Software Engineering PhD Qualifying Examination
March 25, 2009

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

1. Your Research Interests
   a) What area of research most interest you and why?
   b) Describe three important problems in the area along with the current state-of-the-art in those problems.
   c) What is the state of the practice for those problems—i.e., how do practitioners currently deal with the problems?
   d) List five people that are top researchers in your area and discuss, for each of them, what was his/her contribution to the field, why that contribution is important, and how it improved the state of the art and the state of the practice.
   e) What problem or problems in the area do you currently plan to pursue, and what contributions do you plan to make?

2. Research Impact
   Select a recent (published in 2005 or later) software engineering research paper, which comes from an area other than that used in your answer to #1, and which you feel will have long-term impact on both research and practice.
   a) Summarize the paper including its main contributions.
   b) Justify why you think that the paper will have research impact? That is, define what you mean by “research impact” and argue how this particular paper satisfies your definition.
   c) Justify why you think that the paper will have practical impact? Define what you mean by “practical impact” and argue how this particular paper satisfies your definition.
   d) Identify at least two future research directions that build on the results presented in the paper. For each of the directions you identified:
      • Discuss why you think the direction is promising and worth pursuing
      • Discuss the main research challenges for the direction
      • Propose a research plan for investigating the direction and discuss how your research approach addresses the challenges you identified
      • Propose a measure of success for your research plan. That is, how will you validate your research and assess whether it was successful and to what extent? How does your measure relate to your definition of “impact”?

3. General Software Engineering
   a) How would the following non-functional requirements impact a software system's design? You may answer the question for each individual requirement and/or by pointing out appropriate trade-offs among requirements.
      • Security
      • Efficiency
      • Fault-tolerance
      • Adaptability/flexibility/reusability
• Portability
  
b) For each of the non-functional requirements given above, indicate an architectural 
style that would be well suited for supporting the requirement and justify your 
choice.

c) Assume that you are writing a test plan for a system that must satisfy all of the 
non-functional requirements listed in a). How would you describe, in your test 
plan, your strategy for assessing whether such non-functional requirements are 
met? Be detailed and describe clearly how each requirement would be addressed. 
Use specific examples if needed.

4. Testing
  
Testing is performed at many different stages of the software-development lifecycle, 
and may be performed differently at each stage. In particular, consider three stages of 
the lifecycle: development, testing, and maintenance.
  
a) Explain in detail how testing would be performed in each of these stages—for 
example, who would perform the testing at that stage, what kind of testing would 
be done during that stage, how would the test cases be developed.
  
b) Describe one state-of-the-art technique for performing testing at each of the three 
stages, including the names of the researchers or techniques that have been 
developed.
  
c) Describe one state-of-the-practice technique for performing testing at each of 
these stages, including the names of tools that are used.

5. Analysis
  
An output dependence exists between statements $S_i$ and $S_j$ that both define a variable 
$X$. For example, in the figure below, $S_i$ defines $X$ and so does $S_j$. Thus, there is an 
output dependence between $S_i$ and $S_j$.

$$S_i: X = \text{(definition of } X)$$

\[ \text{<------point P immediately after definition of } X \]

$$S_j: X = \text{(definition of } X)$$

One way to determine these dependencies is to compute the sets of definitions that are 
"reachable" from the point P immediately following the definition at $S_i$.
  
a) Give a data-flow framework to compute sets of definitions that can be reached 
from the end of each basic block.
  
1. Indicate if this problem is an any- or all-paths problem.
2. Indicate if the problem is a forward or backward data-flow problem.
3. Give the definitions of the GEN, KILL, IN, and OUT sets for the 
problem;
4. Give the initial values of the sets;
5. Give the algorithm
  
b) Given the reachable definitions for a program, how would you compute its output 
dependencies?
  
c) Provide an example that illustrates your answers to b) and c).

6. Software Architecture
  
Select one large system on which you have worked or that you have developed.
a) Describe in detail the software architecture that was used for the system and create a diagram illustrating the architecture.

b) Describe the architecture as it relates to one of the architectural styles that Shaw and Garlan discussed. With which one is most closely related? Is it a hybrid of several of the styles?

c) Discuss why this architecture was selected; discuss its advantages and disadvantages, etc.

d) If you were designing such a system again, would you use the same architecture? Explain why or why not.

7. **Software Process**
   a) Based on your experience and knowledge of software engineering, what are the best practices that you would use within a project, no matter what specific process is being used for the project? Mention at least three best practices and justify your answer.
   
b) How do agility and rigor/planning trade off in the selection of a software process to use on a project? What is the key business issue in choosing between the two? Pick and describe one agile process and one plan-driven process. Devise a project scenario in which each would be appropriate. Include discussion of the roles of the following factors: project size, criticality, requirements volatility, personnel, and culture/team.

8. **Modeling**
   a) Define and differentiate software analysis, software architecture, and software design? How does UML support each of these three phases? I.e., which diagrams best support each of these phases. For each phase, what are the steps used to derive the diagrams? For each of the three phases, mention another notation that is not part of UML and can be used to support and create models for that phase.
   
b) What are the disadvantages of using UML?
   
c) Take and defend a position on the issue of whether UML should be considered as a formal notation.

9. **Software Generation**
   A variety of techniques beyond traditional coding can be used to produce working software. Consider the following:
   - Domain-specific languages with application generators
   - Object-oriented frameworks
   - Metaprogramming
   
a) Define and describe each technique, including a description of the steps in its use.
   
b) Describe a well-know application or library that has been build using each technique.
   
c) Devise a set of criteria that could be used when given a project definition that could be used to determine which technique to apply.