The Research Council of Norway

Application form 2000

Project number:
For Research Council use

1 Project title (max. 100 characters)
Incremental and component-based software development (INCO)

2 Applicant	Institution/company responsible for the project					Project manager	
Institution/ company (name)	UiO				Project manager (name)	Dag Sjøberg	
Faculty/Institute/ Department	Department of informatics						
Address	P.O. Box 1080 Blindern				Address	Department of informatics, UiO	
City and city code	0316 Oslo						
Web-site	http://www.ifi.uio.no/						
Registration number	97 10 35 854				City and city code	0316 Oslo	
Auditor	Office of auditor general		Other		Position/title	Professor	
Administrative responsibility (name)	Narve Trædal				Academic degree	PhD	
Position/title	Administrative Head of Department				Preferred language	Bokmål x	Nynorsk
Phone and fax	2285 2418	2285 2401			Phone and fax	2285 2742	2284 0072
E-mail	narvet@ifi.uio.no				E-mail	dagsj@ifi.uio.no	

5 Research Council funding division			
Division	NT	Other relevant divisions	
Programme	IKT-2010	Other relevant programmes/disciplines	
Discipline/ specialist field		For additional funding: give project number.	

Executive officer				Activity	Project type
Division	BF	MU	MH		Discipline code
	IE	NT	KS		

3 Principal objective and sub-goals (max. 100 words)

The **principal objective** is to advance the state-of-the-art and -practice for incremental and component-based software development. This includes developing or improving existing methods, models, techniques and guidelines for such development.

Sub-goals

Build a national competence base around these themes and disseminate the knowledge gained in academia and industry:

1. At least 1 method, model, technique and guideline should be developed or improved.
2. At least 8 papers with INCO results should be published at international workshops, conferences or in journals
3. At least 3 companies should be involved in the INCO work.
4. Results from INCO should be taught in at least 1 course at UiO and 1 at NTNU in 2003 and 2004.

4 Project summary (max. 200 words)
The high costs and high failure rate of present software projects call for better technologies and work processes for software development and maintenance. In recent years, incremental and component-based development have been proposed as remedies – separate or combined – to reduce development time and cost, and to increase software quality (especially usability and reliability). However, there are few validated technologies (methods, models and tools) in these areas in industrial use today, reflecting the immaturity of these technologies. Ifi, UiO and IDI, NTNU will in INCO propose, refine and validate improved technologies for incremental and component-based software development, focusing on UML and RUP with time-boxing, and COTS-based development. Technology for planning and managing incremental and component-based software development projects will also be proposed and evaluated. Moreover, the project will develop guidelines for how to train software developers and achieve organisational learning in incremental and component-based development. Empirical studies will be performed both in industry and at universities.

6 Timetable for main activities in the project																				
Project period: From: 01.01.2001 To: 31.12.2004	2000				2001				2002				2003				2004			
Main activities over the project period	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	
1. Definition of hypotheses and planning of empirical studies in companies, literature review/identification of related work.					X	X	X	X												
2. Collection of measurements, data analysis, preliminary results, technical reports and workshop papers							X	X	X	X	X	X								
3. Revision of models/hypotheses, further measurements/ validation/results, papers, material for courses.									X	X	X	X	X	X	X	X	X			
4. Consolidation of results: final INCO report, conference and journal papers, final material for courses.															X	X	X	X	X	X

x

7 Overall costs (in 1000 NOK)	2000	2001	2002	2003	2004	Sum
Personnel costs and indirect costs		1041	1848	1591	763	524
Purchase of R&D services						
Equipment		150	120	70	70	41
Other operating costs		300	367	340	340	134
Project total		1491	2335	2001	1173	700

Specifications

8 Finance plan (in 1000 NOK)	2000	2001	2002	2003	2004	Sum
Own funding						
EU-funding						
Other public-sector funding						
Other private funding						
From Research Council						
Project total						

9 Cost code (in %)	2000	2001	2002	2003	2004
Commercial participants					
R&D Institute sector					
University sector					
Other sectors					
Project total	100	100	100	100	100

10 Research Council application	2000	2001	2002	2003	2004	Sum
Student fellowships						
Doctoral fellowships		595	1251	1251	625	372
Post-doctoral fellowships		378	529	132	0	103
Fellowships for visiting researchers						
Overseas fellowships		0	0	140	70	21
Technical/administrative positions		68	68	68	68	27
Researcher positions						
Hourly-based salary including indirect costs						
Sum personnel costs and indirect costs		1041	1848	1591	763	524
Purchase of R&D services (specified in sect. 7)						
Equipment (specified in section 7)		150	120	70	70	41
Other operating costs (specified in section 7)		300	367	340	340	134
From Research Council		1491	2335	2001	1173	700

Specifications

- A post-doc position may be replaced with visiting researchers.
- The technical/administrative person will be responsible for maintaining web-pages, developing experiment environments, code quality analysers, metrics collection tools, etc.
- Equipment includes PC and software for PhD students and post-docs/visiting researchers
- Other operating costs include travel for project member and visiting researchers, and costs of developing (electronic) teaching/presentation material.

11 Person for whom a fellowship/position is being sought

Name	Personal number (if applicable)	Type position/fellowship	Period	Position-%
3 PhD students, unknown at present		Doktorgradsstipend	-	
1 post-doc or visiting researcher, unknown at present		Postdoktorstipend	-	
1 person in 1/5 position, unknown at present		Teknisk/administrativ	-	

Foreign institution: (name, address, country)

Travelling alone		Travelling expenses	
Travelling with family			

12 Partners

Norwegian companies:
Computas, Bravida, Ericsson, Genera, Icon medialab, FreightConnect, Mogul (Numerica-Taskon), Telenor, Viewpoint, Veritas

International organisations:
Universities of Maryland, Kaiserslautern, Lund, Grenoble, Milano, Glasgow, Carleton (Ottawa), and companies such as Q-labs and VTT.

13 Project publication plan

14 List of enclosures

1. Project description
2. CVs and publication lists of the applicants

15 Signatures

Narve Trædal	Dag Sjøberg	
Date: 28 September 2000 Adm. responsible	Date: 28 September 2000 Project manager	Date: Grant applican

Dag Sjøberg / Reidar Conradi
UiO/NTNU

Oslo/Trondheim, 28. september 2000

Norges forskningsråd
Postboks 2700
St. Hanshaugen
0131 OSLO

Oslo/Trondheim, 28. september 2000

Følgebrev til *revidert* søknad for prosjektet "Inkrementell og Komponentbasert Systemutvikling (INCO)" for 2001—2004 under forskningsprogrammet Grunnleggende IKT-forskning (IKT-2010)

Inkrementell og komponentbasert systemutvikling (INCO)

INCO foreslås herved av Ifi/UiO og IDI/NTNU. NFR-rammen er på 7000 KNOK over fire år, pluss 7284 KNOK dekket av kapitaliserte universitetsmidler.

Planen er et *nasjonalt metodeprosjekt og virtuelt senter for inkrementell og komponentbasert systemutvikling (INCO)* i Norge, representert ved søkermiljøene ovenfor, med flere "ringer i vannet" av nasjonale deltakere, og med internasjonale koplinger.

Vi vil tilstrebe at INCO kan ligge "rygg til rygg" med ett eller flere PROGIT-prosjekter, og med ulike bedriftskontakter (15-20) både innenfor og utenfor slike prosjekter. Vi vil altså bruke industrielle utviklingsprosjekter, både de som er koplet mot PROGIT-prosjekter og andre mer interne, som en *industriell lab*. I tillegg vil vi utnytte våre egne studenter i småskala-eksperimenter. De mest aktuelle PROGIT-prosjektene er PROFIT, DAIM og CoDeVer.

INCO vil nasjonalt koples til våre PROFIT-kollegaer ved SINTEF. Videre vil utnytte våre kontakter med IT-interesseorganisasjoner som ITF, IKT-Norge, NFKL og DND. Endelig vil vi samarbeide med andre norske utdannings- og forskningsmiljøer for spredning av resultater gjennom kurs, seminarer, web-sider osv., jfr. avsluttet NAWUS-prosjekt ("Digitalt bibliotek for digitale læremidler").

INCO søkes internasjonalt koplet med vårt nett av faglige kontakter, f.eks. mot universitetene i Maryland, Kaiserslautern, Lund og Carleton, til Fraunhoferinstituttene i Kaiserslautern og Maryland, til Q-labs, DnV o.l. Blant annet deltar vi i nettverket ISERN (International Software Engineering Research Network, initiert av prof. Basili og Rombach), og flere tilsvarende konstellasjoner planlegges.

Med vennlig hilsen

Dag Sjøberg og Reidar Conradi

PS
Som det fremgår av søknadsskjemaet, har nye etterprøvbare delmål blitt lagt til og budsjettet blitt nedjustert i denne reviderte søknaden.
DS

Vedlegg:
NFR søknadsskjema
Enclosure 1: Project Description
Enclosure 2: Curriculum Vitaes and Publication Lists

Enclosure 1 – Project Description for:

Incremental and component-based software development (INCO)

University of Oslo (UiO), Oslo
Norwegian University of Science and Technology (NTNU), Trondheim

Revised version
28 September 2000

Proposers: Dag Sjøberg, Ifi, UiO (co-ordinator) and
Reidar Conradi, IDI, NTNU

Contents

1 Summary of Incremental and Component-Based Software Development (INCO)

The high costs and high failure rate of present software projects call for better technologies and work processes for software development and maintenance. Recently, incremental and component-based software development have been proposed as remedies – separate or combined – to reduce development time and cost, and to increase software quality (especially usability and reliability). However, there are few such technologies (methods, models, and tools) in industrial use, reflecting the immaturity of these technologies. For instance, incremental development needs better support on overall planning and on estimating risks and costs. Likewise, COTS (Component-Off-The-Shelf) based development suffers from poor integration with requirements engineering, architectural modelling and analysis, and cost/risk estimation.

Overall goals:

G1) Advancing the state-of-the-art of software engineering, focusing on technologies for incremental and component-based software development.

G2) Advancing the state-of-the-practice in software-intensive industry and for own students, focusing on technologies for incremental and component-based software development.

G3) Building up a national competence base around these themes.

G4) Disseminating and exchanging the knowledge gained.

Approach:

1) Propose, refine and validate enhanced technologies for incremental and component-based software development, focusing on UML and RUP with time-boxing, and COTS-based development (R1 below).
2) Derive a menu of PhD study topics in co-operation with industry serving as “software laboratory” (R2).
3) Conduct two phases of technology development and empirical studies, at univ. and in industry (R3).
4) Assemble and disseminate the results (R4).

Close contact with Norwegian, industrial projects (especially PROFIT) is envisaged, although we can run INCO solely based on existing industrial contacts and student projects.

Results – mainly as PhD theses, papers, reports, and Web material – and covering the following:

R1) A set of improved technologies for incremental and component-based software development.

R2) Guidelines for training software developers and for achieving organisational learning in incremental and component-based software development.

R3) A body of experiences from empirical studies of such technologies.

R4) General presentation material, new/revised courses, and a new national seminar on INCO themes

Partners: The software engineering groups at IDI, NTNU and Ifi, Univ. Oslo both have 3-4 teachers, 4-5 PhD students and 20 MSc students involved in related education and research. The two groups have been co-operating for over 7 years, the last 4 years on the national SPIQ and PROFIT projects (on software process improvement), with regular project and research meetings. Both groups have extensive connections to Norwegian industry, partly through the same projects. Together, the two groups have participated in more than a dozen EU projects in software engineering. The groups also participated in the NAWUS project (on software engineering education in “Norgesnettet”) in 1998-1999.

Budget and management: INCO is planned for 4 years, with 3 PhD students and 1 postdoc/guest researcher per year financed by NFR – totally 7000 KNOK (1491 in year 2001). 3 PhD students will be financed internally by the two universities. In addition comes involvement of teachers, guest researchers and the surrounding university environment. Total capitalised effort is estimated to 7284 KNOK from the university side. Two main evaluations are envisaged, respectively after 2 and 4 years. Project management lies at UiO, and four quarterly common project meetings are planned, plus monthly status meetings among the co-ordinators.

Keywords: Software engineering, incremental development, component-based development, software quality, heterogeneous components (COTS), empirical studies.

2 Goals

The overall INCO goals are:

G1) Advancing the state-of-the-art of software engineering, focusing on technologies for incremental and component-based software development.

The focus is two-fold:

– Technologies (methods) for **incremental development** aim to substantially reduce the costs and risks associated with evolving requirements, due to rapid and unforeseen changes in markets or technologies.

– Technologies (methods) for **component-based development** cover both internal components and external COTS components, and aim to build new systems from existing parts in a more cost-effective and reliable manner.

G2) Advancing the state-of-the-practice in software-intensive industry and for own students, focusing on technologies for incremental and component-based software development.

The aim is to use our industrial contact network and our own students to perform a spectrum of empirical studies, from formal experiments to case studies, in order to influence present and future software practice.

G3) Building up a national competence base around these themes.

The aim is to strengthen the Norwegian academic and industrial IT community in the areas of incremental and component-based development.

G4) Disseminating and exchanging the knowledge gained.

The aim is to disseminate PhD theses, papers, reports, educational and presentation material etc. through the Web, in own education, and at appropriate national and international fora.

Note that research activities will have first priority, although we will need (and have) a tight coupling to industry to perform realistic studies and field work. Such practical work may be carried out in other projects under NFR's PROGIT program, or by company-internal projects.

3 Results

The following results are planned:

R1) A set of improved technologies for incremental and component-based software development.

The results will be three-fold:

- **Overall/managerial:** Improved methods and techniques for planning, risk analysis, effort estimation, and management of incremental and/or component-based software projects.
- **Technical:** Improved methods and techniques for requirement engineering, architectural modelling, design, implementation, and testing in incremental and/or component-based software development.
- **Methodological (partly):** Improved or annotated research methods for software engineering in general, as a side effect of trying out the above methods.

Such technologies include process models (methods) and guidelines, architectural models/taxonomies, and cost/risk estimation models. An underlying platform of UML/RUP and COTS is assumed. Note, that there is a strong interaction between incremental and component-based approaches, and between these and requirement engineering / project planning.

These technologies will contribute to reach goal **G1** on advancing the state-of-the-art, and **G3** on developing a national competence base.

R2) Guidelines for training software developers and for achieving organisational learning in incremental and component-based software development.

The results are two-fold:

- Guidelines for applying the R1 technologies for incremental and/or component-based software development, both in an industrial and educational context.
- Guidelines for how to train software developers and to achieve organisational learning in incremental and component-based development.

These guidelines will be synthesised from the outcomes of empirical studies (see R3 below), and be targeted both towards industrial developers and own students.

The guidelines will contribute to reach goal **G2** on advancing the state-of-the-practice, and **G3** on developing a national competence base.

R3) A body of experiences from empirical studies of such technologies.

The empirical studies will involve industrial case studies in the 10-20 companies that we now work with, and more formal experiments involving our own students (2-3 relevant courses at each partner). These studies must be carefully planned and carried out (cf. R2). The raw data and derived experiences/models will partly reside in company-internal *experience bases* and partly in central Web-bases at the universities.

The experiences will give feedback to results R1 and R2 above, and contribute to goals **G1, G2** and **G3.**

R4) General presentation material, new/revised courses, and a new national seminar series.

This presentation and educational material will have the form as course notes, presentation foils, exercises, repeatable experiments etc. It will be used towards own and other students and towards industry. It will be presented at regular and external courses, workshops and conferences, seminars and specialist meetings. Most material will be available from an INCO **Web site** that will be continuously updated.

A new, semi-annual, national seminar on INCO themes and **empirical software engineering** will also be arranged, partly in conjunction to the annual PROFIT seminars.

These results contribute to goal **G3** and to goal **G4** on dissemination.

All the above results will be delivered as parts of PhD and MSc theses, published papers, reports, experience reports, educational and presentation material etc. – mostly accessible over the Web.

It is hard to quantify the economic value of these results, as INCO is meant to be a PhD-type research project. However, considering 60,000 software developers in Norway – if INCO results are influencing 1000 of these and with a 10% “improvement” (which can itself be subject to INCO studies!), the overall effect may be substantial. The indirect results on end-users that rely on working IT systems come in addition.

Further, there is a huge, indirect and long-term benefit by giving more relevant education in and exposure to novel software technologies for university students.

A remark on terminology: when we later say “development”, we will assume “software development”, unless another meaning is explicitly stated.

4 Resources

The table below shows the resources applied for. The units are PY (person-years) and KNOK (1000 Nkr). The effort and thus the resources will roughly be divided equally between NTNU and UiO. The applied amount from NFR is 7000 KNOK. In addition, 7140 KNOK are covered by the university partners on INCO related work. Standard NFR rates have been applied on wages.

Category	**2001**	**2002**	**2003**	**2004**	**Sum PY**	**Sum KNOK**
3 PhD fellows	1,5	3	3	1,5	9	
1 Post-doc/guest researcher	0,75	1	0,25	0	2	
Sum PY, NFR-part	**3**	**4**	**4**	**2**	**13**	
Wages PhD students	595	1251	1251	625		3722
Wages post-docs/guest res.	378	529	132	0		1039
Overseas fellowship	0	0	140	70		210
Technical/administrative position (1/5)	68	68	68	68		272
Guest res./postdoc, travel	0	67	40	40		147
Travels	300	300	300	300		1200
Equipment	150	120	70	70		410
Sum KNOK, NFR-part	**1491**	**2335**	**2001**	**1173**		**7000**
3/4 of 4 PhD fellows	3	3	3	3	12	
PhD advisors	1	1	1	1	4	
Guest res., part-time	0,25	0,25	0,25	0,25	1	
SUM PY, univ.-part.	**4,25**	**4,25**	**4,25**	**4,25**	**17**	
SUM KNOK, univ.-part	**1821**	**1821**	**1821**	**1821**		**7284**

The technical/administrative person will be responsible for maintaining web-pages and developing experiment environments, code quality analysers, data capture tools etc. Equipment includes PCs and software for PhD students and post-doc/visiting researchers. Other operating costs include travel for project member and visiting researchers, and some costs of developing (electronic) teaching/presentation material. Candidates for post-docs may be our present PhD students or those of our colleagues.

As shown, NTNU and UiO each expects to have two additional PhD students financed by internal funds on the INCO topic. They are already in place in Oslo, and scholarships are being advertised at NTNU, thus 3/4 research share for 4 students. One post-doc is also included for two years. Marginal extra costs for guest researchers (two per year and 5 weeks per visit) are also included. The 3 PhD students are each expected to stay abroad for 6–8 months, and extra overseas fellowships of 70.000 KNOK per student are included.

Included in the budget are the contributions by academic employees (prof., assoc. prof.) who supervise PhD-students and conduct own research related to the project. The effect of mobilising own MSc students is not counted, but is expected to be considerable ("light-house effect" of a factor 2-3). No direct support from industry is included.

5 State-of-the-art

This section describes state-of-the-art, previous own work and research trends of respectively incremental development, component-based development, software process improvement and relevant research methods.

5.1 Incremental Development

In incremental development several, smaller parts (increments) of the application system is released instead of one "big bang". Each increment delivers a portion of the functionality of the overall system. Incremental development projects are usually conducted in an evolutionary, that is, non-serial manner. For simplicity, in this application we use the term *incremental development* to denote development projects that are *both* incremental and evolutionary.

Incremental development has been proposed as an efficient way to deal with risks such as new technology and imprecise or changing requirements [Boehm88]. The main idea is to resolve risks early by incrementally evolving the system towards completion instead of relying on the waterfall approach [Royce70]. Thus, an important objective of incremental development is to identify the "real" needs of the customer as the system evolves, cf. *rapid prototyping*. While experience reports show some success in the application of incremental development [Gilb88] [Zamperoni95], for example, Rational's Unified Process (RUP) is being adopted by a growing number software development companies, there are unfortunately few empirically based guidelines on how to apply such processes in different development contexts.

Incremental development may also mean *time-boxing***,** typically with 4–6 months intervals, cf. the DSDM method. Many companies already practice some kind of incremental development, e.g. Computas (using DSDM) Telenor, Cap Gemini, etc. That is, more and more application owners now feel the impact of "new" and rapidly evolving IT technologies, constantly wetting the appetite for new and revised services and products – cf. the mobile phone market. Thus, "Internet time" means that everything changes much faster, both apply to technology and markets. Carefully drafted requirements tend to become obsolete after just a few months. This means a massive amount of changes, both during and after initial development. Time-boxing is an attempt to "freeze" the technology and requirements, at least for a few months, so that we can have a semi-orderly development.

Consider also the "project triangle", with functionality/quality, schedule and effort, respectively, in the three corners. Schedule is often sacred, so the two other corners must adjust. Traditional Telecom has so far escaped these effects, but hardly anymore. Capers Jones estimates from (old) data that requirements "rot" by 1-1.5% per month, so after 3 years about half of the original requirements are gone. "Better" requirements engineering in the first place could not have remedied this, cf. the failed Iridium project which started in 1988 and was delivered in 1998, and that 1/3 of all software developments end up in "dead-before/dead-upon-delivery".

Own work on incremental development:
The Oslo group initiated and managed the interest group on incremental development in the SPIQ project [Dybå00]. The Trondheim group was also active in the interest group. In SPIQ the Oslo group worked closely with the companies Genera, Mogul (Numerica-Taskon) and Ericsson on incremental development [Arisholm98, Arisholm99a, Arisholm99b, Arisholm00a]. Controlled experiments with students at different levels have also been conducted [Arisholm00b].

Research trends on incremental development:

- Estimation, risk analysis, contract specification, project management and configuration of such processes
- Planning and making a stable design/architecture that is suited also for the next set of semi-planned and delayed deliveries
- Prioritisation of requirements
- Down-scaling the large number of prescribed deliverables, roles and activities in RUP to be practical for use in relatively small development projects
- Studies of actual evolution, both in requirements and designs: cause, effect, when, by whom, with what cost

5.2 Component-Based Development

Component-Based Development covers both reuse of traditional libraries and object-oriented frameworks, as well as program/product families and COTS (Commercial-Of-The-Shelf). Cf. also the emergence of software architecture and domain-specific design patterns as a separate branch of software engineering.

The reusable components may be any software artefacts: requirements and analysis documents, software architectures, detailed designs, code (source and binary), test data, and documentation.

The main *benefits* of reuse are more standard solutions and architectures, shorter time-to-market, higher productivity, and better reliability (i.e. lower defect rates). All these benefits have been demonstrated industrially over the last 10 years, see e.g. [Basili91] [Lim94] [Reifer97] [Morisio99]. Typically, a reusable component "earns" its value after 2-4 years, by being reused at least twice. The net economic savings are considerable, typically a 20-80% return on initial investments.

Reuse often implies, that *reuse decisions* (e.g. what libraries/components to possibly develop or reuse) and *requirements decisions* (e.g. what new functionality should be prioritised and when) become entangled. That is, we need a *revised development process for reuse*, with feedback loops to handle project risks, scheduling and decision making.

Classic software (code) reuse assumes *white-box* reuse, where reusable artefacts (libraries, OO frameworks) are compiled into the application code. A common approach has been on developing own or shared *component repositories*, where potential reuse candidates are classified, stored, and later searched for, fetched, and possibly modified. However, the repository-based reuse approaches have had limited success [Poulin95], unless being integrated into a larger reuse process.

Furthermore, attempts at reusing components across product families and/or projects have revealed serious problems of *architectural* and *organisational/economical* nature. First, a component needs an overall architecture (perhaps its own API) to relate to, i.e. which "slot" it is filling. Second, reuse means investing *now* in more generalised components (and in infrastructures such as component repositories), that may not be (re)used at all *later*. Thus, a cost/benefit assessment for reuse must be made. Lastly, Ralph Johnson has emphasised that reusable components are "not (pre)planned, they are discovered (gradually later)" [Gamma95].

Work in the new area of *software architectures* has focused on *design patterns* [Gamma95] [Hofmeister00], architectural styles, connector semantics and special architectural description languages [Shaw96] [Garlan00]. However, the area must still be considered immature. On the other hand, recent work on reuse across pre-planned *product lines* [Jazayeri00] seems to have been successful.

Recent advances in middleware [Emmerich00] (with COM- or CORBA-style wrapping, or via Web servers) have made *black-box* reuse increasingly more attractive. This, again, has made COTS a popular target for component-based development. But again, but COTS-based reuse has proved very, very hard [Garlan95] [Carney97] [Boehm99] [Morisio00]. This is because,, as for classic reuse above, there is no standardised development process for COTS, no shared architectural modelling base, and no applicable cost/risk models [Boehm00]. The extra *glueware* to make individual COTS components play together, may be considerable. Further, we have no control over COTS risk factors, such as incompatible and unexpected "upgrades", vendor delays, or inadequate documentation.

Lastly, in all this, we have to consider normal software configuration and maintenance aspects [Lientz81] [Nosek90].

Own work on component-based development:
NTNU/SINTEF work on component-based development started in the REBOOT/SER EU projects in 1990-95 [Karlsson95] and in an accompanying national strategic technology project (STP) on software reuse. In these two EU projects, 13 empirical studies were performed [Paci97]. See the NTNU publication list for more references, and e.g. [Conradi97] and [Larsen98] on versioning and [Conradi96a] on feedback in reuse processes.

NTNU also participated in the RENAISSANCE EU project in 1996-98 on reengineering legacy software. This project developed a tailorable method, where software reuse was a key element in cost/risk decisions, and where UML was used to describe system architectures [Warren98].

There are also the industrial NFR-projects MAGMA in 1997-99 [Dahle96] and DAIM in 2000-02 [Dahle99], involving a future NTNU colleague, Tor Stålhane (an ex-SINTEF researcher), and many NTNU students supervised by Reidar Conradi. The MAGMA project has developed a standard application architecture and guidelines for business domain models, collected in a method book (see under `www.ikt-norge.no`).

Work at University of Oslo includes work done at Det norske Veritas (DnV) on versioning of component-based development [Arneberg98], on architecture recovery [Bratthall99], and many empirical studies on software evolution/maintenance [Bennett99] and on cost/risk estimation models [Jørgensen95a] [Jørgensen95c] [Jørgensen97] [Jørgensen99].

See also the publication lists for both partners (enclosed).

Research trends for component-based development:
- development models for COTS and reuse development.
- cost/risk estimation models for COTS and reuse development.
- architectural modelling of COTS and reuse software.
- studies of RUP-based development vs. traditional development.
- actual evolution and maintainability of e.g. OO frameworks over time.
- studies of actual evolution and maintainability of web-based solutions.

5.3 Software Process Improvement

Software process improvement (SPI) has been put on the industrial and research agenda for the last 10 years, with many national and international projects and initiatives established – cf. lists in Section 9 and the *Capability Maturity Model (CMM)* effort from Software Engineering Institute (SEI) [Paulk95]. The underlying hypothesis is that *the quality of a product reflects the process by which it was developed.* Since the current software process often is chaotic, a more "structured" process is thought to lead to "better", i.e. more predictable products with respect to end-user quality, cost, or delivery time.

Horror stories about the *software crisis* are exemplified by e.g. the Standish report of 1995, claiming that 31% of all custom-tailored software development projects in the US are stopped before completion ("dead before delivery”), with annual losses of 81 bill. $ or 1% of GNP (cited in the PITAC report, see www.hppc.org).

Many SPI method frameworks have been proposed. Many of these have borrowed concepts from the Total Quality Management (TQM) method [Deming84] [Aune95], originally intended for large manufacturing companies. TQM emphasises long-term management commitment, involvement of all employees, a pervasive focus on end-user quality, and organisational learning (see also below for the latter).

Some SPI frameworks, like CMM, SPICE [Dorling93] and BOOTSTRAP [Haase94], require that a company undergoes a “staged” improvement, typically in five levels, accompanied by regular process assessments to ensure “correct” level number. Each stage may take 2-3 years. ISO-9001 [Ince94] has only one level, and requires that a “certified” quality system (mostly in the form of documented process models) is put in place. All these frameworks have a top-down and management-heavy character.

In contrast, there is the Quality Improvement Paradigm (QIP), developed at the NASA Software Engineering Laboratory (NASA-SEL) with heavy involvement from University of Maryland. QIP, and the associated Experience Factory (EF) [Basili94b] and Goal-Question-Metric (GQM) [Basili94a] methods, has more a bottom-up and developer-oriented focus, and emphasises organisational learning. The EF is an “improvement organisation” within the organisation, although it may be implemented distributedly.

The reason for mentioning SPI in the INCO proposal is that many attempts to introduce incremental or component-based technologies are often part of a company SPI program. Furthermore, the problems with introducing a new technology and in demonstrating its effects are shared with SPI. For instance, there may be no clear baseline to compare with, results may come late or are unconvincing, the current product delivery suddenly gets priority and all “experimentation” is postponed or forgotten, management is semi-committed, or too dependent on local champions. In addition comes the usual organisational and technological turbulence in a highly evolving business place.

Related to SPI is active use of *experience bases* (cf. EF above) to promote *organisational learning* [Nonaka95] [Senge95]. That is, experiences should be systematically collected, refined and generalised, e.g. into guidelines for new work practices, and injected back into the organisation. However, the challenge is to have the new *externalised* models to become *internalised* (ingrained) into the workplace. That is, how to avoid that an experience base turns into an “information graveyard”. However, new Web-technologies have a potential for supporting light and incremental realisation and introduction of such experience bases.

Own work on software process improvement:
Both INCO partners are or have been involved in the industrial, NFR-supported SPIQ [Ulsund96] and PROFIT [Ulsund99] projects in, respectively, 1997-99 and 2000-02. In both SPIQ and PROFIT, about 10 companies have been or are participating – and with SINTEF, NTNU and UiO as research partners. In SPIQ, a down-scaled and pragmatical SPI framework was defined as part of the SPIQ Method Handbook (in Norwegian) [Dybå00]. The Handbook attempts to combine TQM ideas with QIP/EF/GQM. Over 20 small SPI efforts were performed in the participating companies during 3.5 years. Two of them were on incremental development. One was in component-based development, but was stopped by the actual company before getting started, due to local personnel shortage.

Also as part of the SPIQ project, experience bases in four companies were launched, and two have proven successful [Conradi00]. The critical success factors were found to be organisational stability, cultural changes, business value, and an incremental approach [Conradi+00b].

Research trends for software process improvement (excerpt):
- Scaling down and operationalising SPI method frameworks for SMEs.
- Incorporating novel approaches, such as incremental/component-based development, in SPI efforts.
- Studying and managing success factors for SPI/experience bases.

- How to develop successful experience bases as part of learning organisations.

5.4 Research Methods for Incremental and Component-Based Development

The two most relevant research methods for studies on incremental and component-based development are variants of observation and experimentation, see [Cook79], [Brooks80], [Lee89], [Basili86], [Yin94] and [Wohlin00]. Indeed, a new research field, *Empirical Software Engineering,* is emerging. This is especially due to the work of Victor Basili in Maryland and Hans-Dieter Rombach in Kaiserslautern, both with applied research institutes affiliated, and a long shared history for using the NASA-SEL as a "software laboratory".

Observation of real software development work has the advantage of being realistic regarding scale and environment, but there are difficulties due to lack of control of variables, i.e. cause-effect analyses are hard. A spectrum of empirical techniques are available, e.g. formal experiments, case studies, retrospective analysis, even literature studies etc. [Zelkowitz98].

Experiments may have control of variables, but have shortcomings with respect to realism, e.g. unrealistic small software development tasks. We therefore need a combination: To establish reasonable, validated results, we should both carry out research in a controlled environment (as realistically as possible) and in a realistic environment (as controlled as possible). For projects like INCO, it is advisable to combine observation of real projects (often case studies) with "laboratory" experiments (often on own students). A typical research process will be:

1. Formulate research hypotheses and questions based on interviews with software professionals and previous research results.
2. Carry out experiments testing the hypotheses.
3. Carry out observational studies of several industrial projects (introducing measurement and check-points in the development process if necessary) to investigate the validity of the results from the experiments.
4. Discuss the findings with software professionals.
5. Carry out new experiments.
6. Etc.

We have to be very careful in describing the context of the studies, ensuring proper extrapolating of the results only to contexts similar to the contexts studied.

Own work on relevant research methods:
We have carried out a number of observational and experimental studies on software development where results on research method have been a major component (see references in Sections 5.1-5.3).

UiO has published some papers on research methods [Arisholm99a] [Anda00] and [Jørgensen00], and been responsible for several seminars on research methods. In addition, UiO has implemented a measurement program in a large Norwegian software company to assess the impact of incremental development on productivity, time-to-market and delivery precision.

UiO has been working closely with prof. Welland and prof. Atkinson at Glasgow for many years on empirical studies of database schema evolution and the application of persistent language technology. Recently, UiO has also been working with Dr. Lionel Briand at Carleton University, Ottawa on empirical evaluation of metrics for object-oriented systems.

NTNU has for five years used own students in running experiments in a software quality course [Jaccheri00]. It has run many case studies in co-operation with industry, see e.g. the one project evolution [Nguyen97] and on inspections in Ericsson [Conradi99] as part of the SPIQ project. There is also ongoing, joint work with University of Maryland on how to represent repeatable experiments, and in running repeatable experiments on own students [Conradi+00a]. Much empirical work, especially around testing and SPI, has also been done at SINTEF by future NTNU-colleague Tor Stålhane, see [Haugrud95] [Staalhane97] and the publication list.

Research trends for relevant research methods:
There is no clear trend in how software development is studied, and we need several approaches. But we need to understand better their appropriateness for the actual work.

- How to describe and control "context variables"?
- How to combine quantitative and quantitative techniques?
- How should experiences and repeatable experiments best be represented and be available to others (e.g. in experience bases)?

6 Rationale and Approach

6.1 General Focus and Position

The implicit goal of INCO is increased software quality, both on products and processes. The research focus of the project is to apply more rigorous (i.e. scientific) methods to study the effect of incremental and component-based software development.

It has recently been shown that published papers in computer science and software engineering rely on empirical studies to a much less degree than in other disciplines. [Tichy96] have found that only 15% of the research papers in some journals contain any practical validation. Recent studies in 1998 by [Tichy98] and [Zelkowitz98] corroborate these findings. That is, we rely too much on intuition and self-esteem, not scientific probing of stated hypothesis. The same is valid in industry: few companies have any organised recording and synthesis of experiences gained in order to achieve "organisational learning".

For instance, after almost 35 years of object-orientation, there is hardly any empirical paper to assess its advantages and disadvantages in a "controlled" development and maintenance situation over 4–5 years. Recently, UML-related methods, of incremental and component-based software development are among the "in-methods", but strict validation is again lacking.

However, the lack of reproducible and stable "experiments" (projects) is a fundamental problem for software engineering. Thus, TQM principles [Deming84] of Statistical Process Control (SPC) may be applied to manufacturing with stable production lines, but not to software or design work. That is, software development is characterised by unique and highly evolvable projects – with little repetition and stability. We therefore also need to refine our research methods, e.g. in how to define effective and perhaps "soft" metrics and to manage non-repetitive projects. Here, methods from statistics, data mining and social sciences (e.g. [Levin98]) may be applied.

Thus, a focus on empirical evaluation of technology for incremental and component-based software development is of mutual benefit for academia and industry.

6.2 Our Method platform

Both the Oslo and Trondheim research groups have emphasised empirical studies in their former work on incremental and component-based development and on software engineering in general, cf. the SPIQ Method Handbook. We have also been working on guidelines for empirical studies [Arisholm99a] – see also other references to own work in Section 5. In INCO we will keep up our focus on performing solid empirical studies – including formal (controlled) experiments, case studies, interviews and observations (think-aloud protocols, etc.) and "software archaeology".

Among the other relevant work in this area are:

- Methods for empirical studies in software engineering, as in [Basili86] and [Zelkowitz98].

- The database of experimental methods and experiments, found at Univ. Maryland and Univ. Kaiserslautern, see http://www.cs.umd.edu/projects/SoftEng/* and the affiliated Fraunhofer institute of Empirical Software Engineering (IESE), see http://www.iese.fhg.de and http://wwwagse.informatik.uni-kl.de/ISERN/isern.html.
- A textbook in Experimental Software Engineering, edited by prof. Claes Wohlin at Univ. Lund. See http://www.tts.lth.se [Wohlin00].
- Summaries found at European Software Institute (ESI) in Bilbao, e.g. at http://www.esi.es/VASIE.
- Textbooks on how to perform case studies [Lee89] [Yin94], and apply statistics [Miller98].

6.3 Collaboration between Academia and Industry

INCO will be conducted in close co-operation between academia and industry to give a strong focus on practical application of the methods and empirical feedback. Much of the academic work will be done as part of PhD degrees, where each candidate typically will work with one or two specific "case'" companies. The candidates will gather valuable practical experience and empirical data to test the research hypotheses of their theses in a sound way. The companies will achieve better understanding and improvement of their own software development processes, and hence increased competitiveness.

We will also continue our tradition in using MSc students to work together with researchers and PhD students in such empirical work. In addition, each partner has 3-4 courses where more controlled experiments have been carried out for more than 5 years (see T4.3).

INCO intends to create and utilise a synergy with related projects. In addition to working together with software producers on concrete cases, technology transfer to Norwegian organisations will be ensured through "normal" means, as explained in Section 4.

7 Workplan

The project consists of two research groups at NTNU and UiO that constitute a virtual centre for experimental software engineering. Three PhD students, two advisors, ca. 10 MSc students and part-time guest researchers and a shared post-doc (not placed) will have their daily work.

We foresee two one-month visits per year by international researchers, and that the PhD students spend 6–8 months abroad during their study.

At any time we will conduct ca. 5 empirical studies towards our industrial contacts, including formal experiments, case studies / pilot projects, and interviews. The themes (see below) will be research-oriented, thus supplementing the more pragmatic pilots in PROGIT-projects.

There will be joint project meetings every 3rd month. We will also arrange a new national seminar series on INCO themes every six months, co-ordinated with e.g. the ones by the PROGIT-projects PROFIT, DAIM, and similar.

The research will be made available in PhD and MSc theses, reports, internationally published articles, seminar presentations, and in revised university courses (including NAWUS and EEU-courses). A web-node will be present the activities and results of the project.

7.1 General phasing and work structure

The workplan consists of four major phases:

Phase 1 (mainly month 1–6, 2001):

Definition of hypotheses and planning of empirical studies in companies, including experiment design, literature review/identification of related work.

Phase 2 (mainly month 7, 2001 to month 9, 2002):

Collection of measurements, data analysis, preliminary results, technical reports and workshop papers

Phase 3 (mainly month 10, 2002 to month 12, 2003):

Revision of models and hypotheses, further measurements/validation/results, workshop and conference papers, produce material for industrial and academic courses.

Phase 4 (mainly month 1–12, 2004):

Consolidation of results: final INCO report, conference and journal papers, presentation of final results, final material for industrial and academic course.

In practice, these phases will be partly overlapping in time because the PhD students may start at different times, the empirical studies will have varying length, etc.

The workplan consists of four workpackages, WP1-WP4, each composed of tasks named Tn.m:

- **WP1: Project Management** (UiO coord.)
- **WP2: Revised Methods for Incremental and Component-Based Development** (NTNU coord.)
- **WP3: Empirical Studies** (UiO coord.)
- **WP4: Dissemination and contact** (NTNU coord.)

7.2 WP1: Project Management (UiO coord.)

WP1 covers project coordination and reporting:

The groups at Trondheim and Oslo have both demonstrated ability to run international and national research projects (see Section 8 and the CVs in Enclosure 2). They have also demonstrated ability to collaborate, cf. the SPIQ and PROFIT projects, and in many less formal contexts. The project manager prof. Dag Sjøberg and the NTNU contact person, prof. Reidar Conradi, have both been managers of former successful research projects (see CVs).

7.2.1 T1.1 Project co-ordination

There will be monthly status meetings among the project coordinators, four annual common internal review/planning meetings.

Internal deliverables: Internal reports, work notes, drafts of technical reports and articles, presentations.

7.2.2 T1.2 Project reporting

This will be done formally towards NFR each year, and informally towards Norwegian IT-industry (see also WP3).

Internal/external deliverables: Status reports, news sheets – mainly on the Web (T4.4).

7.3 WP2: Revised Methods for Incremental and Component-Based Development (NTNU co-ord.) (for result R1)

"Change is constant, everything else is uncertain", Disraeli (after memory).

This work package will revise and supplement methods for incremental and component-based development, and their interaction. As for other INCO work, we will have a preliminary method revision ready after 0.5 year, and two major revisions after respectively 2 and 4 years. This workpackage will have a strong interplay with WP3 on empirical studies, and partly with WP4 on dissemination.

7.3.1 T2.1 Revised Methods for Incremental Development (UiO)

The revised methods from this task will be validated in T3.2.

Incremental software development needs to be explored regarding:

- When should a project use incremental development, and when are other development methods better?
- What is an adequate process of determining the increments to be delivered?
- Which techniques, processes and tools should be used when applying incremental development?
- How should an organisation apply previous experience to improve the incremental development process?

The three main activities in INCO are:

- **Management of projects based on incremental development.** These activities will focus on how to decide on an optimal software development method (e.g. should incremental development be used) and how to plan software projects based on incremental development. Important issues will be: Risk analysis, cost estimation, project planning.
- **Implementation and use of incremental development in software projects.** Different processes, techniques and tools support incremental development to different degrees. This activity aims at testing and improving the usefulness of existing processes, techniques and tools for incremental development, and to develop and test new ones.
- **Improvement of the incremental development process** An incremental development process opens for learning from each increment, i.e. more frequent feedback compared to a traditional waterfall-process. We will try to instrument the incremental development process with better feedback mechanisms and learning opportunities compared to what is usual today. In addition, we will try to measure the impact of training and education on the quality of the processes and the products.

For all these activities, we will have to consider interplay with component-based methods (T2.2), and in how to manage requirement engineering. The work will further have a focus around UML and in applying and refining RUP.

Internal deliverables: Work notes, preliminary reports and papers etc.
External deliverables: Published articles, MSc and PhD theses, final technical reports etc.
Relevant research challenges:

- How be to combine previous research, the research carried out in INCO and "best practice" in industry to improve the methods for incremental software development.

7.3.2 T2.2 Revised Methods for Component-based Development (NTNU)

The revised models from this task will be validated in T3.3.

Reuse is maybe the most potent of all software engineering technologies, not at least regarding. time-to-market. However, both classical reuse of source code and component-based reuse based on wrapped (COTS) components have faced unexpected and persistent obstacles, especially regarding risks
In INCO, we will have a focus on COTS, using UML and middleware integration techniques à là COM/CORBA. For instance, DnV is already having a component-based framework called Brix for engineering software, using COM.

In addition, we will combine the REBOOT and RENAISSANCE work on component classification and especially on cost/risk models and revised development processes, and some of the MAGMA work on a standard system architecture. We also expect to exchange calibrated cost/risk models for COTS development with University of Maryland working with NASA (Vic Basili), and with University of Rönneby (Jan Bosch).

In addition, we need to work on:

- More advanced risk/cost models for COTS-based development, especially for estimation.

- Architectural modelling for documentation, analysis and configuration, preferably using UML, but we foresee experimenting with research languages like ADL or DARWIN (but no own contribution here).
- The interaction with incremental development (T2.1), and also with requirement engineering and project planning. Here, it is imperative to limit the scope and clarify responsibilities between the methods.
- If we have the capacity: study the impact of product lines in connection with this (depends on industrial opportunities)?

Deliverables: as for T2.1.

Relevant research challenges:
- What is the relation between reuse and incremental development (T2.1)?
- What is an operational cost/benefit metrics for reuse?
- What is the relation between reuse and OO/software architecture?
- How to predict architectural discrepancies and the middleware needed to bridge these?

7.4 WP3: Empirical Studies (UiO coord.) (for result R3, partly R2)

This work package consists of four tasks: T3.1 to define a set of suitable research methods, T3.2 and T3.3 for field studies of the two previous WP2 methods, and T3.4 for refining experiences in an experience base to promote organisational learning. The work will run in two major phases of 1.5 year each: one to apply the initial methods after 0.5 year, and one to use the refined methods after 2 years. Thus, there is a strong interaction with WP2, and some with WP3 on dissemination.

We will conduct *industrial trials* (mostly as case studies), as well as more *formal experiments* using own students. We cannot know in advance *which* companies and/or study profiles that will emerge in 1-2 years from now. However, we feel confident that the relevance of these topics and our previous working relations with industry will secure a good work mode with industry. So, to avoid loosing time, we will have a continuous dialog with relevant companies from the start of the project about possible study themes – finding compromises between hypotheses coming out of WP2/T3.1 and industrial needs and availability.
As mentioned, there is also half a dozen university courses, being candidates for more formal experiments, but we have to plan and make the exercise material almost from scratch.

7.4.1 T3.1 Research Methods for Incremental and Component-Based Development

The methods from this task will mainly be used in T3.2 and T3.3.

This task seeks to define and improve how we systematically can study the impacts of our revised development methods on software processes and products. Planned activities are:

- **Guidelines for context of results.** The empirical research methods need guidelines and/or frameworks on how to describe the context of the results. Currently, it is not unusual that development studies only provide a few lines about the organisation and type of project. This lead to the unfortunate situation that the empirical results with data from one organisation cannot be used by another organisation without a high degree of uncertainty regarding the similarity of the project and the organisation. A better description of the context will improve the reuse of the results from empirical software studies.
- **Guidelines on how to report data quality of research studies**. We need better guidelines on how to report different levels of data quality and methods on how to deal with this difference. This is particularly important since the most commonly used statistical techniques within empirical software engineering studies assume the same level of data quality. For example, [Jørgensen95b] showed that the poor quality of questionnaire based data led to very biased answers regarding software maintenance information.

Even if this task is an auxiliary task, we expect also to do research contributions here.

Internal and external deliverables: Several versions of such methods – as work notes, technical reports, articles, etc.

Some research challenges:
- How to accumulate and represent current knowledge? How to investigate possible cause-effect analyses, considering context variables? How can we know which research method is "best" for a certain technology and study context?
- How different are the research methods (qualitative/quantitative) for formal experiments vs. industrial studies?

7.4.2 T3.2 Empirical Studies of Incremental Software Development (UiO)

This task will try to validate the revised methods from T2.1 and to use research methods from T3.1, giving results to T3.4.

Evolutionary, incremental development processes that integrate object-oriented methods and tools have recently become popular, e.g. Rational Unified Process (RUP), which is becoming a de-facto standard. Unfortunately, there are few scientific, empirical studies on when such a process is suitable, the consequences of adopting such a process, and how it should be adapted to a particular company. The interplay with requirement engineering (e.g. observing the actual change rate of requirements or other change requests) and with component-based approaches both add to the complexity of such studies.

Deliverables: Plans including metrics for studies, data from studies (raw and processed), technical reports, and published papers.

Some research challenges:
- What are the consequences of introducing evolutionary incremental processes on project organisation, customer relation, contracts, effort estimation and risk analysis? (To our knowledge, there are no empirical studies on the consequences of implementing such processes that could result in guidelines for companies that wish to change their process.)
- Does evolutionary, incremental development result in customer satisfaction? Does lead to faster development of "right product"? How important is to focus on uses and to prioritise changed requirements from the users? Does early implementation of such changed requirements affect the stability of the architecture, which in turn may result in more expensive maintenance?
- Does the use of evolutionary incremental processes imply particular requirements on the robustness of the architecture?
- What is the appropriate size of the increments? Do small increments result in code that later is removed, which in turn affects the productivity and product quality? Or does shorter deadlines lead to increased motivation and productivity among system developers?
- How should requirement analysis, design, coding and testing be prioritised? Is it more efficient to emphasise testing early to reduce error correction, or does such an approach result in unnecessary quality assurance of code that eventually will be removed?
- How should configuration management tools and processes be adapted to the continuous changes that occur in evolutionary incremental processes?
- How should tools, e.g. those supporting reverse and forward engineering, be integrated in the process to help ensure up-to-date documentation and consistent code?

7.4.3 T3.3 Empirical Studies of Component-Based Development (NTNU, UiO)

This task will try to validate the revised methods from T2.2 and to use research methods from T3.1, giving results to T3.4.

Such studies are hard to carry out from a university perspective, since we often will need a time horizon of 2-3 years. On the other hand, COTS-based development (and couplings to incremental development in T2.1

and T3.3) has a very large industrial attention. We will (as mentioned) run a continuous dialog with industry on possible themes and project opportunities. A handful companies have already expressed interest, both for COTS-based development and for product lines/families (partly PROFIT and DAIM participants).

As for T3.2, the interaction with incremental development, and with requirement engineering, is very challenging to test in practice. We will also try out modest component repositories, using Web technologies. At the same time we will seek suitable student projects involving software architectures, reuse, and evolution.

Also, as mentioned, we will seek to carry out two phases of studies, both for students and towards industry. We also envisage to exchange student experiments with University of Maryland, Lund and Rönneby, and to provide such experiments to our NAWUS partners in Norway.

Deliverables: similar as for T3.2.

Some research challenges:
- Assessing the actual risks and cost/benefits of component-based development, using COTS?
- How tight is the connection between requirement engineering and COTS selection?
- Can existing architectural languages be used in industry?
- What is an operational structure of a component-based repository, e.g. Web-organisation, keyword-based?

7.4.4 T3.4 Collecting Experiences and Learning from Incremental and Component-based Development Projects (for result R2)

Knowledge management and organisational learning by experience bases is gradually getting into use. This task tries to make the total experiences from INCO available both to industrial developers, researchers and students. The focus should be on learning. This means that the material in the experience bases must be used actively in own and industrial courses, in industrial seminars, ingrained in new work practices etc.

We envisage that the collected experiences from the empirical studies in T3.2 and T3.3 will reside in (modest) company-internal experience bases. A subset of the same information, representing aggregated and generalised data across companies, courses and studies, will also reside in the INCO Web site (T4.4.), accessible by both UiO and NTNU, and partly by any Web user. This means that strict rules for confidentiality must be observed. Some of the company experience bases will be part of SPI efforts in these companies, and currently we have about five companies in the PROFIT project working on experience bases.

Regardless of where the knowledge is stored (INCO Web site, company-internal experience bases), this task will define and promote simple guidelines for how this kind of knowledge should come into practical use. Ideas from the Experience Factory and similar approaches will be used as a starting point.

In contrast, WP4 (and T4.4 on the INCO Web site) has a much more modest scope, focusing on information flow *from* the project. Here, the information flow ultimately will go *inside* a *learning organisation*.

Deliverables: Filled-up experience bases with experimental data and aggregated models, some presentation/educational material, guidelines to promote a learning organisation.

Some research challenges:
- How can we promote learning in the real process – a prerequisite to learn from experience?
- How can we best focus on learning opportunities?
- How much organisational stability is needed to achieve learning?
- How can information overload (“information graveyards”) be avoided, e.g. by Web-technology?

7.5 WP4: Dissemination and Contact (NTNU coord.) (for result R4, partly R2)

The material to be dealt with in WP4 mainly comes from WP2 and WP3, typically as published articles, technical reports, internal research notes, and processed experimental data. This workpackage may do some additional packing and classification of such material, and also produce new presentation material or educational material.All this material will be stored in the project Web site (T4.4), covering all relevant INCO documents. WP4 covers activities on conferences/travels (T4.1), courses (T4.2), a new semi-annual seminar series on INCO-related themes (T4.3), and a Web site (T4.4).

7.5.1 T4.1 International contact: presentations, visits and travels

Our research papers will regularly be presented at scientific seminars, workshops, specialist meetings etc.

Both research groups have a wide collegial network around the research themes of INCO (see Section 9). We expect that the INCO results will be used to consolidate and expand our research network in related areas, and thus improve our standing internationally. The planned guest researchers / postdocs may come from such collegial environments.

Internal deliverables: Regular travel reports.
External deliverables: Articles and reports from WP2 and WP3.

7.5.2 T4.2 New national, semi-annual seminar series on incremental/component-based development and empirical software engineering

Such a semi-annual (twice per year) workshop series will be sought established with other researchers within the IKT-2010 program, and possibly with foreign research groups. It may partly be co-located with the planned, annual IKT-2010 event.

Internal deliverables: Meeting summaries, agendas.
External deliverables: Semi-annual seminar presentations.

7.5.3 T4.3 New and revised educational material and courses

The experiences and new knowledge gained through INCO will be disseminated in several (upgraded) courses in software engineering and remote education given at NTNU and UiO (half a dozen relevant courses) and towards industry (although assuming extra resources from 3rd parties). We can mention courses in software engineering, software quality, object-oriented methods, and software architecture (new one at NTNU). NTNU has also special project courses in the 4th year of the engineering study, covering half a semester, and these project courses can be used for more special studies. There is also a NAWUS "Norgesnett for software engineering education", with UiO and NTNU as the prime partners (see Section 9). We also have an ambition to make a new MSc/PhD-level **course in empirical software engineering**, with emphasis on research methods – possibly in co-operation with Univ. Lund and Maryland.

In general: software engineering education needs interaction with software industry. Software engineering topics, in general, and software quality and improvement ones, in particular, cannot be thought in the same way as programming languages or logic. When teaching programming languages, the model can be: teach a construct; show examples; assign exercises; provide solutions, and iterate. When teaching software engineering, examples and exercises cannot be invented by teachers but must be elicited from real world projects. Moreover, software engineering educational projects can be regarded as experimental laboratories for assessing software technologies in semi-real contexts. The advantages of using student projects instead than real industrial settings are that we have more control on independent variables, such as resource allocation, tool and process choice etc.

External deliverables: New courses and presentation material, on a regular basis.

7.5.4 T4.4 Project Web Site

Establish and maintain a Web-based, project- and document archive – the INCO Web site. This will store all project material, both for internal and external use. Actual filling-up is mostly done elsewhere, such as WP2-3 and T4.1-4.3. As mentioned in T3.4, we have excerpts from the empirical studies available, with aggregated models and summaries. Further, we want to build up some bibliographies and Web pages with references to other research groups. That is, the Web site is considered an important result of the project, intended to be used by the partners also afterwards. Technically, we will reuse some of the Web structures set up in the NAWUS project, and in the work on Experimental packages done at University of Maryland.

Internal deliverables: Web archive templates and guidelines, overall Web structure.
External deliverables: Actual Web contents (mostly from other workpackages and tasks).

8 Partner Description

The partners involved in INCO are:

- **Ifi (UiO),** Oslo, and
- **IDI (NTNU),** Trondheim.

The prime contractor of INCO will be Ifi (prof. Dag Sjøberg), with prof. Reidar Conradi as the NTNU contact.

The Oslo group has been working of persistent languages, schema evolution in databases, software maintenance, incremental development, estimation, software quality, and software process improvement. See Section 8.1.

The NTNU group has previously been working on software reuse, configuration management, software engineering databases, process modelling, software process improvement and software quality, and software engineering education. See Section 8.2.

These two research groups involved have had informal and formal project co-operation for over seven years. We can mention the national SPIQ project in 1997-99 [Ulsund96] [Conradi96b] and PROFIT in 2000-02 [Ulsund99], both on software process improvement, and on the
NAWUS [Sjøberg98] project in 1998-99 on “Norgesnett” for software engineering education. Both groups are membes of the International Software Engineering Research Network (ISERN).

As mentioned, INCO will involve several companies where most of the concrete empirical studies will take place. A list of co-operating companies, many of them SMEs, is listed in Section 9. We also co-operate closely with the IT interest-organisations ITF, IKT-Norge, NFKL, and DND.

See Enclosure 2 for Curriculum Vitaes and publication lists.

8.1 Ifi-UiO: A closer description

The Department of Informatics, University of Oslo, has had a group in software engineering and databases for several years. Recently, the two applicants have formed a new research group, “Industrial Systems Development”, which focuses on empirical studies of system development in industry. The main competence of the new group is on (empirical studies of) methods, techniques and tools for software maintenance and evolution, software processes and effort estimation.

The PhD students in the group are partly funded internally and partly funded externally (the NFR projects SPIQ and GOODS). They already focus on topics that are related to INCO and will thus form a stimulating research environment for new PhD students in INCO. Both the applicants and their students have close contacts with, and conduct studies in collaboration with, companies such as Genera, Numerica-Taskon, Ericsson, Storebrand, Telenor, Computas and Viewpoint. See also Section 9.

The group hosted NWPER'20000 (The Ninth Nordic Workshop on Programming Environment Research at Lillehammer in 28-30 May 2000 and will host POS9 (The Ninth International Workshop on Persistent Object Systems) in 6–8 September 2000.

8.2 IDI-NTNU: A closer description

IDI has had a software engineering group for almost 15 years.

The NTNF-supported EPOS project (1986–91) had software process as one of the foci the last 3 years of the project. IDI has further participated in many international projects (REBOOT, SER, ASSET, PROTEUS, partly PERFECT, RENAISSANCE, PROMOTER, PROMOTER-2, CMEX, AMEX) where process and quality have been a relevant component. Good synergy is planned with PROFIT (SPIQ successor) and related projects (DAIM, CoDeVer).

Both internally and externally financed PhD students already work on topics related to the proposed INCO. We are performing studies in co-operation with companies such as Computas, Ericsson, ISI, Storebrand, Telenor Novit and Telenor 4tel. These studies have provided valuable feedback and experiences. Further opportunities for case studies are being investigated, also in other companies than the above.

The group has acquired much competence and has an internationally acknowledged position in software engineering. It has close contacts with Norwegian software producers and end-users, and with international colleagues. It has arranged international workshops on software configuration management in 1991 (SCM'3) and in process modelling in 1992 (EWSPT'92), and the group leader has been program chair for similar workshops in 1997 (SCM'7) and 2000 (EWSPT'2000). Thus, we feel that the proposed INCO project in 2001–2004 is an excellent way to carry this competence further with the desired result.

9 Associated Activities and Projects

In addition to this INCO proposal and associated, university-funded research activities, we will try to co-ordinate INCO with some of the following ongoing and planned initiatives:

- PROFIT project, under the industrial PROGIT program at NFR.
  Co-ordinator: Tor Ulsund, Bravida. 6–8 MNOK per year for 2000–2002.
- Other NFR industrial projects, like DAIM and partly CoDeVer, both co-ordinated by SINTEF.
- Other Norwegian universities and colleges, e.g. through the existing NAWUS Norges-nettverk for education in software engineering [Sjøberg98].
- New international projects, especially within ESPRIT and ESSI. (Almost 10 EU projects have been conducted by NTNU in related areas since 1990).
- Internal SPI projects in software–producing organisations, such as Computas, DnV, EDB/Telenor 4tel, Ericsson, Genera, IBM, ISI, Kongsberg-Ericsson, Kongsberg-Spacetec, Mogul (Numerica-Taskon), Telenor Novit, SDS, Siemens, Statkonsult, Storebrand, Telenor Geomatikk etc. Links to all of these companies are already established, and the companies welcome further co-operation.
- Other international software engineering institutions and activities: FC-MD in Maryland, IESE in Kaiserslautern, ISERN network node in Kaiserslautern, …
- National projects and networks (e.g. SPINs) abroad, especially in Sweden (NUTEK projects in Lund and Linkøping). Furthermore, several countries have recently initiated large SPI projects, e.g. Canada, Australia, Denmark, Sweden, Germany, and partly USA (PITAC report)
- General academic exchange, by international co-operation and exchange, e.g. with the Universities of Maryland, Aachen, Kaiserslautern, Lund, Manchester, Grenoble, Lancaster, Pisa, Milano, Torino, Aarhus, Berlin, and Glasgow. In addition, we have ongoing contacts with international companies such as Q-Labs, Bull, Siemens, Matra, Cap Gemini, ICL, and Intecs.

Note that INCO is able to deliver relevant results, regardless of the mentioned industrial projects (e.g. PROFIT), but the total effect will be greater if we can involve more actors of this kind.

10 References

[Arisholm98] Arisholm, E., Benestad H.C., Skandsen J., Fredhall, H.:
"Incorporating Rapid UserInterface Prototyping in Object-Oriented Analysis and Design with Genova", Proceedings of NWPER'98 Nordic Workshop on Programming Environment Research, Ronneby, Sweden, August 1998 pp 155-161.

[Anda00] Bente Anda and Magne Jørgensen:
"Understanding Use Case Models", Proc. Workshop No. 5: "Beg, Borrow or Steal" at ICSE'2000, 5 June 2000, Limerick, Ireland.

[Arisholm99a] Erik Arisholm, Bente Anda, Magne Jørgensen, and Dag Sjøberg:
"Guidelines on Conducting Software Process Improvement Studies in Industry", Proc. 22nd IRIS Conference (Information Systems Research Seminar In Scandinavia), Keuruu, Finland, 7-10 August 1999, pp. 87-102.

[Arisholm99b] Erik Arisholm, Jon Skandsen, Knut Sagli, and Dag Sjøberg:
"Improving an Evolutionary Development Process - A Case Study", EuroSPI99, Pori, Finland, 25-27 October 1999

[Arisholm00a] Erik Arisholm and Dag Sjøberg:
"Towards a Framework for Assessing Changeability Decay", Journal of Systems and Software, September 2000, Vol. 53, pp. 3-14.

[Arisholm00b] Erik Arisholm, Dag I.K. Sjøberg and Magne Jørgensen:
"Assessing the Changeability of two Object-Oriented Design Alternatives – a Controlled Experiment", Submitted to Journal of Empirical Software Engineering, 2000

[Arneberg98] Gunnar Arneberg, Knut Sagli, and Dag Sjøberg:
"Configuration Management in Component-Based Development Projects", In John Grundy, editor, Workshop on Component-Based Information Systems Engineering (CBISE'98), pages 94 - 100, Pisa, Italy, 1998.

[Aune95] Asbjørn Aune:
"Kvalitetsstyrte bedrifter", Ad Notam, Gyldendal, Oslo, 1995.

[Basili86] Victor R. Basili, Richard W. Selby, and David H. Hutchens:
"Experimentation in Software Engineering", IEEE Trans. on Software Engineering, SE-12(7):733–743, July 1986.

[Basili91] Victor R. Basili and Hans D. Rombach:
"Support for Comprehensive Reuse" (on TAME project at NASA), Software Engineering Journal (special issue on Software process and its support), 6(5):303–316, September 1991.

[Basili94a] Victor R. Basili, Gianluigi Caldiera, and Hans-Dieter Rombach:
"The Goal Question Metric Paradigm", In [Marciniak94], pages 528–532, 1994.

[Basili94b] Victor R. Basili, Gianluigi Caldiera, and Hans-Dieter Rombach:
"The Experience Factory", In [Marciniak94], pages 469–476, 1994.

[Bennett99] Keith Bennett, E. Burd, C. Kemerer, M. M. Lehman, M. Lee, R. Madachy, C. Mair, Dag Sjøberg, and S. Slaughter:
"Empirical Studies of Evolving Systems", Empirical Software Engineering, 4 (4) pp. 370-380, December 1999.

[Boehm88] Barry W. Boehm:
"A Spiral Model of Software Development and Enhancement", IEEE Computer, May 1988, pages 61—72.

[Boehm99] Barry Boehm and Chris Abts:
"COTS Integration: Plug and Pray?", IEEE Computer, January 1999, pp. 135–138.

[Boehm00] Barry Boehm, Chris Abts, and Elizabeth Bailey Clark:
"Empirical Observations on COTS Software Integration Effort Based on
the Initial COCOTS Calibration Database?",
In John C. Dean, Patricia Oberndorf, and Mark Vigder, editors: "Proc. of Second Workshop on Commercial Off-The-Shelf Software", p. 99-104. (In conjunction with ICSE'2000, Limerick, Ireland).

[Bratthall99] Lars Bratthall and Per Runeson:
"Architectural Design Recovery of a Family of Embedded Software Systems - An Experience Report",
In Proceedings of TC2 First Working IFIP Conference on Software Architecture (WICSA1),
Kluwer Acacdemic Publishers, Norwell, Massachusetts, USA. pp 3-14. 1999.

[Brooks80] R. E. Brooks:
"Studying programmer behaviour experimentally: the problems of proper methodology",
Human Aspects of Computing, 23, 207-213. 1980.

[Carney97] D.J. Carney and Patricia A. Oberndorf:
"The Commandments of COTS: Still in Search of the Promised Land",
Crosstalk, May 1997, Vol.10, No.5, pp 25-30.

[Conradi96a] Reidar Conradi:
"Process Support for Reuse", In Barry Boehm, Marc Kellner, Dewayne Perry (Eds.): ``Proc. 10th International Software Process Workshop (ISPW`10), 17–19 June 1996, Ventron, France",
p. 57-61, IEEE Press, IEEE CS Order Number PR07725.

[Conradi96b] Reidar Conradi.
"SPIQ: A Revised Agenda for Software Process Support".
In Carlo Montangero (Ed.): "Proc. 4th European Workshop on Software Process Technology (EWSPT'96)",
pages 36–41, Nancy, France, 9–11 Oct. 1996. Springer Verlag LNCS 1149.

[Conradi97] Reidar Conradi and Bernhard Westfechtel:
"Version Models for Software Configuration Management",
ACM Computing Surveys, July 1997, Vol. 30, No. 2, pages 232-282.

[Conradi99] Reidar Conradi, Amarjit Singh Marjara, and Børge Skåtevik:
"An Empirical Study of Inspection and Testing Data at Ericsson",
In Markku Oivo and Pasi Kuvaja: "Proc. Int'l Conf. on Product Focused Software Process Improvement (PROFES'99)", Oulu, Finland, 22-24 June 1999, p. 263–284. VTT symposium 195, VTT, Espoo, Finland,
666 p. URL: http://www.inf.vtt.fi/pdf).

[Conradi00] Reidar Conradi and Torgeir Dingsøyr:
"Software Experience Bases: A Consolidated Evaluation and Status Report".
In Frank Bomarius, editor, Proc. Second Int'l Conf. on Product Focused Software Process Improvement (PROFES'2000), Oulu, Finland, 20-22 June 2000. (14 p., forthcoming as a Springer LNCS).

[Conradi+00a] Reidar Conradi (scribe), Victor R. Basili, Jeff Carver, Forrest Shull, and Guilherme H. Travassos:
"Pragmatical Documents Standard for an Experience Library: Roles, Documents, Contents and Structure",
draft Technical Report (V1.54), Univ. Maryland, April 15, 2000, 50 p.

[Conradi+00b] Reidar Conradi, Mikael Lindvall, and Carolyn Seaman:
"Success Factors for Software Experience Bases: What We Need to Learn from Other Disciplines", 6 p.,
Accepted for ICSE'2000 Workshop on "Beg, Borrow or Steal: Using Multidisciplinary Approaches in Empirical Software Engineering Research", Limerick, Ireland, 5 June 2000.

[Cook79] D. Cook and D. T. Campbell:
"Quasi-Experimentation, Design & Analysis Issues for Field Settings",
Houghton Mifflin, USA. 1979. ISBN 0-395-30790-2.

[Dahle96] Hans Petter Dahle at al.:
"MAGMA: Component-Driven Software Development",
November 1996. Proposal to from PROFF to NFR for user-driven project.

[Dahle99] Hans Petter Dahle et al.:
"DAIM: Distribuert Arkitektur, Internett og Multimedia", November 1999.
Proposal from IKT-Norge to NFR's PROGIT program for user-driven project.

[Deming84] W. Edwards Deming:
"Out of Crisis", MIT Center for Advanced Engineering Study, Cambridge, MA, USA, 1984.

[Dorling93] Alex Dorling:
"SPICE: Software Process Improvement and Capability Determination
(ISO 15054)", Software Quality Journal, Vol. 2, 1993, pages 209—224.

[Dybå00] Tore Dybå *et al.:*
"SPIQ metodebok for prosessforbedring" (V3) – in Norwegian, January 2000.
Technical report (forthcoming on a Norwegian Publisher), see also http://www.geomatikk.no/spiq.

[Emmerich00] Wolfgang Emmerich:
Software Engineering and Middleware: A Roadmap.
In Anthony Finkelstein, editor, *The Future of Software Engineering*, ACM Press Order No. 592000-1, pages 119-129.
(Supplementory Proc. at 22th Int'l Conf. on Software Engineering (ICSE'200), Limerick, Ireland, 4-11 June 2000.)

[Gamma95] Erich Gamma, Ralph Johnson, Richard Helm, and John Vlissides:
"Design Patterns – Elements of Reusable Object-Oriented Software", Addison–Wesley. 416 p., 1995.

[Garlan95] David Garlan, R. Allen, and J. Ockerbloom:
"Why reuse is so hard", IEEE Software 12(6):17–28, Nov. 1995.

[Garlan00] David Garlan:
"Software Architecture: A Roadmap". In Anthony Finkelstein, editor, The Future of Software Engineering,
ACM Press Order No. 592000-1, pages 93-101. (Supplementory Proc. at 22th Int'l Conf. on Software
Engineering (ICSE'2000), Limerick, Ireland, 4-11 June 2000.)

[Gilb88] Tom Gilb: 'Principles of Software Engineering Management'. Addison-Wesley 1988. ISBN 0-201-19246-2

[Haase94] Volkmar Haase, Richard Messnarz, Günther Koch, Hans J. Kugler, and Paul Decrinis:
"BOOTSTRAP: Fine-Tuning Process Assessment",IEEE Software, July 1994, pages 25—35.

[Haugrud95] Bjørn Haugrud, Tor Stålhane, and Inger Venås:
"The Effect of Using Object-Oriented Methods in Specification and Design", AQUiS'98 –3rd International
Conference on Achieving Quality in Software, Venezia, Italy March 30 – April 2, 1998.

[Hofmeister00] Christine Hofmeister, Robert L. Nord, and Dilip Soni:
"Applied Software Architecture", Addison Wesley Object Technology Series, 2000, ISBN 0-201-32571-3, 397 p.

[Ince94] Darrel Ince:
"ISO 9001 and Software Quality Assurance", International Software Quality Assurance Series,
Quality Forum, McGraw Hill,London, 1994, 139 p.

[Jaccheri00] M. Letizia Jaccheri, Reidar Conradi, Bård H. Dyrnes:
"Software Process Technology and Software Organisations",
In Reidar Conradi (Ed.): *Proc. 7th European Software Process Workshop on Software Process*

Technology (EWSPT'7), Kaprun near Salzburg, Austria, 21-25 Feb. 2000.
Springer Verlag LNCS 1780, p. 96-108.

[Jazayeri00] Mehdi Jazayeri, Alexander Ran, and Frank van der Linden:
"Software Architecture for Product Families", Addison-Wesley, 2000, ISBN-0-201-69967-2.

[Jørgensen95a] Magne Jørgensen:
"An empirical study of software maintenance tasks",
Software Maintenance: Research and Practice, 7:27–48, 1995.

[Jørgensen95b] Magne Jørgensen:
"The quality of questionnaire based software maintenance studies",
ACSM SIGSOFT - Software Engineering Notes, 20(1):71–73, 1995.

[Jørgensen95c] Magne Jørgensen:
"Experience with the accuracy of software maintenance task effort prediction models",
IEEE Transactions on Software Engineering, 21(8):674–681, 1995.

[Jørgensen97] Magne Jørgensen:
"An empirical evaluation of the MkII FPA estimation model",
In Stein Krogdahl et al., editors, Proc. Norwegian Informatics Conference (NIK'97),
Voss, November 1997. Tapir forlag.

[Jørgensen99] Magne Jørgensen and S. Steinholt Bygdås:
"An empirical study of the correlation between development efficiency and software development tools",
Telektronikk, vol. 95, no. 1,1999, pp. 54-62.

[Jørgensen00] Magne Jørgensen, Dag Sjøberg, and Geir Kirkebøen:
"Human judgement in effort estimation of software projects",
In Janice Singer et al., editors, Beg, Borrow, or Steal Multi-Disciplinary Workshop at the International
Conference on Software Engineering (ICSE'2000), Limerick, Ireland, 5 June 2000. (forthcoming).

[Karlsson95] Even-André Karlsson (Ed.), editor:
"Software Reuse: A Holistic Approach" (The REBOOT Methodology Handbook),
Wiley Series in Software Based Systems. John Wiley. 510 p., 1995.
ISBN 0-471-95819-0, REBOOT report no. 8218.

[Larsen98] Jens-Otto Larsen and Helge M. Roald:
"Introducing ClearCase as a Process Improvement Experiment",
In Boris Magnusson, editor, Software Configuration Management – ECOOP'98 SCM-8 Symposium,
pages 1–12, Brussels, Belgium, 20-21 July 1998. Springer Verlag LNCS 1439.

[Lee89] Allen S. Lee:
"A Scientific Methodology for MIS Case Studies", MIS Quarterly, Vol. 13, No. 1, p. 33-50, 1989.

[Levin98] Morten Levin and Davydd J. Greenwood:
"Introduction to Action Research – Social Research for Social Change", Sage, 1998.

[Lientz81] B. P. Lientz and E. B. Swanson:
"Problems in application software maintenance", Comm. of the ACM, 24(11):763–769, November 1981.

[Lim94] Wayne C. Lim:
"Effects of Reuse and Quality, Productivity and Economics" (from Hewlett-Packard),
IEEE Software (special issue on software reuse), pages 23–30, September 1994.
(Best paper in IEEE Software in 1994).

[Marciniak94] John J. Marciniak, editor:
"Encyclopedia of Software Engineering – 2 Volume Set", John Wiley and Sons, 1994.

[Miller98] J. Miller, J. Daly, M. Wood, M. Roper, and A. Brooks.
"Statistical power and its subcomponents – missing and misunderstood concepts in empirical software engineering research",
Technical report, University of Strathclyde, Glasgow, UK, 1996. Report no. ISERN-96-12.

[Morisio99] Maurizio Morisio, Michel Ezran, and Colin Tully
"Introducing Reuse in Companies: A Survey of European Experiences",
Proc. Symposium on Software Reuse (SSR'99), LA, May 1999, 7 p.

[Morisio00] Maurizio Morisio, Carolyn Seaman, Amy Parra, Victor Basili, Steve Kraft, and Steve Condon:
"Investigating and Improving a COTS-Based Software Development Process",
Proc. 22nd International Conference on Software Engineering (ICSE'2000), pages 31-40,
ACM Order No. 592000.

[Nguyen97] Minh Ngoc Nguyen, Alf Inge Wang, and Reidar Conradi:
"Total Software Process Model Evolution in EPOS", In Richard N. Taylor and Alfonso Fuggetta (Eds.), editor,
Proc. 19th Int'l Conf. on Software Engineering (ICSE'97), pages 390–399.
ACM[IEEE-CS Press, N.Y., May 1997.

[Nonaka95] Ikujiro Nonaka and Hirotaka Takeuchi:
"The Knowledge-Creating Company", Oxford University Press, 1995.

[Nosek90] J. T. Nosek and P. Palvia:
"Software maintenance management: changes in the last decade",
Journal of Software Maintenance, 2(3):157–174, 1990.

[Paci97] M. Paci and Svein Hallsteinsen:
"Experiences in Software Evolution and Reuse", ESPRIT Research reports, Volume 1, Springer Verlag, 1997.
Case studies from 15 European Software Reuse Projects, done in
ESPRIT Project 9809, Software Evolution and Reuse (SER): Bull SA, ATC,
EP-Frameworks, Q-Labs, Sema-Group sae, Sintef, TTB, and TXT Ingegneria.

[Paulk95] Marc C. Paulk, Charles V. Weber, Bill Curtis, and Mary B. Chrissis:
"The Capability Maturity Model for Software: Guidelines for Improving the Software Process",
SEI Series in Software Engineering. 640 p. Addison–Wesley, 1995.

[Poulin95] Jeffrey S. Poulin:
"Populating Software Repositories: Incentives and Domain-Specific Software",
Journal of Systems and Software, 1995:30, p. 187-199.

[Reifer97] Donald J. Reifer:
"Practical Software Reuse: Strategies for Introducing Reuse Concepts in Your Organization",
John Wiley, 1997, 392 p. ISBN 0-471-57853-3.
See Deependra Moitra: "Reuse Strategies for Success" (book review),
IEEE Software, March/April 1999, p. 95-96.

[Royce 70] W. W. Royce:
"Managing the Development of Large Software Systems: Concept and Technique",
Proceedings of WesCon, Aug. 1970, p. 1—9.
(Reprinted in Proc. Int'l Conf. Software Eng., IEEE Computer Society Press, 1987, pp. 328—338.)

[Senge95] Peter M. Senge:
"The Fifth Discipline: The Art and Practice of the Learning Organization", Currency/Doubleday, 1995.

[Shaw96] Mary Shaw and David Garlan:
"Software Architecture – Perspectives of An Emerging Discipline", Prentice Hall, 1996.

[Sjøberg98] Dag Sjøberg and Reidar Conradi:
"NAWUS: NAsjonal Web-database for Undervisning i Systemutvikling",

August 1998. Forslag til Arbeidsgruppen for digitale læremidler, 6 p.

[Staalhane97] Tor Stålhane, Kjell Arnesen, and Per Christian Borgersen:
“In Search of the Customer's Quality View",
Journal of Systems and Software, vol. 38, 1997, pp. 85-93.

[Tichy96] Walter F. Tichy, Paul Lukowicz, Lutz Prechelt, and Ernst A. Heinz"
"Experimental Evaluation in Computer Science: A Quantitative Study",
Journal of Systems and Software, 28(1):9–18, January 1995.

[Tichy98] Walter F. Tichy:
"Should Computer Scientists Experiment More", IEEE Computer, 31(5):32–40, May 1998.

[Ulsund96] Tor Ulsund et al.:
"SPIQ: Software Process Improvement for better Quality", November 1996.
Proposal to NFR for user-driven project.

[Ulsund99] Tor Ulsund et al.:
"PROFIT: Process Improvement for the IT Industry", November 1999.
Proposal to NFRs PROGIT program for user-driven project.

[Warren98] Ian Warren:
“The Renaissance of Legacy Systems: Method Support for Software System Evolution",
Springer Practitioner Series, 1999, 182 p.
(The Method Book from the RENAISSANCE ESPRIT Project in 1996-98, project no. 22010;
NTNU contributed on the architectural modelling and method chapters).

[Wohlin00] Claes Wohlin, Per Runeson, M. Hoest, M. C. Ohlsson, Bjoern Regnell, and A.Wesslen:
“Experimentation in software engineering: An introduction”, Kluwer Academic Publishers. 2000. ISBN 0-7923-8682-5.

[Yin94] Robert K. Yin.:
"Case Study Research Design and Methods", Sage Publications, John Wiley and Sons, London, UK, 1994.

[Zamperoni95] Zamperoni, A., Gerritsen, B. and Bril, B.:
Evolutionary Software Development: An experience Report on Technical and Strategic Requirements.
Technical Report TR-95-25, Leiden University, The Netherlands, 1995.

[Zelkowitz98] Marvin V. Zelkowitz and Dolores R. Wallace:
"Experimental Models for Validating Technology", IEEE Computer, 31(5):23–31, May 1998.

11 Enclosure 2 – Curriculum Vitaes and Publication Lists:

11.1 CV for Dag Sjøberg, UiO

Personalia: Dag Sjøberg, born: 24.01.61

Home address: Niels Henrik Abelsv. 19, 0851 Oslo, Phone: 22-568484.

Education:

- 1990-1993: PhD in Computing Science, University of Glasgow.
  Thesis: "Thesaurus-Based Methodologies and Tools for Maintaining Persistent Application Systems",
  Supervisor: Malcolm P. Atkinson.
- 1986-1987: MSc in Computer Science, University of Oslo.
  Thesis: "Database Concepts discussed in an Object-Oriented Perspective",
  Supervisor: Kristen Nygaard.
- H1984: "Pedagogisk seminar" (half a year full time course in teaching).

Present job and work address: Professor, Department of Informatics, University of Oslo, P.O. Box 1080 Blindern, N-0316 Oslo, Norway, Phone.: 22-852742, E-mail: Dag.Sjoberg@ifi.uio.no.

Language skills: Fluent in Norwegian and English, some German.

Scientific profile:
Software Engineering, Web Engineering, Process Improvement, Evolutionary Development, System Evolution, System measurements/ Empirical Studies, Software Process Improvement, Object-Oriented development, Programming Environments, OODB and Persistent Programming.

Jobs held:

- 1999- present, Professor, Department of Informatics, University of Oslo.
- 1995-1999: Associate Professor, Department of Informatics, University of Oslo.
- 1993-1995: Postdoctoral scholarship from the Research Council of Norway at Department of Informatics, University of Oslo.
- 1990-1993: PhD scholarship from the Research Council of Norway at Department of Computing Science, University of Glasgow.
- 1987-1989: Group leader at Statistics Norway.
- 1986-1988: Consultant at the National University Hospital of Norway.

Projects:

- Local project leader at UiO, National project (NFR) Software Process Improvement for better Quality (SPIQ) (1.1.1997-31.12.1999).
- Local project leader at UiO, Persistent Application Systems, Technologies, Environments and Languages (PASTEL) (Esprit Project 22552),
  Responsible for Research Theme Evaluation Technology and Experiments (1.1.1997-31.12.1999)
- National project (NFR) General Object Oriented Distributed Systems (GOODS), (1.1.1997-31.12.1999)
- Product Model for life-cycle information in the oil and gas industry (POSC/CAESAR) (1.1.1999-31.12.1999)
- National project NAWUS education network for System Development., w/ prof. Reidar Conradi, NTNU (1998-1999).

- National project (SND) Supervision Policies for Automation of Computer Environments (SPACE) (1996-1998).
- Research Council of Norway and British Council, Evaluation Methodology for Software Maintenance Technology, University of Oslo, Norwegian Telecom, University of Glasgow, UK, University of Central Lancashire/University of Lancaster, UK (1995-1996).

11.2 CV for Magne Jørgensen, UiO

Personalia: Magne Jørgensen, born 1964

Home address: Statsråd Ihlensv. 14A, 2010 Strømmen, Phone: 22.311314.

Education:
- PhD thesis: "Empirical studies of software maintenance", University of Oslo Norway, supervisor Arne Maus, 1994.

Present job and work address: Associate Professor at Dept. of Informatics, University of Oslo, Email magnej@ifi.uio.no.

Language skills: Norwegian, English and German.

Scientific profile:
Empirical studies in software Engineering, and software process improvement.
More than 10 international, refereed publications.

Jobs held:
- 1989 - 1998 Research scientist and group leader at Telenor R&D.
- 1998 - 1999 Technical manager at Storebrand in software process improvement.
- 1995 - Assoc. prof. at University of Oslo

Projects:
- Norwegian national project SPIQ (Software Project Improvement for better Quality) (from 1997).
- National project NAWUS education network for System Development., w/ prof. Reidar Conradi, NTNU (1998-1999).

11.3 CV for Reidar Conradi, NTNU

PERSONALIA: REIDAR CONRADI, born 23 July 1946, Oslo. Norwegian

HOME ADDRESS: Skule Bårdsons gate 11, N-7052 Trondheim, Norway. Phone +47 73.935862.

EDUCATION: M.Sc. (1970), Ph.D. (1976) – both from NTH (now NTNU),Trondheim.
Courses in Didactics and Technical writing, 1972-1985.

PRESENT JOB and WORK ADDRESS:
Professor at Dept. of Computer Science and Informatics (IDI).
Norwegian University of Science and Technology (NTNU)
N-7491 Trondheim, Norway
Phone +47 73 593444, Fax +47 73 594466, Email conradi@idi.ntnu.no
Leads software engineering group of 10 people at IDI.

LANGUAGE SKILLS: Fluent in Norwegian and English, good in German, passable in French, some Italian from summer courses in Perugia 1992-94.

SCIENTIFIC PROFILE:
Programming languages, software engineering, database support for CASE, configuration management, software process modelling, software process improvement, software quality, object-orientation and reuse, architecture of distributed software systems.
150.000 lines of pograms written in Fortran, Algol, Simula, Mary, Bliss, Ada, Chill, C/C++, and Java.
Per April 2000: ca. 130 int'l published papers, 20 nat'l ones,
textbooks and 13 booklets, over 20 compendia, almost 100 technical reports.

JOBS HELD and STAYS ABROAD:
1972–75 Research scientist at SINTEF, Trondheim.
1975–79 Assistant professor at NTH, Trondheim.
1979–85 Associate professor at NTH.
1980–81 Visiting scientist at CMU, Pittsburgh.
(1981–83 Partly sick due to a back problem.)
1985- Full professor at NTH.
1985–87 Head of new Dept. of Computer Systems and Telematics (IDT), NTH.
1990 Guest researcher at U. of Pisa and at ENSIMAG, May-June 1990.
1997–98 Head of new Dept. of Computer and Information Science (IDI), NTNU.
1999-2000 Sabbatical at U. of Maryland and Politecnico di Milano.

PROFESSIONAL RESPONSIBILITIES:
1976- Member of IFIP WG2.4 on Programming Technology (prev. Systems Programming Languages).
1984–86 Work to initiate Norway's IT-program in 1987–90.
1985–99 Representative in the Norwegian Informatics Council.
1986 Organizer of International Workshop on Advanced Programming Environments, Trondheim, 16-18 June 1986.
1988–90 Program chair for 1st, 2nd, 3rd Norwegian Informatics Conf.
1990–93, 95-96 Chair for Norwegian Informatics Council.
1991 Conf. chair 3rd Int'l SCM Workshop, Trondheim,12-14 June 1991.
1991 Co-arranger of 1st European SW Process Workshop, Milan, 30-31 May 1991.
1992 General chair for 2nd European SW Process Workshop, Trondheim, 7–8 Sept. 1992.
1997 Program chair for SCM'6 in Boston, May 1997.
1997– Member of ISERN, research network on Software Experimentation, coord. by U. Kaiserslautern.
1998–99 Nat'l NAWUS educ. network for System Dev., w/ prof. Dag Sjøberg, Univ. Oslo.
1999 Member of NFR committee to plan new, strategic IT program.
2000 Program chair for EWSPT'7 in Salzburg, Feb. 2000.
PC member of many workshops/conferences, e.g. the series SCM, EWSPT, SSR, partly ICSE, SIGSOFT and ESEC – usually 4–5 memberships per year. Also in many univ. and nat'l committees.

PROJECTS etc.
1972–99 Textbooks and teaching. Tutored 180 M.Sc. and 12 Ph.D. stud. Responsible for 2 MSc courses and 2 PhD courses.
1972–78 MARY programming language and its operating system.
1980–81 Production Quality Compiler-Compiler (PQCC), and SPICE at CMU.
1983–86 FORTRAN VERIFIER for error diagnosis and documentation of F-77 programs, NTNF-supported.
1986–90 Project leader EPOS project with Veritas Research/Norsk Data, NTNF-supported.
1990–92 Bilateral ACM project NTH and LGI-ENSIMAG.
1991–94 Program leader of REBOOT National Strategic Technology Program in Norway.
1990–95 ESPRIT projects REBOOT, SER, PROTEUS, PROMOTER BRA and ASSET.
1996–99 ESPRIT projects PROMOTER2, CMEX, AMEX, and RENAISSANCE.

1996–99 Nat'l CAGIS basic R&D project on distributed information systems, coordinator.
1996–99 Nat'l SPIQ pre-project and main project on software process improvement, initiator.
1998–2002 Nat'l CSE-project on supercomputing, w/ OO modelling aspects, PhD advisor.
1999–2002 Nat'l KVALIS-project on quality of medical journal systems, co-worker.

CONTACTS:

Especially with U. Karlsruhe, RWTH Aachen, U. Manchester, U. Linkøping, U. Lund, U. Grenoble, U. Pisa, P. di Milano, CMU, U.Maryland, and BPU in Beijing through guest researchers and students.
Also good contact with int'l companies, e.g. Bull, Siemens, Cap, Qlabs, and Intecs.
Many Norwegian industrial contacts in SPIQ project.

11.4 CV for Tor Stålhane, IDI, NTNU

Name: Tor Stålhane.

Year of birth: 1944.

Nationality: Norwegian.

Positions:
Principal Research Scientist, SINTEF Telecom and Informatics.
Adjunct Professor of Computer Science at the Stavanger College since 1997.
Soon to become full professor at IDI, NTNU (Autumn 2000).

Education: M.Sc. Technical University of Norway,
Dept. of Electrical Engineering,
Group for Physical Electronics, 1969.

Ph.D. Technical University of Norway,
Dept. of Mathematical Sciences,
Group for Mathematical Statistics, 1988.

Basic training in Quality Audits, Norwegian Association for Quality (NFK), 1992.

Main fields of competence:

Software Process Improvement:
My work in this area includes participation in ESPRIT projects - PERFECT and Telmet plus improvement projects for several Norwegian companies, such as Norsonic, NFT-Ericsson, Kongsberg Aerospace and Alcatel Telecom. The work has been focused on software measurement. Its practical applications have been process description, definition of improvement experiments, data collection, statistical data analysis and process improvement proposals.

Software Safety and Reliability:
My work in this area includes work for several Scandinavian companies in the area of safety and reliability assessment and estimation. The systems have mainly been transport control systems for road, railway and air traffic. The work related to safety has included modeling, fault tree analysis, failure mode and effect analysis and HazOp analysis of software intensive systems. The work related to reliability has included testing, test plans and reliability estimation based on test results and system analysis.

Research interests:
Improvement of the software development process,
Software Reliability Analysis and Estimation,

Risk Assessments for Software Intensive Systems,
Testing, Debugging, Validation,
Software Certification,
Quality Assurance in software development and maintenance,
Investment problems in software development and use.

Professional memberships:

Member of IEEE, Reliability;
Member of Society of Reliability Engineers (SRE);
Member of Norwegian Statistical Society;
Member of the editorial board for "The Journal of Software Testing, Verification and Reliability";
Member of the board for ITUF - The Information Technology Development Forum 1990 - 1996;
Referee for IEEE Transactions of Software Engineering;
Referee for the Scandinavia Journal of Statistics;
Referee for Engineering and Physical Sciences Research Council, UK;
Member of the organizing committee and technical committee for the EuroSPI conference since 1997.

Experience:

2000-2002: Software process improvement for Norwegian industry - PROFIT, PROcess improvement For IT industry.
2000: Software process improvement for DnB – major Norwegian bank.
1999-2000: Use of case-based reasoning (CBR) in process improvement.
1999: Use of UML for describing and analysing a safety critical system.
1999: Guidelines for Validation of Functional Safety for Systems containing COTS component for the Nordtest organization (project leader).
1999-2000: Procedure for developing IEC 61508 conform component for Siemens Oil and Gas Division.
1998-2000: QIS – Quality in Scandinavia: ESPINODE for project experiments in Norway and Sweden (project leader).
1998: KPB-92 – Technical advice in law suite.
1998: Guidelines for Validation of Functional Safety according to IEC 61508 for the Nordtest organization.
1998: Basic Test methods for Safety Control Systems for the Nordtest organization.
1997-1999: Safety assessment for the SLS-system, a GPS based landing system for NAVIA AVIATION.
1997-1999: Software process improvement for Norwegian industry - SPIQ.
1997-1998: ESSI software process improvement for MaXware - QARI.
1997-1998: ESSI software process improvement for NAVIA AVIATION - AMPIC.
1996-1998: Process improvement at Nera Tele Scada (Bergen) - TELMET.
1997: Methods and standards for safety assessment of railway signaling systems - the ACRUDA project.
1997: Safety assessment for an ILS system.
1997: Process improvement through root cause analysis for Kongsberg Aerospace.
1996: Collection and statistical analysis of software metrics for ESA.
1996: Safety assessment of changes in train radio system for NSB.
1995-1996: SISU - Handbook for integrated software development, including process improvement.
1995-1996: Development of Quality Assurance Manual and course material for subcontractor software development for Securitech.
1995-1996: Process improvement for Norsonic.

1994-1996: PROFF - a research project focusing on certifying software quality according to customer needs and preferences.
1995: Process improvement for NFT-Ericsson.
1995: Safety assessment for an ILS system.

11.5 CV for M. Letizia Jaccheri, NTNU

Personalia: Maria Letizia Jaccheri, born 24 February 1965, Pisa, Italian.

Home address: Osloveien 6, N-7018 Trondheim, Norway. Phone +47 73.537573.

Education:
- Master thesis: "Software Development Environment: A Conceptual Model Representing Software Systems" University of Pisa, Italy, Department of Computer Science, February 1988, supervisor Prof. Vincenzo Ambriola.
- PhD thesis: "Software Process Modeling and Evolution", Politecnico di Torino, Italy, supervisor Prof. Silvano Gai, January 1994.

Present job and work address:
Associate Professor at Dept. of Computer and Information Science (IDI),
Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway,
Phone +47 73 593469, Fax +47 73 594466, Email letizia@idi.ntnu.no.

Language skills: Italian mother tongue, Norwegian and English proficiency.

Scientific profile:
Software Engineering and particularly in software process modelling, and Software engineering education.
37 international refereed publications.

Jobs held:
- Last position (from January 1993 to June 1997):
  Assistent Professor at Politecnico di Torino, Italy (tenure from 1.1.1996).
- From 1988 to 1989: programmer and designer at Tecsiel S.P.A., Pisa Italy.

Projects:
- EPOS (from 1989 til 1992): Software Process Modeling and Execution Environment based rule-base techniques. At NTH, Trondheim, Prof. Reidar Conradi.
- E3 (from 1992 to 1998): Software Process Modeling Environment based on object-oriented modeling, at Politecnico di Torino, Italy
- Italian National Project ex 40% (from 1994 to 1998) Methodologies for Verification and Validation of Software Product and
- Process Promoter ESPRIT BRA 7082 (from 1992 to 1995) Promoter ESPRIT Working Group "European Research Working Group on Software Process Technology" (from 1996 to 2000)
- ACTS AC003 VITAL Validation of Integrated Telecommunication Architectures for the Long term (1995-1998). The project had the goal of validating the TINA (Telecommunication Information Network Architecture) architecture.
- Norwegian national project SPIQ (Software Project Improvement for better Quality) (from 1997).

11.6 Relevant publications for 1993–2000 for ISU, Ifi, UiO

1993:

1. Magne Jørgensen:
"Software maintainability -- on purpose or accidentally?",

Technical Journal of Norwegian Telecom, No. 1, p. 150-156, 1993.

2. Sigrid Steinholt Bygdås and Magne Jørgensen:
”Software development methods and life cycle models”,
Technical Journal of Norwegian Telecom, No. 2/3, p. 44-51, 1993.

3. Magne Jørgensen and Arne Maus:
”A case study of software maintenance tasks”,
Norwegian Informatics Conference (NIK'93), p. 101-112, Nov. 1993.

4. Malcolm P. Atkinson, Dag Sjøberg and Ron Morrison:
”Managing Change in Persistent Object Systems”,
In: Proceedings of the First JSSST International Symposium on Object Technologies for Advanced Software (Kanazawa, Japan, 4th-6th November 1993), pp. 315-338, Springer LNCS 742, Springer-Verlag, 1993.

5. Dag I.K. Sjøberg:
”Quantifying Schema Evolution”,
Information and Software Technology, Vol. 35, No. 1, pp. 35–44, January 1993.

6. Dag Sjøberg:
”Thesaurus-Based Methodologies and Tools for Maintaining Persistent Application Systems”,
PhD Thesis, University of Glasgow, July 1993.

1994:

7. Magne Jørgensen:
”A comparative study of software maintenance effort prediction models”,
5th Australasian Conference on Information Systems, Melbourne, Australia, 1994.

8. Magne Jørgensen:
”Empirical studies of software maintenance”,
Thesis for the dr.scient.-degree, Research Report No. 188, University of Oslo, Department of Informatics, Oct. 1994.

9. Magne Jørgensen:
”Prediction of software maintenance effort”,
Norwegian Informatics Conference (NIK'94), p. 81-92, Nov. 1994.

10. Dag I.K. Sjøberg:
”Managing Change in Information Systems: Technological Challenges”,
In Proc. 17th Information systems Research seminar In Scandinavia (IRIS'17), Syte, Finland, 6th–9th August, 1994, pp. 838–851, University of Oulu, 1994.

11. Dag Sjøberg, Malcolm P. Atkinson, Joao Lopes and Phil Trinder:
”Building an Integrated Persistent Application”.
In: Proceedings of the Fourth International, Workshop on Database Programming Languages (Manhattan, New York City, USA, 30th August – 1st September 1993), pp. 359–375, Springer-Verlag and British Computer Society, 1994.

1995:

12. Magne Jørgensen:
”An Empirical Study of Software Maintenance Tasks”,
Journal of Software Maintenance, Vol. 7, No. 1, pp. 27–48, January–February 1995.

13. Magne Jørgensen:
”Experience with the Accuracy of Software Maintenance Task Effort Prediction Models”,
IEEE Transactions on Software Engineering, August 1995, pp. 674-681.

14. Magne Jørgensen:
”Empirical evaluation of CASE Tool efficiency”,
Sixth Int. Conf. on applications of Software Measurement, Orlando, p. 207-230, 1995.

15. Magne Jørgensen, Sigrid Steinholt Bygdås and T. Lunde:
”Empirical evaluation of CASE tools”,
Norwegian Informatics Conference (NIK'95), p. 71-82, Nov. 1995.

16. Dag I.K. Sjøberg:
”Towards a Methodology for Evaluating Software Maintenance Technology”,
In Proc 18th Information systems Research seminar In Scandinavia (IRIS'18),
Aarhus, Denmark, 11–13th August, 1995, pp. 653–663.

17. Dag Sjøberg, Quintin Cutts, Ray Welland and Malcolm P. Atkinson:
”Analysing Persistent Language Applications”,
In: Sixth International Workshop on Persistent Object Systems (Tarascon, Provence, France, 5th–9th September, 1994), pp. 235-255, Workshops in Computing, Springer-Verlag and British Computer Society, 1995.

18. Dag Sjøberg, Malcolm P. Atkinson and Ray Welland:
”Thesaurus-Based Software Environments”,
Workshop on the Intersec tion Between Databases and Software Engineering (WIBDSE'94), Sorrento, Italy, 16th-17th May, 1994, pp. 41-46, IEEE Computer Society Press, 1995.

1996:
19. Dag Sjøberg, Ray Welland, Malcolm P. Atkinson, Magne Jørgensen, Martinussen and Arne Maus:
”Evaluating Software Maintenance Technology”,
In: Sixth International Workshop on Persistent Object Systems. Also in: Norwegian Conference in Informatics, Alta, Norway, 18-20 November, 1996, pp. 49-61, TAPIR, 1996.

20. Dag Sjøberg, Ray Welland, Malcolm P. Atkinson, Paul Philbrow, Cathy Waite and Stewart Macneill:
”The Persistent Workshop - a Programming Environment for Napier88”,
In: 7th Nordic Workshop on Programming Environment Research (NWPER'96), Aalborg, Denmark, 29–31 May, 1996, pp. 37-52.

1997:
21. Magne Jørgensen:
”Measurement of software quality”
First Int. Conf. Software Quality Engineering, Udine, Italy, pp. 257-266, 1997.

22. Magne Jørgensen:
”An empirical evaluation of the MkII FPA estimation model”,
Norwegian Informatics Conference (NIK'97), Voss, Nov. 1997, pp. 7-18.

23. Dag Sjøberg, Ray Welland, Malcolm P. Atkinson:
”Software Constraints for Large Application Systems”,
The Computer Journal, Vol. 40, No. 10, pp. 598–616, 1997.

24. Dag Sjøberg, Ray Welland, Malcolm P. Atkinson, Philbrow and Waite:

”Exploiting Persistence in Build Management”,
Software - Practice and Experience, Vol. 27, No. 4, pp. 447–480, April 1997.

25. Dag Sjøberg, Ray Welland, Malcolm P. Atkinson, Philbrow, Waite and Macneill:
”The Persistent Workshop - a Programming Environment for Napier88”, Vol. 4, pp. 123–149, 1997.

1998:
26. Gunnar Arneberg, Knut Sagli and Dag Sjøberg:
”Configuration Management in Component-Based Development Projects”,
Workshop on Component-Based Information Systems Engineering (CBISE'98), John Grundy (editor), 8–9 June 1998, Pisa, Italy, pp. 94–-100.

27. Stein Grimstad, Dag Sjøberg, Malcolm P. Atkinson and Ray Welland:
”Evaluating Usability Aspects of PJama based on Source Code Measurements”,
In Third International Workshop on Persistence and Java (PJW3), Tiburon, California, 1–3 September, 1998, pp. 307-321.

28. Magne Jørgensen, Dag Sjøberg and Reidar Conradi:
”Reuse of software development experience at Telenor Telecom Software”,
European Software Process Improvement Conference (EuroSPI'98),
Gothenburg, Sweden, pp. 10.19–10.31, 16–18 November 1998.
Also presented at NIK'98, Kristiansand, Nov. 1999.

29. Dag Sjøberg:
”Software Constraints in Integrated Data Environments”,
In: Fifth Integrated Data Environments Australia Workshop (IDEA5),
Fremantle, Australia, 7–10 February, 1998, pp. 65-68.

1999:
30. Magne Jørgensen:
”Software quality according to measurement theory”,
Telektronikk, Vol. 95, No. 1/99, pp. 12-17.

31. Magne Jørgensen, Dag Sjøberg, Reidar Conradi:
”Reuse of software development experience – a case study”,
Telektronikk, Vol. 95, 1/99, p. 48-53 (special issue on Software Quality in Telecommunications),
Also as NTNU/SU-report 12/99.

32. Magne Jørgensen and Sigrid Steinholt Bygdås:
”An empirical study of the correlation between development efficiency and software development tools”,
Telektronikk, Vol. 95, No. 1/999, pp. 54-62.

33. Lars Bratthall and Per Runeson:
”Architectural Design Recovery of a Family of Embedded Software Systems – An Experience Report”,
Proc. TC2 First Working IFIP Conference on Software Architecture (WICSA'1), Feb. 1999, Kluwer Acacdemic Publishers, Norwell, Massachusetts, USA, pp. 3-14, 1999.

34. Erik Arisholm and Dag Sjøberg:
”Empirical Assessment of Changeability Decay in Object-Oriented Software”,
Proc. ICSE'99 Workshop on Empirical Studies of Software Development and Evolution, pp. 62-69, Los Angeles, CA, 18 May 1999.

35. Erik Arisholm, Bente Anda, Magne Jørgensen, and Dag Sjøberg:

”Guidelines on Conducting Software Process Improvement Studies in Industry”,
Proc. 22nd IRIS Conference (Information Systems Research Seminar In Scandinavia), Keuruu, Finland, 7-10 August 1999, pp. 87-102.

36. Lars Bratthall and Per Runeson:
”A Taxonomy of Orthogonal Properties of Software Architecture”,
Second Nordic Workshop on Software Architecture, Aug. 1999. (no pages numbers in proc.).

37. Erik Arisholm, Jon Skandsen, Knut Sagli, and Dag Sjøberg:
”Improving an Evolutionary Development Process - A Case Study”,
EuroSPI'99, Pori, Finland, 25-27 October 1999, pp. 9.40-9.50.

38. Amela Karahasanovic and Dag Sjøberg:
”Supporting Database Schema Evolution by Impact Analysis”,
Norsk informatikk-konferanse (NIK'99), Nov. 1999, Trondheim, pp. 303-314.

39. Magne Jørgensen:
”Software quality measurement”,
Advances in Engineering Software, Vol. 30, No. 12, December 1999, pp. 907-912.

40. K. Bennett, E. Burd, C. Kemerer, M. M. Lehman, M. Lee, R. Madachy, C. Mair, Dag Sjøberg, S. Slaughter:
”Empirical Studies of Evolving Systems”,
Empirical Software Engineering, 4 (4), pp. 370-380, December 1999.

41. Lionel Briand, Erik Arisholm, S. Counsell, Frank Houdek, P. Thevenod:
”Empirical Studies of Object-Oriented Artifacts, Methods, and Processes: State of the Art and Future Directions”,
Empirical Software Engineering 4 (4), pp. 387-404, December 1999 (also available as ISERN TR 99-12).

2000:

42. Lars Bratthall, Per Runeson, K. Adelsword, W. Ericsson:
”A Survey of Lead-Time Challenges in the Development and Evolution of Distributed Real-Time Systems”,
Accepted for Information and Software Technology, 2000.

43. Knut Kjetil Holgeid, John Krogstie and Dag Sjøberg:
”A Study of Development and Maintenance in Norway: Assessing the Efficiency of Information Systems Support Using Functional Maintenance”,
Information and Software Technology, To appear September 2000.

44. Magne Jørgensen, Dag Sjøberg and Geir Kirkebøen:
”The prediction ability of experienced software maintainers”,
4th Europan Conference on Software Maintenance and Reengineering, Jurgen Ebert and Chris Verhoef (editors), IEEE Computer Society, pp. 93-100, Zurich, Switzerland, Feb 29 - March 3, 2000.

45. Amela Karahasanovic:
”SEMT - A Tool for Finding Impacts of Schema Changes”,
Proc. Ninth Nordic Workshop on Programming and Software Development Environment Research (NWPER'2000), Lillehammer, Norway, 28-30 May, 2000.

46. Magne Jørgensen, Geir Kirkebøen, Dag Sjøberg, Bente Anda, Lars Bratthall:
”Human judgement in effort estimation of software projects”,

Beg, Borrow, or Steal Multi-Disciplinary Workshop at International Conference on Software Engineering (ICSE'2000), June 5, 2000, Limerick, Ireland.

47. Bente Anda, Magne Jørgensen:
”Understanding Use Case Models”,
Beg, Borrow, or Steal Multi-Disciplinary Workshop at International Conference on Software Engineering (ICSE'2000), June 5, 2000, Limerick, Ireland.

48. Lars Bratthall and Claes Wohlin:
“Understanding Some Software Quality Aspects from Architecture and Design Descriptions”,
Accepted for publication at International Workshop on Program Comprehension (IWPC'2000), June 2000.

49. Magne Jørgensen, Dag Sjøberg:
”Empirical studies on effort estimation in software development projects”,
Conf. of Information Resources Management Association (IRMA 2000), Alaska, May 2000.

50. Lars Bratthall, Enrico Johansson, Bjørn Regnell:
”Is a Design Rationale Vital when Predicting Change Impact? – A controlled Experiment on Software Architecture Evolution”,
Int'l. Conf. on Product Focused Software Process Improvement (PROFES'2000), 20-22 June 2000, Oulu, Finland.

51. Erik Arisholm and Dag Sjøberg:
”Towards a Framework for Assessing Changeability Decay”,
Journal of Systems and Software (to appear Sept. 2000).

52. K. A. Mughal, A. L. Opdahl, D. I. K. Sjøberg (eds.):
“Proceedings of NWPER'2000 - The Ninth Nordic Workshop on Programming and Software Development Environment Research, Lillehammer, Norway 28-30 May, 2000”.
Research Report 283, Department of Informatics, University of Oslo, ISBN 82-7368-231-5, ISSN 0806-3036, Research Report, Department of Information Science, University of Bergen.

53. Magne Jørgensen, Dag Sjøberg:
“The importance of NOT learning from experience”,
EuroSPI’2000, Copenhagen, Denmark (to appear)

11.7 Relevant publications for 1993–2000 for IDI, NTNU

1993:
1. Reidar Conradi, M. Letizia Jaccheri, Cristina Mazzi:
”Variability of Process Models and Processes, and the EPOS Solution”,
In Wilhelm Schäfer (Ed.): Proc. 8th International Software Process Workshop (ISPW'8), p. 43–46,
2–5 March 1993, Dagstuhl, FRG, IEEE-CS Press.
Also as EPOS TR 176, NTH, Trondheim, 27. Sept. 1992, 5 p.

2. Reidar Conradi, M. Letizia Jaccheri:
”Customization and Evolution of Process Models in EPOS”, p. 23–39.
In N. Prakash, C. Rolland, B. Pernici (Eds.): ”Proc. IFIP 8.1 Working Conference on Information System Development Process (ISDP'93), Como, Italia, 1–3 Sept. 1993”, North-Holland Publ., Amsterdam, 337 p.
Also as EPOS TR 174, NTH, Trondheim, 30 April 1993 (rev.), 12 p.

3. Reidar Conradi, Christer Fernstrøm, and Alfonso Fuggetta:
”A Conceptual Framework for Evolving Software Processes”,
ACM SIGSOFT Software Engineering Newsletter, Oct. 1993 (Vol. 18, No. 4), p. 26–34.

Also as EPOS TR 187, NTH, 9 Nov. 1992, 26 p., Trondheim.

4. M. Letizia Jaccheri, Reidar Conradi:
”Techniques for Process Model Evolution in EPOS”,
In special issue of IEEE TSE on Software Process Model Evolution, Dec. 1993, p. 1145-1156.
Politecnico di Torino and NTH in Trondheim, 5 May 1993, 18 p. Also as EPOS TR 188.

5. Minh Ngoc Nguyen, Reidar Conradi:
”Workspace Management: Supporting Cooperative Work”,
Proc. International Conference for Young Computer Scientists (ICYCS-93),
15–17 July 1993, Beijing (forthcoming), 4 p.
Also as EPOS TR 186, NTH, Trondheim, 25 April 1993.

6. Tor Stålhane and K.J. Wedde:
”The Quest for Reliability - A Case Study”,
AQUiS'93 - 2nd International Conference on Achieving Quality in Software, Venice, Italy, October 18-20, 1993.

7. M. van der Meulen (NL), Tor Stålhane (N) and R. Code (UK):
”Programmable Electronic System Analysis Technique in Safety Critical Applications”,
Safecomp'93, Poznan-Kiekrz, Poland, 27-29 October 1993.

8. Tor Stålhane (N), M. van der Meulen (NL), R. Cole (UK):
”Reliability assessment for programmable electronic systems using subjective and objective categorical data”,
PCPI'93, Dusseldorf, Germany, March 29-31, 1993

9. Tor Stålhane, Alejandro de Mora, and Alfredo Cosculluela:
”Certification of Reusable Components”,
Second International Workshop on Software Reusability, Pisa, Italy, March 24 - 26, 1993.

1994:

10. Reidar Conradi et al.:
”Object-Oriented and Cooperative Process Modelling in EPOS”,
In PROMOTER book: Anthony Finkelstein, Jeff Kramer and Bashar A. Nuseibeh (Eds.):
”Software Process Modelling and Technology”, 1994, p. 33–70. Advanced Software Development Series, Research Studies Press Ltd. (John Wiley), ISBN 0-86380-169-2, 362 p.

11. Minh Ngoc Nguyen, Reidar Conradi:
”SPELL: A Logic Programming Language for Process Modelling”, 10 p.
Proc. of Workshop on Logic Programming in Software Engineering, 18 June 1994, Santa Margherita Ligure, Genova, Italy (held after the ICLP conference). Also as EPOS TR 220.

12. Minh Ngoc Nguyen, Reidar Conradi:
”Meta-process in a Process Support Environment”, 12 p.
Proc. 6th Nordic Workshop on Programming Environment Research (NWPER'94), Lund, Sweden, 1–3 June 1994. Also as EPOS TR 221.

13. Minh Ngoc Nguyen, Reidar Conradi:
”The Software Meta-process: Taxonomy and Assessment”, p. 167–175,
Proc. 3rd Int'l Conference on Software Process (ICSP'3), IEEE-CS Press, 10-11 Oct. 1994 Washington. Also as EPOS TR 208.

14. Noureddine Belkhatir, Reidar Conradi:
”The Relationship between Software Processes and CSCW”,
Workshop on Links between CSCW and Software Process, October 22, 1994 (held before CSCW'94), Chapel Hill, North Carolina, USA, 2 p. (position paper).

15. Tor Stålhane, E. Ø. Øvstedal:
”Safety Assessment for a Positioning System”,
NSDCS'94, Lyngby, Denmark, August 24-26, 1994.

16. Tor Stålhane and K.J. Wedde:
”The Quest for Reliability - A Case Study”,
Journal of Systems and Software, vol. 26, no. 1, July 1994, pp. 69-76.

1995:
17. Reidar Conradi et al.:
”PSEE Architecture: EPOS process models and tools” (foils),
Alfonso Fuggetta (Ed.): Proc. Workshop on Process-centered Software Engineering Environment Architecture, 20–23 March 1995, Politecnico di Milano, 61 p. SU-report 3/95.

18. Reidar Conradi:
”Session on Change and Meta-process (introduction)”,
In Wilhelm Schäfer (Ed.): ”Proc. 4th European Workshop on Software Process Technology (EWSPT'95), 3–5 Apr. 1995, Leiden, The Netherlands”, Springer Verlag LNCS 913, 1995, ISBN 3-540-59205-9, p. 240-240. SU-report 5/95.

19. Terje Totland, Reidar Conradi:
”A Survey and Classification of Some Research Areas Relevant to Software Process Modeling”,
In Wilhelm Schäfer (Ed.): ”Proc. 4th European Workshop on Software Process Technology (EWSPT'95), 3–5 Apr. 1995, Leiden, The Netherlands”, Springer Verlag LNCS 913, 1995, ISBN 3-540-59205-9, p. 65–69. Also as EPOS TR 232, NTH, Trondheim, Jan. 1995, 5 p. SU-report 6/95.

20. Reidar Conradi, Chunnian Liu:
”Process Modelling Languages: One or Many”,
In Wilhelm Schäfer (Ed.): ”Proc. 4th European Workshop on Software Process Technology (EWSPT'95), 3–5 Apr. 1995, Leiden, The Netherlands”, Springer Verlag LNCS 913, 1995, ISBN 3-540-59205-9, p. 98–118. Also as EPOS TR 231, NTH, Trondheim, Jan. 1995, 19 p. SU-report 7/95.

21. Guttorm Sindre, Sivert Sørumgård, Even-Andre Karlsson:
”Managing a repository”,
In Even-Andre Karlsson (ed.): Software Reuse: A Holistic Approach, Chapter 3 (p. 79–112), ISBN 0 471-95489, John Wiley & Sons., 1995. SU-report 9/95.

22. Tor Ståhane and Magnus Ahlgren:
”Measuring the Effect of Reuse”,
In Even-Andre Karlsson (ed.): Software Reuse: A Holistic Approach, Chapter 4 (p. 113–180), ISBN 0 471-95489, John Wiley & Sons., 1995. SU-report 10/95.

23. Sigurd Thunem:
”Object-oriented development for reuse”,
In Even-Andre Karlsson (ed.): Software Reuse: A Holistic Approach, Chapter 9 (p. 287–342), ISBN 0 471-95489, John Wiley & Sons., 1995. SU-report 11/95.

24. Eirik Tryggeseth, Bjørn Gulla, Reidar Conradi:
”Modelling Systems with Variability Using the PROTEUS Configuration Language”,
In Jacky Estublier: ”Software Configuration Management – ICSE SCM-4 and SCM-5 Workshops, Selected Papers”, Springer LNCS 1005, p. 216–240. ISBN 3-540-60578-9. From SCM-5 workshop 24–25 April 1995 in Seattle. SU-report 12/95.

25. Patricia Lago, Reidar Conradi:
”Transaction Planning to support Coordination” (position paper),
In Jacky Estublier: ”Software Configuration Management – ICSE SCM-4 and SCM-5 Workshops, Selected Papers”, Springer LNCS 1005, p. 145–151. ISBN 3-540-60578-9. SU-report 13/95.

26. Reidar Conradi:
”Introduction to Session on Version Models”,
In Jacky Estublier: ”Software Configuration Management – ICSE SCM-4 and SCM-5 Workshops, Selected Papers”, Springer LNCS 1005, p. 80–80. ISBN 3-540-60578-9. SU-report 14/95.

27. Reidar Conradi, Marianne Hagaseth, Chunnian Liu:
”Planning Support for Cooperating Transactions in EPOS”,
Special Issue of Information Systems, Vol. 20, No. 4 (June 1995), p. 317-336. SU-report 15/95.

28. Jens-Otto Larsen, Bjørn P. Munch, Reidar Conradi, Patricia Lago:
”Improving Cooperation Support in the EPOS CM System”,
In Proc. 8th ERCIM Database Research Group Workshop on Database Issues and Infrastructure in Cooperative Information Systems, p. 135–147. 23–25 Aug. 1995, NTH, Trondheim, Norway.
ERCIM report 95-W002, SINTEF. SU-report 21/95.

29. Guttorm Sindre, Reidar Conradi, Even-Andre Karlsson:
”The REBOOT Approach to Software Reuse”,
Journal of Systems and Software – special issue on software reuse, Vol. 30, No. 3, Sept. 1995, p. 201–212. SU-report 22/95.

30. Kari Juul Wedde, Tor Stålhane, and Inge Nordbø:
”Case Study of a Maintenance Support System”,
International Conference on Software Maintenance. IEEE Computer Society Technical Council on Software Engineering, Opio (Nice) France, October 17-20, 1995.

31. Tor Stålhane:
”Development of a Model for Reusability Assessment”,
Objective Quality – 1995, 29–31 May 1995, Firenze, Italy.

1996:

32. Reidar Conradi, Bernhard Westfechtel:
”Configuring Versioned Software Products”,
In Ian Sommerville (Ed.): ”Software Configuration Management, ICSE'96 SCM-6 Workshop, Berlin, Germany, March 1996 – Selected Papers”, Springer Verlag LNCS 1167, 1996, p. 88-109. SU-report 3/96.

33. Reidar Conradi, Noureddine Belkhatir:
”Support for Cooperative Transactions in Process-centered Software Engineering Environments”,
Proc. Seventh Nordic Workshop on Programming Engineering Environments (NWPER'96), 29–31 May 1996, Aalborg, Denmark, 14 p. SU-report 8/96.

34. Reidar Conradi:
”Process Support for Reuse”,

In Barry Boehm, Marc Kellner, Dewayne Perry (Eds.): ”Proc. 10th International Software Process Workshop (ISPW`10), 17–19 June 1996, Ventron, France”, p. 57-61, IEEE Press, IEEE CS Order Number PR07725. SU-report 9/96.

35. Noureddine Belkhatir, Reidar Conradi:
”SCOOP: A Unified Model for Cooperative Transactions in Software Engineering”,
8th Int'l Conf. on Computing and Information (ICCI'96), 19–22 June, 1996, Waterloo, Canada, 14 p.
SU-report 10/96.

36. Reidar Conradi:
”Metrics – Introduction to Session 5”,
In Carlo Montangero (Ed.): ”Software Process Technology – 5th European Workshop (EWSPT'96), Nancy, France, October, 1996”, Springer Verlag LNCS 1149, 1996, p. 144. SU-report 11/96.

37. Reidar Conradi:
”SPIQ: A Revised Agenda for Software Process Support”,
In Carlo Montangero (Ed.): ”Software Process Technology – 5th European Workshop (EWSPT'96), Nancy, France, October, 1996”, Springer Verlag LNCS 1149, 1996, p. 35–41. SU-report 12/96.

38. Minh N. Nguyen, Reidar Conradi:
”Towards a Rigorous Approach for Managing Process Evolution”,
In Carlo Montangero (Ed.): ”Software Process Technology – 5th European Workshop (EWSPT'96), Nancy, France, October, 1996”, Springer Verlag LNCS 1149, 1996, p. 18–35. SU-report 13/96.

39. Tor Stålhane:
”Process Improvement and Software Metrics”,
chapter in TIMe Electronic Textbook, Richard Sanders (ed.), SINTEF, 1996.

40. Tor Stålhane, Kjell Arnesen and Per Christian Borgersen:
”In Search of the Customer's Quality View”,
AQUiS'96 - 2nd International Conference on Achieving Quality in Software, Florence, Italy January 24 - 26, 1996.

1997:
41. Minh N. Nguyen, Alf Inge Wang, Reidar Conradi:
”Total Software Process Model Evolution in EPOS”,
In Proc. ICSE'97, 21–23 May 1997, Boston, USA, p. 390-399. ACM/IEEE-CS Press. SU-report 8/97.

42. Reidar Conradi, Bernhard Westfechtel:
”Towards a Uniform Version Model for Software Configuration Management”,
In Reidar Conradi (Ed.): ”Software Configuration Management: Proceedings from SCM'7 workshop, Springer LNCS 1235”. 18–19 May, 1997, Boston, USA, p. 1–17. SU-report 9/97.

43. Reidar Conradi (Ed.):
”Software Configuration Management: Proceedings from SCM'7 workshop”,
18–19 May, 1997, Boston, USA. As Springer LNCS 1235, 234 p. ISBN 3-540-63014-7. SU-report 10/97.

44. Reidar Conradi (Ed.):
”Software Configuration Management, 7th International Workshop (SCM'7),
Boston, USA, May 18–19, 1997.
Supplementary Proceedings: additional full papers, all position papers, and address lists”,
Issued by IDI, NTNU, Trondheim, 249 p. SU-report 11/97.

45. Reidar Conradi and Chunnian Liu:
"Revised PMLs and PSEEs for Industrial SPI", 6 p.,
In Jan Bosch and S. Mitchell (Eds.): "Object-Oriented Technology.
ECOOP'97 Workshop Reader", Springer Verlag LNCS 1357, 1997. ISBN 3-540-64039-8.
From ECOOP'97 Workshop on Software Process Technology, 10-11 June 1997, Jyvaskyla, Finland. SU-report 12/97.

46. Vincenzo Ambriola, Reidar Conradi, Alfonso Fuggetta:
"Assessing Process-centered Software Engineering Environments (OIKOS, EPOS and SPADE)",
ACM Trans. Software Engineering Methodology (TOSEM), July 1997, Vol. 6, No. 3, p. 282-328.
SU-report 14/97.

47. M. Letizia Jaccheri, Patricia Lago:
"Applying Software Process Modeling and Improvement in Academic Setting",
Proc. of the 10th ACM/IEEE-CS Conference on Software Engineering Education and Training, Virginia Beach, April 1997. SU-report 20/97.

48. M. Letizia Jaccheri, Patricia Lago:
"Metrics aspects of a software engineering course project",
Proc. Inspire II, 2nd International Conference on Software Process Improvement – Research into Education and Training, Sweden. September 1997. SU-report 21/97.

49. Tor Stålhane, Tore Dybå and Reidar Palmstrøm:
"Introduction of Goal-Oriented Measurement",
STEP'97, London, UK, July 15 - 18, 1997.

50. Tore Dybå, Tor Stålhane and Reidar Palmstrøm:
"Experience of Goal-Oriented Measurement using ami and GQM",
The European Software and Control Metrics Conference, Berlin, Germany, May 26-28, 1997.

51. Tor Stålhane and Kari Juul Wedde:
"Modification of Safety Critical Systems – An Evaluation of three Approaches",
Microprocessors and Microsystems 21, 1998.

52. Tor Stålhane and Kari Juul Wedde:
"Modification of Safety Critical Systems – An Evaluation of three Approaches",
ENCRESS'97, Athens, Greece, May 29-30, 1997.

53. Tor Stålhane, Kjell Arnesen and Per Christian Borgersen:
"In Search of the Customer's Quality View",
Journal of Systems and Software, vol. 38, 1997, pp. 85-93 (reworked from AQUIS'96 paper).

1998:
54. Reidar Conradi, Minh Ngoc Nguyen, Alf Inge Wang, Chunnian Liu:
"Planning Support to Software Process Evolution",
Proc. Eight International Conference on Software Engineering and Knowledge Engineering (SEKE'98), San Francisco, USA, 18–20 June 1998, 16 p. SU-report 1/98.

55. Reidar Conradi:
"Practical Software Experience Databases: Use the Web (position paper)", 3 p.
Proc. Eight International Conference on Software Engineering and Knowledge Engineering (SEKE), San Francisco, USA, 18–20 June 1998, 16 p. SU-report 2/98.

56. Bernhard Westfechtel and Reidar Conradi:
"Software Configuration Management and Engineering Data Management: Differences and Similarities", In Boris Magnusson (Ed.): "Software Configuration Management", Springer LNCS 1439, p. 95–106. ECOOP'98 SCM-8 Symposium, 20-21 June 1998, Brussels. SU-report 3/98.

57. Reidar Conradi, Bernhard Westfechtel:
"Version Models for Software Configuration Management",
ACM Computing Surveys, Vol. 30, Number 2, July 1998, p. 232–282. SU-report 17/97.

58. Alf Inge Wang, Jens-Otto Larsen, Reidar Conradi, Bjørn Munch:
"Improving Cooperation Support in the EPOS CM System",
In Volker Gruhn (Ed.): "Software Process Technology, 6th European Workshop, EWSPT'98, Weybridge, UK, 16-18. Sept. 1998", Springer Verlag LNCS 1487, p. 75–91. SU-report 5/98.

59. Reidar Conradi, Alfonso Fuggetta, Maria Letizia Jaccheri:
"Six theses on Software Process Research",
In Volker Gruhn (Ed.): "Software Process Technology, 6th European Workshop, EWSPT'98, Weybridge, UK, 16-18. Sept. 1998", Springer Verlag LNCS 1487, p. 100–104. SU-report 6/98.

60. Letizia Jaccheri and Patricia Lago:
"How project-based courses face the challenge of educating software engineers",
Accepted at World Multiconference on Systemics, Cybernetics and Informatics (SCI'98) and the 4th International Conference on Information Systems Analysis and Synthesis (ISAS'98), July 12-16, 1998, Orlando, USA. SU-report 8/98.

61. Chunnian Liu, Reidar Conradi:
"Process View of CSCW",
Proc. Int'l Symposium of Future Software Technologies (ISFST'98), Hangzhow, P.R. China, 28–30 Oct. 1998. SU-report 11/98.

62. M. Letizia Jaccheri, Patricia Lago, G.P. Picco:
"Eliciting Software Process Models with the E3 Language",
ACM Transactions on Software Engineering and Methodology, Vol. 7, No. 4, October 1988, pp. 368-410. SU-report 22/97.

63. M. Letizia Jaccheri, Patricia Lago:
"How Project-based Courses face the Challenge of educating Software Engineers",
Proc. of the Joint World Multiconference on Systemics, Cybernetics and Informatics (SCI98) and the 4th International Conference on Information Systems Analysis and Synthesis (ISAS98), Orlando, USA, July 1998.

64. Tor Stålhane, Kari Juul Wedde and Tore Dybå:
"Data Driven Improvement for SMEs",
EuroSPI'98, Gothenburg, Sweden October 17-18, 1998.

65. Tor Stålhane and Kari Juul Wedde:
"Practical experience with the application of HazOp to a software intensive system",
Joint ESCOM and ENCRESS Conference, Roma, Italy, May 27-29, 1998.

66. Bjørn Haugrud, Tor Stålhane and Inger Venås:
"The Effect of Using Object-Oriented Methods in Specification and Design",
AQUiS'98 –3rd International Conference on Achieving Quality in Software, Venezia, Italy March 30 – April 2, 1998.

1999:
67. Reidar Conradi:
”Kommentar til UMLs nye Object Constraint Language (OCL)”,
DND/Faggruppen for metoder (ed.): ”UML – Unified Modeling Language og komponentbasert systemutvikling”, DND, Oslo, foilsamling i eget hefte fra seminar på Geilo, 19-20 April 1999, 44 foiler. SU-report 11/99.

68. Reidar Conradi and M. Letizia Jaccheri:
”Process Modelling Languages”, p. 27–52.
In PROMOTER2 book: Jean-Claude Derniame, Badara Ali Kaba, David Wastell (Eds.): Software Process: Principles, Methodology, and Technology, Springer LNCS 1500, 307 p. SU-report 1/99.

69. Eirik Tryggeseth et al.:
”Chapter 5: Architectural Modeling for Evolution”, 24 p.,
In Ian Warren: The Renaissance of Legacy Systems: Method Support for Software System Evolution” (The Method Book from the RENAISSANCE ESPRIT Project in 1996-98, project no. 22010), Springer Practitioner Series, 1999, 182 p. SU-report 2/99.

70. Alf Inge Wang, Reidar Conradi, Chunnian Liu:
”A Multi-Agent Architecture for Cooperative Software Engineering”,
In Guenther Ruhe (ed.): ”Proc. Eleventh International Conference of Software Engineering and Knowledge Engineering (SEKE'99)”, Kaiserslautern, Germany, 16-19 June 1999, p. 162–169. SU-report 3/99.

71. Reidar Conradi:
”From software experience databases to learning organizations (panel introduction)”,
In Gunther Ruhe (ed.): ”Proc. Eleventh International Conference of Software Engineering and Knowledge Engineering (SEKE'99)”, Kaiserslautern, Germany, 16-19 June 1999, p. 204–206. SU-report 4/99.

72. Reidar Conradi, Amarjit Singh Marjara, Børge Skåtevik:
”An Empirical Study of Inspection and Testing Data at Ericsson”,
In Markku Oivo and Pasi Kuvaja: ”Proc. Int'l Conf. on Product Focused Software Process Improvement (PROFES'99)”, Oulu, Finland, 22-24 June 1999, p. 263–284. VTT symposium 195, VTT, Espoo, Finland, 666 p., ISBN 951-38-5270-9; 951-38-5271-7 (URL: http://www.inf.vtt.fi/pdf). SU-report 5/99.

73. Reidar Conradi, Bernhard Westfechtel:
”SCM: Status and Future Challenges”,
In Jacky Estublier (Ed.): ”Proc. Int'l Workshop on Software Configuration Management (SCM'9)”, Springer Verlag LNCS 1675, ISBN 3-540-66484-X, Toulouse, 5-7 Sept. 1999, p. 228-231. SU-report 7/99.

74. Torgeir Dingsøyr, M. Letizia Jaccheri, Alf Inge Wang:
”Teaching Software Process Improvement Through a Case Study”, In Proc. International Conference on Engineering and Computer Education, Rio de Janeiro, Brazil, 9-12 Aug. 1999”. Publisher: Mundo Virtual (CD-ROM), 5 p.
Also presented at Norsk Informatikk Konferanse (NIK'99), Trondheim, Nov. 1999. SU-report 8/99.

75. M. Letizia Jaccheri:
”A Software Quality and a Software Process Improvement Course”,
Forum for Advancing Software engineering Education (FASE),
Vol. 9, No. 5 (112th issue), 15 May 1999. SU-report 9/99.

76. M. Letizia Jaccheri, Mario Baldi, Monica Divitini:
”Evaluating the requirements for software process modeling languages and systems”,

WORLD MULTICONFERENCE SCI/ISAS'99, International Workshop on Process Support for Distributed Team-based Software Development (PDTSD'99), Orlando, USA, 31 July - 4 August, 1999. SU-report 10/99.

77. Torgeir Dingsøyr:
”A lifecycle process for experience databases”,
In Sasha Schmitt and Ivo Vollrath (Eds.): ”Challenges for Case-Based Reasoning: Proc. International Conference on Case-Based Reasoning (ICCBR'99), Seeon, Germany, 27-30 July 1999”, p. 9-13.
Centre for Learning Systems and Applications, Kaiserslautern, Germany. SU-report 15/99.

78. Roxana Elena Diaconescu, Reidar Conradi:
”An Object-Oriented Approach to Numerical Calculation of Physical Entities”, Proc. EUROPRIME, Warsaw, Poland, 12–13 Sep. 1999, 11 p. SU-report 16/99.

79. Alf Inge Wang:
”Experience paper: Using XML to implement a workflow tool”,
Proc. IASTED International Conference on Software Engineering and Applications (SEA'99), Scottsdale, Arizona, 6-8 Oct. 1999. SU-report 13/99.

80. Reidar Conradi, Amarjit Singh Marjara, Børge Skåtevik:
”An Empirical Study of Inspection and Testing Data at Ericsson, Norway”,
In ”Proc. 24th NASA Software Engineering Workshop, Greenbelt/Washington, USA, 1–2 Dec. 1999”, 16 p. (URL: http://sel.gsfc.nasa.gov/sew). SU-report 18/99.

81. Christian P. Halvorsen:
”A Taxonomy for SPI Frameworks”,
In ”Proc. 24th NASA Software Engineering Workshop, Greenbelt/Washington, USA, 1–2 Dec. 1999”, 4 p. (http://sel.gsfc.nasa.org/sew). SU-report 19/99.

82. Hans Jørgen Lied and Tor Stålhane:
”Experience from process improvement in an SME”,
EuroSPI'99, Pori, Finland, October 26–27, 1999.

83. Tor Stålhane and Kari Juul Wedde:
”SPI – Why isn't it more used”,
EuroSPI'99, Pori, Finland, October 26–27, 1999.

84. Tor Stålhane:
”The role of measurement in software and software development”,
Telektronikk, Vol. 95, 1/99 (special issue on Software Quality in Telecommunication).

2000:
85. Torgeir Dingsøyr, Reidar Conradi:
”Metoder for å høste og spre erfaringer”,
Som kap. 5 i Tore Dybå (red.): ”SPIQ metodebok i prosessforbedring”, Tanum forlag, 2000. SU-report 20/99.

86. Reidar Conradi, Alf Inge Wang:
”Beskrivelse av programvareprosesser”,
Som kap. 10 i Tore Dybå (red.): ”SPIQ metodebok i prosessforbedring”, Tanum forlag, 2000,
IDI-rapport no. 2-2000, ISSN 0802-6394. SU-report 21/99.

87. Reidar Conradi:

”Terminologi”,
Som vedlegg A i Tore Dybå (red.): ”SPIQ metodebok i prosessforbedring”, Tanum forlag, 2000, IDI-rapport no. 2-2000, ISSN 0802-6394. SU-report 22/99.

88. Alf Inge Wang:
”Support for Mobile Processes in CAGIS”,
In Reidar Conradi (Ed.): Proc. 7th European Software Process Workshop on Software Process Technology (EWSPT'7), Kaprun near Salzburg, Austria, 21-25 Feb. 2000. Springer Verlag LNCS 1780, p. 115-130. SU-report 1/2000.

89. M. Letizia Jaccheri, Reidar Conradi, Bård H. Dyrnes:
”Software Process Technology and Software Organisations”,
In Reidar Conradi (Ed.): Proc. 7th European Software Process Workshop on Software Process Technology (EWSPT'7), Kaprun near Salzburg, Austria, 21-25 Feb. 2000. Springer Verlag LNCS 1780, p. 96-108. SU-report 2/2000.

90. Reidar Conradi (Ed.):
”Proc. 7th European Software Process Workshop on Software Process Technology” (EWSPT'7), Kaprun near Salzburg, Austria, 21-25 Feb. 2000, Springer Verlag LNCS 1780, Feb. 2000. SU-report 3/2000.

91. Reidar Conradi:
”Report from SEKE'99 panel on From Software Experience Databases to Learning Organizations”, Accepted for Journal of Software Engineering and Knowledge Engineering, year 2000, 4 p. SU-report 4/2000.

92. Alfonso Fuggetta, M. Letizia Jaccheri:
”Dynamic partitioning of complex process models”,
Journal of Information and Software Technology, 42 (2000), p. 281-291. SU-report 5/2000.

93. Reidar Conradi (scribe):
”Summary from the 7th European Software Process Workshop (EWSPT'7), Kaprun near Salzburg, 21-25 Feb. 2000”,
ACM Software Engineering Notes, May 2000, p. 23. SU-report 6/2000.

94. Reidar Conradi, Minh Ngoc Nguyen, Alf Inge Wang, Chunnian Liu:
”Planning Support to Software Process Evolution”,
Journal of Software Engineering and Knowledge Engineering (SEKE Journal), forthcoming May 2000. SU-report 6/99.

95. Torgeir Dingsøyr:
”Knowledge Management for Software Process Improvement”,
Nordic Workshop on Programming Environment Research (NWPER'2000), Lillehammer, Norway, 28-30 May, 2000. SU-report 7/2000.

96. Reidar Conradi, Mikael Lindvall, and Carolyn Seaman:
”Success Factors for Software Experience Bases: What We Need to Learn from Other Disciplines”, 6 p., ICSE'2000 Workshop on ”Beg, Borrow or Steal: Using Multidisciplinary Approaches in Empirical Software Engineering Research”, Limerick, Ireland, 5 June 2000. SU-report 8/2000.

97. Reidar Conradi and Torgeir Dingsøyr:
”Software Experience Bases: A Consolidated Evaluation and Status Report”,

Accepted at Second Int'l Conf. on Product Focused Software Process Improvement (PROFES'2000), Oulu, Finland, 20-22 June 2000, 14 p. SU-report 10/2000.

98. Alf Inge Wang:
"Experience paper: Implementing a Multi-Agent Architecture for Cooperative Software Engineering", Accepted for Twelfth International Conference of Software Engineering and Knowledge Engineering (SEKE'2000), Chicago, 6–8 July, 2000. SU-report 9/2000.

99. Reidar Conradi and Torgeir Dingsøyr:
"Software Experience Bases: Some Results and Recommendations",
Accepted at 4th International FEAST Workshop, London, 10-12 July 2000, 9 p. SU-report 11/2000.

100. Alf Inge Wang:
"Using Software Agents to Support Evolution of Distributed Workflow Models",
Accepted at "International ICSC Symposium on INTERACTIVE AND COLLABORATIVE COMPUTING (ICC'2000)", at International ICSC Congress on INTELLIGENT SYSTEMS AND APPLICATIONS (ISA'2000), Wollongong (near Sydney), Australia, December 12-15, 2000, 7 pages.

101. Tor Stålhane and Kari Juul Wedde:
"Safety Validation with Focus on Testing",
ICSTEST – International Conference on Software Testing, April 5–7, 2000, Bonn, Germany.

102. Kari Juul Wedde and Tor Stålhane:
"The use of experts in metrics interpretation and analyses",
Proc. combined 11th European Software Control and Metrics conference (ESCOM) and 3rd SCOPE conference on Software Product Quality, April 18–20, 2000, München, Germany.