Software Engineering PhD Qualifying Examination  
October 23, 2009

Answer six of the following ten questions including questions 1 and 2. Include citations to relevant literature where appropriate.

1. **Research Impact**

   It has been claimed that software engineering research has had relatively little impact on actual software development practice. Select three different areas of practice. The first should be an area for which you can show the direct impact of academic software engineering research. For this area, do the following.

   a) Describe the state of the practice in the area in terms of methods and technology (tools) currently used
   b) Indicate the role that software engineering research has had on these methods and technologies citing at least two specific papers. Include a precise statement of the research contribution made in each paper
   c) Argue why you think that research has been able to impact this area of practice

   For the second area, select an area in which academic software engineering research has not had much practical impact. For this area, do the following.

   a) Describe the state of the practice in the area in terms of methods and technology (tools) currently used
   b) Provide at least two major historical research results in this area, including citations of papers and precise statement of the contributions made
   c) Explain why you think that research results have not had an impact on this area

   For the third area, select an area for which you think current research will have a major impact in the future. For this area do the following.

   a) Describe the state of the practice in the area in terms of methods and technology (tools) currently actually used by developers today
   b) Provide at least two recent research results that you believe will have future impact, include citations and a precise statement of research contributions made

   Argue why you think this research will have an impact, including overcoming the kinds of barriers you discussed for the second area.

2. **Testing**

   Test input generation is an expensive activity that involves a considerable amount of human effort. For this reason, a great deal of research effort has been put into the development of techniques that can automate test input generation. One family of such techniques is based on the use of symbolic evaluation—the execution of a program using symbolic, rather than concrete, inputs to identify which inputs can lead to the execution of a specific statement in the program. Although symbolic execution is at least three decades old, recently there has been a renewed interest in this kind of techniques, and researchers have proposed a new wave of symbolic execution techniques that consist in more or less extensive variations of the original idea.

   a) Provide a definition of traditional symbolic execution, and discuss how it works. Use an example to illustrate the technique.
b) Identify two recent papers that, in your opinion, advanced the state of the art in symbolic execution in a considerable way. Summarize the papers, describe how they extended traditional symbolic execution, and state why you think their contributions are important. Discuss any empirical evidence of the usefulness of such techniques.

1. Discuss why the technique they present is unsound and what are the possible consequences of such unsoundness.
2. Provide an example to illustrate the unsoundness and its consequences.
3. Can you define a variation of CUTE that eliminates the unsoundness in the technique? If so, describe the modified technique. If not, justify why you think that is not possible.

3. Object Orientation

Object oriented (OO) methods are now ubiquitous in software engineering research and practice.

a) Characterize, in general, what object orientation is and then indicate at least three areas/activities of software development in which it used. For each area describe how your characterization is realized

b) Summarize the argument on why OO methods are beneficial?

c) Point out any published results, either positive or negative, evaluating the effectiveness of OO. Why is evidence of the value of OO so sparse?

d) Outline a 5-year research program for measuring the cost/benefits of OO methods to industrial practice. Include at least three specific research projects that you would run. Include at least the following items for each. 1) research questions/hypothesis being tested; 2) approach taken and research methods used; and 3) validation procedures

4. Cloud Computing

Cloud computing, also called Software as a Service (SaaS) has been proposed as an alternative delivery vehicle for providing access to software applications.

a) From the point of view of the customer/user, how does SaaS differ from traditional application delivery in terms of non-functional qualities?

b) From the point of view of developers, how does SaaS differ from traditional development and delivery?

c) What sorts of applications do you think can be effectively delivered this way, and what sorts do you think will continue to be delivered traditionally? Give reasons for your choices

d) Describe how software engineering research should address this trend

5. Silver Bullet

Brooks argued more than twenty years ago in his classic "No Silver Bullet" essay that no single innovation would lead to an order of magnitude improvement in software development practice.
a) Summarize his argument. Include definitions of what he meant by the terms *accidental* and *essential*

b) Indicate how his argument might be attacked; that is, give some counter arguments

c) How would you determine whether he was correct of not? That is, how is software development practice measured?

6. **Testing**

Most existing software testing tools use some form of "coverage" as a way of measuring and controlling the testing process.

a) Define the term *coverage*, and give examples of at least three kinds of coverage currently employed in actual practice

b) Cite recent research results that have advanced new approaches to coverage. Summarize the major contribution of this research

c) Coverage measures treats all parts of a program uniformly implying that equal effort will be expended on infrequently occurring situation as on those which are likely to occur frequently. What alternatives to coverage-based testing exist that address this issue? Provide a comparison of these approaches to coverage-based approaches

7. **Software Process**

Many current software development organizations currently make use of agile software processes.

a) Define the term *agile* in this context and characterize the properties of an agile process

b) Describe and differentiate SCRUM and Extreme Programming as agile processes

c) What is the alternative to agility in current practice? Give at least two examples of situations in which agile methods would not be appropriate

d) From a software engineering research point of view, what research questions have been proposed with respect to the role of agility in software process? Cite at least two papers in the area and summarize their contributions

e) Suggest at least two new research questions in this area that have not yet been addressed and indicate how you would study them

8. **Software Generation**

Object oriented frameworks, while providing reusable resources, are notoriously hard to understand and use.

a) Precisely define the term *OO framework*, and cite three important historical research contributions to the development of this area

b) List and explain three reasons why OO frameworks are hard to understand and use

c) Cite recent research addressing this problem. Describe the contribution of at least three publication addressing this problem

9. **Program Analysis**
Program analysis is a fundamental area of software engineering that contributes improvements to the state of the practice in software test and evaluation and program understanding.

a) What is the state of the (research) art in program analysis? That is, what problems have been solved (and by whom) and what problems are currently being worked on? Give at least three examples (with citations) of each.

b) Program analysis is incomplete in the sense that it answers questions about what a program can do, but does not address why (in terms of program requirements) the program does it. Assuming that precise requirements are available, outline a tool that would address *why* questions, in particular, questions about the role a particular program construct plays in satisfying requirements. Include a discussion of technical issues that would arise.

c) What capabilities would such a tool provide to developers? Provide at least three scenarios in which having such a tool would support activities that are currently difficult to accomplish with existing program analyses.

10. Model Driven Architecture
Model driven architecture (MDA) has been proposed by OMG (the sponsors of UML) as a natural extension of modeling to the architectural level.

a) Summarize the substance of the MDA approach.

b) An alternative to MDA is the use of architectural description languages (ADLs). Define the term *architectural description language* and describe and differentiate three example ADLs, including appropriate citations.

c) Compare and contrast MDA (an industrially driven approach) to ADLs (an academically driven approach). To what extent are the approaches compatible/incompatible? Is a synthesis possible? If not, why not; if so, describe what the synthesis would look like.