Bridging the Gap between Academia and Industry

Stevens Award Lecture
by Harry M. Sneed
at the CSMR-2009
Kaiserslautern, Germany
March 25th, 2009

Structure of the Lecture

- 1) The current status of cooperation between Academia and Industry
- 2) The reasons behind the gap between Academia and Industry
- 3) The consequences of the gap
- 4) What can be done to close the gap
- 5) Suggestions for future cooperation between Academia and Industry

CSMR2009-2

Current Status of Cooperation between Academia and Industry

As seen from the side of industry:

- Industrial Users are driven by their vendors and the IT media (newspapers and magazines)
- Industrial participation in academic conferences has been declining for years (ICSE, ICSM and CSMR are good examples of this).
- Industry feels that there is little it can learn from academia especially in the fields of maintenance and reengineering.
- Industrial IT personnel do not access academic journals.
- Industrial IT users are not aware of what academics have to offer
- IT managers distrust academicians, they believe they only want to experiment at their expense.

CSMR2009-3

Current Status of Cooperation between Academia and Industry

As seen from the side of Academia

- Academics are driven by their conferences and technical journals and their need to publish.
- Academics seldom attend industrial conferences as they feel this is below their standard.
- Academics look down upon industrial newspapers and magazines.
- Academics are not aware of the problems and constraints of industry.
- Academics distrust IT managers, who they believe only want to exploit them to achieve their business goals.

Counter Examples
Exceptions confirm the Rules

- Where industry and academia work best together is where they are one.
- This happens when academics found industrial companies to exploit their ideas. Google is a good example of this.
- It also happens when industry finances laboratories in an academic environment. The UBSLab at the ETH Zürich is a good example of that.
- There are institutions which combine both sides like the Frauenhofer Institute in Kaiserslautern and Maryland.
- The TU-Munich is making a concentrated effort to get involved in industrial projects in maintenance, test and reengineering.
- Unfortunately, most of the current software industry leaders grew up without academic involvement (IBM, MicroSoft, Oracle, SAP). Later they tried to compensate that by installing their own research labs, but the spirit of these organizations is basically anti-academic.

Reasons behind the Gap between Academia and Industry

- Academics and Industrialists have a different mind set, they are living in different worlds.
- Academics and Industrialists are pursuing different goals. The Academic is striving for recognition from his peers. The Industrialist is striving to survive.
- Industry thinks in terms of short range goals, Academia has a long range perspective.
- Industry prefers proven solutions with a low risk, Academia is interested in creating new solutions with a high innovation rate.
- Industry seeks for a minimum solution to minimize their risk whereas academia strives for a maximum solution to maximize their recognition.
- Industry is mainly concerned with costs. Academia could care less about costs, it is mainly interested in the benefits.

Reverse & Reengineering as a good Example of the Gap between Academia and Industry

- Industry has to maintain a huge amount of legacy software (over 75% of all software is legacy software). It is not particularly concerned with the quality of that software. It is mainly concerned with finding personnel to maintain it and with migrating it to new platforms.
- Academics are concerned with improving the quality of legacy software, i.e. in reengineering rather than migration. Part of that effort to improve the quality of legacy systems is reverse engineering, a term confusing to industrial users, who don‘t perceive what it is good for. Contrary to the academic belief, reverse engineering is not a prerequisite to transforming or wrapping code, since this is almost always an automated process.

Why industry rejects Reverse & Reengineering Technology

- Industrial users are unwilling to buy into reengineering, i.e. raising the quality of their existing software, without knowing before hand what savings that will bring them. Academia is not able to guarantee them that.
- Industrial users are not willing to finance a reengineering project without knowing beforehand what the ROI will be. Academics are not able to calculate a ROI because they do not have the data to quantify the benefits.
- As a result, it is difficult to sell reengineering and reverse-engineering projects. These activities can only be carried out within the scope of the maintenance budget which is hardly enough to cover the changes and corrections. Reverse and re-engineering are, in contrast to maintenance and migration, often neglected activities in industry, whereas in Academia, they receive a lot of attention.

What Industry is looking for concerning their legacy systems

- There is a need for browsers to search through existing code to find dependencies.
- There is a need for impact analysis to identify what has to be changed.
- There is a need for metrics to estimate the costs of change.
- There is a need for metrics to identify error prone components.
- There is a need for tools to automatically convert old languages into new ones while preserving the functionality of the system.
- There is a need for tools to transform data from one database to another.
- There is a need for tools to wrap existing components, data and procedures for reuse in other environments.
- The success of a migration project – transformation or wrapping – is easily measured. Either it works in the new environment or it doesn't. This is not true of a reverse or reengineering project, where the goal is a quality improvement. There must be a way to measure quality improvement.

On the Discrepancy between Supply and Demand in Software Maintenance

- It is obvious that the academic community spends too much effort on subjects that are not relevant to industry and too little effort on relevant subjects. Thus, the offerings of academia do not match to the demands of industry. To avoid this mismatch, academia should study the requirements of industry.
- On the other hand, industry is often not aware of what they really need. Industry must be educated as to the influence that complexity and quality have on the costs of maintenance and migration. They must also learn what is possible and what is not. It would be the task of academia to educate industry.

Consequences of the Gap

- Academia and Industry tend to drift apart
- Academics live in their own world, they first invent solutions, then they search for a problem that fits to it. They are reluctant to deal with real world problems, since these are usually wicked. As a consequence much of their effort is wasted on toy problems.
- Academia is not teaching students what they need to know about maintaining and evolving existing systems.
- Industry users let themselves be manipulated by scruples venders who like drug dealers are bent on making their customers dependent. Their inability to decide for themselves is due to their lack of education and their lack of trust in the academic community.
- Industry suffers from a lack of awareness as to how they could approach their problems. They do not know what tools and techniques from academia could help them. They also suffer from a lack of standardization and the ability to objectively compare alternate solutions.

Closing the Gap between Industry and Academia

Measures for closing the gap between Academia and Industry

- Academicians should take part in industrial projects
- Professors should be given sabbaticals to work as maintenance programmers
- Practitioners should attend courses at the universities to refresh their knowledge
- There should be common conferences to appeal to both sides in which there are two different standards for selecting papers
- According to a poll by the IEEE Software, both sides seek a stronger collaboration, however for that we need a better mutual understanding.
- There needs to be a common language, i.e. set of terms
- There needs to be a standard set of metrics to measure software systems and projects.

Towards a Common Language

Steps toward a better mutual understanding

- The language used by academicians should be understandable to practitioners
- Reengineering & Reverse Engineering are good examples
- Reverse Engineering means to a practitioner post documenting his system as opposed to pre documenting it.
- Reengineering could mean many things – renovating, restructuring, refactoring, etc. One should be specific.
- Migration needs to be kept apart from reengineering.
- It would be better not to use the terms reengineering and reverse engineering at all since they are confusing. They have been misused to denote too many quite different activities, like business reengineering, business process reengineering architectural reengineering and code reengineering.

Teaching Software Maintenance

- Software maintenance should be taught on real software systems from industry
- Students have to correct errors in both procedural and object-oriented programs
- Students have to make changes to both procedural and object-oriented programs
- Students have to redocument both procedural and object-oriented systems
- Students have to restructure procedural programs and refactor OO-components
- Students have estimate the costs of maintenance tasks
- Students have to design problem reports and change requests
- Students have to design a maintenance process for each type of maintenance task

Teaching Software Migration

- Software Migration should be taught on a real legacy software application
- Students should measure the legacy application and calculate the costs of alternate migration strategies
- Students should have to transform legacy procedural code to a modern OO-language
- Students should have to wrap legacy modules and convert them over into web services
- Students should have to convert old master files over into relational databases
- Students should have to transfer old data interface files over into XML files
- Students should have to perform a regression test upon the migrated system

Expressing Software with Numbers

The role of measurement in increasing mutual understanding

- We should strive to express software in terms of numbers understandable to both academicians and practitioners
- There needs to be a common understanding of what is size, what is complexity and what is quality in software.
- There needs to be a metric standard for measuring quality, complexity and productivity
- Software systems, developed in different modes and languages have to be comparable.
- There should be estimation techniques aimed directly at estimating maintenance and migration projects.

Suggestions for future Cooperation

- Conduct common conferences with a double standard for paper acceptance.
- Firms should invite local universities to participate in industrial projects.
- Universities should offer continual education programs on evenings and weekends.
- Academia and industry should strive to find a common vocabulary by means of standardization.
- Academia should offer industry a simple set of metrics with which they can measure their quality, complexity and productivity.