren frågorna så öppna som möjligt för att följa upp intressanta riktningar samtalen tar. Frågorna måste dock ha någon form av ramverk så att fokus inte går förlorat. Sådana informella intervjuer kan dock innebära vissa problem vid analys då svaren ska struktureras (Starrin & Svensson, 1996).

Vi använde oss bl.a. av intervjuer för att försöka införskaffa den helhetsbild vi ansåg oss behöva för att besvara frågeställningen. Vi ansåg att lärarna och studenterna på Chalmers och på Systemvetarprogrammet som kom att utgöra intervjupersonerna, kunde hjälpa oss med denna helhetsbild om de inte var styrda av alltför detaljerat definierade frågor. Vi använde oss således av ostrukturerade intervjumallar som ramverk för våra frågor för att hålla samtalen inom vissa gränser och kategorier. Avsikten var inte att ställa alla frågor från mallen om det inte behövdes, utan avsikten var istället att utifrån mallen lägga till eller ta bort frågor när vi såg ett behov av det, utifrån hur intervjuerna artade sig.

Vid intervjuer med lärarna användes en gemensam mall (se bilaga 1), den skilde sig dock något beroende på vad de hade för roll på Chalmers eller på Systemvetarprogrammet. Detta gällde även studenternas mall (se bilaga 2) som skilde sig något studenterna emellan beroende på vilken utbildning som undersöktes.

#### **3.5.1 Tillvägagångssätt vid datainsamling**

De vi ämnade intervjua kontaktade vi muntligen, brevlades eller via e-post varpå vi frågade om de hade möjlighet och lust att delta i vår undersökning. Vid den första kontakten presenterade vi oss, syftet med undersökningen samt att vår uppdragsgivare var AstraZeneca. Med dem som tackade ja bestämde vi tid och plats så förmånligt som möjligt för intervjupersonen. Intervjuerna av lärarna genomfördes vid lärarnas arbetsplats på ett ostört ställe som lärarna själva valde. Intervjuerna av studenterna genomfördes i ett konferensrum eller ett grupprum i studenternas närhet.

Före intervjuens genomförande tillfrågades intervjupersonen om vi fick tillåtelse att använda bandspelare. Enligt Kvale (1996) kan användandet av bandspelare frigöra intervjuaren från att samtidigt som samtalen försöka göra anteckningar. För att undvika antecknandet som ett störande moment under intervjun, valde vi således att använda bandspelare.

Vid alla utom två studentintervjuer (Chalmers) genomfördes intervjuerna av oss båda, något vi dock tror inte påvekade resultatet. Intervjuerna tog mellan 20 minuter till 50 minuter att genomföra. Generellt sätt var intervjuerna med lärarna något längre än med studenterna. Intervjuerna genomfördes i februari - april 2003.

---

Gällande genomgång av kurslitteraturen ställde vi frågor till böckerna utifrån en förarbetad, strukturerad frågemall. Frågorna handlade om vad som lärs ut angående dokumentationsarbete (se Tabell 4:1-4:6).

### 3.5.2 Urval av kurslitteratur

Kriterier för vilken kurslitteratur från Chalmers som skulle komma att inkluderas i undersökningen var i första hand att kurserna skulle beröra ämnet systemutveckling. Detta krav gällde alla kurser på samtliga undersökta program. I första urvalet utgick vi således från de kurser som vi ansåg hade med systemutveckling, inklusive projektarbete att göra, varpå flertalet kurser sorterades bort. Av de kurser där innehållet ansågs beröra systemutveckling undersöktes kurslitteraturen utifrån den nämnda mallen (se Tabell 4:1-4:6) för att se vilka som behandlade dokumentering. Böcker som bl.a. behandlade programmering, projektarbete, allmänt om systemutveckling, analys och design behölls i undersökningen. Den kurslitteratur som inte verkade relevant utifrån nyss nämnda kriterier, sorterades bort, som t.ex. böcker som enbart behandlade hårdvara eller algoritmer inom programmering.

På Systemvetarprogrammet gick vi igenom alla informatikkurser som var gemensamma för studenterna. Vi gick även igenom alla obligatoriska kurser inom respektive inriktning, samt den valfria kursen Software Engineering då den bedömdes som extra intressant eftersom den hanterar storskalig programutveckling. Böckerna valdes ut enligt samma principer som nämndes ovan.

### 3.5.3 Urval av intervjupersoner

För att lärarintervjuerna på Chalmers skulle vara relevanta för vår undersökning, tog vi bland andra kontakt med olika studievägledare, prefekter, studierektorer, utbildningsledare, studeranderepresentanter och studieexpeditioner. De guidade oss vidare mot de personer som enligt dem kunde vara av intresse. Utöver det sökte vi namn i Chalmerskataloger samt på Chalmers hemsidor. För att vara aktuella för en intervju krävdes det att lärarna på ett eller annat sätt var drivande i undervisningen av systemutveckling på Chalmers. Efter en tid hade vi ringat in fem personer som alla ville medverka.

Vi önskade intervju en lärare från varje inriktning på Systemvetarprogrammet samt prefekten för Informatikinstitutionen. Prefekten var vår första intervjuperson och gav oss namnen på de tre övriga intervjupersonerna som representerade varsin inriktning på systemvetarprogrammet. Alla tackade ja till medverkan.

För att få tag på Chalmersstudenter att intervju tog vi hjälp av kurslistor som vi fick tillgång till. För att sedan få tag på studenternas e-postadresser då inga adressuppgifter eller personnummer är tillåtna på kurslistorna, utgick vi från ”Den lilla gröna: Chalmers studentkärs katalog 02/03”. Katalogen innehåller e-postadresser till de studenter som valt att offentliggöra dessa. Vi fick inte tillgång till alla kurslistor som vi önskade eftersom en del ansågs sekretessbelagda av personalen på Chalmers. Därmed blev vi ibland tvungna

att skicka förfrågan till studenter där osäkerhet rådde om de deltog i, för vår undersökning, relevant inriktning.

Bell (2000) skriver att forskare och studenter ibland får nöja sig med de intervjupersoner som vill delta, vilket även vi fick erfaras. Av Chalmersstudenterna hade vi velat intervjua fler, men vi blev tvungna att nöja oss med de som hade möjlighet att ställa upp och som var villiga att göra det (se tabell 3:1 för antal tillfrågade och antal intervjuade).

Tabell 3:1 Antal tillfrågade och antal intervjuade

	Antal utskick	Antal som inte gick fram p.g.a. stängd adress	Antal som kom fram till rätt adress	Antal som tackade ja till intervju	Antal som blev intervjuade
Studenter på D-programmet	33	9	24	4	4
Studenter på E-programmet	72*	9*	63	1	1
Studenter på S-programmet	17	2	15	11	7
	* 7 stycken av de 72 utskicken kom garanterat till studenter med inriktningen Dator- och programsystem. Av de 7 kom ingen fram till en stängd adress. Av de övriga studenterna vet vi inte vilken inriktning de har valt. Det finns 9 inriktningar att välja på.				

Patel och Davidson (1994) anser att om bortfallet närmar sig 20 % eller mer bör anledningen undersökas. Vi valde att göra en mindre bortfallsanalys för att få en viss klarhet i varför så få studenter från Elektroteknikprogrammet svarade på våra utskick. Av de 63 studenter på Elektroteknikprogrammet som inte svarade eller tackade ja till en intervju skickade vi ut e-post till elva slumpvis utvalda studenter och frågade varför de inte ville delta i undersökningen. Vi fick fem svar varav fyra stycken skrev att de gick en annan inriktning än Dator- och programsystem. En person skrev att inriktningen var den vi eftersökt, men personen i fråga menade att denne var för stressad för att kunna delta. Se tabell 3:2.

Tabell 3:2 Bortfallsanalys av studenterna på Elektroteknikprogrammet

	Antal studenter som fått förfrågan om att delta i vår undersökning och inte hade stängd adress samt inte tackat ja till intervju	Antal utskick för bortfallsanalys	Svarade: Gick annan inriktning	Svarade: Gick rätt inriktning, men hade ej tid	Ej svarat på bortfallsanalysen
Antal studenter	62	11	4	1	6

Systemvetarprogrammets studenter valdes ut slumpvis från de som vårterminen 2003 läste kursen magisteruppsats. Det slumpmässiga urvalet gick till så att vi började högst upp från listan och arbetade oss neråt. För att inte bara få namn från övre halvan av listan hoppades några namn med jämna mellanrum över för att alla på listan skulle ha en

likvärdig chans att bli utvalda i enlighet med Bells (2000) rekommendationer. Bell (2000) menar att ett slumpmässigt urval innebär att alla aktuella individer har lika stor sannolikhet att bli utvalda. Av de namn som valdes ut såg vi till att det inte var några personer vi kände samt att vi såg till att det var ungefär hälften kvinnor och hälften män.

En förfrågan om deltagande skickades ut till de utvalda studenterna från magisterkursen utifrån e-postlistan. Två personer kontaktade vi dock muntligt. Eftersom vi själva läser på institutionen för Informatik var vi noggranna med att inte få med personer vi innan kursens start var bekanta med. Sjutton förfrågningar sändes ut, varpå elva studenter tackade ja till medverkan. Av dessa elva kom sju med i undersökningen. Efter att ha genomfört sju intervjuer började intervjupersonernas svar likna varandra varpå vi ansåg oss ha en god grund för analys. Vi ämnade kontakta övriga studenter för ytterligare uppgifter om det hade behövts.

De sju studenterna täckte tillsammans in de tre fördjupningsinriktningarna som finns på Systemvetarprogrammet. En student hade gått Business Informatics, tre IT Management och tre hade gått Systems Design and Engineering.

### **3.6 Analys och tolkning**

Kvale (1996) talar om en ad hoc metod för att bearbeta, analysera och tolka kvalitativt insamlat material. Denna metod innebär att blanda olika metoder och tekniker för att strukturera och sortera materialet. En teknik kan innebära meningskoncentrering, där intervjupersonernas meningar sammanfattas i kortare uttalanden så att den väsentliga innebörden framgår i några få ord. En annan teknik kallas meningstolkning och innebär, som ovan, en strukturering av innebörden, men även samtidigt en tolkning av texterna.

Vi kan sägas ha använt en ad hoc metod för bearbetning och analys av data gällande intervjuerna. När en intervju genomförts, transkriberades den för att senare lättare kunna bearbeta och analysera materialet. Därefter läste vi igenom dem för att skapa oss ett helhetsintryck för att sedan ett flertal gånger gå tillbaka till avsnitt av särskilt intresse. På detta sätt hittade vi områden i intervjuerna som inte var uppenbara vid första genomläsningen. Då inte allt material ansågs vara av intresse, kom en del att sorteras bort.

Två viktiga faktorer i ett tolkningsarbete är att dels vara medveten om sitt tankesätt, samt att ha en kritisk inställning till de tolkningar som görs. Ett sätt att minska antalet feltolkningar är att uppmärksamma sådan vaksamhet. Tolkning handlar även om kreativitet då detta främjar upptäckandet av olika förklaringar till undersökt material. (Holme & Solvang, 1997). Vi har i enlighet med detta försökt ha ett öppet förhållningssätt i tolkningsförfarandet genom att utifrån olika perspektiv söka efter mönster som tyder på vissa samband i resultatet. På detta sätt har vi försökt undvika att söka efter bekräftelser på tidigare tankar och förutfattade slutsatser.

Efter att ha sammanställt intervjuresultatet ansåg vi att den mest rättvisa bilden av de svar som gavs innebar att redovisa i antal och inte i procentsats. Vi har således valt att skriva

t.ex. två av sju personer istället för 29 % då procentsatsen inte gjorde sig rättvisa vid så få intervjuade.

Kurslitteraturens resultat sammanställdes i tabeller utifrån varje fråga i frågemallen samt utifrån utbildning och inriktning. Då det blev för komplicerat att förstå en total sammanfattning av alla böckerna, valde vi att redogöra var bok för sig för att sedan analysera och tolka i analys- och diskussionskapitlet.

### **3.7 Trovärdighet**

Graden av vetenskaplighet kan mätas med hjälp av begreppen validitet och reliabilitet (Halvorsen, 1989). Validitetsbegreppet kan översättas med orden giltighet och relevans med avseende på resultatets samt mätinstrumentens giltighet. Validitetsbegreppet används mest av kvantitativa metoder. Reliabilitetsbegreppet å andra sidan avser undersökningsresultatets tillförlitlighet. Vid applicering av kvalitativa metoder är begreppen trovärdighet, autencitet och förståelse mer användbara än validitets- och reliabilitetsbegreppen, menar Starrin och Svensson (1996) som poängterar att reliabiliteten bör ses i sitt sammanhang och bör bedömas utifrån aktuell situation vid varje intervjutillfälle. Av denna anledning talar vi hellre om trovärdighet.

Försök att öka denna studies trovärdighet har gjorts genom att öppet redovisa tillvägagångssättet. Då vi strävat efter att skaffa oss en förståelse kring vårt problemområde valde vi en kvalitativ metod där vi fick möjlighet att skapa ett mer personligt samtal med intervjupersonerna. Vi kunde då ställa frågor direkt om saker vi inte förstod, ställa följdfrågor samt iakta hur intervjupersonerna reagerade på våra frågor. Detta hade vi inte kunnat göra på samma sätt om vi t.ex. använt oss av en enkät. Att lösa missförstånd och oklarheter på plats, omedelbart, anser vi är ett sätt att öka trovärdigheten för resultatet.

Enligt Halvorsen (1989) innebär hög reliabilitet att de olika stegen i mätprocessen ska vara precisa då reliabiliteten ska garantera att data går att lita på. Eftersom vi använt oss av frågemallar med vissa förutbestämda frågor men ändå tillåtit intervjuerna att gå in på andra saker via följdfrågor, så är det svårt att upprepa för någon annan de frågor som hamnade utanför ramen. Samtidigt som detta kan upplevas som negativt sett ur ett perspektiv där undersökningen bör kunna upprepas, så ser vi det som en styrka att på plats kunna få en möjlighet att få veta mer om intervjupersonens tankar och uppfattningar.

Vi har även arbetat för att inte styra resultatet och försökt hålla oss neutrala till arbetet, utan att medvetet lägga in aspekter av egna uppfattningar. Vi reserverar oss dock för omedvetna, subjektiva tolkningar som är svåra att upptäcka och därmed svåra att bortse ifrån och som kan ha påverkat resultatet, vår avsikt till trots.

### **3.8 Metoddiskussion och självkritik**

Kritik som Kvale (1996) tar upp mot den kvalitativa metoden med avseende på intervjuer, som även vi i viss mån håller med om, är angående resultatets

generaliserbarhet. Om det är för få personer med i intervjuerna, kan det vara svårt att göra några generaliseringar, menar Kvale (1996). En enkät hade kanske kunnat ge mer generaliserbarhet och fler personer hade kunnat involveras i studien. Det hade dock krävts av oss en mycket korrekt enkät, med otvivelaktiga frågor, vars innebörd det inte fick råda något tvivel om då vi inte skulle kunna finnas till hands och svara på några frågor. Vi hade i en enkät kunnat ställa liknande frågor och kanske många fler, men samtidigt är problemområdets karaktär av sådan art att djupare frågor inte hade kunnat ställas i enkäten. Inte heller hade vi med en enkät kunnat fråga om sådant vi inte förstår eller om sådant som verkade extra intressant, varpå viktigt material kunnat gå förlorat.

Annan kritik mot den kvalitativa metodens intervju är klassificeringen av informationen och tolkningen av den som kan vara problematisk, subjektiv och tidskrävande (Kvale, 1996). Easterby-Smith et al. (2002) skriver att forskarens tidigare kunskaper och erfarenheter påverkar denne. Ett sätt att försöka undvika att se från bara sitt eget perspektiv är att sträva efter att ha ett öppet sinne och att sträva efter att testa sina egna slutsatser från en helt annan infallsvinkel. Att försöka bortse från vår egen subjektivitet är något vi försökt ha i åtanke under arbetet med vår undersökning även om vi inser att subjektiva tankar sällan går att eliminera.

Enligt Nordlund och Rönnberg (1984) påverkas intervjun av en rad faktorer, däribland av hur intervjupersonen upplever intervjuaren och situationen i sin helhet, vilket i sin tur kan påverka resultatet. Vi försökte tänka på dessa faktorer under intervjuerna, men med våra begränsade erfarenheter inom intervjuområdet kan intervjuerna, särskilt de första, ha påverkats av vår oerfarenhet.

Då de personer vi intervjuat medverkat på frivillig basis kan resultatet ha färgats av detta i viss mån. Den möjliga populationen av studenter var mycket större i förhållande till urvalet, vilket gör det svårt att bedöma hur undersökningen hade utfallit om fler och andra personer hade medverkat. Arbetet kan ur den synvinkeln inte jämföras med en förhoppning om en allomfattande och genomgripande generaliserbarhet. Men förhoppningen har ändå hela tiden varit, att företaget som AstraZeneca av det som framkommit i denna analys, ska kunna få vissa aspekter belysta gällande vad nyutexaminerade studenter har för attityd till och kunskaper om dokumentationsarbete.

Vi såg i denna uppsats inget behov av att förklara de anledningar till att vi inte fick fler svar på studentutskicken av framför allt Elektroteknikprogrammets studenter, utöver den bortfallsanalys vi genomfört (se tabell 3:2). Däremot hade det varit intressant om fler E-programstudenter deltagit eftersom de finns representerade hos AstraZeneca. Fler E-programstudenter hade eventuellt också kunnat öka undersökningens trovärdighet. En möjlig anledning till att fler inte deltog, utöver det tabellen (se tabell 3:2) visar kan vara att en del kanske inte aktivt tittar i sin e-postlåda eller använder sig av andra e-postadresser.

## 4 Resultat

*I detta kapitel redovisar vi den empiriska undersökningens resultat. Resultatet presenteras som en sammanfattning utifrån intervjuerna samt i tabeller över genomgången kurslitteratur. Redovisningen följer de tre teman som återkommer genomgående i uppsatsen.*

### **4.1 Syn på dokumentation och dess roll i systemutvecklingsprocessen**

För att redovisa intervjupersonernas syn på dokumentation inleder vi detta avsnitt med att återge vad lärare och studenter menade med begreppet dokumentation, vilket är centralt för hur dokumentation uppfattas och vad som från lärarperspektivet präglar utbildningen, anser vi. Vi sammanfattar därefter vad som framkom kring dokumentationens syfte och betydelse. Vi ställde även frågor till lärarna angående deras erfarenheter från universitet och näringsliv då deras svar kunde ge en antydning till vilken syn på dokumentationsarbete som präglar lärarna, som i sin tur präglar studenterna.

#### **4.1.1 Vad intervjupersonerna menade med begreppet dokumentation**

På frågan vad lärarna på Chalmers menade med dokumentation gav de svar utifrån vilka dokumenttyper de avsåg. Dokumentationstyper som nämndes var; kravdokument, systemdokumentation, konceptuella modeller, testplaner, UML-diagram, användarmanualer, usecasediagram, projektplaner och tidsplaner. Av de totalt fem intervjuade lärarna från Chalmers arbetade tre med SUIT- och IPOP-kurserna, vilka är två av de mest populära kurserna på Datateknikprogrammets och Elektroteknikprogrammets inriktningar som rör systemutveckling. Dessa tre lärare menade att det fanns två typer av dokumentation vilket var en distinktion övriga Chalmerslärare inte gjorde. Dessa tre lärare menade att dokumentationen kunde delas upp i dels utvecklingsdokumentation och dels projektstyrningsdokumentation som kunde räknas in i dokumentation av betydelse för utvecklingsprocessen. Arbetssätt för framtagning av olika dokument som nämndes var med hjälp av RUP och UML-notation.

Lärarna på Systemvetarprogrammet å andra sidan, talade i huvudsak om kravdokument när det handlade om begreppet dokumentation. Två av fyra personer hänvisade till Mathiassen et als. (2001) två typer av dokumentation; analys- och designdokumentation i objektorienterad analys och design. Tekniker och modeller för framtagning av dokumentation nämndes också som t.ex. Checklands (1999) SSM och CATWOE samt i viss mån även UML-notation. Två av de fyra lärarna menade även att dokumentation är alla de uppsatser och rapporter som skrivs under utbildningens gång, däribland magisteruppsatsen.

Även studenterna gav svar kring vad de menade med dokumentation utifrån olika dokumenttyper. En av de fem intervjuade Chalmersstudenterna uttryckte att dokumentation är allt skrivet material utan att specificera närmare, medan resten av de fem Chalmersstudenterna (fyra av fem) menade att dokumentation främst var analysdokument, kravspecifikation och designspecifikation. En av de nyss nämnda fyra



menade att dokumentation är alla steg och diagram i UML, då UML-notation undervisas på Chalmers.

Andra betydelsefulla dokument som ingår i dokumentationen, som testdokumentation och projektdokumentation, nämndes av tre av fem Chalmersstudenter. Övriga var mer osäkra på om projektdokumentationen och kommenterad kod tillhörde deras definition av dokumentation, men sade samtidigt att båda dessa dokumenttyper var av betydelse för processens del. Bara en av fem menade att användarmanualen är en del av dokumentationen.

Systemvetarstudenterna menade att dokumentation är allt material från analys till implementering. Tre av sju sade att det är allt skrivet material som på något sätt förklarar vad som gjorts. Bland dem uttryckte en person; *”Dokumentering är väl nästan allt man gör, från modellering till objekt-diagram och usecase och alltihopa.”* Dokument som nämndes som viktiga av flest personer var kravspecifikation (fem av sju personer), kommentarer i koden (fyra av sju) och designdokumentation samt systemdokumentation (tre av sju).

#### **4.1.2 Intervjupersonernas syn på dokumentation i samband med systemutveckling**

Lärarna på Chalmers uttryckte alla (fem av fem) att dokumentation var av betydelse under systemutvecklingsprocessen och i underhållsarbetet som följer på utvecklingen. Två lärare på Chalmers uttryckte följande om dokumentation i samband med systemutveckling:

*”Generellt kan man väl säga när det gäller systemutveckling att alla senare faser i utvecklingen vilar på de tidigare. Så ju tidigare ett dokument ligger som är fundamentalt, ju större skada ställer det till om det inte är ordentligt.”*

*”Dokumentation är ju alla underlag du behöver för din konstruktion, de är riktningarna till systemet, de är bruksanvisningarna till systemet. Det är det i särklass viktigaste av allting. Jag brukar säga att när man ska utveckla programvara så är det väldigt viktigt att man inte tror att man ska hålla på med programmering”.*

Nyss nämnda citat uttryckte en syn på att modellering och dokumentation bör vara den största delen av utvecklingsarbetet: *”... programmeringen ska vara 5 % högst 10 % av det totala arbetet, resten ska vara att modellera och beskriva hur du har tänkt det.”* Detta gäller speciellt som utvecklingen går mot att programmerarna håller på att ersättas av kodgeneratorer, som denna person menade. Tidsfördelningen nämndes inte av någon annan men det framgick att konstruktionsarbetet och modelleringen var en stor del av vad alla lärarna på Chalmers prioriterade som viktigt i utvecklingsprocessen.

Även lärarna på Systemvetarprogrammet uttryckte en syn på dokumentation i samband med systemutveckling som kan beskriva uppgiften som betydelsefull för utvecklingsprocessen. Lärarna på Systemvetarprogrammet (tre av fyra) beskrev dokumentation som ett kommunikationsmedel, som ett sätt att förmedla sina tankar kring

vald design och en process under dokumentationsarbetet som innebär att tvingas strukturera och motivera sina tankar och beslut, vilket även flertalet av lärarna på Chalmers menade. De flesta lärarna sammantaget, d.v.s. både bland Systemvetarprogrammets lärare och bland lärare från Chalmers, menade även att dokumentationsarbete är ett sätt att minnas vad som gjorts i arbetet. Att fundera ut vad någon annan gjort eller vad man själv gjort kan bli tidsödande och dyrbart, vilket dock dokumentationen kan lindra. En lärare från Systemvetarprogrammet uttryckte:

*”Dokumentationen är viktig. Det kommer alltid ett tillfälle när du måste gå in och titta i systemet. Det spelar ingen roll om man själv varit med och byggt systemet. Man kan ha glömt efter tre veckor.”*

En av de fyra lärarna på Systemvetarprogrammet menade att dokumentationen är ett sätt att komma överens om vilka beslut som fattats och uttryckte att *”... det är ju mycket svårare om man dokumenterat ner det och alla kommit överens om att så är det, att sen säga att det har ni missuppfattat”*. Dokumentationen beskrevs vidare av två av fem Chalmerslärare som ett av de sätt systemutvecklare har att beskriva system på och de flesta andra lärare på Chalmers menade att dokumentationen tjänar som konstruktionshjälpmedel. Liknelser gjordes med arkitektens husritningar vilket även förekom bland två av Systemvetarprogrammets fyra lärare. En Chalmerslärare menade *”varför ska inte systemutvecklaren arbeta på samma sätt?”* och *”... det är ju löjligt att man tror att man får in allt i ett klassdiagram... eller i programkoden”*. En annan lärare på Chalmers sade att: *”Jämför med ett hus. Det behövs ritningar för mycket, el, rör och så vidare och om det saknas blir det inte ett funktionellt hus.”*

Bland tre av de fem lärarna på Chalmers framhövdes dokumentationen som ett medel att hitta fel i systemet genom granskning gentemot kravspecifikationen, vilket ingen av lärarna från Systemvetarprogrammet sade. Kravspecifikationen framhövdes av Chalmerslärarna som ett viktigt dokument liksom den konceptuella modellen vilket följande tyder på *”... om inte den blir bra blir ju ingenting bra”* och *”... den konceptuella modellen är väldigt dyr att ändra om den blir fel, för då måste allt ändras. Det kanske till och med blir billigare att göra om allt från början.”*

Samtidigt som dokumentation sågs som betydelsefullt poängterade flera lärare på både Chalmers och från Systemvetarprogrammet att dokumentationen bara är till nytta om den är till hjälp och syftet är att den ska användas aktivt; *”... poängen är att dokumentationen ska vara dynamisk och tillgänglig online...”* Alltför många dokument som krävs av organisatoriska skäl men som ingen läser är både onödigt, dyrbart och påverkar motivationen negativt när ingen förstår varför det ska göras, menade sammantaget lärarna från Chalmers. De flesta lärare från Systemvetarprogrammet uttryckte liknande tankar och en lärare sade att *”om man ska förstå dokumentationen likaväl som man förstår programkoden så måste den vara lättläst.”*

En lärare på Systemvetarprogrammet sade angående dokumentation att *”... det är mer komplext än man anar”*. En av fem lärare på Chalmers menade i likhet med detta att *”... dokumentationsområdet är väldigt brett, man behöver inte använda allt”*. Samma person

poängterade att det är viktigt att anpassa utvecklingsmetoden, dokumentationen inkluderad, till den produkt som görs; *"...vissa system behöver inte vissa dokument", "man ska inte ha dem som inte behövs"* och *"... egentligen, vad som ska göras är att varje företag gör sin egen metod"* samt att det är viktigt att inom företaget agera på ett likartat sätt så att det är möjligt för anställda att hoppa in i olika projekt. En av fyra lärare på Systemvetarprogrammet uttryckte något liknande; *"Det är en oerhörd vinst i om man gör på ett likartat sätt, det betyder ju att man kommunicerar saker och ting, den gemensamma kunskapen eller minnet, det går att återanvända."* Således uttalades detta med en egen arbetsmetod för varje företag samt betydelsen att arbeta på ett likartat sätt av en av Chalmerslärarna och en av Systemvetarlärarna.

Angående studenterna å andra sidan sade tre av de fem intervjuade Chalmersstudenterna att dokumentation är *"... viktigt..."* och de övriga två menade t.o.m. att; *"Dokumentation är nödvändigt..."* av olika anledningar. De anledningar som mest tydligt framkom var dokumentationens betydelse ur kommunikativt syfte, för att minnas, för att enas kring en design, för att underlätta för andra utvecklare samt för att själv förstå vad som gjorts. Två av fem menade att dokumentationen kan underlätta kommunikationen mellan utvecklare vilket gör arbetet mer effektivt. En av dessa två uttryckte; *" två och två kan man klara sig utan dokumentation, det kan bli enklare, man behöver inte vara lika stringent, men när man är uppe i fem och fyra då vinner man tid genom att göra det ordentligt."*

Samma person uttrycker dock samtidigt att det är kommunikationen i sig, inte dokumentationen, som är viktig. Dokumentationen kan vara ett verktyg för kommunikation bland flera verktyg. Att dokumentation ger mer kontroll över processen menar dock de flesta av Chalmersstudenterna. Alla diagram och dokument behöver dock inte alltid användas i alla projekt, utan det beror på hur komplext systemet är, menade två av fem av Chalmersstudenterna.

Alla sju av de intervjuade studenterna på Systemvetarprogrammet menade i likhet med Chalmersstudenterna att dokumentation är viktigt eller som en av de sju uttryckte sig *"... väldigt viktigt..."*. Fyra av sju menade dock att betydelsen av dokumentation varierar i takt med typ av system som tas fram och storleken på systemet, samt antalet involverade utvecklare. En person menade att alltför rigorös dokumentering kan kväva kreativiteten och en annan menade gällande dokumentering: *"För det mesta är det ett nödvändigt ont."* Tre av sju menade även att företagen kanske inte prioriterar dokumentation eftersom det kan skapa höga kostnader för kunden eller för det utvecklande företaget. I likhet med Chalmersstudenterna uttryckte fyra av sju av Systemvetarstudenterna att dokumentationens syfte är att underlätta för sitt eget och organisationens minne, vilka designbeslut som fattats och vad det är som gjorts av vilka personer.

Sex av sju Chalmersstudenter menade att dokumentationen är till för att nytillkomna medarbetare lättare ska kunna sätta sig in i utvecklingsprocessen eller designen. En student uttryckte kring detta: *"... hur ska folk som är nyanställda veta, eller sätta sig in i ett system och kanske vidareutveckla det, hur ska de veta alla tankegångar? Det måste vara jättekompext."* Studenten sade också: *"...för hur ska andra folk kunna sätta sig in i*

---

*system om man inte har en bra dokumentering och en bra användarmanual och sånt där?”*

En annan student från Systemvetarprogrammet, med viss erfarenhet angående dokumentation från arbetslivet, uttryckte på ett liknande sätt betydelsen av att dokumentera för andra utvecklares skull och för att själv minnas; ”*Man tror egentligen det är viktigare att få ett system som fungerar. Och det kan det vara i viss mån, men är det odokumenterat så dör det ju när man försvinner. Det är ett sätt att förlänga ett systems livslängd.*”

Två av sju Systemvetarstudenter talade om att dokumentationen kan vara mer eller mindre strukturerad. Strukturerad dokumentation, vilket en av dessa två nämnde, kan vara betydelsefullt ur effektivitetssynpunkt så att alla personer arbetar på samma sätt, där det på förhand är exakt specificerat vad som ska ingå i dokumentationen.

#### **4.1.3 Lärarnas erfarenheter från näringsliv och universitet**

På frågor kring vad lärarna hade för erfarenheter och uppfattningar från sitt arbete angående dokumentation, både inom näringsliv och inom universitet sade en av fem Chalmerslärare att denne inte funderat över det. Övriga fyra menade att branschen i sig mognat och numera lägger mer vikt vid dokumenteringen och mer vikt vid att bygga system provningsbara.

En av de fyra lärarna på Systemvetarprogrammet menade att näringslivets syn är att dokumentation är något viktigt men att det är en kostnadsfråga där dokumenteringen inte prioriteras för att kunden inte vill betala för den delen också utöver själva systemet. Dokumentationen ses således inte av kunderna som en del av systemet, enligt dessa lärare. En annan lärare menade att i näringslivet arbetas det på en mängd olika sätt och på universitetet är syftet enbart att lära studenterna en viss förståelse för dokumentationsarbete. När studenterna sedan kommer ut i arbetslivet kommer de antagligen lära om en del för att passa in i ett visst företags arbetssätt.

Vad gällde lärarnas erfarenheter från universitetet så menade de flesta lärare på Chalmers att synen på dokumentationen som viktig kommer sig av att de flesta som undervisar även är doktorander. Någon uttryckte: ”... *det kommer nog mycket av det sätt man jobbar på som doktorand och forskare, att man hela tiden måste dokumentera det man gör... så länge man inte dokumenterar det på papper så finns det inte.*”

#### **4.2 Helhetssyn och systemutvecklingsprocessen i utbildningarna**

I denna del försöker vi belysa huruvida det finns en helhetssyn i utbildningen. Denna del av resultatet handlar därför om utbildningarnas fokus och innehåll. Utbildningarnas fokus och innehåll samt vilka delar som tas upp i systemutvecklingsprocessen under utbildningen kan ge uttryck för de tankar som präglar utbildningarna och avser att kasta ljus över den syn studenterna har på dokumentation. Helhetssynen är av betydelse för upplevelsen av dokumentation som arbetsuppgift, där en avsaknad av helhetsbild kan

minska förståelsen för dokumentation som arbetsuppgift, vilket i sin tur kan påverka det som upplevs som mindre motiverande med uppgiften.

#### 4.2.1 Lärarnas syn på utbildningens fokus och innehåll

Lärarna på Chalmers undervisar i olika kurser men sammanfattningsvis är inriktningen Systemutveckling på Datateknikprogrammet och inriktningen Dator- och programsystem på Elektroteknikprogrammet präglade av att lära studenterna ett strukturerat konstruktionsarbete, lära studenterna att följa en strukturerad process och utveckla programvara samt skapa modeller. Detta försöker lärarna förmedla genom att försöka skapa en förståelse för utvecklingsarbetet som helhet samt lära studenterna att tänka långsiktigt för att förstå betydelsen av sitt eget arbete. En av fem lärare nämnde att det är viktigt att försöka förmedla vikten av att inte programmera utan istället att fundera över konstruktionsarbetet. En av fem menade dock att det är svårare att undervisa i metodik än i teknik och menade samtidigt att metodikbiten handlar om att lära ut ett professionellt sätt, ett ingenjörsmässigt sätt att arbeta. Gällande kursen IPOP är fokus på processmätning på arbetet i syfte att utvärdera, förändra och förbättra utifrån det. Fokus i kursen SUIIT är på problem som kan uppstå i projektgrupper med avseende på planeringsuppföljning gällande kvalitet och tidsmätning. Det handlar om hur olika dokument kan se ut och användas. Målet är inte att de program studenterna skapar ska vara så bra som möjligt utan målet är att studenterna ska få en så stor förståelse för utvecklingsarbete som möjligt och se vitsen med dokumentation. En lärare på Chalmers menade att *”det är en sak att veta vilka diagram man ska ha, det är viktigt att försöka lära ut hur man får fram dem, vilken ordning man ska ta fram dem, hur man ska tänka och där finns det mycket att diskutera.”*

Lärarna på Systemvetarprogrammet talade om tre olika inriktningar med tre olika fokus, men något som är gemensamt för alla inriktningarna är att förmedla en helhetsbild av systemutvecklingen utifrån olika aspekter som design, arkitektur och informationsteknologins användningsområden. Det handlar om att lära studenterna att hantera ostrukturerade och komplexa problem samt att lära dem ett angreppssätt för att kunna leverera kvalitetsprodukter. En av lärarna talade om yrket systemvetare och menade att i det ingår att se helheten:

*”Den gängse, får jag säga klichén, att en systemvetare är en programmerare, den lever och den är stark. Kommer man ut och man säger att man är systemvetare, då säger de att då kan du programmera. ....Och det är ju ett dilemma därför att en systemvetare är egentligen inte en renodlad programmerare på något sätt i mina ögon, utan en systemvetare är en person som förstår att se till helheten...”*

En lärare på Chalmers talade också om vad som ingår i yrkesrollen att utveckla system och menade att: *”Jag brukar säga att när man ska utveckla programvara så är det väldigt viktigt att man inte tror att man ska hålla på att programmera.”* Läraren menade att olika lärares syn på diverse arbetsuppgifter inom systemutveckling påverkar utbildningen, delvis bort från att handla om konstruktionsarbete och planering: *”Det finns tyvärr väldigt många universitetslärare som tycker att det kreativa arbetet är att skriva program.”*

Tre av de fyra Systemvetarlärarna menade att dokumentationen i undervisningen skulle kunna uttryckas mer nyanserat och ägnas mer tid. En av dessa tre menade samtidigt att det inte är möjligt att lägga mer tid på dokumentation p.g.a. allt annat som också ska läras ut, samt p.g.a. det stora antalet studenter som deltar i undervisningen. Istället överlämnas en del av ansvaret till studenterna själva. Den fjärde läraren uttryckte inte några tankar kring om dokumentationen fick för lite utrymme. Av de som ansåg att dokumentationen kunde uttryckas mer nyanserat, menade en person att det delvis är ett misslyckande ur kurssynvinkel att dokumentationen inte betraktas av studenterna som viktig. Två lärare menade att det kanske är ett beteende hos studenterna man borde grundlägga under utbildningens första år. Att skapa grunden för ett visst beteende redan under studietiden menar en lärare är viktigt för att: *”... om man gjort på ett sätt under hela ens studietid så tror jag inte att man automatiskt går över och gör på ett annat sätt efteråt. Vill man lägga grunden för något så ska man göra det när man studerar.”*

Som det dittills varit, sade en av Systemvetarlärarna, är det modellerna som diskuteras under första året och inte hur dokumentationen ska se ut runtomkring modellerna. En annan lärare menade att tydliggörandet av dokumentation för studenterna har sin grund i att man själv som lärare måste förstå vikten och nyttan av dokumentationen. Först därefter kan denna insikt omsättas i övningar.

#### **4.2.2 Studenternas syn på utbildningens delar**

Studenterna på Datateknikprogrammet och studenten från Elektroteknikprogrammet menade sammantaget att fokus är på konstruktionsprocesserna kring programutveckling och på att lära sig att skapa modeller, dokument av olika slag samt övning i projektarbete. En av Chalmersstudenter menade att det finns två fokus i kurserna inom Systemutvecklingsinriktningen nämligen fokus på processer och dokumentation. En annan student uttryckte fokuset på dokumentation och processer genom att mena att det nästan inte varit någon kodning, förutom i kursen Objektorienterad systemutveckling där det kom in en del kodning p.g.a. som studenten uttryckte sig *”eftersom alla är kodfixerade här”*.

Angående studenterna på Systemvetarprogrammet skiljde sig resultatet naturligt åt utifrån vilken inriktning studenterna valt. De som läste IT Management uppfattade att fokus var på analys och design i den meningen att de fick lära sig att analysera ett företag och träna sig i att se helheten. De studenter som läste Systems Design and Engineering var eniga om att i deras inriktning var programmeringsinslaget fokus.

När Systemvetarstudenterna tillfrågades om fokus i övrigt på utbildningen menade flertalet att det varit *”... svårt att hitta en röd tråd”*, vilket även den student som läste Business Informatics upplevde om sin inriktning. En av studenterna på programmeringsinriktningen menade att programmeringen och introduktionen till hårdvaran fått något för stort utrymme i förhållande till vad en systemvetare behöver veta. En annan student på Systems Design and Engineering menade att *”... på programmeringsinriktningen är man väldigt inriktad på att göra folk till programmerare”* och menade samtidigt att det bara gjorts försök i att lära ut någon

processvariant för systemutveckling. Sådana delar tycktes dock oftast komma i skymundan för de programmeringsprojekt som också skulle göras vilket "... tar bort fokus..." från de böcker som studenterna anvisats att läsa kring utvecklingsprocessen.

Alla studenterna på Systemvetarprogrammet sade att delarna i systemutvecklingsprocessen som de upplevde att de fick lära sig var analys, design och i viss mån implementering. Inte någon av de intervjuade studenterna menade att man fick lära sig något om förvaltning och underhåll av system. Två av de sju studenterna på Systemvetarprogrammet med egna erfarenheter från arbetslivet menade att förvaltning och underhåll är de största bitarna av ett systems livscykel som man som student inte kommer i kontakt med i utbildningen. En student uttryckte sig så här: *"Men den här biten efter sen som testning, förvaltning, som man märker är väldigt stor när man varit ute och jobbat, den berörs inte någonting egentligen."* En systemvetarstudent hade som förslag för att få in underhållsfasen samt aktualisera betydelsen av dokumentation att undervisningen kunde utformas på följande sätt

*"Man skulle ha en labb där man utvecklar nåt fram till en viss punkt och efter halva kursen kör man upp det, testar det och så skriver man vad som funkar och vad som inte funkar. Och sen byter labbgrupperna med varandra och byter programdokument med varandra och fortsätter arbeta från det. Nåt sånt hade varit bra. Då kan man säga till de andra vad man inte förstår. Det ska sluta med nåt som blir helt färdigt, vi har inte gjort det på nån kurs, att göra nåt helt färdigt. Det är också nåt vi aldrig gör här, färdigt."*

Frågor ställdes med syfte att undersöka vilken eventuell helhetssyn studenterna fått eller anser sig ha fått under sin utbildning. De flesta av Chalmersstudenterna talade om dokumentation som ett kommunikationsmedel och som ett sätt att underlätta för sitt eget minne och för andra systemutvecklare. Det här synsättet menade tre av fem Chalmersstudenter att de lärt sig under grupparbeten under utbildningstiden, medan övriga inte satte sin förståelse av dokumentation i samband med sådana erfarenheter. En av de studenter som uppfattade dokumentation som kommunikationsmedel uttryckte: *"Om man jämför med de projekt som vi hade i trean, som inte hade sån bra struktur, så har vi i gruppen mer koll nu när vi tar fram dokumenten. Det blir nu mycket effektivare att arbeta i grupp."*

Av Systemvetarstudenterna var det tre av sju studenter (de från inriktningen Systems Design and Engineering) som satte sin förståelse av dokumentation i samband med sina erfarenheter från utbildningen och från arbetslivet. Övriga fyra gjorde inte det på ett uttalat sätt. De tre studenterna menade sammanfattningsvis att de under kursen Komplexa IT-baserade system varit tvungna att lämna in dokument och att de så småningom upplevt behov av dokumentationen och att den blev värdefull först i efterhand. En av dessa studenter menade:

*"Man såg väl det speciellt i den här komplexa kursen, att så länge man har bra kommunikation i gruppen och man har det man gör i huvudet, då känns dokumentationen ganska överflödig. Då blir det bara en extra grej att göra och det blir ett avbrott i arbetet att skriva ner och skapa dokumenten, osmidigt att gå och titta på dem. Men om nån ny*

---

*ska sätta sig in i eller om man behöver veta vad nån utomstående har gjort då blir det väldigt jobbigt om det inte finns.”*

Två av dessa tre studenter menade att dokumentationen borde få mer eller en annan form av utrymme i undervisningen om det ändå examineras vilket följande uttrycker: *”Det är först när det går upp för en att det här har jag nytta av, som man använder det...”* och *”... vissa delar har man så sakteliga kommit fram till att; vad bra det är när detta finns, vad mycket snabbare det går när det här finns och det är ordentligt gjort.”* Dessa studenter menade att fokus ofta ligger på att få programmet färdigt eftersom det är det enda man kan vara säker på att läraren bedömer och tittar på. En annan student med erfarenhet från arbetslivet menade att: *”Ju mer man har jobbat med det ju mer inser man vikten av att dokumentera.”* och *”Första gången man är ute så vill man liksom... man har inte insett än så länge vikten av dokumentation.”*

En student från Systemvetarprogrammet menade angående hur dokumentation uppfattas av andra studenter under utbildningen: *”Jag tror ändå att man fått det förmedlat på ett sånt sätt att man inte gör det så mycket och då har man nog fått en så...att det inte är så noga med dokumentering.”*

### **4.3 Lärande och utbildning**

Denna del av resultatet handlar om intervjupersonernas uppfattningar kring vad som lärs ut i utbildningarna med avseende på former för undervisning och på innehåll utöver det som nämnts under temat Helhetssyn. Intervjupersonerna uttrycker tankar kring olika undervisningsmoment, feedback och examination samt tankar om kurslitteratur. Dessa delar är intressanta då de kan säga något om vad studenterna lär sig om dokumentationsarbete och vad som ur utbildningshänsyn bidrar till att forma deras syn på dokumentation.

#### **4.3.1 Vad som lärs ut och vad studenterna tycker de har lärt sig**

Alla nio lärare, fyra från Systemvetarprogrammet och fem från Chalmers, menade att dokumentation ingår som moment i deras kurser i en eller annan form. Former som lärarna sammantaget på Systemvetarprogrammet menade att dokumentationsinslaget i undervisningen kan ta sig är bland annat inlämning av olika modeller, kravspecifikationer, programdokumentation samt användarmanualer.

Alla lärarna på Systemvetarprogrammet samt på Chalmers menade att det finns krav på studenterna att de lämnar in olika typer av dokumentation under utbildningens gång. Chalmerslärarna ansåg även att de undervisade i ämnet. Två av lärarna på Systemvetarprogrammet menade å andra sidan att man inte speciellt undervisar i dokumentation. En av dessa uttryckte sig på följande sätt: *”Nog tycker jag att jag trycker på det med dokumentationen. Men jag tar inte upp ett särskilt avsnitt om det, vilket jag kanske måste göra. För att det ska uppfattas, trycket.”* En lärare menade att man på Systemvetarprogrammet försöker fördela dokumentation jämt i kurserna så att alla kurserna bidrar till det totala kunnandet på dokumentationsområdet. Eftersom dokumentationsarbete och systemutvecklingsprocessen är komplex krävs en gradvis



---

påbyggnad av kunskap, menade denne lärare. Två av de fyra lärarna på Systemvetarprogrammet menade dock att lärarna mest säger vad dokumentationen ska innehålla och hänvisar sedan studenterna till att själva studera t.ex. Mathiassen et al. (2001). Detta kan jämföras med att tre av totalt sju studenter på Systemvetarprogrammet uppfattade att lärarna inte utförligt gick igenom före inlämning hur dokumentation skulle se ut och vara utformad. Dessa studenters uppfattning var att det verkar förväntas av dem att de själva ska kunna söka upp resten av informationen de kan behöva i kurslitteraturen, eller att det av lärarna förutsätts att dokumentation tagits upp i tidigare kurser. En student uttryckte detta som ett problem och sade så här om en kurs (där boken studenten nämner avser Mathiassen et al., 2001):

*"... då hade vi den Objektorienterade boken och då var det ingen som gick igenom utan det var mest allmänt prat då. Och då skulle vi dokumentera sen labben efteråt, och då var det ju ingen som tittade på den dokumentationen överhuvudtaget utan vi bara gjorde den utan att veta exakt. Man hade ju läst boken då eftersom vi hade haft den till tentan då, men det var ingen som hade gått igenom att den här delen är lite bättre än andra, eller hur man kunde tänka, utan det var läs boken, gör tentan och sedan kan ni dokumentering."*

Samma student menade att: *"Där hade man kanske behövt mycket mer kunskap om dokumentation så man kunde gjort det, för vi gjorde ju mycket fel, för vi insåg ju inte vad alla delar i Mathiassens bok var för någonting..."*

Flera andra studenter upplevde dock inte detta som ett problem utan ansåg att de fick direktiv inför inlämningar. Den information de saknade kunde de söka upp menade de.

Två av de fem studenterna från Chalmers ansåg i likhet med Systemvetarstudenterna att det inte var tillräckligt klara direktiv i hur dokumenten som produceras ska se ut. Enligt en student präglades arbetssättet därför mycket av ad hoc, där studenterna utgick från vad de trodde var rätt sätt att göra på.

Flera studenter från båda lärosätena upplevde således att det var oklara riktlinjer inför inlämningarna. Samtidigt menade samtliga studenter från Chalmers när de tillfrågades om de i slutändan ansåg ha fått lära sig dokumentera att de hade det, trots att två personer ansåg att riktlinjerna inför inlämningarna varit oklara på vissa kurser. De flesta Chalmersstudenterna menade totalt sett (tre av fem) att de fått lära sig en förståelse eller en känsla för hur dokumentationsarbete kan skötas och ansåg sig ha tillräckligt med självförtroende för att kunna dokumentera i arbetslivet. En av Chalmersstudenterna nämnde t.o.m. att det nog bara är "vi" som har gått Systemutvecklingsinriktningen inom Datateknikprogrammet som är bra på att dokumentera. *"De andra blivande ingenjörerna får bara lära sig kodkommentarer, något om kravspec, men knappast något designarbete."* Samtidigt menade de dock att de inte var färdiglärda utan att lärandet fortsätter i arbetslivet, vilket en student från Systemvetarprogrammet höll med om. En Chalmersstudent sa: *"Man lär sig en förståelse. Det är inte aktuellt att lära sig allt i skolan."* och: *"Man kan inte lära sig alla tekniker och arbetssätt i skolan."*

På Systemvetarprogrammet var svaren mer oeniga. Samtidigt som tre av sju Systemvetarstudenter ansåg att riktlinjerna varit oklara menade totalt sett de flesta (fem av sju) att de fått lära sig dokumentera och hade förutsättningar för att kunna sätta sig in i ett specifikt företags sätt att dokumentera. Två av dessa fem tvekade något men menade att de åtminstone kunde tillräckligt för att klara sig i arbetslivet. De andra två utöver dessa fem uttryckte mer osäkerhet och menade att det finns moment kvar att lära, vilket följande tyder på:

*”Jag tror nog att jag har tillräcklig känsla för hur det funkar vad man behöver kunna. Men hade man gjort mer av det här i skolan och fått lite respons på det, då hade man blivit lite erfaren, nu vet man egentligen inte om... Man har en känsla för det men sen att jobba skarpt med det, det har man inte gjort.”*

Studenterna nämnde ett par dokumenttyper som de ansåg att utbildningen lärt ut och som studenterna ansåg att de fått en känsla kring hur de kan byggas och vad de kan innehålla. Följande dokumenttyper togs upp:

- kravspecifikation
- designspecifikation
- testdokumentation
- projektdokumentation
- kommentar i koden
- tidsplaner för projektarbete
- olika dokument skrivna med hjälp av UML
- usecasediagram
- klassdiagram
- analysdokumentation
- slutlig dokumentation

Dock kan tilläggas att studenterna troligen kan nämna fler dokumenttyper de anser att de fått lära sig. Dessa dokumenttyper var sådana som spontant gavs som svar utan att de tillfrågades om allt de lärt sig.

### 4.3.2 Uppfattningar kring olika undervisningsmoment

När studenterna fick frågan hur de har fått lära sig dokumentationsarbete svarade sex av de totalt tolv studenterna att de fått lära sig framför allt via handledning med feedback och att detta varit det mest värdefulla.

Olika tillvägagångssätt som lärare från båda lärosätena talade om för att lära ut dokumentationsarbete var bl.a.:

- del av föreläsning
- grupparbeten, samt att ha olika roller i en grupp (t.ex. en projektledarroll)
- att skriva rapport/ uppsats
- handledning med feedback

- via böcker
- via iterationer i systemutvecklingen (nämndes enbart av Chalmerslärare)
- inlämningsuppgifter
- göra verklighetstroga projekt
- presentera riktlinjer för hur dokument ska skrivas
- seminarier och gästföreläsningar från näringslivet som kan delge erfarenheter och från arbetslivet angående dokumentation (nämndes enbart av Chalmerslärare)
- lära olika tekniker
- öva praktiskt genom olika övningar

Då alla lärare menade att dokumentation är viktigt, tillfrågades de som följdfråga hur de förmedlade denna uppfattning till studenterna. En lärare från Systemvetarprogrammet sade: *”Ett sätt är att berätta att det finns olika tekniker för dokumentationsarbete och hur man gör, öva praktiskt, och sen använda det i labbarna men också få dem att upptäcka så småningom att man nästan inte kan leva utan.”*

En sak som uttrycktes som positivt gällande föreläsningarna var när dessa inkluderade människor från näringslivet. En Chalmerslärare sade:

*”På kursen Modern Systemutveckling har man seminarier där vi har folk från näringslivet som kommer och berättar om sin systemutveckling, till exempel från Ericsson och Volvo IT, vad de tycker är viktigt att peka på, på sistone har det varit mycket RUP. Jag tycker det är viktigt, för kursen är jobbig men då blir de lite mer motiverade för då vet de vad de måste kunna när de kommer ut.”*

Men lärarna uttryckte även problem kring föreläsningens formen när det handlar om att lära sig dokumentationsarbete. En lärare från Systemvetarprogrammet menade:

*”Vi kan ju föreläsa om det, men det går ju inte in genom att bara föreläsa om det. Man måste genom praktisk handling öva och upptäcka själv att det är ju bra. Då kommer det också sätta sig i kroppen som ett naturligt behov att jobba med, det tror jag.”*

Grupparbeten där avsikten är att öva praktiskt tillämpas både på Chalmers och på Systemvetarprogrammet. Tre av fem Chalmerslärare tog själva upp att dessa grupparbeten är till för att tjäna som praktisk övning och att uppgifterna ska vara så verkliga som möjligt genom förankring i industrin. En Chalmerslärare beskrev att i kursen SUIIT arbetar studenterna i projektgrupper, där studenterna har olika roller och ansvarsområden. Studenterna gör en projektuppgift i iterationer där utvärdering sker efter varje iteration. Studenterna får även förklara sina arbeten för andra på kursen.

Grupparbetena på Systemvetarprogrammet syftar t.ex. till att upptäcka hur komplexa systemen som ska byggas kan vara. Ett annat syfte grupparbetena har på Systemvetarprogrammet är som en av de fyra lärarna på Systemvetarprogrammet sa att i bl.a. programmets introduktionskurs tränas studenterna i att lämna över designdokumentation för ett system de har skapat. Sedan får de olika grupperna opponera på varandras arbeten. De som opponerar ska förklara hur de har tolkat arbetet för att

---

sedan få reda på om de har tolkat det hela rätt. Detta menade läraren ska ge studenterna en tidig förståelse för att tydlig dokumentation är viktig. Problemet med praktiska övningar är som en av Chalmerslärarna uttryckte det ur studenternas synvinkel:

*”Många ser vad de tjänar på att jobba med dokumentation. Men sen är det ju ändå det att dokumentation ses alltid som nåt jobbigt, det är tidskrävande, och det är inte det som konkret är produkten. I skolmiljö kan man inte göra så stora projekt att det kraschar, det funkar inte att göra så stora grejer att det rasar utan riktig dokumentation. Men vi försöker ändå göra som i IPOpen här att vi tittar inte på programkoden, det är inte viktigt för oss. Man kommer då en bit på väg när man gör dokumentationen i sig till examinationen.”*

Två av de fyra lärarna på Systemvetarprogrammet menade att det inte finns tillräckligt med tid i undervisningen för att gå in djupare på dokumentation. En lärare av fyra på Systemvetarprogrammet menade att på universitetet finns det en ambitionsnivå och ett verkligt läge gällande dokumentation i undervisningen. Ambitionsläget är att lära studenterna ett arbetssätt, men det verkliga läget är att resurser och tid begränsar det utrymme dokumentationsarbetet får i undervisningen.

Tiden fördelad på olika undervisningsmoment var tidigare ett problem på Chalmers menade de flesta lärarna (tre av fem) där. En lärare menade kring ett av projekten studenterna nu gör, att när undervisningstiden utökades blev undervisningen bättre. Antal poäng till projektet ökades och det blev möjligt att arbeta i två iterationer. Mer tid till det projektet var positivt menade läraren:

*”... första gången man gör stegen är det väldigt svårt eftersom det är nytt allting, man lär sig väldigt mycket när man gör det andra varvet, man vet vart det syftar då är det mycket lättare. Det var ett väldigt lyft när vi fick två iterationer på den här kursen. Och sen att man kan göra ett helt projekt från början till slut...”*

Projektet innebär numera även obligatorisk närvaro på den handledning som ges en gång i veckan under hela terminen vilket läraren menade förbättrade kvaliteten på det studenterna skapade och som en faktor som ökar studenternas förståelse.

Två (av fem) lärare på Chalmers och tre (av fyra) på Systemvetarprogrammet strävade efter att lära studenterna att deras dokument ska vara så pass tydliga att de kan överlämnas till någon annan utan att dokumentet misstolkas. En av Chalmerslärarna menade att dokumentation är viktigt vid systemutveckling eftersom det mycket sällan är enbart en person som utvecklar ett system. Ett sätt att öva på detta som tillämpas i utbildningen är således grupparbeten och seminarier där oppositioner avklaras.

Att lära sig förmedla ett system till en annan utvecklare ansåg alla Chalmersstudenterna att de hade fått göra. De hade dock inte fått öva sig på att överta eller överlämna ett arbete från/till någon okänd person eller till någon annan grupp utan det har enbart handlat om överlämningar/övertagningar inom den egna gruppen. Inom den egna gruppen är alla redan insatta i vad som har gjorts och vad som ska göras och det är då

enklare än om detta hade gjorts mellan grupper som inte arbetat tillsammans, menade dessa studenter. De flesta (fyra av sju) från Systemvetarprogrammet tyckte att de inte riktigt fått lära sig att förmedla en del av systemutvecklingen (med hjälp av dokumentation) till andra utvecklare. En på Systemvetarprogrammet sa: *"Så man vet ungefär hur man ska göra men man är dåligt tränad på det. Man har ju fått teorier på det."* En annan av systemvetarstudenterna sa angående om de fått lära sig förmedla en del av "systemarbetet": *"Absolut inte. Dokumentationen kommer ju till i efterhand."* och *"Man får inte lära sig att dokumentera så specifikt att en programmerare kan ta över med en gång."*

Lärarna tillfrågades även om hur de tror studenter uppfattar dokumentation i undervisningen. Samtliga av de fem Chalmerslärarna som ingick i studien menade att studenternas syn på dokumentationsarbete är olika för olika studenter. Alla Chalmerslärarna menade dock att även om studenterna under kursens gång tycker det är svårt, jobbigt och ibland t.o.m. ser skapandet av dokumentation som ett straff, så blir studenterna i allmänhet mer positiva till dokumentering efter avslutad kurs. En lärare sa att vissa tar inte till sig det vi förmedlar:

*"Men mycket av det känns som att man predikar mer än att man lär ut det. Vissa lyckas man sälja det till och de förstår det och andra märker man att de tycker att det här var jättejobbigt och försöker komma undan så lätt som möjligt med dokumenten. Det är så det är med studenter på den här nivån att det är upp till en själv om man vill ta sina poäng så lätt som möjligt eller om man vill snappa upp så mycket som möjligt och lära sig."*

En annan lärare menade att: *"Men när de är klara med kursen ser de ju att det hjälpte ju faktiskt. Det var ju inte bara att vi skulle skriva bra dokument. Det har ju faktiskt hjälpt oss också. Det tror jag de alla flesta tycker."*

Alla Lärarna på Systemvetarprogrammet menade i likhet med Chalmerslärarna att det varierar mellan studenterna och menade att det berodde på olika studenters personligheter. En lärare menade att dokumentation uppfattas av en del studenter som det sista som görs i ett projekt. Studenter betraktar inte dokumentation som viktigt eftersom de inte tycker det är lika roligt som programmering och för att de tycker det är svårt, då de ibland säger att de inte fått tillräckligt mycket information om hur dokumentationen ska se ut, menade läraren.

### 4.3.3 Uppfattningar kring feedback

Samtliga lärare som fick frågan om dokumentation någon gång examinerades samt om de gav feedback på skapad dokumentation svarade jakande, men det framkom samtidigt att det förekom varierande grader av feedback. En av Chalmerslärarna menade att ett sätt att få in dokumentationen mer i undervisningen och visa för studenterna att den är betydelsefull är att göra själva dokumentationen till det som examineras, framför t.ex. att program examineras.

---

*” I IPOpen ger vi inte respons på programmet överhuvudtaget, bara på kravspec och designspec och på projektplaneringen, det är egentligen bara på projektdata runt omkring som vi tittar på. Hur de löser själva programuppgiften det bryr vi oss egentligen inte om. Det vi examinerar och bedömer är dokumentationen.”*

Två av de nio lärarna, en från Chalmers och en från Systemvetarprogrammet, tog upp den inre stressen de upplevde med att inte riktigt hinna gå igenom all dokumentation som ska ges feedback på eller examineras. Dessa två lärare sade att de tittar på allt men inte så noggrant som de skulle önska. Chalmersläraren uttryckte att det blivit bättre med tiden, men att det för något år sedan upplevdes på följande sätt: *” Det var kanske lite slappt på den punkten, det var lite för få timmar för det, de var i alla fall en grupp på 170 studenter.”*

Chalmersläraren menade att de försökte att, när det handlade om grupparbeten, enbart ge feedback till projektledare i studentgrupper för att på så sätt ”spara” tid då det oftast går åt mindre tid till att ge feedback till en person än till en hel grupp. Genom detta förfarande kunde lärarna hålla en någorlunda hög standard på feedbacken trots stora studentgrupper.

Övriga lärare på Chalmers poängterade de krav som finns på att dokumentation ska lämnas in varje vecka inför handledning i deras kurser och menade att görs inte det på ett bra sätt får studenterna göra om. I SUIIT kursen går lärarna igenom kravdokumenten, och designdokumenten noggrant. Testplanerna samt spårbarheten mellan dokumenten kontrolleras också, menade en av Chalmerslärarna. För att inte någon av studenterna ska genomgå kursen med låg arbetsinsats finns även ett tentamensmoment. Tentan följer projektgången och: *”...har man inte gjort något i projektet så är det kört”*. Då klarar studenterna inte heller tentan.

En lärare på Systemvetarprogrammet menade att i introduktionskursen ges feedback på dokumentationen, så till vida att vissa saker krävs att det ska vara med. Läraren sa angående detta dock att lärarna vid detta tillfälle inte var så noggranna med att dokumentationen såg ut på ett visst sätt. Läraren menade att i en annan kurs denne var involverad i ges synpunkter på dokumentationen men sa samtidigt att:

*”... vi säger att dokumentationen ska innehålla vissa saker och modeller och sen har man ju då synpunkter dels på innehållet i modellerna och dels hur modellerna är gjorda rent modellmässigt. Men det blir ändå på nåt sätt inte dokumentationen och framställandet av den som blir huvudpoängen fastän att det är det som examineras.”*

En annan lärare menade att man som lärare tittar på dokumentationen, ger kommentarer och hoppas sedan att studenterna gör bättre nästa gång om det vid det aktuella tillfället var mindre bra.

En lärare på Systemvetarprogrammet uttryckte faran med att inte hinna ge bra feedback:

*” När studenterna upptäcker i vissa kurser att läraren har inte tid eller lägger inte mycket tid på att kolla, kontrollera och kommentera, då tycker de att det är meningslöst*

*kanske, det är inte roligt att satsa. De som verkligen har satsat blir besvikna på att de inte fick en reaktion och de som inte satsar de kanske tänker att varför ska vi satsa när ingen läser?”*

Studenterna på Chalmers menade att dokument som skapas i utbildningen kontrolleras av lärarna. Varje vecka skulle det lämnas in vissa dokument i vissa kurser som lärarna sedan gav feedback kring. En av fem Chalmersstudenter ansåg att feedbacken inte var tillräcklig på en av kurserna då studenten inte visste om gruppen hade gjort rätt. Övriga studenter tyckte att feedbacken fungerat bra. Två av Chalmersstudenterna nämnde att det krävs bra dokumentation för att få godkänt. Det räcker inte med att programmet fungerar. En av de två sa *”De flesta lärare säger att icke väldokumenterad lösning rättas ej.”* En student tillhörde en grupp som haft en handledare från näringslivet, vilket studenten ansåg var mycket givande ur feedbacksynpunkt.

Studenterna på Systemvetarprogrammet var delvis av en annan uppfattning än Chalmersstudenterna gällande feedback. Alla Systemvetarstudenter hade någon gång fått feedback på någon form av dokumentation, men överlag ansåg fem av sju att feedbacken i sin helhet under utbildningen inte varit tillfredställande. Tre av de fem studenterna menade att det troligen har med det stora antalet studenter att göra att feedbacken blir sämre på vissa kurser. En av de fem studenterna som menade att feedbacken varit otillfredsställande, sa: *”Har du tur så får du två rader där de har skrivit nåt. Annars får du bara ett G eller ett VG det står inte varför eller nåt sånt. Det är väldigt, väldigt vanligt.”* En annan av de fem studenterna menade angående en kurs att:

*”Man trodde att man skulle få en kommentar på den dokumentationen, men betyget bara ramlade in där plötsligt. Så man visste inte ens om de fått dokumentationen, fast att de tydligt sa att dokumentationen måste vara klar annars får ni inget betyg.”*

En annan student menade i samband med feedback att om ingen av lärarna kontrollerar och ger feedback på den så slängs dokumentationen bara ihop vilket kan få följande konsekvenser: *”Jag tror absolut att det har att göra med vad man lär sig i skolan vad man har för inställning till dokumentation. Man lär sig ju inte på att bara slänga ihop saker.”* En annan student menade att det i realiteten inte ställs några krav på hur dokumentationen ska se ut eller att den ska vara färdig vid inlämning: *”Utan till sist handlar det bara om att få labben färdig och att få den att fungera. Och det är väldigt dålig feedback på de här första stegen man tar i systemutvecklingsprocessen.”*

#### **4.3.4 Uppfattningar kring kurslitteraturen**

Angående kurslitteraturen gav lärarna en blandad bild över vald kurslitteratur. En av Systemvetarlärarna samt en lärare på Chalmers nämnde att det kan vara svårt att få tag på relevant kurslitteratur som speglar kursens innehåll. En av Chalmerslärarna menade att det saknades pedagogiska förklaringar i böckerna vilket gör att kvaliteten brister. I Chalmerskurserna IPOP och SUIT följs kursböckerna relativt exakt, vilket tyder på att böckerna där anses tillfredsställa kraven. Enligt en lärare kan vissa tillägg gällande kravspecifikationens och designspecifikationens utseende vara nödvändiga i SUIT då

dessa saknas i litteraturen. En lärare framhövde Oskarsson (1994, se bilaga 3) som en bok som enkelt förklarar hur dokument kan skapas på ett bra sätt.

Flera studenter (två från Chalmers, tre från Systemvetarprogrammet) nämnde att de lär sig utifrån böckerna. Lärare på Systemvetarprogrammet hänvisade i vissa kurser till Mathiassen et al. (2001). Denna bok nämndes även av studenter på Systemvetarprogrammet som en av de böcker som tar upp dokumentation.

#### **4.4 Resultat från genomgång av kurslitteratur**

Vi presenterar i detta avsnitt sammanställningen av kurslitteraturen i tabeller. Dessa böcker har valts ut från Systemvetarprogrammet, Datateknikprogrammet och Elektroteknikprogrammet med kriteriet att de skulle behandla systemutveckling. I resultatet kommer dessa böcker enbart beskrivas utifrån om de på något sätt behandlar dokumentation. Vi presenterar varje bok för sig för att i kapitel Analys och diskussion analysera helheten av böckernas innehåll med hänseende till dokumentationsarbete. Kurslitteraturens författare och titlar återfinns i bilaga 3.

På följande sidor presenteras nedanstående tabeller:

##### **Systemvetarprogrammet**

Tabell 4:1a	Gemensamma böcker Systemvetarprogrammet
Tabell 4:1b	Forts. gemensamma böcker Systemvetarprogrammet
Tabell 4:2	Böcker för Systemvetarprogrammet, inriktning Business Informatics
Tabell 4:3	Böcker för Systemvetarprogrammet, inriktning IT Management
Tabell 4:4a	Böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering
Tabell 4:4b	Forts. böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering
Tabell 4:4c	Forts. böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering
Tabell 4:4d	Forts. böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering
Tabell 4:5	Bok i kursen Software Engineering, Systemvetarprogrammet



**Chalmers**

De kurser som lärare, studievägledare och som studierektorn ansåg vara av stor vikt och som flest studenter läser, har vi valt att skriva till kursnamnen i tabellerna under författarnamnen på de böcker som ingår i dessa kurser. Detta gäller följande kurser IPOP, OO kursen och SUIT. Vissa kurser är ibland obligatoriska eller valfria enbart för Datateknikprogrammet och ibland enbart för Elektroteknikprogrammet. Under författarnamnen visas vilket program som haft möjlighet att läsa vilka böcker.

Tabell 4:6a	Böcker som läses/kan läsas på D- och/eller E-programmet
Tabell 4:6b	Forts. böcker som läses/kan läsas på D- och/eller E-programmet
Tabell 4:6c	Forts. böcker som läses/kan läsas på D- och/eller E-programmet
Tabell 4:6d	Forts. böcker som läses/kan läsas på D- och/eller E-programmet

Tabell 4:1a Gemensamma böcker Systemvetarprogrammet

Frågor För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Brown (2002)	OO-analys och design behandlas utförligt.	Flera modeller beskrivs.	Modellerna gäller före kodningsfasen.	För användarna samt utvecklarna. dvs sig själv och andra	Det kan bli för lite eller för mycket dokumentation.	För att arbeta fram en lösning till det system som önskas utvecklas.	Ja	Systemutvecklarna
Eklund (2002)	Framst projektplansdokument	Visar t.ex. på projektplaners innehåll.	Under utvecklingens gång.	Definieras ej, olika personer internt. externt	Beskrivs inte.	För att leda projekt.	Ja	Sågs inte direkt vem som ska göra vad.
Kroenke (2002)	Dokumenttyper nämns utan att förklara hur de skapas. Beskriver dock databasmodeller utförligt.	Databasmodeller beskrivs och visas utförligt.	Under utvecklingens gång, olika dokument i olika faser.	För användarna samt utvecklarna.	Beskrivs inte.	Kommunikation mellan alla inblandade i utvecklingen.	Ja	Systemutvecklarna

Tabell 4:1b Forts. gemensamma böcker Systemvetarprogrammet

Frågor För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. I vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Maylor (1999)	Projektokumentation	Flera modeller beskrivs.	Från början till slut.	För användarna samt utvecklarna. d.v.s. sig själv och andra	Beskrivs inte.	Höja kvaliteten på projektet. Visa kund att allt utförts korrekt. Underlätta för liknande projekt i framtiden.	Ja	De som arbetar i projektet.
Nickerson (1998)	Nämner dokumenttyper vid namn, förklarar ej hur de skapas.	Visar ex. på flödes-scheman och kod-kommentarer	Ej specificerat.	För användarna och utvecklarna.	Beskrivs inte.	För att utveckla, använda och underhålla ett IS (Informationssystem).	Ja	Utvecklarna
Skansholm (1999)	Kortfattat om analys och design.	Kod-kommentarer beskrivs utförligt.	Under utvecklingens gång	År den som läser, förändrar eller återanvänder koden.	Beskrivs inte.	För att kunna läsa, förändra eller återanvända koden.	Ja	Programmeraren gör dokumenten som hör till koden.
Sundgren (1992)	Databasmodeller, ej utförligt.	Flera modeller beskrivs.	I början för att förstå problemet.	För utvecklarna.	Beskrivs inte.	Få fram en lösning till systemet som ska utvecklas samt för kommunikation.	Ja	Utvecklarna

Tabell 4:2 Böcker för Systemvetarprogrammet, inriktning Business Informatics

Frågor För- fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Bigus & Bigus (2001)	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Turban & Aronson (2001)	Dokument som nämns är bl.a. manualer, driftsdokument, systemöversiktsdok.	Beskrivs inte.	Implementeringsfasen	Olika dokument för olika personer.	Brist på tid kan göra att dokumentationen blir dålig. Viss dokumentation kräver kunskaper som utvecklarna inte alltid besitter.	För att lättare kunna underhålla systemet.	Det anses farligt att inte dokumentera. Det förklaras ej varför det är farligt.	Rollen dokumentation wriiter nämns som en potentiell roll i ett expert-systemsteam

Tabell 4:3 Böcker för Systemvetarprogrammet, inriktning IT Management

Frågor För- fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avskifter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Boddy et al (2002)	Kort om projektplan.	Kortfattat om projektplan.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Magoulas & Pessi (1998)	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.

Tabell 4:4a Forts. böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering

Frågor	För-fattare							
	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med dokumentation?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Andersen et al (1994)	Projekt-dokument.	Ja, för en del projekt-dokumentation.	Behandlar ej systemutveckling utan projektstyrning.	För medlemmar i projektteamet.	Det kan kännas meningslöst och blir då olustigt. Gemensamma metoder, tekniker saknas.	Skapa bättre projekt där t.ex. nya medlemmar ska kunna sättas in utan problem.	Beskrivs inte i boken.	Beskrivs inte i boken.
Beck & Fowler (2000)	S.k. stories som kan liknas vid en typ av kravspec. Samt användardokument.	Ja, hur s.k. stories ska utföras. (Se fråga 1.)	Stories (Se fråga 1.) i början och användardok. i slutet.	Stories (Se fråga 1.) för programeraren och användardok. för användarna.	Beskrivs inte.	Stories (Se fråga 1.) för att ta reda på kundens önskemål. Användardok. för användning av systemet.	Beskrivs inte.	Stories (Se fråga 1.) skrivs ner av programmerare och kund. Användardok. Av programerare.
Cilliers (2002)	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.

Tabell 4:4b Forts. böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering

Frågor För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Cooper (2001)	Kod samt kommentarer i den.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Programmeraren ska kommentera koden.
Janlert & Wiberg (2000)	Allmänt kort om dok. Från kravspec. till underhållsplan. Mer om designdok.	Nej, inte mer än att de ska vara tydliga för mänskliga och dator (koden).	Designdok. vara färdiga före programmering.	Människan och datorn så att datorn förstår koden.	Beskrivs inte.	För att få bra program som även går att underhålla. Andra ska förstå programmet.	Ja	Utvecklarna.
Kelley & Pohl (1998)	Kod samt kommentarer i den.	Ja, hur det ska programmeras och kommenteras.	Kommentarer skrivs under programmering, inte efter.	För sig själv och andra utvecklare.	Om det inte kommenteras direkt glöms mycket.	Förtydliga koden.	Kommentarer anses bra att göra.	Kommentarer ska programmeraren göra.
Kernighan & Pike (1999)	Kod samt kommentarer i den.	Ja, hur det ska programmeras och kommenteras.	Kommentarer skrivs under programmering, inte efter.	För sig själv och andra utvecklare.	Om det är dåligt kommenterat blir det jobbigt för de som ska underhålla systemet.	Förtydliga koden så att text. underhållsarbete blir enklare.	Kommentarer anses viktiga att göra.	Kommentarer ska programmeraren göra.

Tabell 4:4c Forts. böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering

Frågor För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med dokumentation?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Langsam et al (1996)	Kod samt kommentarer i den.	Ja, hur det ska programmeras och kommenteras.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Macheridis (2001)	Projektdokumentation	Ja, projektrapporters innehåll beskrivs.	Det ska dokumenteras löpande under projektet.	För projektbeställaren och projektleddarna.	Beskrivs inte.	Kunskaps-spridning inom projektgruppen och organisationen. Säkra kvaliteten på projektet.	Det anses viktigt.	Beskrivs inte.
Mathiassen et al (2001)	Olika dokument inom OO analys och design.	Ja det beskrivs utförligt.	Det ska utföras löpande. Olika dokument i olika faser.	För sig själv och andra utvecklare.	Beskrivs inte.	Gemensam referensram för utvecklarna. Skapa kontinuitet, höja systemkvaliteten.	Ja	Utvecklarna
Rescher (1998)	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.



Tabell 4:4d Forts. böcker för Systemvetarprogrammet, inriktning Systems, Design and Engineering

Frågor För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Skansholm (2000)	Kommentarer i koden samt mycket kort om OO-design dok.	Kommentarer i koden samt mycket kort om OO-design dok.	Det ska planeras och dokumenteras väl innan programmering.	För andra utvecklare så att programmet kan förstås.	Beskrivs inte, men det sägs att dokument måste vara tydliga så inget missförstås.	Andra ska kunna förstå systemet.	Ja	Utvecklaren
Wooldridge (2002)	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.

Tabell 4:5 Bok i kursen Software Engineering, Systemvetarprogrammet

Frågor	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
<p><b>För-fattare</b></p> <p>Pfleeger (2001)</p>	<p>Nämner ett stort antal typer i olika faser i systemutvecklingen</p>	<p>Flera utförliga beskrivningar ges.</p>	<p>Löpande, olika dokument vid olika faser.</p>	<p>För sig själv och andra som är involverade i systemutvecklingen. Samt för användarna.</p>	<p>Beskrivs inte.</p>	<p>Höja kvaliteten på systemutvecklingen som i sin tur ger hög kvalitet på det utvecklade systemet.</p>	<p>Ja</p>	<p>Olika dokument görs av olika personer.</p>

Tabell 4:6a Forts. böcker som läses/kan läsas på D- och/eller E-programmet

Frågor / För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Burns & Davies (1993) Valfri D	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Burns & Wellings (1998) Valfri D	Kommentarer i koden.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Hudak (2000) Valfri D, E	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Humphrey (H) (1998) samt Alexander-ssons (A) före-läsningss-bilder (2001). Kurs: JPOP Valfri D, E	H: Projekt-dok. Conceptuella modeller, användardok m.m. A: Dok. på processer, samarbete i grupp, tidplaner, kravdok., designdok.	H: Ja, check-listor för olika dokument, ex. på dokument-m.innehåll m.m. A: Beskriver en del men hänvisar ofta till boken.	H: Under hela utvecklingen. Direkt efter design. A: Under hela utvecklingen, men främst i början så inga missförstånd uppstår.	H: Olika saker för olika personer, både utvecklare och användare. A: Som kommunikation i met och över tid.	H: Uppdatering svår p.g.a. mängden dokument. A: Det kan vara krävande, förslag ges till lösning.	H: Lära av misstag, återanvändning, förklara systemet, för underhåll och vidare utveckling. A: Höja kvaliteten på produkten och projektuppskattningen.	H & A: Ja	H: Utveckla. A: Utveckla och de som ingår i projektet.

Tabell 4:6b Forts. böcker som läses/kan läsas på D- och/eller E-programmet

Frågor	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Författare Humphrey (2000) Kurs: SUT Obl. D, valfri E	Olika inom systemutveckling i projektform.	Ja det beskrivs utförligt.	Löpande under hela processen.	För kund och projektmedlemmar.	För mycket text, tidpunkt att skriva i utvecklingen. Vem man skriver för.	Bl.a. reducera missförstånd och öka återanvändning.	Ja	Utvecklarna inkl. alla projektmedlemmar.
Larman (1998) Kurs: OO-systemutv. Obl. D, valfri E	OO analys- och designdok.	Beskrivs kortfattat.	Boken handlar om analys och design och de dok. ska göras då.	Olika dokument för olika personer.	Svårigheter vid skapandet av designdok.	Analysera och designa före programmering, dock ej utförligt.	Beskrivs inte.	Utvecklarna
Liang (2001) Obl. E	Kommentarer i kod, javadoc, API lite OO-designdok.	Endast hur kommentarer i kod skrivs.	Designdok. klara före programmering, kommentarer i kod under programmering.	Kommentarer i koden för andra programmerare samt för användare.	Beskrivs inte.	Kommentarer i koden för andra programmerare samt för användare ska förstå koden.	Det anses bra att dokumentera	Kodkommentarer ska programmeraren utföra.

Tabell 4:6c Forts. böcker som läses/kan läsas på D- och/eller E-programmet

Frågor För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Oskarsson (1994) Obl. D	Flera som bör finnas med i ett systemutvecklingsprojekt.	Ja, det beskrivs och förklaras.	Löpande hela tiden. Olika dokument vid olika faser.	För sig själv, andra utvecklare och för kunden.	Bl. a. mängden dokumentation (dåligt med både för mycket och för lite.)	Mer effektivt arbete samt högre kvalitet på systemet.	Ja	Utvecklarna
Pfleeger (1997) Valfri D, E	Säkerhetsplan samt lite allmänt om dokumentation.	Ja för en säkerhetsplan.	Beskrivs inte.	Olika dokument för olika personer.	Säkerhetsrisk med dålig dokumentation.	Gör systemet säkrare.	Ja	Beskrivs inte.
Pradhan (1996) Valfri D, E	Kravspec.	Nej	Löpande under arbetet.	Beskrivs inte.	Beskrivs inte.	För att undvika fel i systemet.	Ja, för att upptäcka fel.	Beskrivs inte.
Preece et al (1994) Valfri D, E	Något som kan liknas vid en kravspec.	Beskrivs inte	Beskrivs inte.	För sig själv och andra utvecklare samt för kunden.	Beskrivs inte.	För att samla in krav.	Beskrivs inte.	Utvecklarna samt även annan expertis om det behövs.

Tabell 4:6d Forts. böcker som läses/kan läsas på D- och/eller E-programmet

Frågor För-fattare	1. Vilken typ av dokumentation behandlas?	2. Beskrivs tillvägagångssätt för dokumentation?	3. Vilka faser i systemutvecklingen bör man dokumentera?	4. För vem dokumenterar man?	5. Vilka problem beskrivs med att dokumentera?	6. Vilka avsikter med dokumentation beskrivs?	7. Anses det nödvändigt att dokumentera?	8. Vem/vilka ska utföra dokumentationsarbetet?
Russel & Norvig (1995) Valfri E	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Thompson (1999) Valfri D	Kommentarer i koden.	Ja för att göra kommentarer i koden.	Under tiden man programmerar.	För sig själv och andra som vill/behöver läsa koden.	Beskrivs inte.	För att lättare förstå och förändra ett program.	Beskrivs inte.	Programmeraren
Tucker & Noonan (2002) Valfri D, E	Behandlar ej dokumentation.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.
Xiaoping (2000) Kurs: OO system-utv. Obl. D, valfri E	Analys och design i OO samt javadoc. som t.ex. kommentarer i kod, javadoc.	Ja, Analys och design i OO samt javadoc. som t.ex. kommentarer i kod, javadoc.	Kommentarer i koden under programmering.	För sig själv och andra utvecklare.	Beskrivs inte.	Beskrivs inte.	Beskrivs inte.	Utvecklarna

## 5 Analys och diskussion

*I figur 3:1 beskrev vi vår syn på angreppssätt på metoden. Vi beskrev även hur vi ville göra en återföring i form av diskussion till "Verkligheten", vilket vi försöker göra med detta kapitel. Vi ämnar i detta avsnitt besvara frågeställningen genom att analysera, tolka och diskutera resultatet i förhållande till tidigare forskning där diskussionen följer de tre teman. Det är inte vattentäta skott mellan teman gällande analys och diskussion. Teman hänger samman och går delvis in i varandra, men uppdelningen gjordes för att underlätta förståelsen av texten då den avser att ge struktur. Kapitlet avslutas med viss självreflektion och förslag till fortsatt forskning.*

Vi hade för avsikt att undersöka vilken betydelse undervisningen har inom systemutvecklingsutbildningar för studenters syn på dokumentationsarbete som arbetsuppgift. Metoden för att genomföra denna undersökning har varit av det kvalitativa slaget där syftet varit att utifrån intervjuer med lärare och studenter samt utifrån genomgång av kurslitteratur besvara frågeställningen. Undersökningen är av relevans då utbildningen kan säga något om den syn på och den förståelse av dokumentation studenterna har. Denna vetskap kan vara av intresse för AstraZeneca och andra företag i bemärkelsen vad nyutexaminerade i form av kunskap och uppfattningar kan föra med sig in i företaget. Detta kan få betydelse för planerandet av internutbildningar samt dokumentationsarbetet som arbetsprocess med avseende på effektivitet och utformning.

### **5.1 Diskussion kring syn på dokumentation och dess roll**

Synen på dokumentation som fenomen och vilken roll denna fyller i systemutvecklingsprocessen har vi beskrivit i den teoretiska referensramen för att använda detta som ett sätt att förstå vårt resultat. Synen på dokumentation och dess roll i utvecklingsprocessen tror vi uttrycker vilken betydelse denna uppgift anses ha för utvecklandet av ett system eller ett program. Den syn lärare och studenter beskriver, pekar på vilken uppfattning de kan ha kring dokumentation som arbetsuppgift. Denna syn kan få betydelse för utbildningen på så sätt att lärarna präglas av en syn som de sedan i viss mån förmedlar till studenterna.

Intervjupersonerna gav exempel på vad dokumentation som begrepp innebar genom att beskriva olika typer av dokument. De flesta överensstämde med viktiga dokument som tas upp i den teoretiska referensramen av exempelvis Sommerville (2001) och kommer därför inte diskuteras närmare. Syftet med att beskriva intervjupersonernas uppfattning om begreppet dokumentation och vilka dokument som avses, var att få en bild av vad de menade med dokumentation och huruvida denna bild skiljde sig åt mellan skolorna. Vi fann att de båda lärosätenas beskrivningar ungefärligen överensstämde, en skillnad var dock att på Chalmers talades det mer om projektdokument än vad det gjorde på Systemvetarprogrammet.

Lärarna i undersökningen uttryckte sin syn på dokumentation som att dokumentationen som sådan är viktig för utvecklingsprocessen, en uppfattning som delas av studenterna. En del studenter satte den uppfattningen i relation till erfarenheter från utbildningen. De

menade att de via lärare, kursböcker samt övningar fått bilden att dokumentation är viktigt. Vi tolkar därav att utbildningens betydelse för studenternas syn på dokumentation är att studenterna förmedlas en bild av dokumentation som viktig.

Det fanns dock nyanser i beskrivningarna kring hur viktig dokumentationen var. Någon lärare på Chalmers menade att arbetstiden fördelat på arbetsuppgifter borde vara att programmeringen bara ska vara 5 % - 10 % av utvecklingstiden. Resten ska vara konstruktionsarbete, inkluderat modellering och dokumentering. Detta skiljer sig från den teoretiska referensramen i viss mån som menar att dokumentation, om det inte handlar om system som har betydelse för människors liv och hälsa, inte bör vara mer än 5 % - 10 % av utvecklingsarbetet. Att dokumentation är en viktig del, som tidsmässigt bör beaktas i utvecklingen och som därför tas upp i särskilda kurser i utbildningen, anser vi att Chalmerslärarna i större utsträckning försökte förmedla än Systemvetarlärarna. Vi tolkar därför att det existerar en viss skillnad när det handlar om att få in dokumentation i utbildningen. Känslan från Systemvetarprogrammet är att lärarna menar att det är viktigt med dokumentation, men att det samtidigt råder viss oenighet kring hur mycket tid som kan läggas på detta moment i undervisningen, samt vilka lärare och inom vilka kurser det i så fall ska göras.

Olika anledningar till att dokumentationen ansågs vara viktig uppgavs och vår tolkning är att intervjupersonerna inte var helt eniga om att dokumentationen är en självklart tillhörande del av systemet. Studenterna menade att det är viktigt med dokumentation, men det uppfattas även av vissa som ett nödvändigt ont och som något mindre motiverande att utföra. Anledningar till detta kan vara att dokumentationsarbete upplevs som en extrauppgift och något som avbryter övrigt systemutvecklingsarbete. Bara för att det finns en vetskap om att dokumentation är viktig, är det inte säkert att uppgiften uppskattas, ses som befogad, eller att det finns kunnande i hur den kan utföras, pekar detta på menar vi.

Den teoretiska referensramen uttrycker uppfattningen att dokumentation är en del av mjukvaran och systemet i form av det som beskrivs av Nickerson (1998), Pfleeger, (1997), Pressman (2001) och Sommerville (2001). Sommerville (2001) menar att mjukvarans kvalitet hänger samman med dokumentationen och de utvecklingsprocesser som omger skapandet av mjukvaran. Visconti (1994) menar att huvudorsakerna till olika problem vid utvecklandet eller vid förvaltandet av mjukvaran beror på dålig dokumentation, ouppdaterad dokumentation eller att dokumentation saknas. Samtidigt menar Visconti (1994) att mjukvaruutveckling inte handlar om att skapa dokument utan om att skapa kvalitet, vilket dock kan göras genom att öka kvaliteten på dokumentationen. Således menar vi att det finns risker med att dokumentationen inte ses som en självklar del av systemet. En risk med att en del lärare och studenter inte uttrycker dokumentation som en självklar del av systemet kan innebära att uppgiften ses som betydligt mindre viktig än andra delar i utvecklingsprocessen. Dokumentationen kan då upplevas som en tråkig uppgift eller blir förbisedd i situationer då den tvärtom kan vara av stor betydelse. Pfleeger (1997) menar att dokumentationen är en av systemets tillgångar och att undermålig dokumentation kan utgöra en säkerhetsrisk om det handlar om datasystem vars drift och funktion kan påverka människors liv och hälsa. I



AstraZenecas fall är dokumentationen viktig av just denna anledning då den medverkar till att styra kliniska medicinska tester på människor. Ses dokumentationen inte (trots en uppfattning att dokumentation är viktig) som en självklar del av systemet i detta fall p.g.a. vad utbildningen förmedlat eller lärt ut om dokumentation kan det få negativa konsekvenser för både dokumentationshanteringen och människors hälsa, menar vi.

En annan sida av synen på förhållandet mellan dokumentation och system, kan vara att se dokumentationen som en del av systemet, men ändå inte prioritera det av anledningar som har med vissa faktorer att göra. Sådana faktorer kan vara, som någon lärare uttryckte, att kunder och användare inte ser dokumentationen som en del av systemet och därför ser dokumentationen som en extrautgift i samband med beställning och hantering av det. Lärare som präglas av sådana erfarenheter kanske påverkar sin undervisning med avseende på detta. Innebörden av det kan vara att utbildningens betydelse i den bemärkelsen blir att den förmedlar en syn på dokumentation som mindre viktig när det handlar om att prioritera kostnader och tid, menar vi. Lärarnas erfarenheter tror vi således genom undervisningen kan påverka studenterna och deras syn på dokumentation, vilket även studenterna själva trodde.

Vi menar att det kan vara av vikt i undervisningen att förmedla synen på dokumentation som en del av ett system för att dokumentationsarbetet ska tas på allvar vilket är ett problem med dokumentation som Mathiassen et al. (2001) behandlar. Författarna menar att dokumentationsarbete inte alltid tas på allvar, vilket kan medföra att t.ex. underhållsfasen blir svår att hantera då dokumentationen i föreliggande faser är undermålig.

Det kan tilläggas att intervjupersonerna hade förståelse i sin helhet för att omfattningen av dokumentationen och dess art beror på vilket system som byggs samt vad dokumentationen ska användas till. Detta överensstämmer således med Visconti (1994) som menar att det är *hur* och *vad* som dokumenteras och inte *att* det dokumenteras, som är av betydelse för mjukvarans kvalitet. Syften med dokumentationen som framkom i resultatet liknar Mathiassen et als. (2001) uppfattning där dokumentationen syftar till att tjäna som gemensam referensram för utvecklarna. Mathiassen et al. (2001) skriver att kvalitetsaspekter innebär att dokumentationen bör vara lättläst och tillgänglig samt att den ska användas aktivt. Liknande syften och kvalitetsaspekter beskrevs av lärarna och studenterna som menade att dokumentationen exempelvis tjänar som ett verktyg för överenskommelser, för att minnas och som ett kommunikationsverktyg för inblandade utvecklare.

Den teoretiska referensramen ger dokumentationen i vissa utvecklingsmodeller en betydelsefull roll i systemutvecklingsprocessens underhållsfas, då denna fas kan underlättas om dokumentationen från tidigare faser är ändamålsenligt skapad, uppdaterad och underhållen (Sommerville, 2001). Utbildningens betydelse gällande att förmedla dokumentationens betydelse för underhållsfasen i systemutvecklingsprocessen kanske finns där i teorin inom båda utbildningarna, men inte praktiskt menar vi (något vi återkommer till i Lärande och utbildning, se kapitel 5.3). Detta gäller alla program där ingen av studenterna sade att de undervisades i denna fas, vilket några av dem efterlyste.

I det fallet kan undervisningens betydelse på studenternas syn på dokumentation vara att studenterna missar insikten av betydelsen och nyttan av dokumentation, då utvecklingsfasen inte finns med som ett led i undervisningen. Det kan leda till en syn på dokumentation som mindre viktig när nyttan av den inte upplevs i denna fas.

Vi menar att Mathiassen et al. (2001) definierar god dokumentation i termerna klarhet och elegans. Klarheten avser att dokumenten ska vara lätta att läsa samt hitta mellan och inom. Elegans å andra sidan, handlar om att dokumentationen ska vara relevant samt att den ska bidra till att lösa de problem den är avsedd att lösa. Begreppens innebörd samt andra definitioner och förklaringar av god dokumentation, kan vara bra att känna till för att ha en utgångspunkt för ett dokumentationsförfarande, som vi ser det. Att sådana begrepp finns med i utbildningen är positivt då studenterna genom dessa kan komma i kontakt med förklaringar hur ändamålsenlig dokumentation kan skapas. Utbildningens betydelse för studenternas syn på dokumentation i detta avseende är att studenterna utsätts för tankar kring hur god dokumentation kan skapas, vilket kan bidra till en klarare syn på tillvägagångssätt och angreppssätt för att skapa ändamålsenlig dokumentation.

Vi anser att utbildningen försöker förmedla dokumentationens syften och dess roll i systemutvecklingsprocessen, vilket har betydelse för studenternas syn på och förståelse av dokumentation som uppgift. Det kan dock diskuteras hur starkt studenterna påverkas av det.

## **5.2 Diskussion om helhetssyn på systemutvecklingsprocessen**

Med detta tema ville vi undersöka vilken betydelse i utbildningen en eventuell närvaro/frånvaro av en helhetsbild av systemutvecklingsprocessen kan få på studenternas syn på dokumentationsarbete. Avsaknad av en helhetsbild på systemutvecklingsprocessen i utbildningen tror vi kan minska förståelsen för dokumentation som arbetsuppgift, vilket i sin tur kan påverka det som upplevs som mindre motiverande med uppgiften. Att uppgiften upplevdes som mindre motiverande var något som framkom av Alexandersson och Johanssons (2002) uppsats och är ett problem AstraZeneca vill lindra.

Resonemanget ovan kring helhetssyn, knyts samman med den teoretiska referensramen på så sätt att Sommerville (2001) motiverar behovet av en helhetsförståelse på systemutvecklingsprocessen för att den kan bidra till att lösa sådan komplex problemlösning som systemutveckling kan vara. Ytterligare en dimension, till Sommerilles (2001) tekniska beskrivning av systemutveckling vilket bidrar till komplexiteten, är vad Checkland (1999) menar den mänskliga aspekten. Denna aspekt bör beaktas vid systemutveckling då den på olika sätt kan påverka utvecklingens förutsättningar och förhållanden. Den mänskliga aspekten gör att analysmetoder i systemutveckling blir ett tillvägagångssätt för att organisera diskussioner kring lösningar. Vi menar att underlaget till denna diskussion kring lösningar i analysen och i senare skeden kan utgöras av dokumentation. Dokumentationen utgör då ett sätt att kommunicera, bl.a. med syftet att, som Checkland (1999) menar, ha kontroll över den rådande situationen och uppföljningen av den, men även för att enligt Sommerville (2001) utväxla information mellan utvecklare. Mathiassen et al. (2001) hävdar att dokumentationen även kan skapa förutsättningar för bättre samordning och för att stärka

banden mellan utvecklingarna. Bättre kommunikation kan skapas genom reflektion kring systemutvecklingsprocessen och reflektion kan således förbättra förståelsen för processer och utveckling av komplexa objekt, menar Hazzan (2002). Vi menar att en helhetsbild, med en förståelse för processens olika delar, dokumentationen inkluderad, lättare kan skapa en förståelse för nyttan med dokumentation. En förståelse för nyttan med dokumentation tror vi kan göra att arbetsuppgiften upplevs mer motiverande att utföra, speciellt också om syftet, mottagaren och läsarens kunskaper är kända.

Vi bygger resonemanget att en helhetssyn kan motivera, på Hackman och Oldhams teorier (refererade i Bakka et al., 1993) kring faktorer som påverkar arbetstillfredsställelsen. Vi menar att dessa teorier således även kan tillämpas på dokumentationsarbete. Hackman och Oldham menar exempelvis att arbetstillfredsställelse och motivation gällande arbetsuppgiften påverkas bland andra faktorer av om arbetet anses ha konkret inverkan på andra människors situation. Vi relaterar detta till undersökningens resultat och gör tolkningen att studenterna menar att det först är när nyttan med dokumentationen för sin egen skull och andra utvecklare skull, blir till en insikt som dess syfte verkligen kan förstås och uppfattas positivt. Det i sin tur påverkar motivationen inför uppgiften, menar vi.

Studenterna påpekar samtidigt, främst på Systemvetarprogrammet, att de upplever att de inte avslutar uppgifter och övningar på ett bra sätt då diverse kursmoment inte kan ges mer tidsutrymme. Vi menar att detta kan göra att helhetsperspektivet i undervisningen i praktiken får stå tillbaka p.g.a. resursbrist i form av tid, vilket kan påverka motivationen inför dokumentationsarbete negativt. Detta kan jämföras med Hackman och Oldham (refererade i Bakka et al., 1993) som menar att i vilken utsträckning arbetet kräver att den anställde gör klart ett helt arbete, från början till slut med synligt resultat, påverkar arbetstillfredsställelsen. För att optimera möjligheter att skapa god arbetstillfredsställelse bör alla fem av de aspekter Hackman och Oldham presenterar uppfyllas. För att uppfylla dem alla samtidigt bör således övriga aspekter beaktas. En aspekt som behandlar betydelse av feedback för arbetsmotivationen, återkommer vi till i temat Lärande och Utbildning (se kapitel 5.3).

Studenter med erfarenhet från arbetslivet menar att underhållsfasen är en del i yrkeslivet som är stor och viktig, men som inte behandlas i dagens utbildning. Studenterna menade att denna del skulle kunna införas på olika sätt. Övningar i utbildningen föreslås gå över längre tidsperioder där det ges möjlighet att överta andra studenters arbeten, dokumentationen inkluderad, för att fortsätta utifrån det. Ett sätt att lära sig vikten av att skapa tydliga dokument är att som i Systemvetarprogrammets introduktionskurs ha ett moment som innebär att studenter lämnar designdokument till en opponentgrupp som opponerar på lösningen och dess tydlighet. Det kan förbättra förutsättningarna för att insikten ökar att andra utvecklare måste kunna förstå ens arbete. Detta i sig kan öka chanserna att helhetssynens betydelse infinner sig i studenternas medvetande. På Chalmers ansåg lärarna att de tidigare hade problem med tidsresurserna till sina kurser, men att detta blivit bättre vilket är viktigt för studenternas förståelse. Vi tror i likhet med Hazzan (2002) att reflektion kan leda till bättre förståelse och att utbildningens betydelse är att då övningar tidsmässigt är för korta minskas studenternas möjlighet till reflektion,

vilket i sin tur kan påverka studenternas syn på dokumentation. Betydelsen, som möjlighet till reflektion i undervisningen kan ha för studenternas syn på dokumentation, är att förståelsen för uppgiften kan öka.

Gällande utbildningarnas fokus menade lärarna från båda lärosätena att de försöker förmedla en helhetsbild. Någon lärare talade om yrket att utveckla system och menade i likhet med Sommerville (2001) att en helhetssyn är viktig när det handlar om komplex problemlösning som systemutveckling innebär. På både Systemvetarprogrammet och Chalmers talades det om att lägga grunden till ett visst beteende tidigt i utbildningen då detta senare kommer att påverka studenternas yrkesutövande. Frågan kvarstår huruvida detta kan genomföras i praktiken, särskilt gällande Systemvetarprogrammet som har stort antal studenter. Utbildningens betydelse i detta sammanhang är således att det finns en möjlighet och en vilja hos de flesta lärare att genom utbildningen förmedla en helhetssyn till studenterna som gör att studenterna ser nyttan med dokumentationen. En lärare på Chalmers menade dock att det i skolmiljö inte går att göra så stora projekt att de rasar utan riktig dokumentation vilket gör att en helhetssyn i viss mån missas. Vi anser att faktorer som tidsresurser, utbildningens avsikt samt viljan hos studenterna själva, spelar in för att lära ut en helhetssyn. Dessa faktorer kan bidra till att alla studenter kanske inte efter genomgången utbildning har denna helhetssyn. Generellt sett är det fler studenter på systemvetarkurserna än vad det är på Chalmerskurserna, vilket kan påverka huruvida det är möjligt att förmedla en helhetssyn till alla. Det bör dock tilläggas att studenterna på Chalmers inte läser lika många poäng relaterade till systemutveckling totalt under sin utbildning som systemvetarstudenterna gör. Vi kan dock med denna undersökning inte säga om det har någon betydelse för vilken syn studenterna har på dokumentationsarbete.

### **5.3 Diskussion om lärande och utbildning**

Detta tema avser att belysa vad i utbildningarna, med avseende på utformning och innehåll, som bidrar till att forma studenters syn på dokumentation.

Sommerville (2001) tar upp en diskussion kring yrket software engineer och menar att software engineering är en disciplin som innefattar alla aspekter kring produktion av mjukvara. Yrkesverksamma inom software engineering har ansvar gentemot yrket och samhället i form av professionalism och de bör inte enbart beakta tekniska förhållanden och ämnen. Aspekter kring professionalism berör åtaganden att engagera sig i analys, specifikation, design, utveckling, testning och underhåll, vilket vi tolkar även inkluderar dokumentation. Vi menar att utbildningar bör präglas av en syn där dessa aspekter tas upp, vilket vi anser att även forskning efterfrågar för att svara upp mot industrins krav och behov. Hazzan (2002) menar exempelvis att studenter inte bara måste vara tekniskt kunniga utan även bör besitta kunskaper inom arkitektur, design samt även bör kunna lyssna, övertyga och vara professionella. Således bör utbildningarna inte enbart vara inriktade på tekniska delar och självklart inte enbart heller på dokumentationsarbete, då dokumentation enbart är en del av utvecklingsprocessen, som vi ser det.

Med utgångspunkt i resultatet anser vi att utbildningsinnehållet avser att ha en god avvägning mellan metodik och praktiskt utförande. Avsikten överensstämmer emellertid inte alltid med verkligheten. Någon lärare på Systemvetarprogrammet menade att strävan

efter att lära in ett visst beteende kanske inte alltid görs under utbildningen när det gäller dokumentationsarbete. Två av lärarna på Systemvetarprogrammet menade att de inte speciellt undervisas i dokumentation utan att det kommer in automatiskt i undervisningen och att fördelningen av dokumentationsinslag blir jämn utslagen på hela utbildningen. Lärare på både Chalmers och Systemvetarprogrammet menade att metodikdelen är svårare att förmedla till studenterna. Av resultatet framgår att Chalmersstudenterna sammantaget uppfattade att lärarnas undervisning i dokumentering fungerat tillfredställande, medan Systemvetarstudenterna var mer oense utifrån inriktning. Studenterna från inriktningen Systems, Design and Engineering menade att de fått för lite undervisning, för lite riktlinjer inför inlämningar, samt att deras uppfattning var att examinationen mest prioriterade genomförandet och uppvisandet av program. Någon student upplevde att det förväntades av dem efter första året att de kände till hur dokumentering kan utformas.

Humphrey (1998) betonar metoder och projektplanering som betydelsefulla för att förbättra utvecklingsprocesser och som något studenter bör lära sig som ett förberedande moment inför yrkeslivet. Vi menar att dokumentation även kan räknas dit. Ett vanligt problem är att studenter under utbildningstiden koncentrerar sig för mycket på produkten och ignorerar processen, menar Humphrey (1998). Vi menar att utbildningens betydelse för studenters syn på dokumentation kan hänga samman med att studenter mer koncentrerar sig på produkten än på processen eftersom studenterna uppfattar produkten som det som examineras i realiteten. Vad som examineras kan således ha betydelse för studenters syn på dokumentation, vilket vi återkommer till nedan. Givetvis bör det som examineras relateras till kursens innehåll. Vi anser dock att om det finns krav på att dokumentation ska lämnas in, kan dokumentationen i sig göras till det som examineras. Annars kan det resultera i att dokumentationen som en student uttryckte det: ”... *slängs ihop ...*” eller upplevs som en onödig extrauppgift, för att dokumentationen ska vara med trots att den inte egentligen examineras. En del studenter upplever att det är programmet som examineras och inte dokumentationen, trots att kravet finns att dokumentationen ska vara med. Att dokumentationen ibland upplevs som en onödig extrauppgift tror vi kan bero på olika faktorer. En faktor kan vara att studenterna i sig mer intresserar sig för andra arbetsområden inom systemutveckling och då upplever dokumentation som ett avbrott i annat arbete. En annan orsak till att studenter upplever uppgiften på ovannämnda vis kan vara att dokumentation i utbildningen förmedlas som en uppgift med lägre status beroende på inställningen hos de som utformar och påverkar undervisningen. En tredje orsak kan vara en blandning av ovan, där varken studenter eller lärare intresserar sig närmare för uppgiften vilket kan innebära att dokumentation som utbildningsmoment mer eller mindre faller bort. Detta kan vara allvarligt med hänsyn till eventuella konsekvenser som bristfällig dokumentation kan skapa och som orsakas av att dokumentationen ses som mindre relevant eller mindre motiverande att utföra.

Akademiker och yrkesutövare inom området värderar aspekter avseende utbildningarnas innehåll som olika viktiga, vilket tyder på en diskrepans mellan vad akademi och industri anser att utbildningar bör innehålla (Lee et al., 2002). Den ökade komplexiteten i systemutvecklingsprocessen och kravet från industrin på bättre kvalificerade och bättre förberedda studenter borde innebära att läroplaner inom systemutveckling relateras till

tillämpningen av ämnesområdet (Humphrey, 2000). Det viktigaste i utbildningar inom software engineering är således att förbereda studenterna för storskalig utveckling vilket kan göras genom att föra in industrirelevans, menar Wohlin och Regnell (1999). Hur det är med detta införande av industrirelevans i de båda utbildningarna kan besvaras utifrån resultatet. Chalmerslärarna menar exempelvis att detta görs via gästföreläsningar där dokumentationsförfarande inom yrkeslivet beskrivs. Andra sätt industrirelevansen finns med i undervisningen på är i form av grupparbeten där studenterna ges möjlighet att öva praktiskt, i likhet med vad Wohlin och Regnell (1999) rekommenderar. Detta gäller således civilingenjörsutbildningarna och Systemvetarprogrammet. Huruvida gästföreläsningar på Systemvetarprogrammet behandlar dokumentationsarbete framgår inte av denna undersökning. Om dokumentationsarbete inte behandlas i gästföreläsningar skulle införandet av ett sådant moment ytterligare kunna vara ett bra sätt att få in dokumentationsarbete i undervisningen på Systemvetarprogrammet, förutsatt att viljan samt tidsresurser för detta finns, anser vi. Mer involvering av industrin tror vi kan motivera studenterna, då vi instämmer med Wohlin och Regnell (1999) som menar att studenternas motivation kan öka när de upptäcker att industrin brottas med samma problem och tillämpar samma lösningar som i utbildningen. En lärare på Chalmers menade i likhet med detta att realistiska övningar samt gästföreläsare från näringslivet som berättar om sitt dokumentationsförfarande, är av betydelse då det kan höja motivationen hos studenterna. Utbildningens betydelse ur denna aspekt är att visa på utbildningens relevans i förhållandet till industrin och på så sätt bidra till en syn på dokumentation hos studenterna som innebär att de ser dokumentationsarbete som en relevant uppgift att utföra. För att involvera industrin än mer än vad som görs idag så skulle skolan även kunna komma till arbetslivet istället för att enbart arbetslivet kommer till skolan i form av gästföreläsare. Vi tror att det skulle vara nyttigt för studenterna och företagen om det på utbildningarna infördes någon form av praktik.

Vi menar att dokumentation lärs ut på ett eller annat sätt inom båda lärosätena och att det finns krav på inlämning av dokumentation, men graden av undervisning, graden av kvaliteten på uppföljningen och på den feedback som ges kan diskuteras, liksom vad som examineras. Undervisningsmoment som inkluderar handledning med feedback tyckte studenterna på båda lärosätena var positivt gällande att lära sig att dokumentera. Vi drar således slutsatsen att feedback är viktigt för studenternas syn på dokumentation.

Vår syn på feedbackens betydelse kan stärkas med Hazzans (2002) tankar som menar att studenter inte kan bli lärda vad de behöver kunna utan de kan bara bli guidade genom t.ex. feedback i en viss riktning. Även Humphrey (1998) menar att feedback uppmuntrar till motivation och fortsatt lärande och förståelse hos studenter vilket vi således anser är av betydelse när det handlar om att lära studenter ett förhållningssätt till dokumentationsarbete. Hackman och Oldham (refererade i Bakka et al., 1993) menar att feedback kan påverka arbetstillfredsställelsen på så sätt att direkt eller klar information ökar förutsättningarna för ökad tillfredsställelse. Studenterna på de olika lärosätena hade olika uppfattningar kring feedback i utbildningen. Chalmersstudenterna ansåg att den feedback de erhållit fungerat bra överlag, särskilt bra hade det fungerat menade en student som under någon kurs haft en person från näringslivet som handledare. De flesta av Systemvetarstudenterna ansåg att feedbacken i sin helhet under utbildningen fungerat

otillfredsställande. Brister i feedbacken på Systemvetarprogrammet menade både lärare och studenter beror på svårigheten från lärares sida att hinna med p.g.a. det stora antalet studenter. Dock bör tilläggas att på Chalmers är antalet studenter i aktuella kurser oftast lägre, vilket lättare möjliggör feedback. En lärare på Chalmers som tidigare upplevt liknande problem med att hinna med feedback, nämnde att försök hade gjorts när det handlade om grupparbeten, att enbart ge feedback till projektledaren i gruppen för att på så sätt ”spara” tid. Oftast går det åt mindre tid till att ge feedback till en person än till en hel grupp. Genom detta förfarande kunde lärarna hålla en någorlunda hög standard på feedbacken trots stora studentgrupper.

Wohlin och Regnell (1999) menar i likhet med detta att då utbildningarna ges för hundratalet studenter är det svårt att genomföra individuell undervisning. Vi är medvetna om detta problem. Vi menar ändå att det kan vara svårt att hålla motivationen uppe för olika inlämningsuppgifter gällande dokumentation om inga kommentarer lämnas, vilket kan påverka synen på dokumentation negativt. Dokumentationen kan ses som något onödigt, tråkigt och trivialt, vilket Mathiassen et al, (2001) hävdar kan bli ett problem i senare utvecklingsfaser.

En lärare menade att dokumentationen på Chalmers hade gjorts till ett moment som examineras. Vi tror att ett sådant tillvägagångssätt kan öka dokumentationens status då studenterna på så sätt får feedback samt lär sig betydelsen och kanske uppfattar syftet med dokumentationen. Genom att förbättra dokumentationsarbetets status kan kostnader minskas för dyra misslyckade processer som orsakats av dåligt utförd dokumentering, menar vi.

Underhållsarbete är en del som inte behandlas närmare i undervisningen, vilket vi beskrivit ovan i avsnittet om Synen på dokumentation och dess roll i systemutvecklingsprocessen (kap 5.1). Att underhållsarbete inte är en del av undervisningen kan tyda på, som Saiedian (2002) menar, att underhållsarbete ses som en andraklassuppgift. Saiedian (2002) och Lee et al. (2002) förespråkar införande av undervisning kring denna fas. Sommerville (2001) hävdar att underhållsfasen är en stor del av ett systems livscykel och denna fas påverkas av den tidigare processen. Det innebär att om dokumentationen är ändamålsenligt utförd kan den förbättra underhållsfasens förutsättningar och tvärtom. Trots underhållsfasens betydelse finns det sällan några incitament att spendera mer ekonomiska resurser under systemutvecklingsprocessen för att reducera kostnaderna av systemförändringar i framtiden (Sommerville, 2001). Vi tolkar detta som att det inte dokumenteras av den grad det skulle behöva göras, eftersom det är en dyrbar process. Den enda långsiktiga lösningen på problemet att underhållsfasen bortses ifrån är att acceptera och inte förneka att system sällan har en definierad livstid, menar Sommerville (2001). Vi menar att ett sådant synsätt kan införas i utbildningarna mer nyanserat än vad det idag gör hos de båda lärosätena för att förebygga dyra underhållskostnader i framtiden som orsakas av brister i dokumenteringen. Att införa ett sådant synsätt kan aktualisera vikten av dokumentation om ämnet då behandlas mer strukturerat i undervisningen. Vi finner det ändå vara av godo att undervisningen inom kurser på Systemvetarprogrammet försöker förmå studenter att reflektera genom att försöka påvisa hur komplex systemutveckling kan vara, vilket vi tror skapar en syn hos studenter på

dokumentation som viktig. Utbildningen försöker förmedla en helhetssyn samt att utvecklingsarbetet är en komplex process. Utbildningen försöker på så sätt förklara att dokumentation behövs, vilket kan öka studenternas förståelse för dokumentationsarbete som arbetsuppgift.

#### **5.4 Diskussion kring kurslitteraturen**

Vi anser att kurslitteraturen mycket översiktligt kan delas in i tre kategorier utifrån vad som tas upp angående dokumentering. Vi menar att den första kategorin av böcker innebär att dokumentation inte nämns då det visade sig att böckerna behandlade andra saker. En annan kategori där vi anser att flest böcker hamnar, är en där dokumentation enbart nämns och/eller där dokumentation kortfattat tas upp. Dessa böcker behandlar exempelvis kortfattat avsikten med dokumentationen och förklarar i vissa fall när dokumentation är viktig att skapa. Andra gånger behandlar böckerna inte alls varför dokumentationen behövs utan skriver bara att det är viktigt. Den tredje kategorin av böcker är en kategori som beskriver vikten av dokumentation, som visar på checklistor inför dokumentationsarbete, som förklarar vad dålig respektive ändamålsenlig dokumentation är och som exemplifierar hur dokumentation kan se ut och struktureras.

Innehållsmässigt menar vi att de böcker som behandlar dokumentation, d.v.s. böckerna i den andra och tredje kategorin, oftast beskriver dokumentering som bra eller nödvändig för utvecklingsprocessen. I de böcker som är gemensamma för Systemvetarprogrammets studenter nämns det på ett eller annat sätt att det är bra eller nödvändigt att dokumentera. Även i de kurser som är mest populära på Chalmersutbildningarna som ingår i undersökningen beskrivs dokumentation som bra eller nödvändig i de flesta böcker. Detta innebär att studenter som läser dessa kurser har en god chans att påverkas mot en positiv syn på dokumentationsarbete via det de läser i kurslitteraturen. Vi har inte stött på någon kursbok som hävdar att dokumentation inte bör göras eller är dåligt att göra. Det har dock nämnts att det i vissa lägen kan vara mindre bra att dokumentera för mycket.

Det är inte många böcker som tar upp problem med att dokumentera. Om böckerna mer skulle belysa problem som kan uppkomma vid dokumentationsarbete kanske det skulle bli lättare för studenterna att veta vad de ska tänka på vid dokumentationsarbete. Chalmersstudenterna har något oftare via kurslitteraturen kommit i kontakt med en diskussion om vad som kan vara problematiskt. Vi tycker det är bra att alla studenter på både Chalmers och Systemvetarprogrammet kommer i kontakt med flera böcker som beskriver hur dokumentation bör utföras. Alla studenter kommer även i kontakt med kurslitteratur som beskriver avsikten med att dokumentera. För studenterna blir det då mer motiverat att lära sig om de får veta syftet med det som lärs ut.

Självklart kan inte och ska inte alla böcker behandla dokumentering. Syftet med den strukturerade genomgången var att undersöka kurslitteraturens innehåll med avseende på dokumentation för att utifrån innehållet diskutera hur det tillsammans med undervisningen kan påverka studenternas syn på dokumentation. Innehållet kan således prägla studenterna på olika sätt. Att studenterna får syftet beskrivet för sig är visserligen viktigt, men det hjälper dem inte vidare när det handlar om att lära sig bra respektive dålig dokumentation och hur den kan utföras. Det finns dock böcker inom båda



lärosätena som ger tydliga exempel på hur dokumentation på ett ändamålsenligt sätt kan skapas och struktureras. Böcker som studenter och lärare nämnde i detta avseende var Oskarsson (1994), Brown (2002), Humphrey (1998), Humphrey (2000) och Mathiassen et al. (2001) (se bilaga 3).

### **5.5 Sammanfattning av analys och diskussion**

Lee et al. (2002) menar som tidigare nämnts att en student som examineras bör ha förberetts väl för den ingångsnivå som krävs för att kunna arbeta inom yrket, vilket påminner om motivet för denna undersökning. Vi hade för avsikt att undersöka vilken betydelse undervisningen har inom systemutvecklingsutbildningar för studenters syn på dokumentationsarbete som arbetsuppgift. Utbildningen kan säga något om den syn på och den förståelse av dokumentation studenterna har. Vi finner således sammanfattningsvis att studenterna har förberetts inför dokumentationsarbete i yrkeslivet på ett eller annat sätt. Vi anser likaså att undervisningen har betydelse för vilken syn studenterna får på dokumentation under utbildningen. Som vi ser det utifrån våra intervjuer samt genomgången av kurslitteraturen görs mycket från lärarnas sida för att studenterna ska få en syn som är positiv till dokumentationsarbete. Det finns dock saker i utbildningarna som kan göras bättre gällande dokumentationsarbete som vi ser det. Det bör påpekas att dokumentationsarbete enbart är en del av flera delar i systemutvecklingen och därför bör prioriteras i utbildningarna utifrån det.

Studenternas syn på dokumentation är att den teoretiskt sett är viktig. Vi menar att det är den syn som grundläggs i utbildningarna. Hur studenterna sedan tillämpar dokumentationen och var kunskapsnivån egentligen ligger, är svårare att avgöra då utgångspunkten för resultatet är studenternas subjektiva uppfattningar. De anser sig ha en uppfattning kring hur de kan göra och menar att resten kan läras i arbetslivet. Skillnaden ligger egentligen lite i självförtroendet mellan utbildningarna och även inom olika inriktningar. Chalmersstudenterna menade totalt sett att de fått lära sig en förståelse eller en känsla för hur dokumentationsarbete kan skötas och ansåg sig ha tillräckligt med självförtroende för att kunna dokumentera i arbetslivet. På Systemvetarprogrammet var svaren mer oeniga. Vi menar dock att bara för att det finns en vetskap om att dokumentation är viktig, är det inte säkert att uppgiften uppskattas av studenterna, att den ses som befogad, eller att det finns reellt kunnande i hur den kan utföras. Av analysen och diskussionen framkom att en rad faktorer påverkar studenternas syn på dokumentation vilket vi presenterar nedan:

Saker som påverkar studenterna positivt till dokumentationsarbete:

- Lärarna är medvetna om att dokumentationsarbete är viktigt vid systemutveckling
- Studenterna är medvetna att dokumentationsarbete är viktigt vid systemutveckling
- Flera böcker tar upp vikten av dokumentationsarbete
- Alla studenter kommer i kontakt med någon bok som lär ut dokumentationsarbete och vikten av att dokumentera
- Att lärarna examinerar olika dokument vid systemutveckling

Saker som påverkar studenterna negativt till dokumentationsarbete:

- Ofullständig feedback på inlämnad dokumentation
- Alla lärare hinner inte med feedback
- Stora undervisningsgrupper
- En del böcker nämner inte dokumentationsarbete eller nämner det kortfattat, vilket gör det lätt att missa
- Underhållsarbete lärs inte ut

Om källkoden var tillräckligt kommenterad skulle dokumentationsförfarandet vara överflödigt, vilket skulle kunna vara ett sätt att resonera kring dokumentation. Humphrey (1998) menar tvärtom att källkoden som sådan inte är tillräckligt tydlig för att använda som enda dokumentationsmaterial. Om källkoden används som enda dokumentation och den då är mycket lite kommenterad är den dessvärre otillgänglig för någon annan än designers och implementörer. Humphrey (2000) anser att dokumentation är en essentiell del av varje mjukvaruprodukt och är i många fall således även viktigare än själva programkoden. Någon form av överblick behöver göras, en slags avbild av systemet i form av dokumentation. Detta tror vi dock kan vara mycket svårt att skapa så att dokumentationen på ett optimalt sätt gör sig rättvisa. Alltför mycket dokumentering i alltför komplicerade modeller kan göra det som var tänkt att skapa överblick istället till något oöverblickbart menar vi, i likhet med Dawsons (2003) tankar kring Bonini's paradox. Denna paradox används som förklaring inom datorsimulering för att förklara att i försök att detaljerat beskriva något kan överblickbarheten gå förlorad då modellen blir lika svår att förstå som det den är skapad för att förenkla och förklara. Blir dokumentationen oöverblickbar tror vi att den som arbetsuppgift kan upplevas som tråkig och omotiverande, tror vi.

Det närmaste vi tror att det går att komma god dokumentation är att beakta liknande riktlinjer och begrepp Mathiassen et al. (2001) behandlar som exempelvis klarhet och elegans. Att beakta dessa begrepp är ett sätt att skapa överblickbarhet och relevans i dokumentationen. Att dokumentera för mycket kan häva kreativiteten anser vi vilket även studenter och lärare menade. Blir det alltför mycket och rigorös dokumentering kan det upplevas negativt, speciellt när mottagaren och dennes förkunskaper är okända eller om känslan finns att ingen kommer ta del av dokumentationen. Detta kan återigen jämföras med Hackman och Oldham (refererade i Bakka et al., 1993) som menar att om ens arbete inte har betydelse för någon upplevs arbetsuppgiften inte som motiverande och viktig.

Vi har tidigare nämnt att studenternas syn är av intresse för AstraZeneca och andra företag p.g.a. den kunskap och de uppfattningar det kan tänkas föra in i företaget. Vi har som åsikt att det kan vara värdefullt att veta att studenterna ser dokumentation som viktig. Det kan även vara intressant att känna till vad som på ett ungefär behandlas om dokumentation i kurslitteraturen. Annat av intresse för planerandet av exempelvis internutbildningar kan vara att känna till att studenterna upplever en viss osäkerhet kring hur dokumentation på ett ändamålsenligt sätt kan skapas. Det kan även vara värdefullt att veta vilka principer som figurerar i kurslitteraturen gällande vad som anses vara god respektive dålig dokumentering.

Vår uppfattning är att synen på dokumentation även kan ses ur ett större perspektiv. Vi anser att det bör föras en diskussion kring vilka ämnen som ska tas upp i utbildningarna, vilket även bör gälla dokumentation. Systemutveckling handlar om en hel process som kräver olika färdigheter. Tekniska delar ska inte tas för givet, som vi ser det. I det professionella ingår enligt Sommerville (2001) att stödja kollegor samt delta i livslångt lärande gällande sitt yrke samt att beakta processen, dokumentation inkluderat, då den hanteras av människor. Som ett led i att bli mer professionell skriver Sommerville (2001) att yrkesutövare bör beakta ett ständigt lärande, ett hänsynstagande till och en respekt för andra utvecklare. Vi anser även att Sommerville (2001) menar att yrket är en slags helhet som sådant, där uppgifter och ansvar inte kan avfärdas som tråkiga utan det handlar således om att uppfylla sin yrkesroll och agera professionellt, menar vi. Således ligger ansvaret att uppfylla detta hos den enskilde individen. Utbildningens roll kan dock peka på att i yrket finns arbetsuppgifter som både är roliga och tråkiga, men att de ändå måste utföras för helhetsresultatets skull. Det är ett led i att agera professionellt. Vi uppfattar Sommerville (2001) som förespråkare av att det förs en diskussion kring vilken syn som bör präglade yrket, vilket även är en diskussion som bör präglade utbildningarna. Vår uppsats kan ses som ett inlägg i en debatt som handlar om utbildningarnas innehåll, då uppsatsen hävdar att innehållet bör diskuteras och utformas utifrån vad yrket kräver av såväl tekniska som metodiska och moraliska delar. Vi menar att då dokumentation beskrivs i litteratur och av lärare som viktig så bör det i undervisningen uttryckas mer nyanserat, särskilt då det ingår i examinationen att kunna dokumentera.

### **5.6 Självreflektioner samt förslag till fortsatt forskning**

De som läst de aktuella kurserna på Chalmers har antagligen valt det för att de är intresserade av Systemutveckling och kanske även dokumentering. Det gör en jämförelse lite svårare att göra med Systemvetarprogrammet, där valfriheten inte finns på samma sätt. Lärarna på Chalmers sade att många studenter innan kursens start och under pågående kurs, upplevde kursen som tung och svår. Först efter genomgången kurs fann studenterna kursen lärorik och nyttig med avseende på dokumentationsarbete, vilket talar emot att studenterna skulle vara mycket positiva till dokumentation före kursstart.

Att antalet intervjuade studenter kan ha påverkat resultatet har redan nämnts. Något annat som kan ha påverkat resultatet är att studenterna givetvis påverkas av andra faktorer än undervisningen gällande deras syn på dokumentation. Försök gjordes att ställa frågor med avsikten att utröna om andra erfarenheter påverkat deras syn på dokumentation.

En annan faktor som kan ha påverkat undersökningen kan vara om andra utbildningar inkluderats som exempelvis Chalmers Lindholmen där IT-relaterade utbildningar ges. Vi valde dock bort dessa utbildningar då de är relativt nya och då flera är påbyggnadsutbildningar och således är fortsättningar på exempelvis Systemvetarprogrammet.

Då vi funnit att feedback inom utbildningarna har betydelse för studenternas syn på dokumentation anser vi att detta utgör ett område kring vilket fortsatt undersökning skulle kunna utföras. Undersökningen skulle kunna utföras med avseende på vad studenterna upplever otillfredsställande kring den feedback som ges, på vilka sätt

feedbackmöjligheter skulle kunna förbättras samt huruvida detta är möjligt inom utbildningens ram så som den idag är utformad.

## 6 Slutsats

Frågeställningen var:

*Vilken betydelse har undervisningen inom systemutvecklingsutbildningar för studenternas syn på dokumentation?*

Vi besvarade frågeställningen på detta sätt:

- Vi tolkar att utbildningens betydelse för studenternas syn på dokumentation är att studenterna förmedlas en bild av dokumentation som viktig, vilket gör att studenterna har en syn på dokumentation som viktig teoretiskt sett.
- Utbildningens betydelse för studenternas syn på dokumentation är att studenterna utsätts för tankar kring hur god dokumentation kan skapas, vilket bidrar till en klarare syn på tillvägagångssätt och angreppssätt för att skapa ändamålsenlig dokumentation.
- Ett par av studenterna uttryckte inte dokumentation som en självklar del av systemet vilket kan innebära att uppgiften ses som mindre viktig än andra delar i utvecklingsprocessen. Det finns dock de som direkt hävdar motsatsen. Utbildningens betydelse för en syn på dokumentation som ej tillhörande systemet tror vi delvis är ett resultat av tidigare erfarenheter hos dem som utformar undervisningen som sedan influerar undervisningen.
- Vi menar att en helhetsbild, med en förståelse för processens olika delar, dokumentationen inkluderad, lättare kan skapa en förståelse för nyttan med dokumentation. En förståelse för nyttan med dokumentation tror vi kan göra att arbetsuppgiften ses som mer motiverande att utföra. Vi menar att utbildningens betydelse ur detta perspektiv innebär att när den förmedlar dokumentationens syften genom att förmedla en helhetssyn, har den betydelse för studenternas syn på och förståelse av dokumentation.
- Examination samt feedback har betydelse för studenternas syn på dokumentationsarbete. Om dokumentation examineras och om feedback ges tror vi att studenterna kommer lära sig mer och uppleva dokumentationsarbete mer motiverande att utföra.
- Gästföreläsare och övningsuppgifter med förankring i näringslivet och som berör näringslivets dokumentationsförfarande, är av betydelse då det kan höja motivationen hos studenterna. Utbildningens betydelse ur denna aspekt är att visa på utbildningens relevans i förhållandet till industrin och på så sätt bidra till en syn på dokumentation hos studenterna som innebär att de ser dokumentationsarbete som en relevant uppgift att utföra.

Vi hoppas med denna undersökning kunna bidra med information om vad som lärs ut om dokumenteringsarbete samt information om hur undervisningen kan påverka studenternas syn på arbetsuppgiften. Studenter ser dokumentation som viktigt i teorin, men det finns hos dem en viss osäkerhet kring deras egentliga kunskaper. De anser sig ha en uppfattning om hur dokumentationsarbete kan utföras och menar att de kan lära sig resten i arbetslivet. Självförtroendet inför uppgiften skiljer sig något mellan utbildningarna. Vi menar att för att förbättra självförtroendet och för att lindra synen på dokumentation som en onödig extra arbetsuppgift, kan utbildningarna ses över med avseende främst på feedback och med avseende på vilken status dokumentationen har i utbildningarna. Vi har även i undersökningen funnit att dialogen mellan näringsliv och akademi kan förbättras gällande vad utbildningarna bör innehålla med avseende på dokumentering. Att det förekommer en skillnad mellan hur dokumentering prioriteras i näringsliv och inom akademien är något vi finner till viss del även i denna undersökning. Något vi också finner vara rimligt med hänvisning till det som framkommit i teorin, är kravet från näringslivet att höja dokumentationsarbetets status.

## Referenser

Alexandersson, M., & Johansson, A-B. (2002). *Att dokumentera vid systemutveckling – en arbetsuppgift sedd ur systemutvecklarens perspektiv* (Magisteruppsats, Göteborgs universitet, Institutionen för Informatik).

Andersson, C. (2000). *Kunskapssyn och lärande – i samhälle och arbetsliv*. Lund: Studentlitteratur.

AstraZeneca. (2003a, februari 26). *AstraZeneca R&D Mölndal* [www dokument]. URL <http://www.astrazeneca.se/iSverige/molndal.asp>

AstraZeneca. (2003b, februari 26). *Korta fakta* [www dokument]. URL <http://www.astrazeneca.se/iSverige/kortafakta.asp>

Backman, J. (1998). *Rapporter och uppsatser*. Lund: Studentlitteratur.

Bakka, J. F., Fivelsdal, E., & Lindkvist, L. (1993). *Organisationsteorier, struktur, kultur, processer*. Malmö: Liber- Hermods.

Bell, J. (2000). *Introduktion till forskningsmetodik* (3:e uppl.). Lund: Studentlitteratur.

Brookshear, J. G. (1997). *Computer Science an Overview* (5<sup>th</sup> ed.). Reading, Massachussetts, USA: Addison-Wesley.

Brown, D.W. (2002). *An Introduction to Object-Oriented Analysis: Objects and UML in Plain English*. New York, USA: John Wiley & Sons.

Bunse, C., Verlage, M., & Giese, P. (1998). Improved software quality through improved development process descriptions. *Automatica*, vol. 34, s. 23-32.

Chalmers (2003). *Chalmers programkatalog 2003* (2:a uppl.).

Checkland, P. (1999). *Systems thinking, systems practice: a 30 year retrospective*. Chichester: John Wiley & Sons.

Dawson, M. R. W. (2003, april 23) *Chapter 2: Advantages And Disadvantages Of Modeling* [www dokument]. URL [http://www.bcp.psych.ualberta.ca/~mike/Pearl\\_Street/PSYCO452/pdfstuff/Chapter2.pdf](http://www.bcp.psych.ualberta.ca/~mike/Pearl_Street/PSYCO452/pdfstuff/Chapter2.pdf)

Easterby-Smith, M., Thorpe, R., & Lowe, A. (2002). *Management Research: An introduction* (2<sup>nd</sup> ed.). London: Sage Publications.

Göteborgs Universitet. (2003). *Kurser & program 2003-2004 A-Ö*.

- Halvorsen, K. (1989). *Samhällsvetenskaplig metod* (S. Andersson övers.) Lund: Studentlitteratur.
- Hazzan, O. (2002). The reflective practitioner perspective in software engineering education. *The journal of systems and software*, vol. 63, s. 161-171.
- Holme, I. M., & Solvang, B.K. (1997). *Forskningsmetodik - om kvalitativa och kvantitativa metoder* (2:a uppl.). Lund: Studentlitteratur.
- Humphrey, W. S. (1998). *A discipline for software engineering*. Reading, Massachusetts, USA: Addison Wesley Longman.
- Humphrey, W. S. (2000). *Introduction to the team software process*. Reading, Massachusetts, USA: Addison Wesley Longman.
- Institutionen för Informatik, Göteborgs Universitet. (2003, april 11). *Utbildningsplan för systemvetarprogrammet* [www dokument]. URL [http://www.informatik.gu.se/utbildning/svp/svp\\_plan.shtml](http://www.informatik.gu.se/utbildning/svp/svp_plan.shtml)
- Kvale, S. (1996). *Den kvalitativa forskningsintervjun* (S. E. Thorsell övers.).Lund: Studentlitteratur.
- Lee, S., Koh, S., Yen, D., & Tang, H-L. (2002). Perception gaps between IS academics and IS practitioners: an exploratory study. *Information and management*, vol. 40, s. 51-61.
- Lethbridge, T. C. (2000). Priorities for the education and training of software engineers. *The journal of systems and software*, vol. 53, s. 53-71.
- London, K. R. (2000). Documentation i Ralston, A., Reilly E., & Hemmendinger, D. (Eds.), *Encyclopedia of Computer Science* (4<sup>th</sup> ed.) (s. 602-608). London, England: Nature Publishing Group.
- Mason, R. O. (1989). MIS experiments: a pragmatic perspective, in Benbasat, I. (Ed.): *The information systems research challenge: Experimental research methods*. Harvard Business School Research Colloquium, v.2, Harvard Business School, Boston, s.3-20.
- Mathiassen, L., Munk-Madsen, A., Nielsen, P.A., & Stage, J. (2001). *Objektorienterad analys och design*. Lund: Studentlitteratur.
- Nationalencyklopedin* (1991). Höganäs: Bra Böcker.
- Nickerson, R.C. (1998). *Business and information systems*. Reading, Massachusetts, USA: Addison-Wesley.
- Nordlund, O., & Rönnberg, S. (1984). *Att forska i utbildning, vård och samhälle - en introduktion*. Lund: Studentlitteratur.



Parnas, D. L., & Madey, J. (1995). Functional documents for computer systems. *Science of Computer programming*, vol. 25, s. 41-61.

Patel, R., & Davidson, B. (1994). *Forskningsmetodikens grunder: Att planera, genomföra och rapportera en undersökning*. Lund: Studentlitteratur.

Pfleeger C. P. (1997). *Security in Computing* (2<sup>nd</sup> ed.). Upper Saddle River, New Jersey, USA: Prentice Hall.

Polak, W. (2002). Formal methods in practice. *Science of Computer programming*, vol. 42, s. 75-85.

Poulymenakou, A., & Holmes, A. (1996). A contingency framework for the investigation of information systems failure. *European Journal of Information Systems*, vol. 5, s. 34-46.

Pradhan, D., K. (1996). *Fault-tolerant computer system design*. Upper Saddle River, New Jersey, USA: Prentice- Hall.

Pressman R. S. (2001). *Software engineering: A practitioner's approach* (5<sup>th</sup> ed.). New York, USA: McGraw-Hill.

Procaccino, J. D., Verner, J. M., Overmyer, S. P., & Darter, M. E. (2002). Case study: factors for early prediction of software development success. *Information and software technology*, vol. 44, s. 53-62.

Saiedian, H. (2002). Bridging academic software engineering education and industrial needs. *Computer science education*, vol. 12, s. 5-9.

Sommerville, I. (1996). *Software engineering* (5<sup>th</sup> ed.). Wokingham, England: Addison-Wesley.

Sommerville, I. (2001). *Software engineering* (6<sup>th</sup> ed.). Harlow, England: Addison-Wesley Publishers.

Starrin, B., & Svensson, P-G. (1996). *Kvalitativa studier i teori och praktik*. Lund: Studentlitteratur.

Visconti, M. A. (1994). *Software system documentation process maturity model (Documentation)* (doktorsavhandling, Oregon state university).

Wohlin, C., & Regnell, B. (1999). Strategies for industrial relevance in software engineering education. *The journal of systems and software*, vol. 49, s. 125-134.

Xiaoping, J. (2000). *Object - oriented software development using Java: principles, patterns and frameworks*. Reading, Massachusetts, USA: Addison Wesley Longman.

## Bilaga 1. Mall för intervjuunderlag till lärare

- Vilket ämne/vilka kurser undervisar du i?
- Var ligger tyngdpunkten i de kurser du undervisar i?  
Vad ska man kunna efter att man gått denna kurs?  
Förmedlas det något helhetsperspektiv kring systemutveckling?  
Är det några speciella delar i systemutvecklingsprocessen ni lär ut eller handlar det om hela?  
Förmedlas kvalitetstänkande?
- Ingår dokumentation som moment i era kurser?
- Om ja vid punkten ovan: Vad anser du om dokumentationsarbete som moment i era kurser för systemutveckling?
- Utifrån vilka kriterier bestäms kursinnehållet?  
Vem bestämmer det?  
Hur bestäms det?  
Önskemål från industrin?
- Hur ser du på dokumentering i samband med systemutveckling?  
Vilka dokument är viktiga? Varför är de viktiga?  
Om det inte är viktigt, varför tycker du så?
- Vad har du för uppfattning att den här institutionen och näringslivet har för inställning till dokumentering?
- Hur tror du studenterna ser på dokumentering?
- På vilket sätt får de lära sig om de får lära sig?  
Några särskilda modeller, tekniker?
- Får studenterna lära sig att förmedla sitt system till t.ex. en annan systemutvecklare som ska underhålla systemet eller till en användare?
- Examination av dokumentation
- Feedback på inlämnat material?
- Speglar kurslitteraturen och kursinnehållet det studenterna bör lära sig kring dokumentation som du ser det? (om intervjupersonen anser att studenter behöver kunna dokumentera)  
Tar du upp/hoppar över nåt annat än det som står i böckerna?  
Vem väljer kurslitteraturen?
- Vet du någon annan kurs som lägger extra tyngd vid dokumentering?

## Bilaga 2. Mall för intervjuunderlag till studenter

- Vilken inriktning går du på din utbildning?
- Har du arbetat inom denna bransch tidigare?
- Vilka delar i systemutvecklingsprocessen har du fått lära dig?
- Var ligger fokus på det ni får lära er?
- Får ni lära er att förmedla era system till andra utvecklare?  
Om ni får det, på vilket sätt ska det förmedlas?
- Vad har ni gjort för att bibehålla förståelse för de program ni utvecklat?
- Vad menar du med dokumentation? Vilka dokument är viktiga tycker du? (Om intervjupersonen anser att det finns några viktiga)
- Får ni lära er att dokumentera ett system?  
Vilka typer av dokument handlar det om då?
- Hur ser du på dokumentation i samband med systemutveckling?
- Tror du att det du får lära dig är det du behöver kunna när du börjar arbeta när det gäller dokumentation?
- Vad är det för rapporter och dokumentation ni lämnar in?
- Får ni lära er hur dessa ska se ut?
- Tekniker/modeller?
- Får ni feedback på dessa rapporter?  
Examineras de?
- Har du något mer att tillägga?

## Bilaga 3. Litteraturlista över undersökta böcker

Efter förlag beskrivs kurs, inriktning och program som boken tillhörde.

Alexandersson, R. (2001). *Föreläsningbilder IPOP* (Chalmers tekniska högskola, 412 96 Göteborg). Komplement till kurslitteratur för kursen Individuell programvaruutveckling och processförbättring (IPOP). Valfri kurs på D-programmet och E-programmet.

Andersen, E. S., Grude, K. V., & Haug, T. (1994). *Målinriktad projektstyrning*. Lund: Studentlitteratur. Kursbok i kursen Komplexa IT-baserade system, inriktning Systems Design and Engineering på S-programmet.

Beck, K., & Fowler, M. (2000). *Planning extreme programming*. USA: Addison-Wesley. Kursbok i kursen Komplexa IT-baserade system, inriktning Systems Design and Engineering på S-programmet.

Bigus, P., & Bigus, J. (2001). *Constructing intelligent agents using java* (2nd ed). New York: John Wiley & Sons. Kursbok i kursen Beslutsstödsystem, inriktning Business Informatics på S-programmet.

Boddy, D., Boonstra, A., & Kennedy, G. (2002). *Managing Information Systems an organisational perspective*. Harlow, England: Pearson Education. Kursbok i kursen Informationssystemmiljöer och kursen IS/IT planering och ledning. Båda kurserna inom inriktningen IT Management på S-programmet.

Brown, D.W. (2002). *An Introduction to Object-Oriented Analysis: Objects and UML in Plain English*. New York, USA: John Wiley & Sons. Kursbok i kursen Introduktion till informatik som alla läser på S-programmet.

Burns A., & Davies, G. (1993). *Concurrent Programming*. Harlow, England: Addison-Wesley. Kursbok i kursen Parallell programmering och kursen Realtidssystem, valfria kurser på D-programmet.

Burns A., & Wellings, G. (1998). *Concurrency in Ada (2<sup>nd</sup> ed.)*. Cambridge, Storbritannien: Cambridge University Press. Kursbok i kursen Parallell programmering, valfria kurs på D-programmet.

Cilliers, P. (2002). *Complexity and postmodernism*. London: Routledge. Kursbok på kursen Komplexa IT-baserade system, inriktning Systems Design and Engineering på S-programmet.

Cooper, W., J. (2001). *Java design patterns*. Boston , USA: Addison-Wesley. Kursbok på kursen Komplexa IT-baserade system, inriktning Systems Design and Engineering på S-programmet.

Eklund, S. (2002). *Arbeta i projekt – en introduktion*. Lund: Studentlitteratur. Kursbok i kursen Databaser och systemutveckling som alla läser på S-programmet.

Hudak, P. (2000). *The Haskell school of expression: learning functional programming through multimedia*. Cambridge, England: Cambridge University Press. Kursbok i kursen Avancerad funktionell programmering, valfria kurs på D-programmet.

Humphrey, W. S. (1998). *A discipline for software engineering* . Reading, Massachusetts, USA: Addison Wesley Longman. Kursbok i kursen Individuell programvaruutveckling och processförbättring (IPOP), valfria kurs på D-programmet och E-programmet.

Humphrey, W. S. (2000). *Introduction to the Team Software Process*. Reading, Massachusetts, USA: Addison-Wesley. Kursbok i kursen Systemutveckling i team (SUIT), obligatorisk kurs på D-programmet (inom inriktningen systemutveckling) och valfria kurs på E-programmet.

Janlert, L-E., & Wiberg, T. (2000). *Datatyper och algoritmer (2:a uppl.)*. Lund: Studentlitteratur. Kursbok i kursen Avancerad programkonstruktion, inriktning Systems Design and Engineering på S-programmet.

- Kelley, A., & Pohl, I. (1998). *A book on C. Programming in C (4<sup>th</sup> ed.)*. Reading, Massachusetts, USA: Addison-Wesley. Kursbok i kursen Avancerad programkonstruktion, inriktning Systems Design and Engineering på S-programmet.
- Kernighan, B.W., & Pike, R. (1999). *The practice of programming*. Reading, Massachusetts, USA: Addison-Wesley. Kursbok i kursen Avancerad programkonstruktion, inriktning Systems Design and Engineering på S-programmet.
- Kroenke, D. M. (2002). *Database concepts*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kursbok i kursen Databaser och systemutveckling som alla läser på S-programmet.
- Langsam, Y., Augenstein, M. J., & Tenenbaum, A. M. (1996). *Data Structures Using C and C++ (2<sup>nd</sup> ed.)*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kursbok i kursen Avancerad programkonstruktion, inriktning Systems Design and Engineering på S-programmet.
- Larman, C. (1998). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kursbok i kursen Objektorienterad systemutveckling, obligatorisk kurs på D-programmet (inom inriktningen systemutveckling) och valfri kurs på E-programmet.
- Liang, Y. D. (2001). *Introduction to Java programming (3<sup>rd</sup> ed.)*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kursbok i kursen Programmeringsteknik, obligatorisk på E-programmet.
- Macheridis, N. (2001). *Projektspekter: kunskapsområden för ledning och styrning av projekt*. Lund: Studentlitteratur. Kursbok på kursen Komplexa IT-baserade system, inriktning Systems Design and Engineering på S-programmet.
- Magoulas, T., & K. Pessi (1998). *Strategisk IT-management* (avhandling för doktorsexamen, Göteborgs universitet). Kursbok i kursen Informationssystemmiljöer, inriktningen IT Management på S-programmet.
- Mathiassen, L., Munk-Madsen, A., Nielsen, P. A., & Stage, J. (2001). *Objektorienterad analys och design (2:a uppl.)*. Lund: Studentlitteratur. Kursbok i kursen Objektorienterad systemutveckling, inriktning Systems Design and Engineering på S-programmet.
- Maylor, H. (1999). *Project Management (2<sup>nd</sup> ed.)*. Harlow, England: Pearson Education. Kursbok i kursen Informationsteknologi som alla läser på S-programmet.
- Nickerson, R.C. (1998). *Business and information systems*. Reading, Massachusetts, USA: Addison-Wesley. Kursbok i kursen Introduktion till informatik som alla läser på S-programmet.
- Oskarsson, Ö. (1994). *Programutveckling i liten skala – en praktisk handbok*. Lund: Studentlitteratur. Kursen Projektkurs, obligatorisk kurs på D-programmet (inom inriktningen systemutveckling).
- Pfleeger, C. P. (1997). *Security in Computing (2<sup>nd</sup> ed.)*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kursbok i kursen Tillämpad datasäkerhet, valfri kurs på D-programmet och E-programmet.
- Pfleeger, S. L. (2001). *Software Engineering: Theory and Practice. (2<sup>nd</sup> ed.)*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kursbok i kursen Software Engineering, valfri kurs på S-programmet.
- Pradhan, D. K. (1996). *Fault-tolerant computer system design*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kursbok i kursen Tillförlitliga och feltoleranta datorsystem, valfri kurs på D-programmet och E-programmet.

- Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human computer interaction*. Harlow, England: Addison-Wesley. Kursbok i kursen Människa- datorinteraktion, valfri kurs på D-programmet och E-programmet.
- Rescher, N. (1998). *Complexity, a philosophical overview*. New Jersey: Transaction Publishers. Kursbok på kursen Komplexa IT-baserade system, inriktning Systems Design and Engineering på S-programmet.
- Russel, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jersey, USA: Prentice Hall. Kurbok i kursen Artificiell intelligens, valfri kurs på E-programmet.
- Skansholm, J. (1999). *Java direkt*. Lund: Studentlitteratur. Kursbok i kursen Programkonstruktion som alla läser på S-programmet.
- Skansholm, J. (2000). *C++ direkt (2:a uppl.)*. Lund: Studentlitteratur. Kursbok i kursen Avancerad programkonstruktion, inriktning Systems Design and Engineering på S-programmet.
- Sundgren, B. (1992). *Databasorienterad systemutveckling: Grundläggande begrepp, Datamodellering, Systemkonstruktion*. Lund: Studentlitteratur. Kursbok i kursen Databaser och systemutveckling som alla läser på S-programmet.
- Thompson, S. (1999). *Haskell: the craft of functional programming (2<sup>nd</sup> ed.)*. Harlow, England: Addison Wesley Longman. Kursbok i kursen Avancerad funktionell programmering, valfri kurs på D-programmet.
- Tucker, A., & Noonan, R. (2002). *Programming languages principles and paradigms*. New York: The MacGraw-Hill Companies. Kursbok i kursen Programspråk, valfri kurs på D-programmet och E-programmet.
- Turban, E., & Aronson, J. E. (2001). *Decision support systems and intelligent systems (6<sup>th</sup> ed.)*. New Jersey: Prentice Hall. Kursbok i kursen Beslutsstödsystem, inriktning Business informatics på S-programmet.
- Wooldridge, M. (2002). *An introduction to multiagent systems*. West Sussex, England: John Wiley & Sons. Kursbok i kursen Komplexa IT-baserade system, inriktning Systems Design and Engineering på S-programmet.
- Xiaoping, J. (2000). *Object - oriented software development using Java: principles, patterns and frameworks*. Reading, Massachusetts, USA: Addison Wesley Longman. Kursbok i kursen Objektorienterad systemutveckling, obligatorisk kurs på D-programmet (inom inriktningen systemutveckling) och valfri kurs på E-programmet.