PhD defense

Collaborative learning in software development:
An investigation of characteristics, prerequisites and improvement

Tor Erlend Fægri
March 13, 2012
Outline of talk

- Motivation
- Conceptual point of origin
- Related work
- Research problem and research questions
- Research approach
- Research contribution
- Implications for practice and further research
Introduction

- Software is vital infrastructure in our society
  - Software’s flexibility, versatility and pervasiveness enable the construction of advanced and innovative systems with great impact for business and personal life
- Yet, these features make software development inherently challenging:\n  - Much uncertainty
  - High complexity
  - High degree of interdependence
- Coping with these challenges demands high capacity for flexible, integrative learning²

---

Motivating question

If software development demands a high capacity for flexible, integrative learning ...

... how can we conceptualize it, enable it and improve it?
Collaborative learning: The conceptual origin

Collaborative learning entails *learning in joint work*
Related work

- Several scientific discourses gives insight into aspects of collaborative learning:
  - *Software development*: software practice, SPI;
  - *Learning literature*: education and CSCL, creativity research;
  - *Organization literature*: team research, organizational learning, Communities of Practice;
  - *Management literature*: knowledge management, innovation management;

- However, a lack of *integrative research* that elaborates and conceptualizes what collaborative learning is

- This is problematic for software development
  - Growing appreciation of the pivotal role of learning in software development
  - With increasing pervasiveness of software based systems integrative research becomes more important
  - Few studies of actual change and adoption
Research problem and research questions

The research problem

*How can capabilities for collaborative learning in software development be improved?*

Research questions

1. *What are the characteristics of collaborative learning in software development?*
2. *What are the prerequisites for collaborative learning in software development?*
The interpretive research approach

"Interpretive research does not predefine dependent and independent variables, but focuses on the complexity of human sense making as the situation emerges; it attempts to understand phenomena through the meanings that people assign to them."1

An interpretive approach was my choice because:

- I wanted to explore the complexity of the collaborative learning phenomenon
- In particular – it’s role in software practice
- It can accommodate a flexible research process

Outline of research process

Stage 1
- Problem framing
- Case study: Investigation A

Stage 2
- Elaborating action research
  - Investigations B, C

Stage 3
- Conceptual framework development

Stage 4
- Analysis and discussion
- Conceptual framework development
- Elaborating action research
  - Investigation B, C
- Problem framing case study:
  - Investigation A

Literature study
Problem framing case study: Investigation A

Context: Medium-sized, Norwegian software product company (CompNN) in transition to evolutionary development.

Primary objective: Exploring the impact of evolutionary, customer-centric development processes.

Research approach: Longitudinal case study.


Paper covering the use, experiences and adjustment of collaborative learning processes in evolutionary product development.


Paper describing the orchestration of collaborative learning processes to accommodate CompNN’s strategic, tactical and operational concerns.
Elaborating action research: Investigation B

Context: Medium-sized, Norwegian software product organization delivering geographical information systems (MapIT).

Primary objective: Exploring the effects of job rotation on the development of breadth knowledge.

Research approach: Action research covering the introduction of job rotation among developers to customer support.


Paper discusses merits of job rotation to improve breadth knowledge among developers. It uses organizational learning as the theoretical framing.


Paper elaborates upon the knowledge redundancy aspect, i.e. that people have overlapping skills and competence. The paper also discusses the socio-technical considerations of sustainable development of knowledge redundancy.
Elaborating action research: Investigation C

**Context:** A product group (CardPay) within a large, Norwegian ICT solution provider in the banking sector.

**Primary objective:** Exploring social and organizational impacts of specialist organizational culture on collaborative learning.

**Research approach:** Action research covering the introduction and adoption of Planning Poker in CardPay.


Paper describing the full action research cycle that lasted approximately a year. The most significant contribution of the paper is its description of how practice-based techniques – such as Planning Poker – can become a vehicle for surfacing, discussing and overcoming organizational barriers to collaborative learning.
Conceptual framework development

Legend:
(1) Identification of main elements in conceptual framework.
(2) Theoretical contribution to analysis and discussion.
(3) Empirical contribution to analysis and discussion.
(4) Existing theory gave direction to empirical work. In seeking to explain empirical findings, particular theoretical fields became relevant.
(5) Empirical work gave confidence to conceptual framework development.
Collaborative learning is a type of *social object*:\(^1\):

- Socially constructed. The object is brought into being by people but may act independently of particular people.
- Contingent. Creation of social objects is non-deterministic – only probabilistic.
- Dynamic. The creation of a social object is causally dependent on other social objects.

Applied to the research questions and the research problem:

- Characteristics: “How can we recognize collaborative learning social objects?”
- Prerequisites: “What are the ‘creators’ of collaborative learning social objects?”
- Improvement: “How can we support the creation of more collaborative learning social objects?”

The trio of characteristics, prerequisites and considerations for improvement, is considered the *pillars* of the conceptual framework

Conceptual modeling: The influence of practice

Engagement with software practice was very helpful

- Created awareness of the dimensions of collaborative learning
- Broadened the scope of relevant literature
- Gave better meaning to the literature

Central theoretical strands

- The social construction of software through collective, situated performances
- Empowerment resulting from emergent perception of problem-solving capability
- The surfacing of hidden values and norms through practice assisted in the change process

Example; introduction of planning poker at CardPay (investigation C):

The technique strikes a delicate balance between mere presence and actual joint, collaborative development
Main contribution: The conceptual framework

**Collaborative learning in software development**

**Characteristics**
- Collective, situated performance
- Purposeful
- Cooperative behavior
- Coordinated action
- Collective power

**Prerequisites**
- Participation
- Opportunities for learning
- Knowledge diversity
- Knowledge redundancy
- Proximity; physical or virtual

**Improvement**
- A re-orientation of values
- Creating learning opportunities
- Socio-technical reconciliation
Example empirical foundation: Improvement, socio-technical reconciliation

The combining of social and technical concerns creates a range of challenges for collaborative learning in software practice. These challenges are situated in the organization’s context and encourage local adaptations. The challenges may even fluctuate over time and thereby require continuous attention.

1: Common technical expertise
2: ...
3: ...
4: Experimentation with different strategies may give the necessary leverage for sustained use

Observation: Despite the group estimation technique, which ‘democratized’ input from each participant, there were moderately challenging tasks for the individuals. All tasks pertained to a product with which all had associations. Most of the tasks had some relevance to the participants’ former technical experience.

Observation: The CardPay group decided to implement local adoptions to the planning technique. Only between a third and half of the tasks were subjected to group-based estimation. This created a useful compromise between time use and effect in knowledge redundancy.
Limitations

- Flexible research design
  - Research questions were fixed during the last phase of empirical work
  - May endanger depth of empirical work
- Generalizability
  - Generalizing from case studies more difficult (‘useful exemplars’)
  - Limited contextual richness on psychological traits
  - Case selection
- Completeness and representativeness of the conceptual framework
  - The conceptual framework has not yet been used by others
- Breadth at the expense of depth
  - ‘Connecting’ rather than integrating and synthesizing
Implications for further research

- Clear conceptualizations are useful vehicles to support continued research
  - Qualitative research
  - Quantitative research
- Contribution to the socio-technical aspects of learning in software development
  - Holistic learning strategies – embracing both social and technical systems
- Learning techniques and practices
  - Could be an interesting starting point for making research more relevant for software practice
Implications for practice

- Conceptualization improves understanding of collaborative learning in software development
  - What it is
  - How it is enabled
  - How to improve it

- Practical management of software development
  - Project staffing
  - More holistic strategies of product portfolio management, product management

- Use of learning practices
  - Learning strategy; learning techniques
Thank you for your attention!