Software Development: a social activity with technical practices

Helen Sharp

Centre for Research in Computing
The Open University, Walton Hall
Milton Keynes MK7 6AA, UK

ACCU 2006, 20th April

Acknowledgements to:

Hugh Robinson, Stephanie Chamberlain, Fiona Hovenden,
Mark Woodman, all study participants

Overview

- **Three studies, three sets of findings illustrating specific social influences in software development:**
  - **Technical practices of software development (XP studies)**
  - **Influence of individuals (software quality study)**
  - **Strength of communities (origins of object-orientation)**

- **Think about:**
  - **How do these (and other) interactions affect what developers (you) do as professionals in a highly technical field?**

Social activity

- **Interactions between people**
- **Importance of social/human not revolutionary**
  - **Psychology of Programming (e.g. Weinberg)**
  - **Cognitive (e.g. Detienne)**
  - **Team composition, 'right' people (e.g. Belbin)**
  - **Managerial, workload, career (e.g. DeMarco & Lister)**
  - **Exceptional designers (e.g. Curtis and MCC)**
- **... but our perspective is different**

Research Approach

- **Focus on software practitioners**
- **Qualitative data gathering: interviews, observation, documentation, diaries**
- **Emergent themes**
  - **Data-driven approach**
  - **No *a priori* hypotheses to prove or disprove**
  - **Viewing everything as 'strange'**
- **Social interactions**
  - **what is influential?**
  - **paying attention to technical context**

Studies of mature eXtreme Programming (XP) teams

Focus:

To understand how XP works in practice, e.g. the interplay between the technical practices

Social observation:

Technical practices rely on social interactions

Data gathering

- **Observational studies of at least one week**
- **Six teams: five mature, one ‘lapsed’**
  - **Financial institution**
  - **Bank**
  - **Document management**
  - **Travel websites**
  - **Web adverts**
- **(Further three teams in one media organisation, studied for 6 months)**

Findings

- **Various areas:**
  - **Team characteristics**
  - **Organisational culture**
  - **Information flows within and around a team**
  - **Details of different practices**
- **Social interactions affect technical practices and how they are conducted**

Social interactions affect ...

- **Code production through pair programming**
  - **three-way conversation**
  - **peripheral awareness leads to expanded conversation**
  - **stressful & intense**
- **Refactoring**
  - **when, what, who**
- **Maintenance**
  - **pool resources**

Social ⟷ Technical

Social observation:

Social interactions influence technical practices

Technical practices influence social interactions

The adoption and evolution of Software Quality Management Systems

Focus:

To identify the non-technical factors influencing the adoption and evolution of SQMSs (e.g. ISO9000, TiCKIT)

Social observation:

Influential individuals such as the ‘maverick’ developer

Data gathering & analysis

- **Interviews with quality managers in 5 companies**
  - **quality-related aspects such as organisation structure, SQMS history, gurus and publications with influence**
- **One-week field study in one company (interviews and observation)**
- **Discourse analysis of interview and meeting transcripts framed by observations**

Findings

- **Key influence was peer network**
  - **Consult internal experts rather than gurus**
  - **Publications and conferences little influence**
- **Individuals have strong influence**
  - **The charismatic leader**
  - **The quality manager**
  - **The ‘maverick’ developer**

The ‘Maverick’ Developer

- Technically very good and innovative
- Highly productive problem solver
- Great respect from colleagues
- Work to their own rules (hiding, ownership)
- Highly influential:
  - Commands approval
  - SQMS was not considered by others
  - Technical decisions taken only with his blessing

The ‘Maverick’ Developer

“Tell him what you want, three months later, on schedule he produces what you wanted and it’s usually good... very few problems in it. Yet, say to him well look we just want you to put down beforehand what you’re going to do, and then document when you’ve finished what you’ve done, he says that’s going to take me three times as long. I mean, what’s the approach to these people? How can we sell the principle ... so their hearts and minds are with it?”

IEE Colloquium on SQMS in London, 1993

Influence of peers and maverick developers is strong

Social observation:

‘Maverick’ developers influential but not always explicit

A social history of object orientation

Focus:

To trace the history of object orientation as seen through contemporaneous documents

Social observation:

Communities shape the adoption and emergence of technical innovation

Data gathering & analysis

- **Studied articles, announcements, adverts from:**
  - ***Byte*** **(1985 to 1992)**
  - ***IEEE Computer*** **(1978 to 1992)**
  - ***IEEE Expert*** **(1986 to 1992)**
  - ***ACM SIGPLAN Notices*** **(1978 to 1992)**
- **Looked for**
  - **Mention of 'objects'**
  - **Significance attached to 'object'**
  - **Articles relating to object-oriented 'anything'**
- **Quantitative counts and qualitative analysis**

Findings

- **Three phases in acceptance of object-orientation**
  - **Interpretative flexibility (unsure of its significance; to mid-80s)**
  - **Community and dissemination (building a community, conferences etc; mid to late 80s)**
  - **Pervasiveness (throughout software; late 80s and early 90s)**
- **Observations**
  - **Social history differs from intellectual history of idea**
  - **Influence of community in growth and acceptance of new technology**

Example facts (first phase)

- **In the 6½ years from June 1978 to December 1985, there are only 9 papers (out of 400) in *ACM SIGPLAN Notices* where some substantive mention was made of object-orientation**

- **As late as August 1987 *IEEE Computer* contains articles which use the word 'object' as an everyday synonym of 'thing' e.g., 'engineering object' referring to artefacts such as widgets, '3D object' referring to a three-dimensional structure, etc., with no qualification of its use**

Example facts (first phase)

- **The earliest reference to an object-oriented *paradigm* was in 1982**
- **The August 1981 Special Issue of *Byte* was devoted to a series of articles on the Smalltalk-80 system, all authored by individuals from the Xerox Palo Alto Research Centre (PARC) where the system was developed. The articles range across various aspects of the Smalltalk-80 system, but there is an emphasis on the system's own GUI, and its ability to support good interface design**
- **Adele Goldberg's 1981 Smalltalk talk**

ABSTRACT: The Xerox Smalltalk-80 system is the latest effort in a long-term research program to explore new approaches to computer software design. It differs from other Smalltalks in its basic syntactic structures, in the design of the interpreter and choice of primitive routines, and in the selection of system class descriptions.

To test its transferability to a wide variety of hardware systems, the Smalltalk-80 system has been documented in the form of books and a magnetic tape, which are under review for more general release later this year. The tape contains all of the source code, written in Smalltalk itself.

This talk will summarize the main features of this release of Smalltalk, including an analysis of how the software transfer is progressing.

BIOGRAPHY: Dr. Adele Goldberg is manager of the Xerox PARC Learning Research Group and a Principal Scientist. She has been with Xerox Corporation since 1973, working on the development of software systems for personal and school use. Previously a Research Associate at the Institute for Mathematical Studies in the Social Sciences, Stanford, she received her Ph.D. in Information Sciences from the University of Chicago. Dr. Goldberg is currently Editor-in-Chief of the ACM Computing Surveys.

1986 turning point ...

- **The (USA) OOPSLA series began in 1986**
- **A 'C++ Available' announcement in *SIGPLAN Notices* of February 1986:**

  AT&T has announced that the C++ programming language is now available to both educational and non-educational institutions. ... Distribution is handled by AT&T Software Sales and Marketing ... or by your local sales organization for the UNIX system.

- **The (European) ECOOP series began in 1987**
- **The Journal of Object-Oriented Programming (JOOP) was founded in 1987**
- ***SIGPLAN Notices* has 286 papers in the period January 1986 to December 1989 (4 years)**

… so what happened?

- **No evidence of persuasion through documents**
- **Community already persuaded**

  “If the conference attendance and the volume of papers submitted are any indication, we have tapped an eager and growing population” Bobrow & Purdy, 1986
- **Strength of community (HICOOPS)**

Sponsored by SIGSPOOF and SPOOKS

Preliminary Program:

- A.Trendie: *Towards a Theory of Object-Oriented Programming Frameworks.* (invited paper).
- T.Ank, J.B.DeFeet, M.A.Drumett, R.A.Y.Gunn, and C.Major: *Merging ADA[1] and DB5X+[2] for Object-Oriented Tactical Database Management Systems.*
- G.Buzzword and G.E.T.Rich: *An Object-Oriented Technology for Quality Control in a High-Tech Production Environment.*
- C.P.A.Petty: *Standardization of level 3 Object Representation Specification Languages.*
- J.Dullman: *Object-Oriented Programming in the Context of Extended Abstract Data Type Theory.*
- G.R.Sue, R.E.Grabitt, and G.E.Runne: *Legal Aspects of Object Oriented Prolog.*
- A.Bashful, B.Doc, C.Dopey, D.Grumpy, E.Happy, F.Sleepy, G.Sneezy: *A Conceptional Design for an Object Oriented Graphics Interface for an Intelligent User-Friendly Expert Workstation.*
- A.M.Tired and G.Oldie: *Object-Oriented Software Engineering for the Nineties.* (invited paper).
- I.Hill and I.H.West: *An Integrated Test-bed for Object-Oriented 'C' Programs under the UNIX[3] Operating System.*
- A.Prof, O.O.Grad, and P.D.Intern: *A Meta-Language for Object-Oriented Virtual Program Synthesis for Large Parallel Architectures.*
- P.D.Quick: *What I did in my Summer Vacation.* (shurly shome mistake -ed.).
- A.Wild, B.Wully, and C.Weird: *ExtraLisp – A General and Efficient Object-Oriented Extension of CommonLisp with Parallel Write-Only Semantics.*
- P.S.DeGroan Jr.: *Architectural Support for Expert System Development using High-Performance Professional Workstations in a Secure Integrated Network File System Environment.*
- N.S.A.Float and N.R.O.Double: *Object Oriented Programming in Fortran-77.*
- G.R.Emlin: *Artificial Intelligence Techniques for Object-Oriented Software Engineering.*
- Z.Z.Dull and S.O.Boring: *ISSOOSK – Design of an Operating System Kernel for an Integration Support System.*
- N.O.T.Sensible: *An Object-Oriented Design Methodology for Cobol.*

Influence of a strong community

Social observation:

Communities shape the adoption and emergence of technical innovation

Social history differs from intellectual history of an idea

Summary

- **Examples of three different social influences**
  - **Technical practices rely on social interactions**
  - **‘Maverick’ developers influential but not always explicit**
  - **Communities shape adoption & emergence of innovation**
- **How do these (and other) interactions affect what developers (you) do as professionals in a highly technical field?**

Thank you

Any questions/comments?

h.c.sharp@open.ac.uk