



BRITISH COMPUTER SOCIETY

Kingston & Croydon Branch

THE IMPACT PROJECT

Professor Alex Wolf, Imperial College, London

Tuesday 13th November 2007 at 7.30pm
at BT Delta Point, Croydon

1 Introduction

Tonight's talk is about the Impact project of ACM SIGSOFT and illuminates the question, "does software engineering research have impact on software engineering practice?"

1.1 Assumptions

- There is a software engineering community, which practices software engineering
- There is software engineering research, which asks questions about how to do this better
- There is widespread scepticism

Witness the reduction in funding for both academic and industrial research.

2 Where does software come from?

- Facile answers are misleading

Ideas have many parents!

Researchers, scientists, open source developers...

Contributions range from conceptualisation via evangelism, prototyping, public communication, nurturing via community activities, education, product commercialisation...

2.1 Why should we care?

- Facile answers are misleading

Software isn't all good. Some of it is very good.

How do we evaluate a contribution?

Ideally there would be a formal process – qualitative, quantitative. This is a challenging task! How do you objectively measure such very different contributions? (Attribution is difficult, and timescales can be

very long – 10 or 15 years is not uncommon. People forget and it becomes hard to trace impact of each contribution).

Example: UK Government seeks accountability for research spending. Why? To demonstrate the effective exploitation of research outputs.

DTI report in 2006 on “increasing the economic impact of the research councils” (i.e. get them to prove that they have any reason to exist). The report

- Provides several recommendations on how to consider economic impact in funding decisions
- Recommends that each review panel should have at least one person competent in measuring economic impact

But measuring is easier said than done! The EPSRC (also in 2006) produced an “International Review of ICT Research in the UK”. They noted: “It is difficult to conduct a macro-economic analysis of ICT economic impact”.

In 2007, the Russell Group repudiated the DTI report: “There is no evidence to date of any rigorous way of measuring economic impact other than in the very broadest of terms and outputs”.

Economists didn’t even recognise that computing had any economic impact at all until very recently! (Thankfully, it was positive!)

3 The Impact Project

3.1 Goals

- Scholarly, objective and case-based evaluation
- Desired deliverables
 - Peer-reviewed papers
 - Presentation materials and outreach activities
 - Expertise in conducting such studies
- Community building
- Prospective for future research investment
- Lessons learned for “successful” research
 - But only with respect to transfer into practice!

Software engineering research has the lowest success rate in funding applications out of all of computing science!

There are several possible approaches: tracing back from successful practice, or tracing forward from research initiatives. The former approach is more objective.

3.2 Administration

Impact is an initiative of ACM SIGSOFT, who provided some funding too. Funding also provided by various national government agencies in the US, Europe, and Japan.

3.3 Method

Teams were formed around particular practices

- Important and widespread
- Examples: configuration management, programming languages, middleware, assertions, walkthroughs, testing...
- Find and recruit researchers (both academic and industrial) and practitioners

Start from the practice and trace backwards

Use accepted historical tools (advised by Professor Michael Mahoney – science historian)

- Qualify conclusions by solidity of evidence, use references, oral histories etc.

3.4 Some results

The first completed report was on Software Configuration Management in 2005. By the most recent report, the techniques had improved enormously.

4 Software Configuration Management

Investigated the research origins of the leading vendors' SCM products (circa 2003).

Lead authors were Jacky Estublier and David Leblang. Collaborators were G. Clemm (IBM – Clearcase), R. Conradi (Univ. Trondheim), A. van der Hoek (U. California), W. Tichy (Univ. Karlsruhe, inventor of RCS), D. Wiborg-Weber (Telelogic – Continuous)

Practice was measured by the success of vendor products.

Ovum and Gartner estimates of the value of the SCM product market in 2003 were between \$3bn and \$6bn.

This approach ignored the success of open source systems such as CVS and Subversion, which have eroded the vendors' market shares since 2003 but which have similar technical origins.

4.1 Application of the historical method

- Examine the characteristics and features of the leading products
- Assume that the products are used in products

- Trace back to research ideas and prototypes
- Make arguments for and against the influence of research on practice (via products)

Incidentally, many research ideas closely associated with the researchers that produced the winning ideas were found *not* to have been adopted by commercial tools. Vendors found that the products became too complicated to sell if they incorporated these ideas.

It turns out that in this field, industrial researchers made the earliest breakthroughs. SCCS was developed at Bell Labs in 1972 (followed by diff in 1976 and make in 1977). Variants were the first academic contribution, with RCS in 1980.

How is industrial research made visible? In the case of Bell Labs, it published the software as part of the UNIX system.

The team also found that many times, products started to emerge with innovative features some time after a research prototype was published with those features. The timing between these events appeared to be significant. The Impact researchers were able to tie the product architects and the prototype developers together via the attendance lists at various SCM workshops from 1988 onwards.

5 Middleware Technology

Investigation into the research origins of successful middleware technology (circa 2007).

Lead Authors were Professors Wolfgang Emmerich, Mikio Aoyama and Joe Svitek.

5.1 Is there a practice?

Gartner measured the market for middleware licences: in 2006, IBM alone earned over \$3bn from middleware licensing.

5.2 Application of the historical method

Very exhaustive – from minutes of standardisation committees to interviews.

Impact Trace Graphs were devised that traced citations, document references, technology uses etc. from products to standards, minutes, papers, articles, PhD theses and prototypes. There was a huge amount of material to process and some of the graphs became extremely complex.

5.3 Key findings

5.3.1 Technical development is interdisciplinary

Impacts often cross computer-science disciplines. E.g. for middleware: software engineering, networking, programming languages, distributed systems and databases. Often the impact is felt more in another field than where the technology originated.

5.3.2 Technology takes time to mature

Typically, commercial exploitation requires 15 to 20 years.

5.3.3 Technology transfer needs commitment

Often, transfer is the result of people movements. There were very many specific examples of this (e.g. Nelson from CMU to PARC to DEC Research, where he wrote the influential Network Object paper with A. Birrel that eventually led to Java RPC).

5.3.4 Postgraduate Students are critical sources

Investment in PhD research is a very important mechanism for generating commercially valuable ideas.

What's the "success rate" for PhD students in terms of the proportion of them that will end up being influential on software engineering practice? Probably very low. There is no predictive model that lets us select which students will produce the influential ideas. Compare artists – the majority will not be hanging paintings in the National Gallery, yet the 90%+ who do not are still economically valuable in whatever walk of life they choose.

The time scale is also significant in two ways – either you accept that it takes 15 to 20 years to find out the answer, or you find a way to reduce it to a shorter time.

Another interesting question might be to try to predict which areas of research might bear fruit in the future. Alex finds that this is not usually possible. Reid (CMU, 1981) insisted in the face of opposition from his tutors on working on Scribe – which was groundbreaking and, even though it is no longer used widely, was very influential on the development of mark-up languages.

In many cases also, something that is very important at the time it is being done turns out to be irrelevant 15 years on. That doesn't make the research any less valuable within its original context.

The "best" idea (defined how?) doesn't always necessarily succeed – see Betamax.

6 Impact Reports Roadmap

Two papers have been published so far, the Middleware technology paper is due in 2008, and at least four more are planned.

7 Conclusions and “Meta Findings”

There is very little surprising in the findings:

1. Lasting impact comes from repeated and sustained interactions. What kind? Workshops are one important vector. Governments have a fetish about academic/industrial partnering on research projects – these are marriages of convenience and they do not work in terms of repeated and sustained interactions
2. Interplay can be difficult to determine precisely and communicate clearly. The middleware report is more successful in this regard
3. More benefit comes from nurturing many and varied ways of interacting than from a single approach. Making products of research freely available, people movement, symposia and workshops, collaborative research...
4. Communities need support to maintain a nurturing environment
5. Broad-based community funding may be more effective than targeted transitions – because personal relationships and people movement worked best in transferring ideas

8 Further Information

<http://www.acm.org/sigsoft/impact>