# Multiple-Choice Tests in teaching Software engineers: how applied and how applicable?

Tamara Matveeva
Faculty of Computing
London Metropolitan University

*"MCTs provoke so much debate and controversy because they happen to be the most common format of so-called "standardized tests."*
(Korsunsky, n.d., Part1)

## Introduction

Although computer-aided assessment has been developed since late 1980's, advantages and disadvantages of multiple-choice tests (MCTs), as opposed to traditional academic techniques (such as essays), are still greatly debated. This paper overviews current practice of application of MCTs in Computing and generally outlines their applicability in teaching Software Engineering, as a subject area in HE.

## MCTs in Computing

Software Engineering (SoftEng) is a relatively new area, closely linked to Computer Science, Systems Engineering, IT Management and some others that address various aspects of 'creating computer software'. In the last two decades a number of international standards were adopted, as well as Software Engineering Body of Knowledge (SWEBOK, 2004), and so SoftEng appears to be a discipline, mature enough to be taught academically.

It should be noted that academic teaching of SoftEng adopts much from older 'adjacent' disciplines, in particular, regarding the practice of assessment. Since this case study is focused on MCTs, the first step was to review the current practice and analyse the actual 'coverage' of MCT-based assessments in Computing. From several sets of MCTs, (such as BIT 2001-2006, Hambling *et al.,* 2007, Williams, 2005 and Collins, 2005), the following typology of questions has been constructed by the author of this paper as a tool to analyse the MCTs:

C1: to test the ability to recall, but not necessarily use or explain,

C2: to test the ability to apply a topic in a practical setting,

C3: to test the ability to solve problems/make decisions by classifying information, making evaluative comparison and/or judgement (and sometimes, to explain reasoning).

The C1-category covers, *per se*, testing of cognitive skills at the two lowest levels of the Bloom's taxonomy (Bloom, 1956), whereas C2 generally corresponds to Bloom's third level ('application'). C3 potentially allows covering other levels, even though questions to test 'highest' skills (synthesis and evaluation) are relatively rare. It is known that devising MCTs to test higher levels of intellectual skills is a difficult task. One practical approach is to write a question to pose with a 'typical' decision with some evidence upon which the testee can base the decision, interpreting the evidence and applying appropriate decision-making methodologies. (La Trobe, 2007).

In the context of devising summative assessment, it should be also noted that these are not levels of difficulty, as distracters make some C1-questions more difficult to answer than C2/C3 ones.

One of the recognised benefits of MCTs is that they are also highly appropriate for formative assessment, because personalised feedback can be provided immediately and personally. Figure 1 presents an example of a C3-question taken from Hambling *et al.* (2007). The question is to test the ability to identify correct combination of information. A set of feedback elements is designed to give personalised feedback for any wrong combination, addressing particular mistakes or misinterpretation.

MCTs are also very suitable for diagnostic purposes. In particular, a short MCT can be given when beginning a new topic to inform lecturers about students' knowledge and understanding of the area. Lecturers can then design relevant learning activities for students and engage them in learning at an appropriate level. An interesting example of a session is provided by (REAP, 2006), where the delivery of a topic is followed by an MCT which explores the students' conceptual understanding. The results are collected through electronic voting system and the resulting bar chart is displayed showing the distribution of the student answers. If there is a disparity, students are asked to discuss the question and 'to convince the group that you have the right answer'. The peer-to-peer discussion lets students explore their own thinking and reasoning behind their answer and to reinforce their understanding of addressed concepts. Then students are asked to vote again, either on the same question or on a slightly different question on the same concepts.

*Which of the following are the <u>most</u> important factors to be taken into account when selecting test techniques?*

(i) Tools available.
(ii) Regulatory standards.
(iii) Experience of the development team.
(iv) Knowledge of the test team.
(v) The need to maintain levels of capability in each technique.

    a. (i) and (ii)
    b. (ii) and (iv)
    c. (iii) and (iv)
    d. (i) and (v)

<u>Feedback elements</u>:
Answer (i) looks tempting, and the availability of tools might make the use of techniques more or less attractive, but it would NOT be decisive in the way that regulatory standards and tester knowledge are.

Answer (v) is a factor in managing the test team, and experience would need to be maintained, but it should NOT influence the selection of technique.

*Figure 1. MCT with two examples of feedback elements.*

Also, recent publications present a number of refined MCT-techniques of diagnostic assessment, for example, Ordered Multiple-Choice Items (Briggs et al, 2006), which leaves room for further experimentation by HE educators within their subject areas.

So, the current practice clearly demonstrates that MCTs are widely used and still have a great potential for Computing disciplines to assess cognitive skills up to higher levels – summatively, formatively and diagnostically. This statement is not to veil their known disadvantages: MCTs are time consuming and relatively difficult to set; they bring   ambiguity which potentially leads to 'multiple guess' scenarios; their focus on recognition (choosing an answer) rather than on recall (constructing it) (Collins, 2005) induce academic suggestions that MCTs encourage students to take a surface approach to learning, *etc.*; a summary of common criticisms can be found, for example, in (Korsunsky, n.d., Part1). Rather the intention here is to point that, like any other general-purpose tool, MCTs can only be good where appropriate and when applied with full understanding of their peculiar limitations.

## MCTs in teaching SoftEng – where applicable and in what extent?

Before considering the application of MCTs to SoftEng, it is necessary to clarify what this discipline entails. Table 1 outlines the area of SoftEng, classifying the full set of SE processes at the top level. Every process is then broken down into activities and every activity in turn - by its tasks.

| Primary life cycle processes | Supporting life cycle processes | Organisational processes |
|---|---|---|
| *Acquisition process* <br> *Supply process* <br> Development process <br> Operation process <br> Maintenance process | *Audit process* <br> Configuration Management <br> *Joint review process* <br> Documentation process <br> *Quality assurance process* <br> *Problem solving process* <br> *Verification process* <br> *Validation process* | *Management process* <br> Infrastructure process <br> *Improvement process* <br> Training process |

*Table 1. Software Engineering processes (ISO/IEC 12207)*

It can be seen that industrial standards involve much more than 'writing computer programs'. In particular, programming in the old sense is just one small brick in the Development process that contains 12 other activities, besides Coding, such as Requirements Analysis, Architectural Design or Qualification Testing. So, along with technical processes, SoftEng covers many others that actually make an 'umbrella' and support technical ones. Many of them (shown in Table 1 in italic) require 'softer' skills. Some require the ability to conduct professional assessment as a key skill, for example, Joint review or Validation and Verification (V&V) processes that imply that any product in the Development process should be somehow reviewed, verified and/or validated.

Thus, students have to be taught to gain all key skills, essential in professional practice, and then be qualified to become software engineers.

In the context of this consideration, the questions to rise are: where MCTs are appropriate in teaching SoftEng and where they are of limited use or not applicable, in terms of validity.
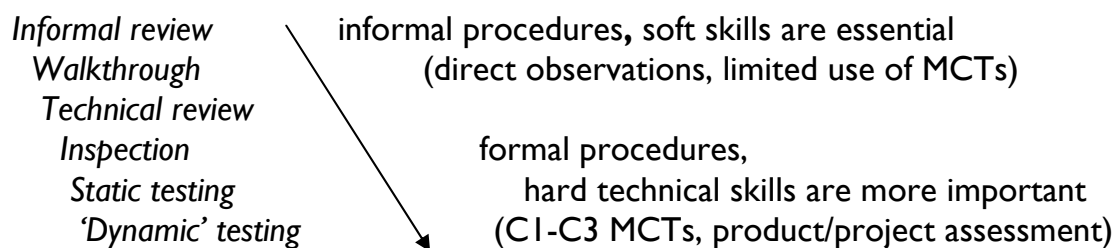
Two points are to be addressed here:

1. Although MCTs can potentially support assessment of all cognitive skills, valid assessment of upper-intermediate skills of *practitioners* still requires alternative techniques. Good illustration of this is provided by the ISEB qualification procedure for Software Testing, where the procedure includes 50% of C1, 30% of C2 and 20% of C3 MCTs for the Foundation level; shifting to higher proportion of practical questions at the next (Intermediate) level. However, at the highest (Practitioner) level, MCTs-based assessment is replaced by a scenario-based essay

(two hour exam), plus 18-month practical experience in a testing role is required, as a prerequisite (Hambling, 2007).

In this context, the possible 'substitution of notions' should be strongly avoided, as Bloom's *application* level that actually combines comprehension and ability to apply 'new information, rules, methods, concepts, principles, laws, and theories' (Blooms, 1956) in concrete (more or less similar) settings is <u>not</u> the same as *application* skills that are expected by employers from software engineers as 'core' or 'hard' ones (for example, 'proficiency with software applications' or 'software design' skills). The latter normally imply problem solving and/or decision-making (i.e., Bloom's higher order levels) by transferring knowledge and skills in new, sometimes unique, situations, and so cannot be validly assessed with only C1/C2-types MCTs.

2. MCTs are rarely applicable for affective and psychomotor domains, so the more influential 'soft' skills are the issue under assessment (see italicised processes in Table 1), and here the more alternative techniques should be involved. This can be illustrated more specifically by the example of V&V techniques (here cognitive skills of higher levels are mostly essential, as V&V are to provide professional judgement and recommendations):

*Informal review*       informal procedures, soft skills are essential
  *Walkthrough*                 (direct observations, limited use of MCTs)
   *Technical review*
    *Inspection*                    formal procedures,
     *Static testing*                  hard technical skills are more important
      *'Dynamic' testing*           (C1-C3 MCTs, product/project assessment)

Above, main software V&V techniques are ordered by the level of their formality (Hambling *et al.*, 2007). On the 'informal' end of the scale, MCTs (applied summatively) are unable to provide enough predictive validity, as the issue under assessment involves 'softer' skills and is not defined formally (such as, for example, the ability to act professionally and effectively in a given role during a peer-to-peer technical review meeting). Alternative assessment techniques should be involved, such as direct observations of the process (e.g., modelled review meetings) with different settings, as many as possible to provide more reliable results (Knight, 2001). However, for the 'formal' end of the scale MCTs are quite appropriate, if properly combined with traditional product assessment techniques typical of computing disciplines.

**Conclusion and future directions**

MCTs have a great potential for educational practice, but since they can also lead to poor assessment, every educator should be aware of their limitations in the context of their subject discipline to not compromise validity. MCTs are most appropriate

for testing factual, conceptual, or procedural information, but for more complex cases and in many real contexts they should be supported or replaced by other techniques, especially where the so-called 'soft' skills are involved.

Current software industry requires from SoftEng graduates to have more than just traditional 'hard' skills, which raises a challenge for educators. The VLE provided by London Metropolitan University for teachers and students (WebLearn) usefully supports easy creation and flexible use of MCTs, but many professional skills (such as peer review skills for software engineers) still need to be taught and assessed by using more 'time-consuming' techniques, such as direct observation in different contexts that require more efforts and assessment skills from assessors themselves. Peer to peer reviews, observations, cross reviews and other techniques should be integrated with a reasonable amount of MCTs. The author's future intention is to design a series of workshop sessions for the whole range of software V&V techniques, where MCTs are not only used for providing (via Weblearn) pre-/post-session diagnostic/formative assessment quizzes, but also support group interaction during a session, quickly highlighting main controversial views of a topic and possible misunderstanding. Also, as MCTs are provenly good and natural means for implementing V&V inspection questionnaires, some MCTs-based online training can preface and support other forms of assessment (such as student peer reviews/inspections of development documentation produced by their peers, against standards and normative requirements). It should be noted that industrial standards and practical guides provide a very useful and up-to-date source of information of this kind that can be relatively easy converted to good *educational* materials.

## Acknowledgments

## References

BIT (2001-2006) *BIT Examination Papers* (Semester1 – Semester 6) (Online resources) University of Colombo. Available at: http://www.bit.lk/examination/Semester1.html (Accessed 03/2009)

Bloom B.S. (ed.) (1956) *Taxonomy of Educational Objectives: the classification of educational goals*. Susan Fauer Company.

Briggs, D.C., Alonzo, A.C., Schwab, C., and Wilson, M. (2006) Diagnostic Assessment With Ordered Multiple Choice Items, *Educational Assessment*, Vol. 11, Issue 1, pp. 33–63.

Carneson J., Delpierre. G., and Masters, K. MCQs and Bloom's Taxonomy. *Designing and Managing Multiple Choice Questions* (Online handbook) Available at: http://www.le.ac.uk/castle/resources/mcqman/mcqappc.html (Accessed 03/2009)

Collins, J., (2005) Writing Multiple-Choice Questions for Continuing Medical Education Activities and Self-Assessment Modules. (Online) *Education Techniques for Lifelong Learning* Available at: http://radiographics.rsnajnls.org/cgi/content/full/26/2/543#R14 (Accessed 03/2009)

Hambling, B. (ed.) et al. (2007). *Software testing: an ISEB Foundation*, British Computer Society, Swindon

ISO/IEC 12207 (1995). *Information technology – Software life cycle processes.* Geneva: ISO.

Knight, P. (2001). A Briefing on Key Concepts: Formative and summative, criterion & norm-referenced assessment. *LTSN Generic Centre Series* No 7. York, UK: Learning and Teaching Support Network

Korsunsky, B. Multiple-Choice Tests: Why the Controversy? (Part 1), MCTs: Good for Assessment, Good for Research, Good for Teaching (Part 2) (Online). Available at: http://apcentral.collegeboard.com/apc/members/courses/teachers_corner/40989.html (Accessed 03/2009)

LaTrobe (2007). *A Guide to using Multiple Choice Questions and other Objective Test Questions.* (Online) La Trobe University, CTLC. Available at: http://www.latrobe.edu.au/teaching/assets/downloads/Guide%20to%20objective%20test%20item%20writing.pdf (Accessed 03/2009)

REAP (2006). *A REAP project case study* (Online) University of Strathclyde. Available at: http://www.reap.ac.uk/public/CaseStudy/SU_MechEng_CS1.pdf (Accessed 03/2009)

SWEBOK (2004): *A Guide to the Software Engineering Body of Knowledge* (Online). Available at: http://www.swebok.org/htmlformat.html (Accessed 03/2009)

Williams, T.J., (2005). *Multiple-choice questions in Computer Science*, 2nd ed., Tata Mcgraw-Hill

## Biographical note

Tamara Matveeva is part-time lecturer in the Faculty of Computing. Previously, Tamara worked as a senior researcher for the Russian Academy of Science. She obtained her PhD in Computing from the Institute of Automation and Control Processes (IACP). Her main research interests include: software testing, validation and verification (V&V) techniques, software measurement, and software quality metrics.
Contact: t.matveeva@londonmet.ac.uk