Understanding the University/Industry Gap

Jorge L. Díaz-Herrera†, Mark Ardis†, Deborah Coleman†, and Maria Makabenta-Ikeda††
† B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology, Rochester, N.Y., USA.
†† School of Applied Information Technology
Kyoto College of Graduate Studies for Informatics, Kyoto, Japan.

Abstract

In this presentation we discuss a project to study best practices in industry and compare these with curricula recommendations for software engineering programs. We have designed a survey on software development practices and methods to be administered as widely as possible within the US and in other countries. This would pave the way to gain a better understanding of the perceived gap between industry and academia. We would like to present this idea to the workshop and to solicit feedback on the survey design; we would also be seeking additional contributors.

1. Introduction

Software engineering is an emerging discipline with an expanding body of both theoretical and practical knowledge that aims to provide means to produce reliable software products that predictably meet intended requirements, including schedule and budget. When the term Software Engineering (SE) was introduced in the late 1960s, it "was deliberately chosen as being provocative, in implying the needs for software manufacture to be based on the types of theoretical foundation and practical disciplines, that are traditional in the established branches of engineering." [1] The term is now widely used in industry, government and academia: Thousands of computing professionals go by the title software engineer; numerous publications, groups and organizations, and professional conferences use the term software engineering in their names; and there are many educational courses and programs on software engineering. However, SE is not widely considered an engineering discipline and the vast majority of software engineers are not registered or licensed engineers.

The question becomes whether the practice of SE and the education of its practitioners is adequate from the point of view of engineering education and the practice of engineering. What does a software developer do that can be considered engineering practice? In terms of engineering education, a typical engineer today is a college graduate who is familiar with mathematics and the physical sciences, in addition to the engineering design and problem solving approaches particular to the corresponding branch of engineering. How does this compare to what software engineers are being taught in today's degree programs?

In this paper we present our intentions to conduct a survey among software practitioners in various countries and organizational settings to study best practices in industry and compare these with curricula recommendations for software engineering programs. Our goal is to gain a better understanding of the perceived gap between industry and academia. We would like to present this idea to the workshop and to solicit feedback on the survey design.

The rest of the paper is organized as follows:

- Motivation and background
- A description of the survey
- Summary and appendix

2. Motivation and Background

Our project started when one of the authors spent time in Japan in the summer of 2005 teaching an advanced software engineering seminar, and inspired by Cusumano's analysis [2], began to study the differences between the Japanese software industry and that of the US [3].

Japanese software firms have performed well in some areas, and it is clear that the quality of the software they produce is very high, even for Japan's low CMM-rated companies [4], and that ad-hoc software development is not the norm. Over the years, there have been several attempts to analyze the software practice in Japan and to draw comparisons with standard US practices [5, 6, 7, 8, 9, 10]. Such analyses are based on comparing the processes employed, the product standards in place, the quality of the software generated, the business approach, etc. Analyses have been made based on completed software

projects [11], but to our knowledge, a comprehensive survey has not been carried out [12].

- What can we learn from this dichotomy and other relevant aspects of the software industry?
- Does the same apply to other countries and organizations?

Comparisons across countries, organizations, and even across vertical markets can provide useful information in understanding the perceived gap between the state-of-the practice and the state-of-the-art in software engineering.

We propose to administer a comprehensive survey of software practices and methods covering as many aspects of software development and management as possible. A major goal of our project is to collect evidence as to whether those practices and methods represent the current state-of-the-practice.

We based our survey on the descriptions of practices and methods found in the latest version of SwEBoK [13]. Given the influence of SwEBoK on software engineering educational curricula, we wish to ensure that it is as accurate as possible.

The next step is to have some volunteers take the survey and give their feedback about whether the questions are good (i.e. describe the practices of a software developer), and whether the set of questions, taken as a whole, are reasonable.

3. The Survey

The survey on software development practices tries to identify whether "best practices," as defined by SwEBoK, are in use and the level of familiarity software professionals have with specific software methods –in a given organization, country or other comparative groups. Table 1 lists the main categories of questions in the survey, while the complete survey can be found in the appendix.

Practices are types of activities performed by the software engineer/developer as part of their software practitioner role in the organization. For example, design practices include ad-hoc, text-only, graphical design artifacts, etc.

Methods refer to specific approaches to carry out a practice activity. For example, data flow diagramming, class modeling, etc., are methods for design practices. Methods can also range from overarching approaches to very specific techniques as for example, boundary value analysis for testing, and formal specification (using a given technique such as Z) for requirements practices.

Topic area	No. Questions
Professional information (Job title, organization type, etc.)	11
Requirements Practices	6
Design Practices	9
Analysis and Design Methods	30
Construction Practices	9
Test Practices	20
Test Methods	17
Maintenance Practices	6
Configuration Management Practices	6
Software Quality Practices	7
Software Process Practices	10
Software Management Practices	10
Other	2
Total	147

Table 1: Survey categories

4. Summary

A body of core knowledge exists for SE curricula and is being addressed in the several volumes of computing curricula. There is no question that the definition of a professional engineer applies to software engineering. There still is, however, some confusion about what the field is and how it relates to other engineering practices. Particularly lacking is a widespread recognition of software professionals as professional engineers. This is currently the subject of a fair amount of discussion among the various professional groups involved, although certain legislating bodies have proceeded with registration and certification of software professionals.

Approaches to certify individuals vary from country to country. To establish a basis for certification at the international level, the IEEE Computer Society has made available, worldwide, the Certified Software Development Professional (CSDP) designation [14].

The CSDP professional certification has three critical components:

1. Exam-based testing demonstrating mastery of a Body of Knowledge (BOK). The CSDP examination, based on SwEBoK and SE2004, and covers fifteen knowledge domains (see table 2).
2. Extensive experience base in the performance of the work or profession being certified. To achieve this designation you have to have 9000 hours of software engineering work experience.
3. Continuing professional education, measured and relevant to the BOK. To achieve this designation you have to have a university degree.

We intend to find out how widespread are the fundamentals of software engineering in the practice of software development across countries, organizations, vertical markets, etc.

Knowledge
I. Software Requirements
II. Software Design
III. Software Construction
IV. Software Testing
V. Software Maintenance
VI. Software Configuration Management
VII. Software Engineering Management
VIII. Software Engineering Process
IX. Software Engineering Methods
X. Software Quality
XI. Software Engineering Professional Practice
XII. Software Engineering Economics
XIII. Computing Foundations
XIV. Mathematical Foundations
XV. Engineering Foundations

Table 2: CSDP Exam Specification

5. References

[1] Naur, P. and Randell, B. (Eds.) Software Engineering: Report on a Conference Sponsored by the NATO Science Committee (7 – 11 October 1968). 1969.

[2] Cusumano, M. The Japanese Software Puzzle." Communications of the ACM, vol. 48, no. 7, 2005:25-27

[3] Díaz-Herrera, J. L., Kashihara, H., Makabena-Ikeda, M., and Nakaguchi, T. "The Japanese Software Puzzle Revisited."Nippon Association for Information System (NAIS) Journal, Vol 3, 2008 (forthcoming).

[4] Matsubara, T. "Japan: A Huge IT Consumption Market," IEEE Software, Sept/Oct 2001: 77-80.

[5] Humphrey, W. S., Kitson, D. H., and Gale, J. "A Comparison of US and Japanese Software Process Maturity." Proceedings of the 13th International Conference on Software Engineering, pages 38-49, 1991.

[6] Yasuda, K. and Yamada, S. "The Concept and Practice of Software Quality Assurance in Japan in the Global Era." Proceedings of the IEEE 7th International Symposium on High Assurance Systems Engineering (HASE'02). 2002.

[7] Duvall, L. M. "A Study of Software Management: The State of Practice in the United States and Japan," Journal of Systems Software, Vol. 31, No. 2, 1995:109-124.

[8] Katsuro, I. and Mikio, A. "A Status Report on Research and Development of Software Process in Japan," Software Process Newsletter, No. 1, 1994.

[9] Tajima, D. and Matsubara, T. "Inside the Japanese Software Industry," IEEE Computer, March 1984:34-43.

[10] Isoda, I. and Saeki, M. "Software Engineering in Asia," IEEE Software, November, 1994: 63-68.

[11] Glass, R. L. "The State of the Practice of Software Engineering." Special issue, IEEE Software, Nov/Dec 2003.

[12] Glass, R. L. "One Man's Quest for the State of Software Engineering's Practice." Communications of the ACM, Vol. 50, No. 5. May 2007:21-23.

[13] Bourgue, P. and Dupuis, R. (Ed.s) Guide to the Software Engineering Body of Knowledge: 2004 Version, PIEEE Computer Society, April 2005, http://www.swebok.org.

[14] "Certified Software Development Professional." IEEE Computer Society, Los Vaqueros, CA. http://computer.org/certification/

Appendix: Survey (to be distributed at the workshop)

Clipboard R·I·T Online

Survey Listing :: **Preview Survey**

Software Development Practices

Printer-Friendly Version

Instructions:

Thus survey covers a broad collection of software development practices. Do not worry if some of them seem unfamiliar. Please just answer as many questions as you can.

1. **1. Which of the following best describes your job title?**

- Business Analyst
- Database Administrator
- Data Analyst
- Interface Designer
- Network Engineer
- Programmer
- Programmer/Analyst
- Project Leader
- QA/Software Test Engineer/Analyst
- Requirements Analyst
- Requirements Engineer
- Software Architect
- Software Developer
- Software Engineer
- Systems Programmer
- Systems Analyst
- Technical Support Analyst
- Webmaster
- Web/Internet Developer
- General IT Staff (multiple responsibilities)
- Chief Architect
- Chief Information Officer
- Chief Scientist
- Chief Technology Officer
- Director