Does Software Engineering Research have Impact on Software Engineering Practice?

A Brief Introduction to the Impact Project

Alexander L. Wolf
Imperial College London
Assumptions

There is a software engineering practice
- engineers apply techniques, use tools, and follow processes

There is a body of work that can be identified as software engineering research
- both academic and industrial

There is skepticism about the impact of research on practice
- witness: the relatively low level of academic and industrial research funding vis-à-vis the increasingly critical role played by the practice
From where does technology come?

Facile answers are misleading

It comes from...
- Sun, Microsoft, IBM, Google, “the Web”, ...

Yes, but!
- from where did they get it? and how?

It comes from...
- Dr. X, who published a seminal research paper and produced a popular prototype

Yes, but!
- we didn’t read the paper
- it was only a prototype, not engineered/licensed for industrial use

It comes from...
- being “in the air”, everybody knows it

Yes, but!
- how did it get there? who nurtured it?
Ideas have many parents...

Researchers
Scientific and technical communities
Technology transfer agents
Students with new degrees
New hires with different perspectives
Early adopters
Commercializers
... and their contributions differ

Initial conceptualization of idea
Evangelism
Prototype demonstration
Public promulgation
Nurturing by community activities
Education, training, and indoctrination
Product commercialization
Why should we care?

Some technology is not very good
- why are we stuck with it?
- why is it not better?

Some technology seems useful
- how can we get more of it?
- how can we speed its appearance?
- are there approaches that need to be strengthened/nurtured, despite their slow adoption?

Facile answers are misleading
How do we evaluate the contributions?

Facile answers are misleading

Ideally
- qualitatively
- quantitatively

But a challenging task
- different parties have different motivations
- apportioning contributions is difficult
- long timescales attenuate measurements and memories
CiP: Pressure to measure in the UK

UK Government is seeking accountability
- demonstrating economic impact of tax-payer investment in basic research and improving exploitation of research outputs

2006 DTI report on “Increasing the Economic Impact of the Research Councils”†
- provides several recommendations on how to consider economic impact in funding decisions
- example: an individual competent in the economic impact of research should be accommodated on each review panel

But measuring is easier said than done

2006 EPSRC report on “International Review of ICT Research in the UK”†
- panel noted the difficulty in conducting macro-economic analysis of ICT commercial impact

2007 Russell Group response to 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.”

†http://epsrc.ac.uk/ResearchFunding/Programmes/ICT/ReviewsAndConsultations/InternationalReview/
Goals of the Impact Project

Scholarly, objective, case-based evaluation

Deliverables
- peer-reviewed papers
- presentation materials and outreach activities
- expertise

Community building

Prospective for future research investment

Lessons learned for “successful” research
- but only with respect to transfer into practice
  (there are other measures of research success)
Administration

An initiative of ACM SIGSOFT

- volunteers mostly pay their own way
- modest funding from US NSF, UK IEE, and various agencies in Italy, UK, Germany, and Japan

International executive committee

Leon Osterweil  Carlo Ghezzi  Jeff Kramer  Alexander Wolf
Method

Form teams around practices
- important and widespread
  » examples: configuration management, programming languages, middleware, assertions, walkthroughs, …
- recruit volunteers, including researchers (academic and industrial) and practitioners

Start from practice and trace backward

Use accepted historical tools
- qualify conclusions by solidity of evidence; use references, oral histories, …

Michael Mahoney
Professor of (Science) History
A quick tour of two completed reports

Software configuration management
- first report to emerge from the project
- ACM TOSEM, October 2005

Middleware technology
- latest report to emerge from the project
- ACM TOSEM, to appear October 2008
Software configuration management

Investigation into the research origins of successful SCM vendor products (ca. 2003)

- version control
- product models
- change control
- composition/selection
- build management
- workspace management

Lead authors

Jacky Estublier  David Leblang

Other team members

- G. Clemm, IBM (ClearCase)
- R. Conradi, U. Trondheim
- A. van der Hoek, U. California
- W. Tichy, U. Karlsruhe (RCS)
- D. Wiborg-Weber, Telelogic (Continuus)
Is there a practice?

Practice measured by sales of vendor products

  - 25% mainframe
  - 15%-20% workstations
  - 5%-10% PC

BTW: this ignores the “sales” of freeware and shareware

- examples: CVS, Subversion
Historical method

1. Examine characteristics/features of leading products in SCM market
2. Assume products are used in practice
3. Trace product characteristics/features back to research ideas and prototypes
4. Make arguments for/against influence of research on practice via products
## When was it introduced?

<table>
<thead>
<tr>
<th>Academic Research</th>
<th>Industrial Research</th>
<th>Industrial Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1972</td>
<td>SCCS (Bell Labs)</td>
<td></td>
</tr>
<tr>
<td>1976</td>
<td>Diff (Bell Labs)</td>
<td></td>
</tr>
<tr>
<td>1977</td>
<td>Make (Bell Labs)</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>Variants, RCS (Purdue)</td>
<td>Change-sets (Xerox Parc)</td>
</tr>
<tr>
<td>1982</td>
<td>Merging, and/or graph (Purdue)</td>
<td></td>
</tr>
<tr>
<td>1983</td>
<td></td>
<td>Change-sets (Aide-de-Camp)</td>
</tr>
<tr>
<td>1984</td>
<td>Selection (Grenoble)</td>
<td></td>
</tr>
<tr>
<td>1985</td>
<td></td>
<td>System model (DSEE)</td>
</tr>
<tr>
<td>1988</td>
<td>Process support (Grenoble)</td>
<td>NSE Workspaces (Carnegie Mellon, Sun)</td>
</tr>
<tr>
<td>1990</td>
<td></td>
<td>nDFS file system (Bell Labs)</td>
</tr>
<tr>
<td>1994</td>
<td>Virtual file system and MultiSite (ClearCase)</td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>Activities (Asgard, Bellcore)</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>WebDAV (California, Microsoft, ClearCase, …)</td>
<td></td>
</tr>
</tbody>
</table>
An argument: research/product timing

Research “initiative” was shared between academia and industry. Some research tools were seriously used in practice:
- Make, RCS, Odin, Adele, ...

Research prototype
Prototype/product
Practical product

ClearCase Continuus

NUCM
Proteus Vesta
Dacs ICE
Asgard

CVS

EPOS

Adele

SCCS

Make

Odin Voodoo
Jasmine
RCS

PVCS CCC/Harvest
NSE
DSEE

Progress
Time
An argument: professional interaction

SCM research community organized regular workshops beginning in 1988

Product architects were present at all
- Cagan, Clemm, Dart, Leblang, Wiborg-Weber, ...
- Some presented work, while others simply attended and participated in discussion

The meetings put ideas “in the air”, and helped to keep them there
The role of creativity: vendor’s view

Vendors tend to consider that research impact is restricted to...

*algorithms* (e.g., differencing)

*pieces of reusable code* (e.g., RCS)

and not...

*concepts* (e.g., hierarchical workspaces)

*architectures* (e.g., peer-to-peer repositories)

which are often seen as “engineering common sense”
The role of creativity: researcher’s view

Researchers tend to consider that...

precedence
concepts
prototypes
are sufficient as impact and ignore...
efficiency
usability
reliability
dismissing them as “engineering common sense”
Both are right and both are wrong

A good idea is had more than once

Vendors have disincentives for distributing credit for ideas

Researchers have incentives for claiming credit for ideas

Research and productization both require engineering creativity
Middleware technology

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

- web services
- application servers
- transaction monitors
- distributed object systems
- message queues
- remote procedure call systems

Lead authors

Wolfgang Emmerich  Mikio Aoyama  Joe Sventek
Is there a practice?

Middleware License Market in 2005 [Gartner 2006]
Historical method

1. Seek sources
   - market analysis reports
   - professional articles
   - technical reports
   - standards documents
   - minutes of standards meetings
   - people movement
   - PhD theses
   - software
   - interviews

2. Derive “impact trace graph”
Trace: Simple Object Access Protocol

- [IBM 2004] WebSphere
- [Microsoft 2004] BizTalkServer
- [Apache 2004] Axis
- [BEA 2004] WebLogicServer

[Box et al, 2001] Soap History

- [Gudgin et al, 2003] SOAP 1.2
- [Box, 2001] SOAP 1.1

[OMG, 1995] CDR, IIOP, & GIOP

[SUN, 1988] XDR & ONC

[ISO, 1986] SGML

[Bray, 1998] XML

[OpenGroup, 1995] DCE & NDR

[Winer, 1999] XML RPC


[Reid, 1981] Scribe

[Reid, 1981] Scribe

[Rigorous SW Engineering]

[Bray, 1998] XML

[ISO, 1986] SGML

[OpenGroup, 1995] DCE & NDR

[Winer, 1999] XML RPC


[Reid, 1981] Scribe

[Rigorous SW Engineering]
Trace: Messaging in App. Servers
Trace: Dist. Objects in App. Servers

[Sun, 2003] J2SE 1.3 RMI
[Waldo 1998] RPC and RMI
[Wollrath et al 1996] RMI
[Birrel et al 1993] Network Objects
[OMG, 1995] CORBA 2.0
[OMG, 1991] CORBA 1.0
[Microsoft, 1995] DCOM 1.0
[Bal 1989] Shared Objects
[Bal et al 1988] Orca
[Hutchinson 1988] Emerald
[Dixon et al 1989] Arjuna
[Jul et al 1988] Emerald
[Black et al 1987] Emerald

[Birrel&Nelson 1984] Implementing RPC
[Liskov 1988] Arden
[Shapiro et al 1985] SOS
[Almes et al 1985] Eden
[Almes et al 1985] Eden
[OMG, 1995] CORBA 1.0
[Microsoft, 1995] DCOM 1.0
[Sun, 2003] J2SE 1.3 RMI
[Waldo 1998] RPC and RMI
[Wollrath et al 1996] RMI
[Birrel et al 1993] Network Objects
[OMG, 1995] CORBA 2.0
[OMG, 1991] CORBA 1.0
[Bal 1989] Shared Objects
[Bal et al 1988] Orca
[Hutchinson 1988] Emerald
[Dixon et al 1989] Arjuna
[Jul et al 1988] Emerald
[Black et al 1987] Emerald

Product
Standard
Article
PhD thesis
Prototype
M/W: Some key findings/confirmations

Technology development is interdisciplinary
- often winds back and forth among disciplines

Technology maturation needs time
- 15-20 years between first publication of an idea and widespread availability in products

Technology transfer needs commitment
- people movement is most successful vehicle

PhD students are critical sources of ideas
- almost all impact traces lead back to PhD theses

Standards are critical enablers of ideas
- without widespread agreements on ideas there is no widespread adoption
Tech. development is interdisciplinary

Impact traces often cross CS disciplines

For middleware...
- software engineering
- networking
- programming languages
- distributed systems
- databases

Impact sometimes larger in area other than first publication
- e.g., message queues

Example: RPC IDLs
- infom. hiding [CACM 15(5), 1972]
- MIL [IEEE TSE SE-2(2), 1976]
- Mesa [ICSE-4, 1977]
- Cedar RPCs [ACM ToCS 2(1), 1984]
- Sun RPC [IETF RFC 1057, 1987]

Example: dist. transactions
- operating systems (Gray, 1976)
- nested transactions (Moss, 1981)
- concur. control (Bernstein, 1987)
- Arjuna (Dixon, 1989)
- OSF ODTP/XA (1991)
- CORBA CCS, OTS (1994)
- J2EE JTS, JTA (2001)
### Technology maturation needs time

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RPCs</strong></td>
<td><strong>Dist. Transactions</strong></td>
<td><strong>Dist. Objects/RMI</strong></td>
<td></td>
</tr>
<tr>
<td>- idea of module interconnection languages</td>
<td>- research on non-standard transactions</td>
<td>- basic research and prototypes (Argus, Eden, Emerald)</td>
<td></td>
</tr>
<tr>
<td>- research on RPC systems</td>
<td>- standardization by IETF and OSF</td>
<td>- consolidation as “network objects”</td>
<td></td>
</tr>
<tr>
<td>- release of RPC into Apollo and Sun OSs</td>
<td>- widespread use in application servers</td>
<td>- standardization through JCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- widespread use in Java and .NET</td>
</tr>
</tbody>
</table>
Technology transfer needs commitment

B. Nelson from CMU to Xerox PARC
  - wrote definitive paper on RPCs with A. Birrel

B. Nelson and A. Birrel to DEC Research
  - wrote Network Object paper providing the basis for Java RMI

A. Watson from APM to OMG
  - lead CORBA standardization

A. Herbert from Cambridge to APM
  - devised ANSA

G. Dixon from NCL to Transarc
  - wrote OMG CORBA OTS and CCS service specs

J. Waldo from UMass, J. Sventek from APM to HP
  - wrote CORBA 1.0 spec

J. Waldo from HP to Sun
  - wrote RMI spec
PhD students are critical sources

Remote procedure calls
- architecture and failure semantics: Nelson (CMU 1981)
- orphan detection: Panzieri (Newcastle 1985)

Distributed transactions
- nested transactions: Moss (MIT 1981)
- object transactions: Dixon (Newcastle 1987)

Distributed object models
- general object models: Snyder (MIT 1978)
- RMI object model: Hutchinson (UW 1987) and Bal (Vrije 1989)

Web services
- Scribe: Reid (CMU 1981)
Impact reports roadmap

Modern programming languages
Middleware technology
Software architecture
Reviews and walkthroughs
Runtime assertion checking
Software testing

Software configuration management
Preliminary project “meta” findings

1. SE research has had impact on SE practice
2. Lasting impact comes most readily from repeated and sustained interactions
3. Interplay can be difficult to determine precisely and communicate clearly
4. Substantially different mechanisms have been successful at causing impact
5. More benefit from nurturing many and varied ways than single approach to impact
6. Community needs support to maintain the nurturing environment
Further information and reading

Impact web site
- http://www.acm.org/sigsoft/impact/
- contains links to published reports on
  » software configuration management (ACM TOSEM)
  » modern programming languages (ACM TOSEM)
  » middleware technology (ACM TOSEM)
  » run-time assertion checking (ACM SIGSOFT SEN)

Overview article