

Software Testing Education and Training in Hong Kong^{1 2}

F. T. Chan³, T. H. Tse

The University of Hong Kong, Pokfulam, Hong Kong
ftchan@hkuspace.hku.hk, thtse@cs.hku.hk

W. H. Tang

The Kowloon-Canton Railway Corporation, Hong Kong
dwhtang@krc.com

T. Y. Chen

Swinburne University of Technology, Australia
tychen@swin.edu.au

Abstract

While the use of computer applications is widely spread in every business and, hence, the reliability of software is critical, it is believed that many organizations involved in software development do not take software testing sufficiently seriously as an important task. It is worthwhile to find out how far organizations are carrying out software testing in a systematic and structured manner or still taking on an ad-hoc approach. A survey was conducted to understand the software testing practices and the level of related education and training in Hong Kong. It was found that most testing team members did not have formal training in software testing. University curricula generally did not prepare graduates with enough coverage in software testing. It is proposed that a review of the current software engineering curricula in the universities to examine the coverage of software testing will be useful to the development of quality software.

Keywords: *software testing, software engineering curriculum*

1. Introduction

With the widespread use of computer applications in almost every business, the reliability of software becomes critically important. It does not only affect the operation of a business but can have implications on human lives or the stability of society. While software testing [4, 6] is one of the most effective measures in assuring and improving the reliability of software, it is believed that not all software developers take a formal and systematic approach in carrying out this important software development task. Many organizations may still take an ad-hoc approach towards software testing and may not provide adequate resources to accomplish this task. A major reason is believed to be the lack of proper education in software testing in university software engineering programmes and the lack of proper professional training in software testing when these graduates join the workforce.

In a survey aimed at understanding the situation of software testing in the software development environment in Hong Kong, we attempted to understand the extent where software testers had received formal education and training in software testing. From these findings, the

¹ © 2005 IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Personal use of this material is permitted. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder. Permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

² This research is supported in part by a grant of the HKU SPACE Research Fund.

³ All correspondence should be addressed to Dr. F. T. Chan, School of Professional and Continuing Education, The University of Hong Kong, Pokfulam, Hong Kong. Tel. (+852) 2587 3288. Email: ftchan@hkuspace.hku.hk .

availability and popularity of software testing education in university curricula as well as software testing courses in professional development training were inferred. This article also reports some common testing practices that were adopted in Hong Kong.

2. Background of the survey

A survey questionnaire was designed to gather information from software testing practitioners and software project managers in the areas of software testing standards, practices, automated tools, and training and education. The survey was planned in various countries and cities in the Asia-Pacific region, including Australia [7], China, Hong Kong, Malaysia and Singapore. To facilitate consistency and data integrity of the responses to the questionnaire, face-to-face interviews were conducted with the respondents by two trained research assistants.

People with responsibilities for managing software development and/or software testing from organizations of different backgrounds were invited to take part in the survey interview. A total of 34 interviews were successfully conducted. The participating organizations are classified into government (14.7%), public non-commercial (17.6%), local commercial (50%) and overseas-based commercial (17.6%). The sectors included banking, finance and insurance (11.8%), education and training (11.8%), manufacturing and engineering (2.9%), research and development (8.8%), software house and IT consultancy (35.3%) and others (29.4%).

In terms of the sizes of the organizations, over half of the organizations in this survey (55.9%) employed more than 500 full-time staff and 11.8% had a workforce of 100 to 499 full-time staff. 2.9% employed 50 to 99 full-time staff and 5.9% had 20 to 49. The remaining 23.5% were smaller ones employing fewer than 20 full-time staff. Regarding the number of full-time IT staff, 20.6% employed 250 or more IT staff, 29.4% had 50 to 249, 11.8% had 25 to 49, 14.7% had 10 to 24 and 23.5% had less than 10.

3. Some basic findings on software testing practices

All the organizations in the survey had over 2 years of software development experience. Half of them had over 10 years of experience. The majority of the applications developed were either Windows-based or Internet-based. Slightly over half of the organizations (56%) went for complete in-house development. The rest had varying degrees of out-sourcing.

Before looking at the software testing education and training situation, it is worthwhile to have an overview on

the general environment of software testing practices in these organizations. The information allows for possible linkage between education and training on software testing with actual practices.

In these organizations, system specifications were mainly represented by semi-formal methods. Nearly all the organizations (97.1%) claimed to practise software testing management. The most experienced ones, having more than 20 years of experience, accounted for 23.5% of the organizations in the study. 14.7% had 11 to 19 years, 11.8% had 6 to 10 years, 38.2% had 2 to 5 years and a small proportion (8.8%) had less than 2 years. The weighted average experience of software testing management was over 9 years.

38% of the organizations appointed a designated person solely responsible for the management of software testing activities. The use of independent testing teams [4] with testers who were not directly involved in the development of the software systems was reported by 35.3% of the organizations. User Acceptance Test (UAT) was found to be a very common practice, with a majority of 76.5% of the organizations in the survey indicating that they would establish a formal UAT team for every software application developed.

Almost all organizations in this survey (97.1%) adopted one or more structured testing methodologies in software testing in the past 3 years. The methodologies used included static analysis, dynamic analysis, selection of test cases, mutation analysis, data flow analysis, symbolic analysis and program instrumentation. Most of the organizations (73.5%) had a high proportion (80–100%) of software development projects tested using one or more of these methodologies. Another 14.7% of the organizations used them in 60–79% of their projects. 78.2% of the software development projects were tested based on testing methodologies.

In the selection of test cases, the most commonly used methods were based on black-box testing [1, 6], also known as functional testing or specification-based testing. It was more popular than white-box testing [2, 6], also known as structural testing or code-based testing. 91.2% of the organizations reported using the former and only 35.3% were using the latter. Among the black-box techniques, contrary to popular belief, random testing (58.8%) was the most commonly used, followed by boundary value analysis (50%). Error guessing (35.3%), cause-effect graphing (27%) and equivalence partitioning (20.6%) were less common. Among the white-box techniques, branch coverage analysis was by far the most common one, used by 9 (75%) of the 12 organizations doing white-box testing. The slightly less common ones include predicate coverage analysis (50%) and statement coverage analysis (50%).

Over half of the organizations (55.9%) in this survey had a large proportion (80–100%) of test cases derived from specifications [8]. Those who had 60–79% test cases

derived from specifications accounted for 26.5% organizations. Only a few organizations (5.9%) had less than 20% specification-driven test cases. The weighted average percentage of test cases derived from specifications in this study was 73.5%.

58.8% of the organizations made use of some form of metrics to measure and record the effectiveness of testing. The majority of the organizations, accounting for 80% of those using software metrics, counted the number of defects detected. Other metrics used included defect detection rate (40%), defect density (10%) and defect repair rate (10%) [4].

26 organizations (76.5%) adopted some form of software testing standards. Seven of them (27%) used both published and in-house developed standards. Sixteen (62%) used only in-house developed standards and three (11%) used solely published standards.

Nine (35%) of the 26 organizations had their software development processes accredited. ISO 9001 [3] series was the most common accreditation standard, acquired by seven organizations. CMM [9] Level 2 accreditation was acquired by two organizations and CMM Level 5 by one. There was one organization acquiring both the ISO 9001 and CMM accreditations.

4. Analysis of education and training in software testing

For the 12 (35.3%) organizations that had independent testing teams, when asked about the proportion of testers who had completed formal training in software testing, they gave varying responses. 33.3% had less than 20% of their testing team members having formal training in software testing. Another 16.7% had 40–59% of trained testers, 8.3% had 60–79% and 33.3% had 80–100%. The weighted average percentage of testing team members who have completed formal training in these 12 organizations was only 47.5%. Although organizations without independent testing teams were not asked to respond, it was believed that the extent of training in software testing would even be lower. Among the 12 organizations that had an independent testing team, the majority (83%) provided allowances/assistance for relevant software testing staff to receive training. For those organizations that did not have independent testing teams (22 organizations), only 13 (59%) of them provided such support in training. The overall percentage of organizations that provided software testing training was 68%, comparable with the 72% in the Australian study [7].

A majority (91.3%) of the 23 organizations that provided allowances/assistance to staff in software testing training recognized the value of formal training in improving staff productivity. They also pointed out higher morale (30.4%) and staff retention (26.1%) as the benefits

of providing formal training in software testing to their staff. Only 13% of these organizations used training courses provided by universities. The figure was comparable to the 10.7% reported in the Australian study [7].

Cost and time were the most significant factors that hindered the provision of training. 23 (67.6%) and 19 (56%) out of the total 34 organizations cited these factors, respectively. Sixteen (47.1%) organizations indicated the lack of appropriate courses as one of the barriers in providing training in software testing.

4.1. Software testing in university education

The study also revealed that most companies had only a small proportion of testing team members who had completed formal training in software testing during their university education. Over half of these organizations (53.8%) that had independent testing teams indicated a low percentage of testers (less than 20%) did so. 25% of the organizations reported having 40–59% and only one company reported a high percentage (80–100%) of their testers had completed formal training in software testing at universities. The weighted average percentage of testing team members who had received formal training in software testing from universities was 28%.

We note that, in a survey conducted by Lethbridge in 1998 [5], most software professionals indicated they were either “vaguely familiar” or only “learned the basics” about software testing, verification and quality assurance. Software metrics and process standards such as CMM and ISO 9000 were among the least learned topics in their formal education. The findings of the two surveys indicate a general inadequacy internationally of addressing software testing in the curricula of software engineering or related programmes in formal university education.

Percentage of testing team members having formal software testing training at universities	Frequency (%)
Less than 20%	7 (53.80%)
20 to 39%	0 (0.00%)
40 to 59%	3 (25.00%)
60 to 79%	0 (0.00%)
80 to 100%	1 (8.30%)
N/A	1 (8.30%)

Table 1: Percentage of testing team members having formal software testing training at universities.

4.2. In-service software testing training

Out of the 23 (67.6%) surveyed organizations that provided allowances/assistance for relevant staff to receive training in software testing, external training courses offered by commercial organizations and internal training courses were the most popular sources of training, amounting to 65% and 61% of these organizations, respectively. Self-study was also a common form of training, reported by 39% of the organizations. Higher diploma courses of the vocational education institutes and short courses offered by universities were less common. These institutions were used by 17% and 13% of the organizations, respectively.

4.3. Extent of software testing training in relation to organization type and size

Cross tabulations of data revealed that the provision of training was most prevalent in the banking, finance and insurance sector, with all the organizations in this sector providing training to their software testing staff. 75% of the organizations in the education and training sector offered software testing training to relevant staff while 50% of the organizations in the research and development sector and the software house and IT consultancy sector provided such training. Little training in software testing was reported by the manufacturing and engineering sector.

In terms of the organization size, smaller organizations sampled in this survey provided training to a lesser extent when compared to larger ones. Of all the organizations with fewer than 20 staff, only 37.5% provided software testing training. For those employing over 500 staff, 68.4% provided such training to their staff. It so happened that all organizations employing 20–499 staff in the survey provided software testing training to their staff.

In terms of the size of the IT team, the provision of training tended to be more prevalent as the size of the IT team increased. Among organizations with the smallest IT team size (fewer than 10 staff), 50% provided allowances/assistance to staff in software testing. This percentage increased to 60% for IT team sizes of 10–24 and 75% with those with 25–49 IT staff. Among organizations with 50–249 staff in their IT teams, 90% of them had provisions for training. However, this proportion dropped to just 57% among organizations with the largest IT team size (250 staff or more) in the survey.

5. Conclusion

From this survey, we find that less than half of the testing team members have received formal training in software testing. The provision of training support by the

organizations varied from one industry sector to another. All the surveyed organizations in the banking, finance and insurance sector provided software testing training to their staff. This is a good sign, reflecting that the sector pays proper seriousness towards the importance of software testing. The other sectors, in particular the software houses and IT consultancy companies, are very much lagging behind in their efforts to support staff training in software testing. It is important to note that education and training in software testing may not be adequately addressed in the formal education system. Most of the software testing education and training that IT staff received were through courses offered outside the universities and other tertiary institutions. While this appears to be an international trait, we must caution ourselves that it is not a reason for complacency in Hong Kong.

It is suggested that a review of the current software engineering curricula in the universities will be necessary to find out whether the coverage of software testing needs to be strengthened further. In the continuing and professional development area, software testing training courses had better be promoted through customized courses according to the needs of individual companies, as training courses tailor-made for individual organizations' needs were preferred over general programmes open to public audiences.

Acknowledgement

The authors are grateful to the assistance and support of the organizations and people who have participated in the survey.

References

- [1] B. Beizer. *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. Wiley, New York, NY, 1995.
- [2] L. Copeland. *A Practitioner's Guide to Software Test Design*. Artech House, Norwood, MA, 2004.
- [3] A. Gillies. *Software Quality: Theory and Management*. Thomson Computer Press, London, UK, 1997.
- [4] M. Hutcheson. *Software Testing Fundamentals: Methods and Metrics*. Wiley, Indianapolis, IN, 2003.
- [5] T. Lethbridge. What knowledge is important to a software professional? *IEEE Computer* 33 (5): 44–50, 2000.
- [6] G. Myers. *The Art of Software Testing*. Wiley, Hoboken, NJ, 2004.

- [7] S.P. Ng, T. Murname, K. Reed, D. Grant, and T.Y. Chen. A preliminary survey on software testing practices in Australia. In *Proceedings of the 2004 Australian Software Engineering Conference (ASWEC '04)*, pages 116–125. IEEE Computer Society, Los Alamitos, CA, 2004.
- [8] R.M. Poston, editor. *Automating Specification-Based Software Testing*. IEEE Computer Society, Los Alamitos, CA, 1996.
- [9] J. Raynus. *Software Process Improvement with CMM*. Artech House, Norwood, MA, 1999.