



Evaluering av studieprogram: **Felles mastergrad i programutvikling (UiB/HiB)**

25. april 2016

Programsensor: Torgeir Dingsøy
sjefforsker, SINTEF IKT

professor II, Institutt for datateknikk og informasjonsvitenskap, NTNU

Agenda

1. Om evalueringen
2. Sammenheng mellom læringsutbytte i program og læringsutbytte i fag
3. Evaluering av læringsutbytte og faginnhold i MOD251
4. anbefalinger om rekruttering
5. Tiltak for jevnere fordeling av studenter mellom UiB og HiB
6. Forslag til videre forbedringsarbeid
7. Oppsummering; prioritering av tiltak
8. Eventuelt

1. Om evaluering

Evaluering basert på:

- Møte med programstyre 9. mars 2016
 - Presentasjon av fellesgraden
 - Presentasjon av HiB-fag
 - Presentasjon av UiB-fag
 - Kvalitets/forbedringsarbeid
- Materiale fra websider
- “Learning outcome description: joint masters degree in software engineering”
- Kursmateriale fra Moderne systemutviklingsmetoder (MOD251)
- Curriculum guidelines for Graduate Degree Programs in Software Engineering, Stevens Institute of Technology

Bakgrunn, programsensor

- Academic experience:
 - Chief scientist at SINTEF ICT, Software Engineering
 - Adjunct Professor at NTNU, Department of Computer and Information Science
 - Dr. Ing. from NTNU in Software Process Improvement
- Industrial experience:
 - Applied research projects with Acando, Bekk, Ericsson, Geomatikk, Kantega, Kongsberg Group, KnowIT, Sopra Steria, Sticos.
 - Certified Scrum Master, PRINCE2 Foundation
- Current teaching:
 - Project course for bachelor students, Department of Computer and Information Science, NTNU
 - Agile project Management, Westerdals College, Oslo
- Research interests:
 - Knowledge Management in Software Engineering, Software Process Improvement, Agile Software Development, Teamwork in Software Development, Large-scale agile development
- Professional service:
 - Academic program co-chair for the 16th International Conference on Agile Software Development (XP2015)
 - Editor of four special issues, two scholarly books
 - Co-organizing chair for the 11th International Conference on Agile Software Development (XP2010)

2. Program og fag

*“Dette er eit studieprogram som er ein fellesgrad med Høgskolen i Bergen. I dette masterprogrammet kan ein velge mellom spesialisering i **programvareutvikling og programutviklingsteori.**”*

Ca. 40 studenter, 30 på programvareutvikling (PU) og 10 på programutviklingsteori (PT)

Program ble formelt etablert 2011.

Beste naturvitenskapelige program i studiebarometeret.

God søkning til programmet, men få kvalifiserte internasjonale søkere.

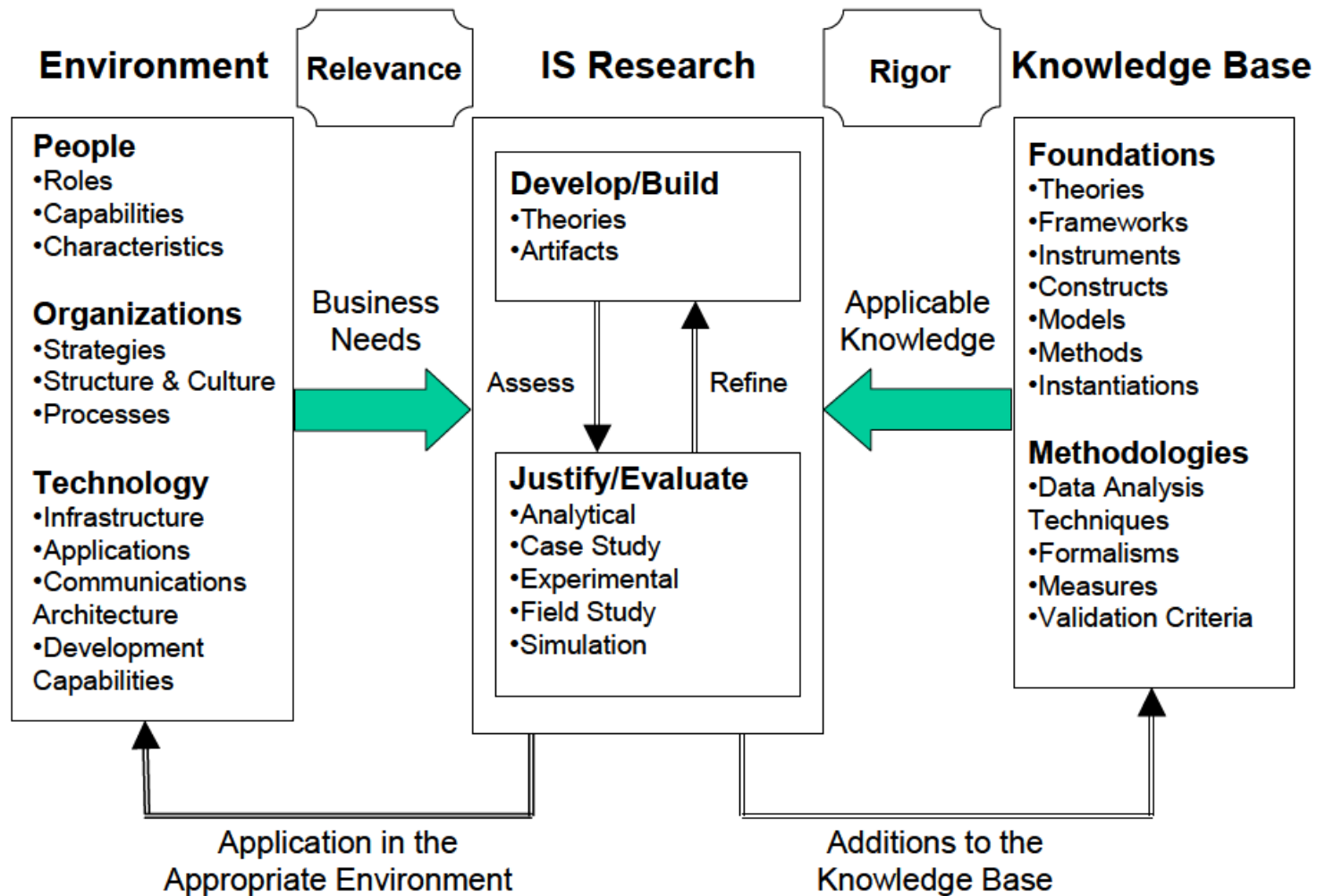
Kunnskapsmål

*“Dei ferdige kandidatane skal ha **solide vitenskapleg funderte kunnskapar** og evne i informatikk og programutvikling. Dei skal kjenne godt til dei vitenskaplege arbeidsmåtene, og dei har trening i sjølvstendig arbeid med omfattande og krevjande faglege oppgåver.”*

-> design science

-> empiriske studier

Design science



*Hevner *et al.*, (2004) Design Science in Information Systems Research, *MIS Quarterly*, 28(1): 75-105.

Ferdighetsmål

“Etter fullført felles mastergrad i programutvikling skal kandidaten:

- 1. sjølvstendig kunne **utforme, spesifisere og programmere** mellomstore informasjonssystem*
- 2. delta i **store programutviklingsprosjekt** i næringsliv og forvaltning*
- 3. sjølvstendig kunne **analysere, teste og validere** store informasjonssystem*
- 4. kunne bruke **moderne verktøy** for programutvikling*
- 5. kunne **forstå og bruke resultat av vitenskapelig forskning** innan programutvikling”*

Uklart hvordan retning PT tilfredsstiller punkt 1-2, se diskusjon om 2 og 5 på punkt 3 i agendaen.

Generelle kompetansemål

“Ein ferdig mastergradskandidat i programutvikling, skal:

- 1. ha tileigna seg **kompetanse for vidare sjølvstendig arbeid**, til å kunne utvikle og forbedra hennes/hans profesjonelle og faglige ferdigheter.*
- 2. vere i stand til å **nytte tileigna kunnskap**, forståing og problemløysningsmetodar i nye kontekstar.*
- 3. kunne **arbeide sjølvstendig og individuelt**, og samstundes bidra aktivt i **gruppesamarbeid**”*

Styrke fokus på gruppesamarbeid (PT).

Learning outcome description I

Knowledge

The candidate...

- has broad knowledge within the field of software engineering and specialized insight in a limited area of **application or theory of software engineering**.
- has thorough knowledge of the theories and methods used in the field of software engineering.
- be able to apply knowledge, understanding and problem-solving abilities in new contexts.
- can analyze academic problems within software engineering on the basis of the history, traditions, distinctive character and place in society of the academic field.

Bruke områder fra ACM Curricula recommendations

Learning outcome description II

Skills

The candidate...

- can analyze and deal critically with various sources of information and use them to structure and formulate scholarly arguments.
- can analyze existing theories, methods and interpretations in within software engineering and work independently on practical and theoretical problems in the field.
- can use relevant methods for research and practical development work in an independent manner.
- can carry out an independent, limited research or development project in accordance with applicable norms for research ethics.
- can independently design, specify and program medium-scale information systems.
- can independently analyse, test and validate large-scale information systems.

Forhold til brukere? Prosjektarbeid?

Learning outcome description III

General Competence

The candidate...

- can analyze relevant academic, professional and **research ethical problems** tied to the field of software engineering.
- can apply his/her knowledge and skills in new areas of software engineering in order to carry out advanced assignments and projects.
- can communicate extensive independent work and masters language and terminology used in the field of software engineering.
- can communicate about academic issues, analyses and conclusions within software engineering, both with specialists and the general public.
- can contribute to new thinking and innovation processes.

Bransje-etiske problemer?

Teamarbeid?

Spesialisering Programutviklingsteori (PT)

*“Spesialiseringa innan **programutviklingsteori** legg vekt på dei **teoretiske grunnprinsippa og metodane** som ligg under konstruksjonen og analysen av komplekse datasystem. Målet er å utdanne kandidatar med spisskompetanse innanfor avansert programmeringsteori, der hovudvekta ligg på fleksible løysingar med omsyn på teknologiske endringar og utvikling.”*

Kurs:

- Algoritmer (10SP)
- Valgfritt: (Datamaskint teori / Programspesifikasjon / Innføring i programomsetjing / Kategoriteori / Innføring i logikk / Utvalde emne i programutviklingsteknologi) (30 SP)
- Valgemne (20 SP)
- Masteroppgave (60SP)

Spesialisering Programvareutvikling (PU)

*“Spesialiseringa innan **programvareutvikling** legg vekt på opplæring i og bruk av **moderne systemutviklingsmetodar og teknologi**. Målet er å utdanne kandidatar med spisskompetanse innanfor avansert datateknologi med fokus på praktiske problemstillingar.”*

- Algoritmer (10SP)
- Avansert programvareteknologi (10 SP)
- Moderne systemutviklingsmetoder (10 SP)
- Valgfritt: (Agentteknologi / Grid-basert databehandling / Modellbasert programutvikling / helseinformatikk? / grafikk?) (10 SP)
- Valgemne (20 SP)
- Masteroppgave (60SP)

Program versus curriculum guidelines; Software engineering

- **Ethics and professional conduct**
- System engineering
- Requirement engineering
- Software design
- Software construction
- Testing
- **Software maintenance**
- Configuration management
- **Software engineering management**
- Software process
- Software quality

Curricula Recommendations



<http://www.acm.org/education/curricula-recommendations>

Læringsutbytte program - fag

- Algoritmer (10SP)
- PU: (Avansert programvareteknologi / Moderne systemutviklingsmetoder)
- PT: (Datamaskint teori / Programspesifikasjon / Innføring i programomsetjing / Kategoriteori / Innføring i logikk / Utvalde emne i programutviklingsteknologi)
- Godt valg av Algoritmer som felles kurs
- Dekke mer av software engineering-feltet for alle
- Kurs med fokus på **software engineering management**
- For stor valgfrihet i PT?

3. Evaluering av MOD251

Modern Software Development Methods

“**Objective:** learn about and get hands-on experience with selected **modern methods** in software development as well as getting an understanding on **how to do research on software engineering**.”

Topics: agile software development (scrum, XP), test-driven development, acceptance testing, continuous integration, design principles and patterns, research methods in software engineering.

Teaching and learning methods:

6 hours of combined lectures and hands-on exercises per week.

A set of mandatory assignments (groups, oral presentations).

A larger software development project applying agile methods.

Writing of a master's thesis project proposal

(joint lectures, iterations with feedback from the supervisor(s)).

Assessment: 5-hour written exam.”

MOD251: Modern Software Development Methods

- Pensum:
 - Få spor av forskningsbasert innhold
 - Bra materiale om forskningsmetode, men kun Shaw-referanse fra systemutviklingsfeltet
- Studieopplegg:
 - Vanskelig å vurdere om læringsmål er oppnådd med dagens eksamensform. Eksamen fra 2013 teori-orientert.
- Undervisning:
 - Mye øvelser og gruppearbeid, god kobling til Booster i 2016. Mange tema og lite forskningsbasert pensum en trussel mot kompetanse og ferdigheter.
- Kobling mot program:
 - Ingen spor av fokus på “store programutviklingsprosjekt” (large-scale agile development, global software development)
 - Lite om “moderne systemutviklingsmetodar”, nye trender som Kanban, Lean software development, DevOps. Bra på “eldre trender” som XP og Scrum.
 - Et kurs hvor evner i “gruppesamarbeid” kan utvikles mer. Lite fokus på metoder for læring, som retrospektiver i Scrum.

Research on agile software development

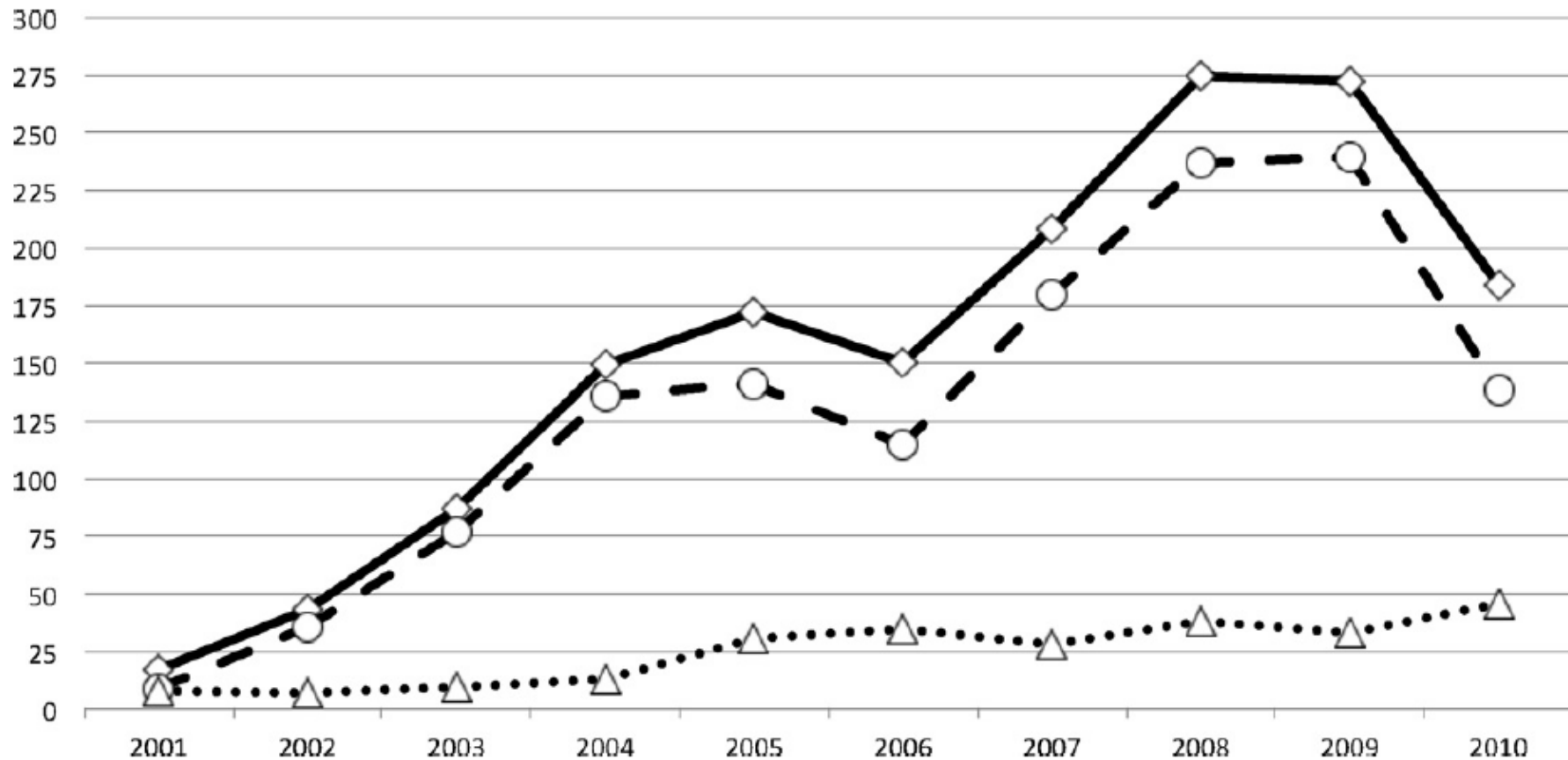


Fig. 1. Publications on agile software development from 2001 to 2010, total number (top), conference papers (middle) and journal articles (bottom).

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B., "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software*, vol. 85, pp. 1213-1221, 2012.

Trender innen systemutvikling



that certain developments and transitions will occur, even if it's not clear when. Here, I summarize six trends influencing the evolution of software engineering practices. The trends are organized from high level and societal to more specific and technological.

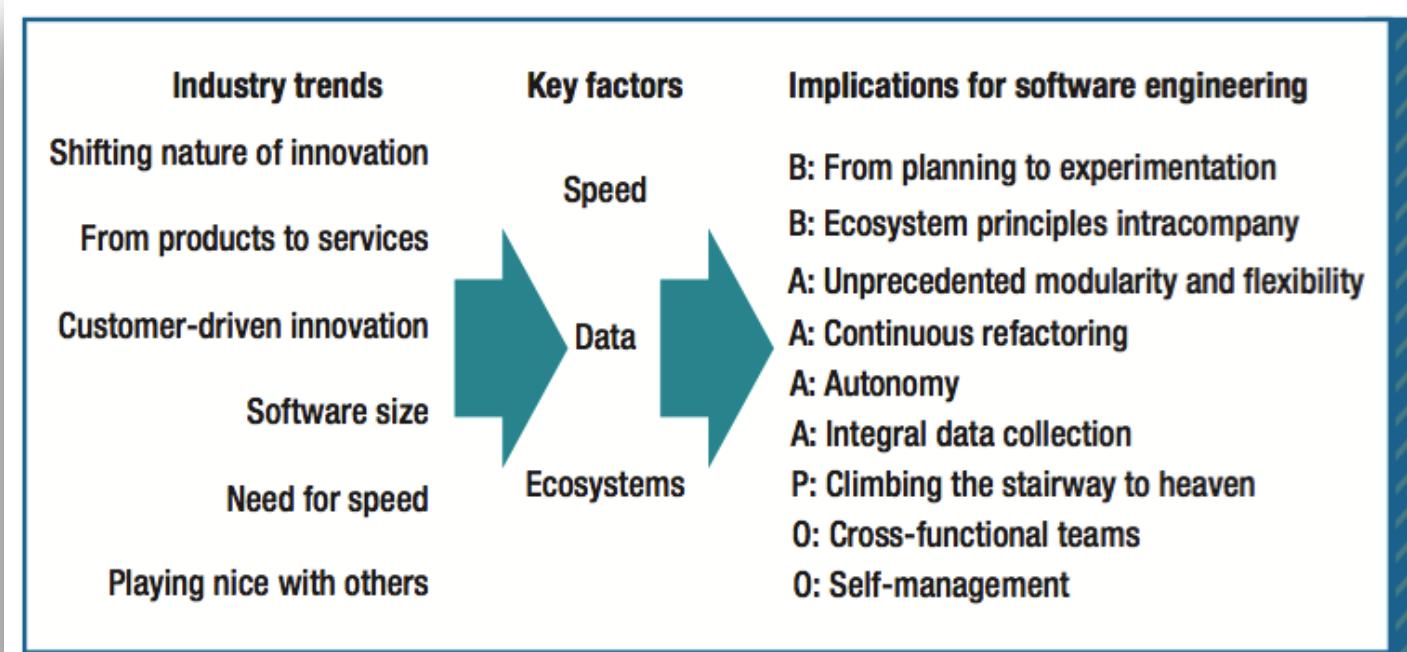


FIGURE 1. The trends and factors influencing software engineering's future, and the implications for business (B), architecture (A), process (P), and organization (O).

4. Anbefalinger om rekruttering

Utfordring

- Kvinneandel
- Mange ukvalifiserte internasjonale søkere

Anbefalinger

- Eget program for å styrke rekruttering av kvinner
- Velge strategi: Utnytte internasjonal profil
- Involvere studenter i rekrutteringstiltak

5. Tiltak for jevnere fordeling

Utfordring

- Skjev fordeling mellom PU og PT

Anbefalinger

- Relevans av PT: Klarere kobling mellom faginnhold og praktisk arbeid. Kobling til lokalt næringsliv.
- Undervisningsform PT: økt bruk av gruppearbeid, vurderingsform. Innspill fra PU på form.
- Involvere studenter i rekrutteringstiltak.

6. Forslag til videre forbedringsarbeid

Dagens status:

- Emneevaluering hvert semester
 - PT: sluttevaluering med spørreundersøkelse
 - PU: midtveis og sluttevaluering
- Lite henvendelser fra studenter til programstyre, men velfungerende fagutvalg

Forslag til tiltak:

- Bruk av referansegrupper for å få tilbakemelding i løpet av semesteret, eventuelt adoptere HiB sitt system på PT
- Større eksperimentering med vurderingsformer
- Involvering av studenter i programstyrearbeid
- Fagfellevurdering av undervisning på tvers av UiB / HiB

7. Oppsummering, prioritering av tiltak

Generelt:

- Populært program, god gjennomstrømning
- Stor valgfrihet (PT) og god fokus på praksis (PU)

Viktigste muligheter for utviklingsarbeid:

1. Kontinuerlig forbedringsarbeid
2. Relevans (PT)
3. Faglighet (PU)
4. Nytt obligatorisk fag; Mer i tråd med ACM Curriculum Guidelines
5. Synergieffekt av miljøene
6. Øke basis i systemutvikling
7. Øke studentenes ferdigheter i gruppesamarbeid (PT)
8. Prosjektfag