

Värderingsförmåga och förhållningssätt i Software Engineering

Daniel Einarson

Introduktion

Datavetenskap vid Högskolan Kristianstad (HKR) är sedan några år tillbaka med i det världsomspännande ramverket och nätverket CDIO (Conceive, Design, Implement, Operate, [1]). Grunden till CDIO utgörs av kritik från teknikindustrin där man sagt att ingenjörsutbildningar varit alltför teoretiska och behöver tydliga moment av träning i verklighetsnära projekt med tillräcklig komplexitet. Vad vi ser här är alltså ett behov av ett samverkansnära utbildningssystem, med flera syften, såsom relevans, verksamhetsintegrerat lärande, och studenters anställningsbarhet. CDIO som utbildningsramverk utgörs dels av CDIO 12 Standards, som fungerar som vägledande för utövare av CDIO-baserade teknikutbildningar, och dels av CDIO Syllabus ([2]) som utgörs av ett antal lärandemål. Kategoriseringen av CDIO Syllabus utgörs av fyra sektioner enligt nedan, där nedanstående uppställning även omfattar lärandemål under de fyra sektionerna.

1 DISCIPLINARY KNOWLEDGE AND REASONING

1.1 KNOWLEDGE OF UNDERLYING MATHEMATICS AND SCIENCE

1.2 CORE FUNDAMENTAL KNOWLEDGE OF ENGINEERING

1.3 ADVANCED ENGINEERING FUNDAMENTAL KNOWLEDGE, METHODS AND TOOLS

2 PERSONAL AND PROFESSIONAL SKILLS AND ATTRIBUTES

2.1 ANALYTICAL REASONING AND PROBLEM SOLVING

2.2 EXPERIMENTATION, INVESTIGATION AND KNOWLEDGE DISCOVERY

2.3 SYSTEM THINKING

- 2.4 ATTITUDES, THOUGHT AND LEARNING
- 2.5 ETHICS, EQUITY AND OTHER RESPONSIBILITIES
- 3 INTERPERSONAL SKILLS: TEAMWORK AND COMMUNICATION**
- 3.1 TEAMWORK
- 3.2 COMMUNICATIONS
- 3.3 COMMUNICATIONS IN FOREIGN LANGUAGES
- 4 CONCEIVING, DESIGNING, IMPLEMENTING, AND OPERATING SYSTEMS IN THE ENTERPRISE, SOCIETAL AND ENVIRONMENTAL CONTEXT**
- 4.1 EXTERNAL, SOCIETAL AND ENVIRONMENTAL CONTEXT
- 4.2 ENTERPRISE AND BUSINESS CONTEXT
- 4.3 CONCEIVING, SYSTEMS ENGINEERING AND MANAGEMENT
- 4.4 DESIGNING
- 4.5 IMPLEMENTING
- 4.6 OPERATING

CDIO tar upp flera lärandemål som ligger nära Högskoleförordningens ([3]) tredje kunskapsform (Värderingsförmåga och förhållningssätt), såsom 2.5 ETHICS, EQUITY AND OTHER RESPONSIBILITIES, och 4.1 EXTERNAL, SOCIETAL AND ENVIRONMENTAL CONTEXT. Sådana samverkansnära utbildningsambitioner ligger väl i linje med Högskolelagen ([4]), den senaste forskningspropositionen ([5]), och även med Högskolan Kristianstads ambitioner, genom HKRs valda uttalade riktningar mot VFU och anställningsbarhet.

CDIO framhäver även aktiva och integrerade lärandeformer (CDIO Standard 7 och 8, [6]), där aktiva lärandeformer innebär att undervisningen går alltmer från traditionella föreläsningar till att mer aktivera studenterna i deras lärandesituation, medan integrerat lärande innebär att hitta former där flera typer av kompetenser, såsom ur samtliga CDIOs sektioner (se ovan) tränas i samma undervisningsmoment. Det senare förutsätter mer eller mindre ett studentaktivt lärande i sig självt, såsom diskuteras i ([7]).

Aktiva lärandeformer beaktas inte bara i samverkansnära syften, utan även för att ge fördjupad kunskap i sig självt. Detta kan i sig vara ett sätt att ge större värde åt Högskoleförordningens första kunskapsform (Kunskap och förståelse).

Det har visats ([2]) att CDIO Syllabus svarar mot Högskoleförordningens examensmål angående ingenjörsutbildningar. Vidare har det visats att om man har CDIO Syllabus som utgångspunkt för lärandemål på kurser och utbildningsprogram så svarar man även upp mot högskoleförordningens examensmål för kandidatutbildningar ([8]). Datavetenskap vid HKR utvecklar sina kursplaner och utbildningsplaner utifrån CDIOs principer, och svarar på så vis upp mot examensmålsobligatoriet.

Kursexempel

Exempel på hur den tredje kunskapsformen, värderingsförmåga och förhållningssätt, realiseras i dessa kommer här att göras genom kurserna Software Engineering 2 från kandidatprogrammet för Datasystemutveckling, och magisterprogrammets kurs Utveckling av inbyggda system ([9]). Båda dessa kurser är starkt projektbaserade, dvs. kretsar kring färdigställandet av ett projekt. Båda kurserna är, mer eller mindre, CDIO-anpassade. Detta innebär att ett huvudsakligt perspektiv ligger i att ha utbildningsinslag som ska vara teknikindustrinära, eller bas för verksamhetsintegrerat lärande, vilket också framhävs i ([10]). I detta fall gäller det såväl arbetssättet, dvs. processen att nå fram till projektresultatet, som projektets komplexitetsgrad i sig självt. Man kan alltså se att redan dessa utgångspunkter visar på potentialer att adressera alla tre av Högskoleförordningens kunskapsformer. Nedan kommer särskilt den tredje av dessa, värderingsförmåga och förhållningssätt, att adresseras.

Software Engineering 2

I ([10]) diskuteras hur kursen Software Engineering genomgått tre år (ett initialt år, samt två på varandra följande år) av förändringar, där dessa delvis kan hänföras till värderingsförmåga och förhållningssätt. I en nuvarande form av kursplanen gäller för denna kunskapsform:

Värderingsförmåga och förhållningssätt

Efter genomgången kurs ska studenten

- a. kunna värdera styrning av projektprocessen för att färdigställa projektets produkt
- b. kunna värdera betydelsen av projektarbeten och samverkan med andra personer
- c. visa förmåga att göra bedömningar inom mjukvaruprojekt med hänsyn till relevanta vetenskapliga, samhälleliga och etiska aspekter samt visa medvetenhet om etiska aspekter på forsknings- och utvecklingsarbete.

Dessa kursmål examineras på olika sätt, det i sammanhanget mest intressanta är dock Projektuppföljning, som examineras individuellt genom skriftlig och muntlig presentation av deldokumentationer vid obligatoriska projektmöten. Via iterativa möten mellan studenter och lärare betraktas dokumenten och diskuteras, för att senare betygsättas.

Kursprojektet ska simulera ett storskaligt projekt, där en större grupp om ca 15 studenter samverkar för att utveckla smarthustekniker för personer med funktionella nedsättningar. Man ska kunna styra lampor, värmeelement, fläktar, lås, mm., i ett mindre modellhus via appar på mobiltelefoner, eller webbgränssnitt. För att klara detta behöver studentgruppen dela upp sig i mindre grupper om ca 3 personer, där dessa delgrupper löser delproblem, som sedan integreras i den större lös-

ningen (för mer information om detta, och hur kursen genomfördes som stöd för efterföljande diskussioner, se exempelvis [10]).

För att utveckla projektet behöver man uppnå målen a, b, och c. Målet c, bemöter man genom att koppla systemutvecklingen till dess tänkta användning, dvs. stöd för personer med funktionshinder, liksom att man ser på vetenskapliga- och forskningsutvecklingsmässiga aspekter. De etiska aspekterna blir här dessutom väsentliga. En fråga handlar dock om vilka värdedjup man når i dessa mål. Här blir det treåriga utvecklingsarbetet intressant.

Insatserna under de tre på varandra följande år som beskrivs i ([10]), resulterade i tre huvudsakliga utgångspunkter, alla med inspiration från Software Engineering, dvs. med anknytning i programvaruindustrin, som studenterna skulle förhålla sig till i utvecklingen av sina projekt:

1. En i grunden etablerad processmodell.
2. Capability Maturity Model, CMM, nivå 3.
3. The Code of Ethics.

Den etablerade processmodellen vägleder studenterna i deras arbetssätt. Man arbetar i iterativa steg, där man för varje iteration beaktar projektets status, med avseende på krav, design, risker, mm. detta sätter i sig stora krav på kommunikation och samarbete studenterna emellan.

CMM ([11]) är en nivåvis mognadsmodell för hur en utvecklingsorganisation förhåller sig till processen. Nivå 3 innebär att organisationen utvecklar processmodellen så att denna blir anpassad till organisationen. Studenterna tvingas här att utveckla formerna för processen utifrån sina egna behov och önsk-

ningar. Detta kommer att ställa ytterligare krav på kommunikation och samarbete, där överenskommelser är väsentliga för att få projektet att fortskrida.

Code of Ethics för Software Engineers ([12]), ses som guidelines angående förhållningssätt hos programvaruutvecklare, och tar som primär punkt upp hur man bör förhålla sig till allmänheten, eller avnämarna av ett system samt sekundärt hur man förhåller sig till sitt arbete, enligt uppställningen nedan. Det blir i detta sammanhang särskilt viktigt att se båda aspekterna.

PUBLIC - Software engineers shall act consistently with the public interest.

CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.

MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.

SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Studenterna ska här reflektera över dessa, och hur man ställer sig till dessa skriftligt, samt jobba med utgångspunkt i dessa under projektets gång. Särskilt viktigt blir att se produkten de

utvecklar i förhållande till den tänkta avnämargruppen (personer med funktionella handikapp). Men även förhållandet kollegor emellan, och hur man förhåller sig till sig själv och sitt eget ansvar. Särskilt denna tredje utgångspunkt, the Code of Ethics, ligger väldigt nära den tredje kunskapsformen Värderingsförmåga och förhållningssätt. I ([10]) diskuteras också hur den grupp studenter som var bäst på att jobba med the Code of Ethics, också var den som lyckades bäst på projektet och kursen. Man kan alltså här se en korrelation mellan mognadsgraden i förhållandet till Högskoleförordningens tredje kunskapsform, och kursresultatet. För denna typ av projekt är alltså värderingsförmågan och förhållningssättet av signifikant betydelse.

Utveckling av inbyggda system

Då ansvarig lärare för magisterkursen Utveckling av inbyggda system (även här samma som författaren) utvecklade denna kurs gjordes detta med utgångspunkt i en fördjupning av kursen Software Engineering 2. Projektet hade ett liknande upplägg fast med mer avancerade utmaningar. Arbetssätten var i stort de samma, med iterativa möten med lärare för diskussion av dokument, svarande mot projektets status. Även här användes the Code of Ethics. Vad det gäller denna, så presenteras den i två nivåer med anseende på detaljrikedom (för mer information, se [12]), Uppställningen ovan är den minst detaljerade och den som presenterades för studenterna på kandidatnivå, medan den mer detaljerade versionen presenterades för magisterstudenterna.

Från kursplanen ([9]) för Utveckling av inbyggda system:

Värderingsförmåga och förhållningssätt

Efter genomgången kurs ska studenten

- kunna värdera styrning av projektprocessen för att färdigställa projektets produkt,

- kunna reflektera över samhällliga och etiska aspekter i samband med utveckling av inbyggda system,
- kunna värdera betydelsen av samverkan med andra personer i samband med projektarbeten,
- visa förmåga att göra bedömningar inom inbyggda system projekt med hänsyn till relevanta vetenskapliga, samhällliga och etiska aspekter samt visa medvetenhet om etiska aspekter på forsknings- och utvecklingsarbete.

Kursens lärandemål liknar de för kandidatprogrammet, men progression och fördjupning realiserar så som beskrivits ovan. Till detta kommer ett moment av rapportläsande och – skrivande för att fördjupa kunskaper i förhållande till forskning och utvecklingsarbeten inom området.

Referenser

[1] CDIO, Conceive, Design, Implement, Operate real-world systems and products, www.cdio.org

[2] The CDIO Syllabus v2.0 - An Updated Statement of Goals for Engineering Education, Edward F. Crawley, Johan Malmqvist, William A. Lucas, Doris R. Brodeur, The CDIO Syllabus, accessed through <http://www.cdio.org/>

[3] Högskoleförordning (1993:100), Svensk författningssamling 1993:100, http://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/hogskoleforordning-1993100_sfs-1993-100

[4] Högskolelag (1992:1434), Svensk författningssamling 1992:1434, http://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/hogskolelag-19921434_sfs-1992-1434

[5] Forskningspropositionen, Kunskap i samverkan – för samhällets utmaningar och stärkt konkurrenskraft ID-nummer: Prop. 2016/17:50

[6] CDIO Standards. <http://www.cdio.org/implementing-cdio/standards/12-cdio-standards>

[7] Einarson D. Saplacan D., ADDRESSING INTERGRATED LEARNING THROUGH PROJECT-BASED COURSES – FIVE YEARS OF IMPROVEMENTS, submitted to CDIO 2017, <http://www.cdio.ca/>.

[8] Einarson D. CDIO, en introduktion – diskussioner kring ett ramverk för verksamhetsintegrerat lärande, Högskolepedagogisk debatt, Nr 2 2015. Kristianstad University Press 6:2015, ISBN: 978-91-981338-8-2.

[9] Exempelkurser: Software Engineering 2, <http://www.hkr.se/kurs/DA326A/>, och Utveckling av inbyggda system, <http://www.hkr.se/kurs/DT574C/>.

[10] Einarson D., Progressioner inom en projektbaserad kurs baserad på principer från Software Engineering och CDIO, Lärlärdom 2013.

[11] Sommerville I., Software Engineering, Addison Wesley, 2010.

[12] ACM, Association for Computing Machinery, Advancing Computing as a Science & Profession, "Software Engineering Code of Ethics and Professional Practice", <http://www.acm.org/about/se-code>