Software Engineering
PhD Qualifying Examination
October 28, 2010

Rules:

- Answer six of the following nine questions including questions 1 and 2.
- Include citations for relevant literature where appropriate, including items from the reading list.
- Include a copy of the questions with your answers.

1. Your Research Interests – You must answer this question
   a) What area of research most interests you and why?
   b) Describe three important problems in the area along with the current state-of-the-art addressing 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 both 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. Test-input Generation – You must answer this question
   Over the years, researchers have defined a large number of test-input generation approaches. Two important classes of approaches in this area are (1) those based on symbolic evaluation and (2) those based on random generation.
   a) Select one representative approach from each of these two classes that was published in the last five years and with which you are already familiar.
   b) For each of the two approaches you selected:
      1) Provide a summary of the approach including its main characteristics.
      2) Describe what is the novel contribution of the approach over previous work in the same class (i.e., test-input generation based on symbolic evaluation or random generation).
      3) Discuss what are the major specific strengths and weaknesses of the approach.
      4) Discuss what are the major general strengths and weaknesses of the corresponding class of approaches.
   c) Describe a possible way in which these two classes of approaches could be combined. It is OK to refer to existing work in the answer, in case combined approaches already exist, or state that the two classes of approaches cannot be combined. In this latter case, provide a convincing argument of why this is the case.

3. Debugging
Debugging is time consuming, and thus, there has been much research to improve the debugging process.

a) Discuss the activities of the debugging process and give an example of recent research to address each activity.

b) One of the activities is locating the faults in the program and much research has addressed this activity.
   1) Discuss two static approaches to locating faults in a program.
      i. For each approach, give a brief overview of the approach, discuss its strengths and weaknesses, and given an example of how it works.
      ii. Discuss one area for future work that could improve the approach.
      iii. Compare and contrast the approaches with each other.
   2) Discuss two dynamic approaches to locating faults in a program.
      i. For each approach, give a brief overview of the approach, discuss its strengths and weaknesses, and given an example of how it works.
      ii. Discuss one area for future work that could improve the approach.
      iii. Compare and contrast the approaches with each other.
   3) Discuss one approach based on remote monitoring, give a brief overview of the approach, and discuss future work that could improve it.

c) Based on your discussion in b), what do you think is the most promising approach for locating faults in programs and why?

4. Software Process

Imagine that you have been hired as a consultant for a large (over 500 developers) software company. The company has been awarded a contract to deliver the software parts of a chemical process control system to an industrial client. The system includes various sensors (temperature, pressure, flow rates, etc.) as well as valves for controlling chemical flows. The system will manage the production of toxic chemicals, so safety is a critical requirement.

a) List and describe at least three approaches to software safety that you would consider and select one for use. Give reasons for your choice.

b) Some of the developers with whom you speak have had positive experiences using agile software development methods while others describe the need for more rigorous approaches. What factors should you take into account in deciding between these methods to recommend? With respect to design validation, describe at least one agile method and one rigorous method. Compare the two enlisting the factors you selected, make a recommendation and justify the recommendation in terms of the factors.

c) Some of the developers recommend the use of formal modeling. What opportunities (steps in the software process) for formal modeling does this scenario provide? Pick one such step and describe at least two modeling formalism that could be applied, including in your description their strengths and weaknesses.
5. No Silver Bullet

In 1986, Fred Brooks, in "No Silver Bullet" states with respect to software engineering that "There is no single development, in either technology or management technique, which by itself promises even one order-of-magnitude improvement within a decade in productivity, in reliability, in simplicity."

a) Was Brooks correct in his prediction? Justify your answer.
b) Describe three advances that have occurred in software engineering practice since 1986 that have demonstrated improvements in either the productivity, reliability or simplicity of software development practice. Describe the improvements and provide a reasoned numerical estimate of the extent of improvement that each engendered?

c) Define what Brooks meant by "accidental" and "essential" complexity and give an example of each in terms of some currently prominent software system.
d) Provide and justify three current research approaches to dealing with accidental complexity that you deem promising.
e) Outline a software engineering research program for dealing with the problem of essential complexity. Include research questions, methodology, and validation.

6. Software Architecture

a) Define the term software architecture, citing appropriate literature from the qualifier reading list.
b) Give at least three common types of diagrams that are used to convey architecture and critique them with respect to your definition.
c) Consider the following conjecture: "The dominant factor in architecting a significant software system is how it will deal with the system's non-functional requirements." Using the Apache web server as an example, answer the following questions:
   1) Describe the major structural elements of its architecture.
   2) Describe the three most important non-functional requirements that it has to deal with.
   3) Support or refute the conjecture with respect to your answers in 1) and 2).

7. Data-flow Analysis

A dead definition is a definition of a variable that is never referenced (i.e., there is no def-clear path from that definition to a use of the same variable).

a) Formulate the problem of detecting dead definitions as a data-flow problem that can be solved using a generalized iterative data-flow analysis framework. Assume that the data-flow analysis will be performed on programs in Static Single Assignment (SSA) form (i.e., that every variable in the program is assigned exactly once). Specifically:
1) Specify whether it is a forward or backward data-flow problem and, based on that, which visiting order could improve the worst-case time complexity of the analysis.

2) Define GEN(B) and KILL(B) for the problem, where B is a block in the program's CFG.

3) Define the initial value of IN(B) and OUT(B).

4) Define the equations to compute IN(B) and OUT(B).

5) Define how the result is computed, that is, which set would contain the dead definitions at the end of the analysis.

b) Is it possible to formulate this problem as an instance of the generalized iterative DF analysis framework, as you did above, for programs not in SSA form? Justify your answer.

c) Exhibit a program for which your analysis would require four iterations to converge using the visiting order of Point a.I. (Convergence in four iterations means that the data-flow facts computed in the first three iterations are different, but the data-flow facts computed in the fourth iteration are identical to those computed in the third iteration.)

8. Points-to Analysis

Consider the following two programs P1 and P2:

<table>
<thead>
<tr>
<th>P1 ()</th>
<th>P2 ()</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 if (...)</td>
<td>int <strong>foo (int</strong> i) {</td>
</tr>
<tr>
<td>2 p = &amp;x;</td>
<td>6 (**i)++;</td>
</tr>
<tr>
<td>3 else</td>
<td>7 return i;</td>
</tr>
<tr>
<td>4 p = &amp;y;</td>
<td>}</td>
</tr>
<tr>
<td>5 x = &amp;a;</td>
<td></td>
</tr>
<tr>
<td>6 y = &amp;b;</td>
<td></td>
</tr>
<tr>
<td>7 *p = &amp;c;</td>
<td></td>
</tr>
</tbody>
</table>

Program P1 | Program P2

a) For P1
1) Compute and show the results of *flow-insensitive may points-to analysis.*
2) Compute and show the results of *flow-sensitive may points-to analysis.*
3) Compute and show the results of *flow-sensitive must points-to analysis.*

b) For P2
1) Compute and show the results of *flow-insensitive, context-sensitive may points-to analysis.*
2) Compute and show the results of *flow-sensitive, context-insensitive may points-to analysis.*
3) Compute and show the results of *flow-sensitive, context-sensitive may points-to analysis.*

9. Research Impact

a) 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.

1) Describe the state of the practice in the area in terms of methods and technology (tools) currently used.
2) 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.
3) Argue why you think that research has been able to impact this area of practice.

b) 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.

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

c) 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.

1) Describe the state of the practice in the area in terms of methods and technology (tools) currently actually used by developers today.
2) Describe at least two recent (last four years) research results that you believe will have future practical impact. Include citations and a precise statement of research contributions made for each.
3) Argue why you think this research will have an impact, including overcoming the kinds of barriers you discussed for the second area.