1. (15%) You are the chief architect in a software company. Can you explain when you can apply “Web Services” in software development, and when you cannot? Name 3 software in your company that can be reengineered with Web services, and 3 software that cannot. Support your arguments with +/- evidences.

**Answer.**

4% (a) Use “Web Services” when the software development desires or has:

1. high interoperability (or protocol standardization) w/o -2
2. high reusability (or components-off-the-shelf (COTS)) w/o -2
3. componentization (or modularization, information hiding, large-scale complexity, need for middleware) with +0.5
4. dynamic matching or deployment (or networked) with +0.5

3 x 1.5% (b) 3 example software satisfying the above requirements

For each example (1.5%):

the mark = \min(1, (satisfaction mark / 4))

Where satisfaction mark is determined by criteria in bullet (a).

Example. A heterogeneous database system needs high interoperability (2%) and high reusability (2%), needs to hide implementation details (+0.5%). Since it has 4.5%, 1 mark is earned.

Explained with positive/negative reasons. w/o -0.5

2% (c) Do not use “Web Services” when the software requires

1. demanding performance (or real-time) w/o -1
2. can not use any net protocol (or extreme high security) w/o -0.5
3. too simple (or single component, no need for middleware) w/o -0.5

3 x 1.5% (d) 3 example software satisfying the above requirements (similar to marking scheme in (b), replace criteria (a) with criteria (c))

2. (15%) What are the risks in a software development? How to evaluate the risk of an earthquake to the software development? Name 4 risk factors in the OmniEditor project and indicate measures that can prevent or mitigate them?

**Answer.**

3%(a) Give a definition of the risk in software development (w/o -2) and list some common risks (w/o -1).

2%(b) Give the reasoning for the earthquake risk (w/o -2):

Impact is big to personnel, small to software storage (w/o -1); Probability is very small (w/o -1).

4x2.5%(c) Sum risk factor mark for at most 4 factors. Each factor is marked by:

1. relevant to OmniEditor (w/o -0.5)
3. (15%) A competitor company A has a killer application K that dominates the market. You are recently employed by company B as a senior consultant, their application K' always has all the functionalities of K. What technical advices will you give to help take away K's market share?

Answer.

5% (a) Quality (non-functional issues) improvements, w/o -5
5% (b) Break down of the quality into a number of issues, such as performance, usability, security, etc. w/o -5
5% (c) Study the company A's software, find its weakness, w/o -5
5% (d) Measuring the qualities attributes +1
5% (e) Open-source or use open standard with +1
5% (f) Give a concrete and reasonable example +1

4. (25%) You are managing the development of a large-scale software system which already has Y out of X components developed. According to your project estimation, the undeveloped components still need T person-month to finish, while you just have N team members including yourself. Now you are pressed to complete the system within as short as S months and you can and only can recruit M more junior developers. Will you meet the time pressure? If you can, how will you reorganize the team? If you cannot, why? Please answer according to the following three scenarios:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>T</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>(1)</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>(2)</td>
<td>30</td>
<td>2</td>
<td>100</td>
<td>5</td>
</tr>
<tr>
<td>(3)</td>
<td>3</td>
<td>0</td>
<td>15</td>
<td>5</td>
</tr>
</tbody>
</table>

Answer.

5% (a) Use the numbers in estimating the time needed for the new project w/o -5
4% (b) Scenario (1) is not feasible because the time available is below 75% of the estimated time; otherwise -4
4% (c) Scenario (2) is feasible by the estimation, if hierarchical team organization is used; it is not feasible if training efforts is huge by explicit assumption; otherwise -4
4% (d) Scenario (3) is not feasible because the time available is far below 75% of the estimated time; otherwise -4
4 x 2% (e) If the calculations in the discussions make following mistakes, such as: mistake "person-month" for "month".
-2 per mistake: if there are more than 4 minor mistakes, -8.
Here is a list of unfinished requirements specifications for the “Meeting Scheduler” system (including software and human as parts of the system):

<table>
<thead>
<tr>
<th>Functional Requirements</th>
<th>Non-functional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. Schedule Meeting</td>
<td>1. Minimal effort</td>
</tr>
<tr>
<td>0.1 Collect Timetable</td>
<td>1.1 Collection effort</td>
</tr>
<tr>
<td>0.1.1 By Person</td>
<td>1.2 Matching effort</td>
</tr>
<tr>
<td>0.1.1.1 By email, fax and letters</td>
<td>2. Good Quality</td>
</tr>
<tr>
<td>0.1.1.2 By email</td>
<td>2.1 Minimal conflicts</td>
</tr>
<tr>
<td>0.1.2 By System</td>
<td>2.2 Good participation</td>
</tr>
<tr>
<td>0.1.2.1 Have updated time table</td>
<td></td>
</tr>
<tr>
<td>0.1.2.2 Collect them</td>
<td></td>
</tr>
<tr>
<td>0.2 Choose Schedule</td>
<td></td>
</tr>
<tr>
<td>0.2.1 Manually</td>
<td></td>
</tr>
<tr>
<td>0.2.2 Automatically</td>
<td></td>
</tr>
</tbody>
</table>

(a) For each functional requirement to the left, express its goal, input, output, pre/post conditions, exceptions.

(b) Relate the tasks at the left hand side with the criteria at the right hand side using positive [+] or negative [-] relations (E.g. 0.1.1 [-] → 1.1), then indicate which functional tasks are needed to fulfill the top level requirements “0” and “1”, regardless to “2”.

Answer

3% (a) If not all functional requirements are expressed, -3

2x3%(b) For 0, 0.1.2 (AND decompositions): check if a parent goal’s precondition implies the first sub goal’s precondition (not -1), and the first subgoal’s postcondition implies the second subgoal’s precondition (not -1), and the second subgoal’s postcondition implies the parent goal’s postcondition (not -1).

3x2%(c) For the 0.1, 0.1.1, 0.2 (OR decompositions): check whether each subgoal’s postcondition implies the parent goal’s postcondition (w/o -1), and the parent’s precondition implies the subgoal’s precondition (w/o -1).

10%(d) For the second question, check list:

0.1.1 or \{0.1.1.1, 0.1.1.2\} \([-]\)→ 1.1, w/o -2
0.1.2 or \{0.1.2.1, 0.1.2.2\} \([+]\)→ 1.1, w/o -2
0.2.1 \([-]\)→ 1.2, 0.2.2 \([-]\)→ 1.2, w/o -2
0.2.1 \([-]\)→ 2.1, 0.2.2 \([-]\)→ 2.1, w/o -2
0.2.1 \([+]\)→ 2.2, 0.2.2 \([-]\)→ 2.2, w/o -2

5%(e) if 2 is not concerned, one can remove the last 4 relations and prune the functional requirements based on the other relations. The resulting functional requirements should only have:

\{ 0, 0.1, 0.1.2, 0.1.2.1, 0.1.2.2, 0.2, 0.2.2 \}

If the answer has only the leaf-goals \{0.1.2.1, 0.1.2.2, 0.2.2\}, it is also right.