Cross-disciplinary challenges in Open Source Software (OSS)

GoOpen 2010 conference attn/ Friprogsenteret
19-20 April 2010 (rev. 4 May 2010)
Aker brygge, Oslo

Reidar Conradi, Daniela S. Cruzes
Department of Computer and Information Science (IDI)
NTNU, NO-7491 Trondheim

http://www.idi.ntnu.no/grupper/su/oss/goopen-rc-20apr10-v2.ppt
conradi@idi.ntnu.no, dcruzes@idi.ntnu.no

20.04.2010
Table of contents

1. IDI and NTNU: who, what, …
2. Software reuse and CBSE: the precursors
3. The Open Source Software (OSS) phenomenon
4. OSS selection: no formal classification, so ”situated”
5. OSS project work flow: from linear to networked
6. OSS: leads to Open Innovation, new innovation and business models
7. Future R&D and cooperation
8. Referances
FYI: IDI Department at NTNU

- **IDI (2009):** 135 employees, NTNU’s largest – 31 nationalities!
- 45+ teachers (faculty w/ six women), 22 tech./adm., 55 PhD fellows, 20 temporary researchers/postdocs/teachers incl. 8 adjunct teachers (I’llere).
- 6000 individual exams per year, 800 full-time students, participating in 7 study programs.
- 125 master candidates and 10 PhD candidates per year.
- Important **value chain:** teachers -> postdocs -> PhD candidates -> master candidates -> bachelor candidates – all important for innovation in IT industry.
- 11 research groups, incl. SE group below.
- Budgets 2008: 74.8 MNOK from NTNU + 23.3 MNOK by projects.
- Budgets 2009: 80.1 MNOK from NTNU + 26.0 MNOK by projects.

- **Software engineering (SE) group in 2009:**
  - 5+3*0.2 = 5.6 teachers, 5 researchers, 17 PhD fellows – 13 nationalities!!
  - All papers: 44/61(#counting/#all in 2007), 40/47 (2008), 45/56 (2009); i.e. 25 % of IDI total, ca. 20 each year w/ foreign colleagues; 500 papers in last 10 years.
  - Ca. 25 master candidates/year; 3 PhDs in 2009, 4 planned in 2010.
  - 7 MNOK in external projects (40% of IDI total.)
1. Research fields of the SU group (2)

- Software 
  quality
- OSS/COTS: CBSE, Evolution, SCM
- Software reuse and architecture
- Reliability, safety, maintainability
- Distributed Software Engin.
- Mobile Tech. for Learning
- Co-operative work
- SPI, learning organisations, SE education
- Software and Art; Computer games

20.04.2010
1. Research profile of SE group (3)

- Empirical research methods (all members)
- Quality, QA, SPI, knowledge management (Conradi, Stålhane)
- Reliability and safety, testing (Stålhane)
- Software reuse, component-based development, open source / digital content (Conradi, Jaccheri)
- Cooperation technologies, learning, awareness, virtual 3D (Divitini, Prasolova-Førland)
- Mobile and ambient technology (Divitini, Wang)
- Computer games for higher learning (Wang)
- Software and art (Jaccheri)
1. Research projects in SE group (4)

• EVISOFT on Industrial SPI, NFR BiA, 2006-2010 (Conradi, Stålhane)
• NordicOSS network, Nordforsk, 2009-2012 (Conradi)
• CESAR on Software Safety, EU IP, 2009-2013 (Stålhane)
• LIKT ("Læring med IKT") research activities like MOTUS, NTNU, 2005-2012 (Divitini)
• FABULA. NFR VERDIKT, 2008-2012 (Divitini)
• ASTRA on awareness during cooperation, EU FET, 2007-2010 (Divitini)
• TESEO, future umbrella project in cooperation technologies (Divitini)
• MOSS on MOBILE and Social games, NTNU, 2008-2012 (Wang)
• Lecture games, NTNU, 2008-2012 (Wang, Trætteberg)
• 3D VR, LIKT NTNU, 2010-2011: Virtual Campus, Virtual Forskningstorg, Virtual Eidsvoll;
• Two EU projects: TARGET, Travel in Europe, 2009-2011 (all Prasolova-Førland)
• SART on Software and ART, IDI, 2006-2014 (Jaccheri)
• Art and Technology - ArTe www.artenu.com, May 2009 - March 2010, NFR Proreal (Jaccheri)
• KKK: Komputer + Kunst = Kreativitet, 2010, Norw. Culture Council (Jaccheri)
• REMICS, EU STREP on cloud computing, 2010-2013 (Mohagheghi)
• Migration to services, internal SINTEF project (Mohagheghi)
• Enterprise modeling and cooperation (Sobah A. Petersen)
2. Classic software reuse inside companies (1)

- **Software reuse: assets** – e.g. class libraries in internal repositories
- Develop “for” reuse: make generalized assets / components
- Develop “with” reuse: CBSE, using these assets, “inner source”
- **Advantages of CBSE**, either with internal or external OTS components:
  - Cheaper, earlier, better, … software
  - Spread novel work practices & software architectures, leading to standardization
  - Re-user communities for experience mgmt/support
  - I.e. software development gets “modernized”
- **Mainly cross-disciplinary obstacles**: price/licensing, ROI, (re)negotiate requirements, unclear responsibility, “not-invented-here”, company policies.
- **Now OSS: massive global reuse** - over half million software components.
3. The OSS phenomenon (1)

- **OSS** recently: a benign “tsunami”; enabled by the Internet and massive investments by companies.
- Cross-disciplinarity of “social computing” and normal SW development: distributed, peer reviews, incremental.
- Based on launched OSS **projects**, each with an open community of involved persons.
- Many **licensing models**: Strict GPL (contagious “copyleft”) vs. generous BSD (any after-use is allowed).
3. The OSS phenomenon (2)

• **OSS**: changes the whole paradigm of software development and associated economic patterns and interactions – by cooperative, distributed innovation.

• Norwegian software-intensive companies and public institutions must undergo this economic and cultural *revolution*:
  – novel Innovation Models: new products and services
  – novel Business Models: make money on these

• Need **professional partnership communities** to establish and evolve their needed software, itself being OSS.

• Move away from “unpredictable” volunteers working for free.

• **OSS**: adopted by *private and public policy makers*: e.g., IBM and Sun Microsystems, IKT-Norge, …, Skattedirektoratet, KS, …

• St.meld. 17 (2006-07) by Norwegian Government recommends OSS and open standards.

• **www.FriProg.no** competence center. Also **www.FriNett.no** (NFR).

20.04.2010
3. OSS: origin and now (3)

- **1650?**: free flow of ideas begins in emerging scientific community.
- **1967-1977**: DoD/DARPA’s development of Arpanet / Internet TCP/IP protocols -- open and distributed work, anti-authoritarian “counter culture”.
- **1978-85**: Openness spreads via Univ. Berkeley’s adaptation of Unix, with built-in Internet protocols and generous BSD licenses with all source provided.
- **1985**: Free Software Foundation: by Richard Stallman from MIT AI Lab; ideological ”CopyLeft”-inspired licenses.
- **1998**: Open Software Initiative: by Eric Raymond from IT industry in US, pragmatic BSD-inspired licenses. ”The Cathedral and the Bazaar”:
- **2000**: FLOSS = FSF + OSI; IFIP WG2.13 uses OSS as common denominator.
- **2000**: LAMP platform (Linux, Apache, MySQL, PHP) – 3/4 Nordic!
- **2005**: Social computing and creative commons for knowledge work – standardized formats for “zero-cost” digital information, the world is ”flat”.
- **2008**: 1,5 billion Internet users, 3 billion w/ mobile phones.
3. OSS facts and challenges (4)

Facts:

• **Ex.** Over 50% of Norwegian companies use OSS to make new software, 16% are community members, 6% have themselves “gone open”.
  Ex. eZ has had 2.3 mill. downloads, over 100 plug-ins made by community.
  Ex. Adobe Acrobat similarly over 500 mill. downloads.

• **OTS = COTS and OSS**: many similarities, mostly used as black box.

• Over 30 portals with domain-specific COTS/OSS software.

Three X-disciplinary Challenges:

• **Situated selection** of promising components, auto-generated ontologies.

• **New ways of cooperation and sharing**: new work organization.

• **Innovation model**: combine N open technologies into marketable products, & **Business model**: - and making money of this!

20.04.2010
4. “Ladder” of OSS roles (1)

- **OSS (component) provider**: Leads a “gone-open” software project, ex. Linus Thorvalds w/ Linux.
- **OSS co-developer** or **participant**: assists the provider in more technical work – the OSS “idea” of joint, altruistic work.
- **OSS advanced integrator**: reports error reports and change requests for new features, sometimes changing the code.
- **OSS integrator**: makes new software system by **re-using** OSS software/components, makes use of OSS CASE tools (Eclipse, SVN) and techniques.
- **OSS customer**: specifies and finances new software, later published as OSS?
- **OSS user**: may download “free” (not open) binary software on a private computer, e.g. Acrobat.

- **OSS skills**: technical, commercial, organizational, creative.
4. Roles in OTS-based reuse (2)

Development with OTS components

OTS component-based development process

Component selection
Component integration
Component maintenance

OTS component-related knowledge

Component provider
Integrator
Customer

20.04.2010
4. OSS selection by Integrator (3)

1. Elicit and formulate requirements, R.

2. Then, check if you or your colleagues have concrete experience with some partially matching OSS components.

   Ex. there is an entire product P that solves 90% of your R now, plus 500+++% of what you never thought of. What to do:
   • Develop own P’ from scratch (6-24 months; risky), modify P, or just settle with P?
   • Don’t even bother with an explicit R; just “fish” for something that might lead to interesting Ps.

3. OSS component selection more general:
   3.1 Search for OSS components (Cs) that matches R, using e.g. Google.
   3.2 Evaluate the most relevant Cs – download hottest 2-6 candidates.
   3.3 Choose the most suitable C (“first fit”) – a risky decision with considerable time pressure, lack of info and uncertainties.
4. Component classification (4)

1. Next idea: annotate components with keywords (meta-data) according to a (formal) classification scheme or ontology. Four basic kinds of meta-data:
   1. Functional attributes – may automate ontology build-up/evolution and annotation?
   2. Quality attributes – uncertain and diverse info!
   3. Platform attributes – simple in theory
   4. Policy attributes – tricky?

2. Apply semantic search tool with weights (web 2.0), a la Facebook.
3. Sorry, does not work! – both practical and theoretical hinders.

Rather “situated selection”, using the actual OSS community:

1. You will know about relevant Cs and own R-intentions in a gradual and mutual learning process. Renegotiate R? Reuse and share experiences with others, use wikis? Later evolution of R and Cs?
4. OSS and ontologies (5)

- **Ontologies**: exciting field, but heavily oversold by AI and linguistics.
- Basic problem to have *one* agreed-upon ontology, due to different and irreconcilable worldviews [Bowker99]:
  - Ex. book as a legal object (IPR), information carrier (on paper), investment object, art, fuel, weapon, back-cover in bookshelf, slide projector fundament, …
  - Ex. Homosexuality as a (curable) “decease”, “democracy” in previous DDR, …
  - Ex. NIH medical taxonomy has 1.8 million terms – how to use consistently?
  - Ex. Misleading “tag” classification of newspaper articles by journalists.
  - Ex. MIS Quarterly recently refuses to have keywords on submitted papers.
- To not miss finding some “textual” object: 1) Have *automatically generated and applied ontologies*; or 2) Let powerful search engines chew through *all text* in (software) components.
- In all this - versatile support and pragmatic experience is needed.
5. From linear to networked paradigm (1)

Before:

Now:
5. Networked development actors (2)

Emerging network paradigm for OSS will affect:

- Long-term vs. short-term planning,
- Cross-company cooperation and dependence: (re)negotiation vs. dictate,
- Control mechanisms: many risky parts,
- (Open) Innovation: new combinations, spin-offs, …
- Legal: IPRs and licensing,
- Economical: new business models,
- Marketing: close to development,
- => Requirement Engineering reborn!
- See [Ayala09] [Hauge10] [Lindman09].
6. Open Innovation (1)

Proposed by Henry Chesbrough (2003, 2006)

Scenarios Before and Now:
6. Open Innovation – by combinatorics (2)
6. Open Innovation (3)

Closed Innovation

Open Innovation

boundary of the firm

Ideas, Artifacts

current market

new market

boundary of the firm

Ideas, Artifacts

current market

20.04.2010

X-disciplinarity in OSS; GoOpen, 20 Apr. 2010
6. Altruism promotes capitalism! (4)

- **Traditional innovation** models: based on the “egoistic” protection of Intellectual Property, i.e. cannot capture value from OSS for companies. So your cards lie with “face down” and contents hidden, as a private and short-term sub-optimalization.

- **Novel innovation** models leave your cards “face open” with contents accessible for everyone. I.e., such altruism promotes maximum combinatorics and thus future innovation around OSS, i.e., public and long-term benefits for both society and companies – a win-win.

- Guidelines for revised development processes from “private investment” to “collective action” hardly exist.

• Several issues need to be addressed:
  – How to succeed in attracting co-developers when “going open”?
  – How can traditional (i.e. paid software development by software companies), coexist and even amplify the benefits for both the “old” and “new” work mode?
  – So rely less on classic OSS idealism, rather refine and commercialize more cooperative work modes.

• Learning ecosystem to systematize the (meta-) services that might be needed (by Friprog, ch. 6.8):
  – How to establish such an ecosystem? Roles?
  – How to build competence?
  – How to offer legal or economic advise?
  – How to make decisions about changes in old vs. new systems?
  – How to build and maintain expert communities?
  – How to manage experience bases (avoid the “white elephants”)?
  – Practically: if (most of) the code is shared and “free”, where is the profit coming from?
6. OSS Business Models (6)

- In general, to enter in the OSS field, companies have to manage Intellectual Property (IP) differently and to innovate more:

![Diagram showing business models and revenue streams for OSS]

- Own market revenue
- Internal development costs
- License
- Spin-off
- Sale/divestiture

New revenues
Cost and time savings from leveraging external development

Chesbrough2006
6. OSS Business Models (7)

- From COSI ITEA2-project:
6. OSS Ecosystem for support - by Friprog (8)
6. Summary of X-disciplinary aspects in OSS (9)

- **Situated selection** of promising components, pragmatic approach, later auto-generated ontologies for functionality properties.

- **New ways of cooperation and sharing**: new work organization, very interesting paradigm – “The world is flat”, global cooperation. Ex. Cooperate on shared, domain-specific middleware.

- **Open Innovation model**: altruism gives the highest long-term profit, by combining several open technologies into new products and services [Browning08] [Chesbrough03] [Chesbrough03a] [Chesbrough06].

- **Business model**: and making money of this - proven to work!
  - Apache: open and shared source supported by a cooperative foundation (“IBM”); separate payable services.
  - eZ: dual model with free previous version, payable current version plus services and support.
7. Future R&D and coop. on OSS (1)

How to succeed with OSS?

- Experience to NTNU from industry-related research
  - What should we focus on in future research?
- Experience to you from theory-driven research at NTNU
  - How to validate in practice?
- Experience to NTNU about our education
  - What should students know about OSS?
7. Ex. NTNU Resources on OSS (2)

- Software engineering group (SU in Norwegian)
  - publications, master student reports, and courses
  - www.idi.ntnu.no/grupper/su/ (SU group homepage)
  - www.idi.ntnu.no/grupper/su/publ/INT-PUBL.php3 (publ. list)
  - www.idi.ntnu.no/grupper/su/oss/ (SU on OSS)

- Wiki on open source research at NTNU
  - http://research.idi.ntnu.no/oss

- The ITEA2 COSI Project (2006-2008)
  - Norwegian partner IKT-Norge w/ subcontrcator NTNU
  - http://www.itea-cosi.org/

- The Nordic OSS Network, Nordisk råd (2009-2012)
  - Seven partners incl. NTNU, coord. by Bjørn Lundell at U. Skøvde
  - http://www.??.his.se
7. Ex. Open Source follow-up (3)

- Previous **Norsk COSI**, 2006-2008, NFR BiA. Shared Open Source Software into European ICT industry. Totally 8.5 MNOK.
- Contact: **Reidar Conradi**.
- Prev. PhD students/researchers: Carl F. Sørensen, Sven Ziemer, Øyvind Hauge.
- PhD student: Odd Petter Nord Slyngstad (SEVO project on software evolution); revised thesis in June 2010.
- Researcher: Jingyue Li, Postdoc: Daniela S. Cruzes.
- NordicOSS academic network: strategic meeting place.
- - Applying to NFR-VERDIKT program about research projects in cooperation with Norwegian industry and foreign colleagues …
7. Ex. NordicOSS research network (4)

- Nordic Council network project, 2009-2012, 1.2 MNOK, only for travel and meetings, 7 partners.
- Try to initiate common research in OSS: joint papers, new projects especially from EU, workshops and similar events.
- Affiliated: PhD stud. Øyvind Hauge, postdoc Daniela S. Cruzes.
7. Ex. Also EVISOFT project industrial SPI (5)

- EVISOFT, NFR BiA, 2006-2010, 8 MNOK/year.
- Experience-based SPI in Norwegian IT industry. 4th similar project in a row.
- Telematikk IKT (coord.) + 10 companies + SINTEF, UiO, NTNU (research partners).
- Focus: Scrum, testing and defect analysis, use-case based estimation.
- Faculty: Reidar Conradi, Tor Stålhane
- Researcher: Jingyue Li ("Bill")
- PhD students: Geir Kjetil Hanssen (agile methods), Børge Haugset (acceptance testing) - both SINTEF.

20.04.2010
7. Possible future NTNU Resources (6)

- NFR-VERDIKT proposal by R. Conradi, Daniela S. Cruzes (both NTNU), and Tonje Osmundsen (NTNU Samfunns-forskning (Eds.): “Maintaining Software Systems with an evolving network of open source providers (MAIN-SOFT), 25. Nov. 2009. REJECTED!

- NFR-SFI proposal by SINTEF, UiO, NTNU (Dybå, Sjøberg, Conradi) + 14 partners: Research in SE (RISE) - on agile methods, OSS, application portfolio complexity. 21 April 2010.

- Revised education plans
  - National research school
  - International master in OSS
8. References (1)