

Civilingenjör i Mjukvaruteknik vid Linköpings Universitet – Mål, Design och Erfarenheter

Fredrik Heintz och Inger Erlander Klein
Linköpings universitet

Abstract—Hösten 2013 startade Linköpings universitet den första civilingenjörutbildningen i Mjukvaruteknik. Utbildningens mål är att bland annat ge ett helhetsperspektiv på modern storskalig mjukvaruutveckling, ge en gedigen grund i datavetenskap och computational thinking samt främja entreprenörskap och innovation. Studenternas gensvar har varit över förväntan med över 600 sökande till de 30 platserna varav 134 förstahandssökande. Här presenterar vi programmets vision, mål, designprinciper samt det färdiga programmet. En viktig förebild är ACM/IEEE Computer Science Curricula som precis kommit i en ny uppdaterad version. Tre pedagogiska idéer vi har följt är (1) att använda projektkurser för att integrera teori och praktik samt ge erfarenhet i den vanligaste arbetsformen i näringslivet; (2) att undervisa i flera olika programspråk och flera olika programutvecklingsmetodiker för att ge en plattform att ta till sig det senaste på området; och (3) att införa en programsammanhållande kurs i ingenjörsp professionalism i årskurs 1–3 som ger studenterna verktyg att reflektera över sitt eget lärande, att jobba i näringslivet samt sin professionella yrkesroll. Artikelns avslutas med en diskussion om viktiga aspekter som computational thinking och ACM/IEEE CS Curricula.

I. INTRODUKTION

Vi lever i en värld som ständigt förändras. Vårt samhälle blir alltmer beroende av datorer och mjukvara. Förändringstakten ökar hela tiden, med teknik som en viktig drivkraft. För att hantera detta behövs mjukvaruingenjörer med de nödvändiga kunskaperna, färdigheterna och förhållningssätten för att kunna utveckla och underhålla de allt större och komplexare mjukvaruintensiva systemen. För att möta detta behov har Linköpings universitet utvecklat Sveriges första civilingenjörutbildning i Mjukvaruteknik.

I den här artikeln beskriver vi visionen, målen och designen av programmet. Vi har ansträngt oss för att formulera mål och principer som kan användas av andra liknande program, även om förutsättningarna är annorlunda. Detta gör innehållet intressant för en större publik. Slutligen diskuterar vi viktiga aspekter som Computational Thinking och relationen till ACM/IEEE CS Curricula, vilken även används för att jämföra Mjukvaruteknik med liknande program i Sverige.

En anledning att skriva den här artikeln är för att bjuda in till diskussion och konstruktiv kritik kring frågor om hur man designar ett modernt civilingenjörprogram inom datavetenskap och programvaruteknik. Vi uppmuntrar er att kontakta oss om ni har frågor eller kommentarer.

II. LIKNANDE UTBILDNINGAR

A. Liknande utbildningar vid Linköpings universitet

Linköpings Tekniska Högskola som är en del av Linköpings universitet har i dagsläget fem program inom

dataområdet förutom civilingenjör i Mjukvaruteknik. Två civilingenjörprogram, Datateknik och Informationsteknologi, ett högskoleingenjörprogram i Datateknik, ett treårigt kandidatprogram i Innovativ Programmering samt ett tvåårigt masterprogram i datavetenskap. När det gäller civilingenjörprogrammen så innehåller Datateknik ungefär lika delar hårdvara och mjukvara samt lägger stor vikt på samspillet mellan dem medan Informationsteknologi fokuserar på infrastruktur för att överföra, strukturera och presentera information. Med Mjukvarutekniks fokus på datavetenskap och mjukvaruutveckling får Linköpings universitet ett heltäckande utbildningsutbud från elektronik och hårdvara till mjukvara.

B. Liknande utbildningar i Sverige

Alla tekniska högskolor har minst ett civilingenjörprogram inom dataområdet. De har varierande inriktningar, från hårdvaruorienterade till mer mjukvaruorienterade. De program som mest liknar Mjukvaruteknik är Datateknik på KTH och Informationsteknik på Chalmers. Även om det finns likheter så finns det också viktiga skillnader. Till exempel så är Informationsteknik på Chalmers huvudsakligen ett program i programvaruteknik med relativt lite datavetenskap och fokuserar på ett programspråk (Java).

2011 gjordes en jämförande studie mellan 10 av Sveriges civilingenjörprogram inom dataområdet utifrån ACM/IEEE CS Curricula 2008 [1]. En slutsats är att det finns en stor variation av program, från hårdvara till mjukvara, från teori till praktik, och från allmänna till mer specifika ingenjörprogram. En jämförelse mellan det nya programmet och dessa 10 program görs i sektion V.

III. VISION OCH MÅL

Visionen för programmet i Mjukvaruteknik är enkelt: utbilda Sveriges bästa mjukvaruingenjörer. Utbildningen ska ge de bästa förutsättningarna för en spännande, framgångsrik och givande karriär inom det väldigt breda området mjukvaruteknik. Färdiga mjukvaruingenjörerna kommer ta ledande och drivande roller både i näringslivet och i samhället samt starta nya företag baserade på innovation och entreprenörskap.

Programmet ska ge de nödvändiga kunskaperna, färdigheterna och förhållningssätten för att kunna utveckla alla typer av mjukvaruintensiva system, använda alla utvecklingsmetodiker och viktigast av allt att kunna välja rätt språk och metodik för varje given situation. Programmet ska förbereda studenterna på en professionell karriär där mjuka

färdigheter som kommunikation, personligt ledarskap och lagarbete i tvärfunktionella team är absolut nödvändiga.

De konkreta målen för programmet är att:

- 1) ge en gedigen grund i datavetenskap och matematik;
- 2) ge förståelse och erfarenhet av "Computational Thinking";
- 3) ge ett helhetsperspektiv på modern mjukvaruutveckling;
- 4) ge förståelse och erfarenhet av flera olika programmeringsspråk/paradigm och industriella utvecklingsmetoder;
- 5) ge förståelse och erfarenhet av viktiga applikationsområden som moderna storskaliga distribuerade och inbyggda system, mobila och sociala applikationer, samt AI, datadrivet beslutsfattande och robotik;
- 6) bygga på projektkurser med aktuellt teknikinhåll som integrerar och applicerar teori;
- 7) främja entreprenörskap och innovation;
- 8) lyfta fram globala utmaningar, hållbar utveckling, och den globala arbetsplatsen.

IV. DESIGN

På två år har vi gått från tanken på att starta ett nytt program till att de första studenterna har börjat. Den riktiga designprocessen startade Januari 2011 och det har tagit ungefär 18 månader att nå en stabil version av de tre första åren. Designgruppen har bestått av representanter från datavetenskap, systemteknik, matematik och studenterna. Det fanns bivillkor som att minimera antalet helt nya kurser. De största utmaningarna har varit att välja ut de obligatoriska kurserna de tre första åren, att hitta en bra balans mellan datavetenskap, datorteknik, systemteknik, programvaruteknik och matematik särskilt med tanke på att det är ett civilingenjörsprogram, samt att ordna kurserna så att alla förkunskapskrav uppfylls och det blir en tydlig progression genom programmet. Vi är väldigt nöjda med slutresultatet, trots en del kompromisser.

Många ingenjörsprogram börjar med grundläggande matematik- och teknikkurser de tre första åren för att studera tillämpningar och specialiseringar de två sista åren. Ett stort problem med detta är att studenterna inte ser hela bilden förrän på slutet. Det finns tydliga indikationer på att många studenter som hoppar av gör det för att de saknar helhetsbilden. Då studenter utvecklas och mognar väldigt mycket under utbildningen innebär det också att de inte är redo att fullt förstå ämnet i början. Mjukvaruteknik gör därför tre iterationer genom de breda och djupa ämnena datavetenskap och programvaruteknik. Genom att ha flera iterationer, på olika detaljnivåer, är vi övertygade om att studenternas förståelse kommer öka markant samtidigt som antalet avhopp kommer att minska.

Den första iterationen är den första terminen som ger en bred översikt över hela området och hela utbildningen. Den andra iterationen är de följande 5 terminerna som tillsammans ger en solid grund inom datavetenskap, programvaruteknik, matematik och relevanta områden inom systemteknik. Iterationen avslutas med ett kandidatarbete där studenterna demonstrerar vad de lärt sig de första två iterationerna. Den tredje iterationen är masterprofilen de två sista åren där studenterna

fördjupar sina kunskaper inom viktiga områden baserat på sina intressen och fallenheter. Linköpings universitet är stort och kan erbjuda ett stort urval av avancerade kurser med aktiva forskare inom området. Den tredje iterationen avslutas med exjobbet där studenterna visar att de uppfyller alla kraven på en civilingenjör i Mjukvaruteknik.

En översikt över programmet ges i Figur 1. Ljusgrå lådor är datavetenskap, mellangrå är andra ingenjörskurser och de mörkgrå är matematikkurser. Varje termin består av två läsperioder och totalt 30 högskolepoäng (hp). Kurser är normalt 6hp och ges inom en läsperiod. Förkortningarna inom parentes refererar till kunskapsområden inom ACM/IEEE Computer Science Curricula 2013 [2]. Ämnena som nämns för de två projektkurserna beskriver innehållet i lite mer detalj (summan av delpoängen är samma som för hela kursen).

A. Designprinciper

Designen är baserad på följande generella principer:

- 1) Det ska finnas en tydlig progression genom programmet, både vad gäller huvudämnet datavetenskap och vad gäller hur arbetet görs och vad som krävs av studenterna.
- 2) De tre första åren ska bestå av obligatoriska kurser som ger en gemensam grund medan de två sista ska vara så fria som möjligt för att ge studenterna möjlighet att utveckla sina intressen och specialisera sig.
- 3) Det ska normalt sett vara tre parallella kurser, varav en matematikkurs och en programmeringskurs.
- 4) Varje termin de två första åren ska fokusera på ett programmeringsspråk. Studenterna ska lära sig flera olika språk och paradigm med ett tydligt fokus på koncept och principer som gör att de snabbt kan ta till sig nya programmeringsspråk. De ska åtminstone lära sig funktionell, imperativ och objektorienterad programmering i Python, C/C++ och Java.
- 5) Varje termin ska ha någon form av projektkurs. En projektkurs karaktäriseras av större och relativt öppna uppgifter som studenterna ska genomföra, antingen individuellt eller i grupp. Projektkurser ska integrera och tillämpa teoretiska koncept och tekniker genom utveckling av mjukvarusystem. En projektkurs kan också innehålla teori.
- 6) Projektkurser ska använda olika metoder för att ge erfarenhet och förståelse av flera olika industriella utvecklingsmetoder. Det är viktigt att valet av metod passar projektet för att ge rätt förståelse. Precis som med programmeringsspråk är det de bakomliggande koncepten och principerna som är viktigast samt att få studenterna att reflektera över hur valet av metod påverkar resultatet.
- 7) Matematik är en central och integrerad del av programmet. Det är viktigt att lyfta fram de starka kopplingarna mellan matematik och datavetenskap då dessa ger grunden för "Computational Thinking". Det första året ska fokusera på diskret matematik och logik som är direkt relevanta för mjukvaruteknik. Andra året ska ge nödvändiga kunskaper i kontinuerlig matematik. Tredje året ska innehålla tillämpade kurser som direkt använder den kontinuerliga matematiken, där sannolikhetslära och

	Hösttermin		Vårtermin	
	Period 1	Period 2	Period 1	Period 2
År 1	Diskret matematik (6hp DS)	Logik (6hp DS)	Dator teknik (6hp AR)	
	Perspektiv på datavetenskap (7hp SDF++)		Projekt: Mobila & Sociala Applikationer (11hp; NC 3hp, PBD 8hp)	
	Datorsystem och programmering (4hp SDF)	Funktionell och imperativ programmering i Python (6hp SDF)	Internet (3hp NC) Webbprogrammering (3hp PBD)	Mobilprogrammering (5hp PBD)
	Ingenjörsp professionalism (2hp SP)		Objektorienterad programmering och Java (6hp PL)	Formella språk och automatateori (6hp; AL 3hp, PL 3hp)
År 2	Programutvecklingsmetodik (4hp SE)	Inledande matematisk analys (6hp)	Envariabelanalys 1 (6hp)	Flervariabelanalys och Diffekvationer (4hp)
	Datastrukturer, algoritmer och programmeringsparadigmer (11hp; 6hp AL, 5hp PL)		Projekt: Distribuerade och inbyggda system (13hp; NC 5hp, PD 6hp, SF 2hp)	
	Linjär algebra (8hp)		Operativsystem (6hp OS)	Datomät (5hp NC) Distribuerade system (5hp PD) Multicore (1.5hp PD) Inbyggda system (1.5hp SF)
	Ingenjörsp professionalism (2hp SP)			
År 3	Sannolikhetslära & Statistik (6hp)	Fysik & Mekanik (6hp)	Systemteknik (8hp)	Reglerteknik (6hp)
	AI (6hp IS)	Databasteknik (6hp IM)	Kandidatprojekt (15hp SE)	
	AI-projekt (5hp IS)			
	Ingenjörsp professionalism (2hp SP)			
År 4	Profilkurser och Valfria kurser (30 hp)		Profilkurser och Valfria kurser (30hp)	
	Profilkurser och Valfria kurser (30hp)		Exjobb (30hp)	

Fig. 1. En översikt över programmet. Ljusgrå lådor är datavetenskap, mellangrå är andra ingenjörskurser och de mörkgrå är matematikkurser.

statistik är väldigt viktiga både för datavetenskap och för systemteknik.

- 8) De tre första åren ska ha projekt inom de applikationsområden som nämndes i målen, dvs. moderna storskaliga distribuerade och inbyggda system, mobila och sociala applikationer, samt AI och datadrivet beslutsfattande.
- 9) Ämnen som testning, användbarhet och säkerhet är grundläggande i modern mjukvaruutveckling och ska ha en naturlig plats i de flesta mjukvaruorienterade kurser, särskilt projektkurserna.
- 10) Programmet ska ge en grundläggande förståelse för datorarkitektur, elektroteknik och reglerteknik som tillsammans ger en grundläggande förståelse för det mycket intressanta samspelet mellan datorer och den omgivande världen. Detta är extra viktigt eftersom det är ett civilingenjörsprogram.

B. År 1

De två första terminerna består av två matematikkurser, diskret matematik och logik, tre kurser i datavetenskap, perspektiv på datavetenskap, datorsystem och programmering samt funktionell och imperativ programmering i Python. Perspektivkursen ger en bred översikt över datavetenskap och relaterade områden. Programmeringskurserna introducerar Unix och programmering i Python. Funktionell och imperativ programmering går igenom med fokus på både rekursiv och iterativ problemlösning. Rekursion, induktionsbevis och relationen mellan dem är centrala koncept. Studenterna får också

prova på att programmera i Prolog, Haskell samt SQL. Syftet är de tidigt ska få en känsla för olika språk och stilar.

Under andra läsperioden, i perspektivkursen, gör studenterna sitt första projekt. Syftet är att de ska göra något häftigt och inspirerande. Projekten är ganska stora och görs i grupper om tre studenter som var och en förväntas lägga 130 timmar på projektet. Projektet börjar ungefär som en labbserie som sedan blir mer och mer fri och till slut förväntas studenterna själva bestämma hur de vill utöka sitt projekt. Exempel på projekt är att programmera humanoida robotar (Nao), Lego Mindstorm, skriva styrprogram till X-Pilots, och att göra en webbaserad karttjänst med hjälp av Google Maps API. Ett viktigt mål med projektet är att studenterna ska inse att de faktiskt kan utveckla ett större program samtidigt som de ska inse att det finns mycket mer att lära. En viktig sidoeffekt är att de har något konkret att visa upp för familj och vänner på jullovet, vilket är bra både för dem själva och för att rekrytera nya studenter till programmet.

Slutligen har vi kursen i Ingejörsp professionalism som sträcker sig över alla de tre första åren. Detta är en program-sammanhållande kurs där grupperna består av studenter från de tre första årskurserna. Inspirationen kommer från en liknande kurs på KTH och är en kurs med många funktioner [3]. De två huvudsakliga är att ge studenterna en helhetsförståelse för programmet och att lära ut mjuka färdigheter som personligt ledarskap, etik och att jobba tillsammans med andra. En viktig och spännande förändring jämfört med kursen på KTH är att vi använder dialogseminariemetoden för att reflektera över och lära från erfarenheter [4]. Metoden är utvecklad av forskare på KTH tillsammans med Combitech AB, ett konsultföretag

med kopplingar till SAAB. Combitech är också delaktiga i utvecklingen av kursen och i träningen av kursens lärare. Vi tror att kursen kommer uppfattas som mycket mer trovärdig då den utvecklas tillsammans med industrin som verkligen vet vad som är nödvändigt för professionella ingenjörer.

Efter den första terminen ska studenterna ha bra allmän förståelse för datavetenskap, programmering och dess formella grunder i diskret matematik och logik.

Den andra terminen består av en teorikurs, en datorteknikkurs, en programmeringskurs och den första projektkursen. Kursen i formella språk och automatateori ger en introduktion till teoretisk datalogi och fortsätter det diskreta mattespåret. Kursen i datorteknik ger en introduktion till digitalteknik, komponenterna i en dator och hur de kopplas samman samt assemblerprogrammering. Kursen i objektorienterad programmering och Java ger studenterna ytterligare ett språk och ett paradigm. Över hela terminen ligger också projektkursen i mobila och sociala applikationer. Den introducerar ett viktigt applikationsområde och ett systematiskt sätt att utveckla mjukvara. Fokus är på den individuella utvecklaren som designar och utvecklar användbara mobila applikationer med sociala funktioner. Viktiga principer som tas upp är prototypning vid design, användbarhet, enhetstestning, och kodgranskning.

C. År 2

Den tredje terminen består av en kurs i programutvecklingsmetodik, en stor kurs i datastrukturer, algoritmer och programmeringsparadigm som använder C++, en kurs i linjär algebra och en introduktionskurs i analys. Delen om programmeringsparadigm är särskilt viktigt då den samlar upp alla paradigmerna och relaterar dem till varandra. Den visar också hur samma koncept kan realiseras på många olika sätt. Efter den här terminen ska studenterna vara duktiga programmerare.

Det är ett genomtänkt val att ha kursen i programutvecklingsmetodik efter det första projektet. Tanken är att studenterna ska ha erfarenhet med mjukvaruutveckling innan de lär sig teorin. På så sätt är studenterna mentalt förberedda på innehållet och har redan stött på några av de problem som utvecklingsmetodikerna försöker hantera.

Den fjärde terminen består av analys i en och flera variabler, operativsystem, samt en projektkurs i distribuerade och inbyggda system. Koncept relaterade till datorsystem studeras från många olika perspektiv. Grunden är datorer som kopplas samman i nätverk där förståelse för operativsystem och datornät är väldigt viktiga. Utifrån den här grunden är det naturligt att ta upp utökningar som multicore, parallella datorer, samt inbyggda system. Ett viktigt steg är att se datorer sammankopplade i nätverk som distribuerade system samt att designa och utveckla program för dessa. Viktiga begrepp är skalbarhet, feltolerans, säkerhet och testning av distribuerade system.

D. År 3

Det tredje året består av kurser i AI och databaser, som är två viktiga områden inom datavetenskap, sannolikhetslära och statistik som är centrala t.ex. för empiriska aspekter

av ingenjörsvetenskap, samt fysik och mekanik som ger en grundläggande förståelse för den fysiska världen. Den här kunskapen utökas och appliceras i kurserna i systemteknik och reglerteknik där studenterna lär sig grunderna för elektriska kretsar, signaler och system inklusive att mäta, modellera och styra dessa. Projektkursen i termin fem fokuserar på AI och datadrivet beslutsfattande. Kursen kommer tillämpa mycket av innehållet i mattekurserna då maskininläring och dataming baseras på linjär algebra, sannolikhetslära och statistik.

Kandidatprojektet motsvarar hälften av termin sex. Detta är ett större mjukvaruutvecklingsprojekt där studenterna jobbar i grupper om 6–8 studenter tillsammans med en extern kund. Målet är att utveckla det kunden beställer. Det betyder att studenterna måste välja den metod som lämpar sig bäst och sedan följa den. De måste hantera saker som att kunderna ändrar sig, att medarbetare blir sjuka eller kanske försvinner (uppnås genom att byta personer mellan grupperna). Varje student skriver en kandidatuppsats baserat på sitt arbete.

När studenterna har fullföljt de tre första åren så kan de ta ut en kandidatexamen. Detta gör det möjligt för dem att byta universitet för sina masterstudier. Vår erfarenhet är att väldigt få gör detta.

E. År 4–5

De två sista åren motsvarar ett masterprogram och består huvudsakligen av valfria kurser. För att få en masterexamen behöver studenterna ta 90hp på avancerad nivå inklusive 30hp exjobb. För att hjälpa studenterna att välja finns det ett antal profiler. Varje profil består av en mängd kurser varav studenten måste ta minst 36hp där minst 30hp är på avancerad nivå.

Två nya profiler kommer att utvecklas för programmet: Industriell mjukvaruutveckling samt AI och datadrivet beslutsfattande. Det finns även existerande profiler från programmen i Datateknik och Informationsteknologi som t.ex. Programmering och algoritmer, Spelprogrammering, Säkra system och International Software Engineering tillsammans med Harbin Institute of Technology som ger dubbel examina.

För profilen i Industriell mjukvaruutveckling kommer åtminstone två nya viktiga kurser att utvecklas. Den första är en kurs i storskalig mjukvaruutveckling baserad på open-source-projekt och är central för att studenterna ska få erfarenhet av att jobba med riktigt stora projekt tillsammans med många andra. Viktiga färdigheter som tränas är att sätta sig in i, felsöka och vidarutveckla existerande program. Att hantera legacy-system är en kompetens som är starkt efterfrågad. Enligt våra näringslivsrepresentanter är det här en färdighet som de flesta studenter saknar eftersom de är vana vid att skriva saker från grunden, vilket sker väldigt sällan i industrin. Den andra kursen handlar om entreprenörskap och innovation. Mjukvaruområdet är en väldigt spännande bransch där steget från idé till kommersialiserad produkt är väldigt kort. Ett nystartat företag kan direkt få global distribution antingen genom Internet eller genom en appstore. Mjukvarubranschen är extremt expansiv och rätt idé, rätt genomförd, vid rätt tidpunkt kan leda till ett miljardföretag inom väldigt kort tid (exempel Instagram).

Profilen i AI och datadrivet beslutsfattande fokuserar på de väldigt spännande områdena artificiell intelligens, mask-

inlärning, datamining, och datadrivet beslutsfattande. Profilen har en väldigt stark forskningsanknytning vid Linköpings universitet och är ett samarbete mellan institutionerna i datavetenskap och systemteknik.

V. DISKUSSION

A. Computational Thinking

“Computational thinking” eller datalogiskt tänkande är centralt för programmet. Här ingår rekursiv nedbrytning av problem i mindre och mer hanterbara delar; att hitta och generalisera mönster genom att exempelvis systematiskt lösa mindre fall; att använda simulering för att antingen approximera lösningar eller för att generera och testa hypoteser. Här används idéer från både datavetenskap och matematik, och programmering används ofta som en delösning.

En viktig utmaning är att knyta samman kurser i programmering och matematik. I programmeringskurserna försöker lärarna att visa hur ämnena kopplar till varandra explicit genom att t.ex. peka på likheterna mellan induktiva bevis och rekursiva beräkningar, och genom att använda matematiska funktioner som programmeringsexempel. Härigenom visas fördelarna med matematik för programmering. Vår nuvarande utmaning är att visa fördelarna med programmering i matematiken. För att uppnå detta undersöker vi hur “computational thinking” kan användas i diskret matematik, vilket är den första matematikkursen på Mjukvaruteknik. Som en del i detta har vi undersökt studenternas attityd till matematik och programmering samt deras syn på samspelet mellan matematik och programmering. Preliminära data visar att studenterna är mycket positiva till matematik och extremt positiva till programmering. Dessutom inser studenterna att det finns en viktig koppling mellan ämnena, men de saknar ännu förmågan att kombinera matematik och programmering. För att visa studenterna hur viktigt och användbart detta samspel är, samt att lära ut vissa grundläggande kunskaper, så har inlämningsuppgifterna utvidgats med förslag på hur programmering kan användas både som ett verktyg för att undersöka matematiska problem och för att finna möjliga lösningar. Programmering är bara ett verktyg, de måste fortfarande bevisa matematiskt att deras svar är korrekta. Vi hoppas härigenom att studenterna blir mycket bättre på att använda matematik som ett verktyg vid programmering och programmering som ett verktyg för att lösa matematiska problem.

B. ACM/IEEE CS Curricula

ACM/IEEE CS Curricula [2] är ett viktigt och imponerande dokument som tydligt beskriver kunskaperna som varje datavetare (computer scientist) bör ha. Då de har gjort ett mycket bra jobb så har vi använt läroplanen som en guide och en måttstock för att säkerställa att Mjukvaruteknik täcker in alla relevanta aspekter av datavetenskap. I översikten över programmet (Figur 1) har vi klassificerat innehållet i kurserna enligt ACM/IEEE CS Curricula 2013. Genom att jämföra innehållet i kunskapsområdena med innehållet i programmet ser vi tydligt att programmet täcker det allra mesta.

Karlfeldts exjobb har som vi tidigare nämnt jämfört 10 svenska civilingenjörutbildningar liknande Datateknik på KTH

relativt ACM/IEEE CS Curricula 2008 [1]. Figur 2 visar för varje kunskapsområde hur Mjukvaruteknik ligger till i förhållande till genomsnittet av de 10 utbildningarna som Karlfeldt studerat. Det är viktigt att notera att figuren endast tar upp obligatoriska kurser under de tre första åren. Det betyder att kurserna de två sista åren inte är med. Figuren använder procent då olika utbildningar har olika stor del obligatoriska kurser. Figuren är en utökning av Figur 1 i Karlfeldts exjobb.

Två viktiga slutsatser kan dras från Figur 2. För det första att Mjukvaruteknik uppfyller ACM/IEEE CS Curricula 2008 under de tre första åren. För det andra att Mjukvaruteknik innehåller betydligt mer datavetenskap än de andra programmen. Det finns självfallet program med mer innehåll inom specifika områden, men inget annat program har samma breda och djupa täckning av datavetenskap.

C. Ingenjöraspekter

Alla ingenjörutbildningar vid Linköpings universitet bygger på CDIO-konceptet (Conceive, Design, Implement, Operate) [5]. CDIO är ett koncept för att systematiskt utveckla, genomföra och evaluera ingenjörutbildningar enligt deras syllabus, som specificerar kunskaper och färdigheter som alla studenter förväntas ha. Ingenjörutbildningarna på Linköpings universitet använder en variant av CDIO syllabus som kallas för LiTH syllabus. Den har fyra huvudrubriker:

- 1) Matematiska, natur- och teknikvetenskapliga kunskaper;
- 2) Individuella och yrkesmässiga färdigheter och förhållningssätt;
- 3) Förmåga att arbeta i grupp och att kommunicera;
- 4) Planering, utveckling, realisering, drift och affärsmässigt förverkligande av tekniska produkter, system och tjänster med hänsyn affärsmässiga och samhällliga behov och krav.

Det finns en direkt koppling mellan LiTH syllabus och de nationella målen för civilingenjörutbildningar, vilket betyder att om en utbildning uppfyller LiTH syllabus så uppfyller de även de nationella målen. Genom Mjukvarutekniks ämnesinnehåll, de många projektkurserna och den nya kursen i ingenjörsp professionalism uppfyller programmet LiTH syllabus och därmed de nationella civilingenjörsmålen.

VI. ERFARENHETER

I introduktionen nämnde vi ett antal utmaningar, inklusive att välja vilka kurser som ska ingå och i vilken ordning. Här diskuterar vi några erfarenheter utifrån dessa utmaningar.

Normalt sett är kurser 6hp och går i en läsperiod. Detta gör det svårt att få plats med allt material vi vill ha med i programmet. För att komma runt det skapade vi större kurser som går över en hel termin och som består av flera mindre delar. En konsekvens är att programmet blir mer beroende av att lärarna på dessa kurser täcker det material som vi har räknat med. För att säkerställa att detta sker har vi skapat en arbetsgrupp med examinatorerna för dessa kurser samt några andra viktiga kurser som de inledande programmeringskurserna och kursen i ingenjörsp professionalism. Ordförande i programplangruppen för Mjukvaruteknik är även ordförande i arbetsgruppen. På så sätt kopplas design och genomförande av programmet tydligt

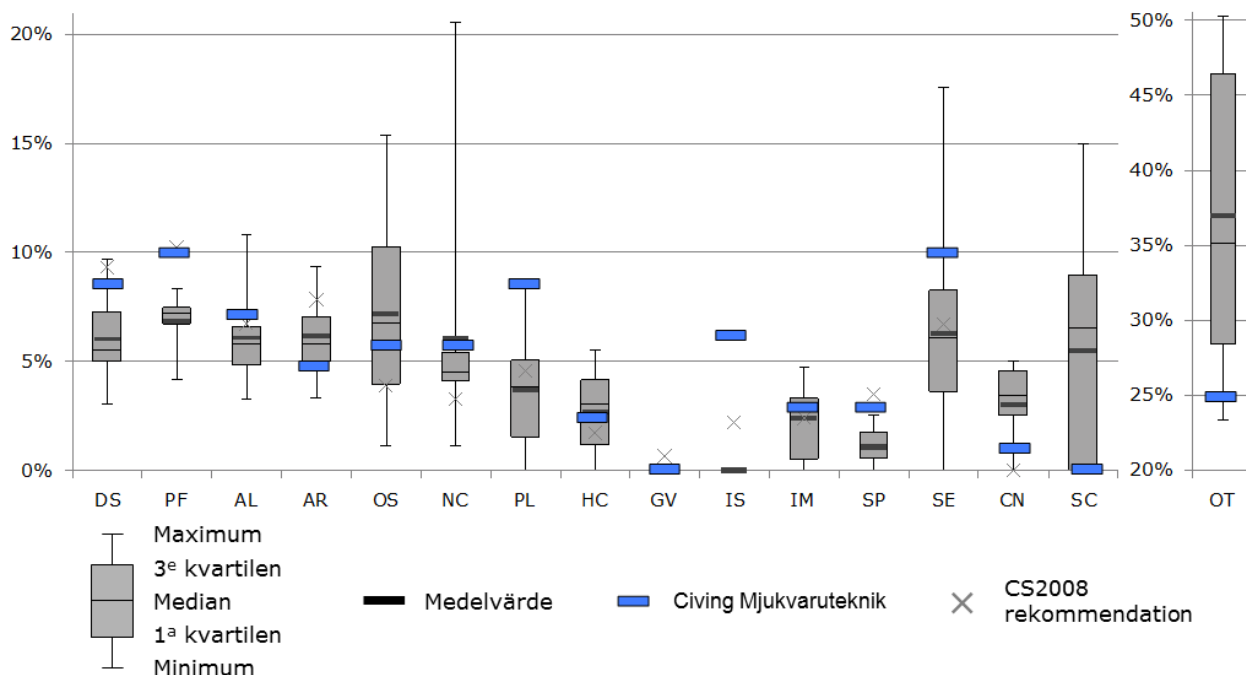


Fig. 2. En jämförelse mellan svenska civilingenjörsutbildningar med inriktning mot datavetenskap utifrån ACM CS Curricula 2008, baserat på Figur 1 i [1].

samman. En annan konsekvens är att det blir svårare att hålla reda på förkunskaper. Tidigare definierade kursnamnen oftast innehållet, men nu blir namnen mer generella och det krävs att man tittar på vad kursen faktiskt innehåller för att avgöra om studenter har uppnått förkunskapskrav till senare kurser. Samtidigt är en lärdom från designprocessen att kursnamn kan ge ett intryck av vad kursen innehåller, medan kursplanen ger ett annat intryck.

Från början ville vi att fysik, mekanik, systemteknik, reglerteknik och AI skulle läsas samtidigt och tydligt knyts samman i ett robotikprojekt. Tyvärr skulle det kräva förändringar med alltför många nackdelar, främst på grund av förkunskapskraven för de här kurserna.

Vad saknas? De tre första åren täcker flera programmeringsspråk och programmeringsparadigm vilka ger en väldigt bra grund. Dock är paradigmdelen ganska liten. Dessutom hade det varit bra med en grundläggande kurs i programspråksdesign där grunderna i parsning och tolkning av språk tas upp samt domänspecifika språk vilka blir allt viktigare. Det finns dock kurser i årskurs 4 och 5 i kompilatorkonstruktion samt data- och programstrukturer som till stor del täcker behovet. En annan svaghet är att det inte finns obligatoriska kurser i optimering och grafteori som båda är väldigt grundläggande och centrala för stora delar av modern datavetenskap.

En kompromiss vi fick göra är att placera kursen i sannolikhetslära och statistik i termin 5, även om vi ville ha den tidigare. En fördel med detta är att studenterna då har läst flervariabelanalys, vilket gör att kursen kan komma lite längre. I slutändan är det antagligen bra att studenterna är mer matematiskt mogna när de tar den här viktiga kursen.

VII. SAMMANFATTNING

Vi har beskrivit visionen, målen och designen av Sveriges första civilingenjörsprogram i Mjukvaruteknik (Master of Science in Computer Science and Software Engineering). Målen med programmet är att ge ett helhetsperspektiv på modern storskalig mjukvaruutveckling, att ge en bred och djup förståelse av datavetenskap och “computational thinking”, samt främja innovation och entreprenörskap.

Att utveckla ett helt nytt civilingenjörsprogram är väldigt spännande, utmanande och tillfredsställande. Vi anser att vi har lyckats göra bra avvägningar i valet och ordningen av kurser så att målen uppnås. Studenternas respons har varit väldigt positiv med mer än 600 sökande till de 30 platserna, varav mer än 130 förstahandssökande. När programmet startade i höst antogs 40 personer, var 39 hade Mjukvaruteknik som sitt förstahandsval. 38 studenter kom till uppropet och efter en läsperiod finns drygt 30 studenter kvar. Vi jobbar nu hårt för att programmet verkligen ska leva upp till visionen och till de högt ställda målen och verkligen vara världens bästa mjukvaruingenjörsutbildning.

REFERENCES

- [1] J. Karlfeldt, “Den svenska dataingenjören – en jämförande studie av 10 svenska civilingenjörsutbildningar inom det datavetenskapliga området,” Master’s thesis, KTH, 2012.
- [2] M. Sahami, S. Roach, *et al.*, “Computer Science Curricula 2013 – Ironman Draft Version 1.0,” Feb. 2013.
- [3] V. Kann, “En programsammanhållande kurs med många funktioner,” in *Proc. 3:e utvecklingskonferensen*, 2011, pp. 153–156.
- [4] B. Goranzon, R. Ennals, and M. Hammeron, *Dialogue, skill and tacit knowledge*. Wiley.com, 2006.
- [5] “The CDIO initiative, <http://www.cdio.org>.”