Software Engineering (Adv CS II, 320212)

Final Exam Spring 2006

Logistics

- You have 90 minutes (sharp) for the test.
- You can reach 82 points if you solve all problems. You will only need 78 points for a perfect score, i.e. 4 points are bonus points.
- Mark all sheets you deliver with your name to make sure it can get graded. We cannot grade if not present or illegible!

Name:

1 Language Processing

Task 1.1 (5 pts): What are the components inside a compiler?

Task 1.2 (8 pts): Write down the yacc/bison code to determine for a given arithmetic expression whether its opening and closing parentheses are balanced in number, ie, the number of '(' characters is equal to the number of ')' characters. The parsing routine, *yyparse()*, shall return 1 (i.e., true) if the paratheses in the input expression is well-balanced, and 0 (i.e., false) otherwise.

Hints:

- This input line is balanced: `(1+2)*((3+4)*(5+6))`
- This input line is balanced: `)*(`
- This input line is not balanced: `(1+2`
- You do not need to provide a complete yacc/bison source, just the grammar rule section including the associated action code (no token definitions, no *main()*, no error handling, no (f)lex scanner code).
- Syntax is not graded per se, however, the concepts must come across clearly and the use of yacc/bison and C/C++ language must be adequate.

2 User Interface Design

Task 2.1 (2 pts): What are the advantages of using analogue rather than digital presentation of information?

Task 2.2 (4 pts): List 4 techniques of user interface evaluation.

Task 2.3 (3+3 pts): Name the 3 “Golden Rules” for User Interface Design, as cited by Pressman. Give a **counter** example for each rule – i.e., some (real or imaginative) UI scenario where the rule clearly is violated. Be very specific in your scenario (something like “*I imagine a GUI where the user is not in control*” is not accepted) and make clear how the rule is violated.

3 Graphical User Interface Technology

Task 3.1 (4 pts): Explain the concept of even-driven programming in the context of GUIs.

Task 3.2 (3+1+1 pts): Describe the Model-View-Controller paradigm; of which component can you have multiple instances *according to lecture*? Give a concrete example of such multiple instances.

4 Web Engineering

Task 4.1 (3+3 pts): Explain the standard Web retrieval cycle by naming the steps and by drawing a UML sequence diagram for client and server.

Task 4.2 (5 pts): You are hired by GrandpaSoft, a long-established software company that wants to extend, with your guidance, into the Web software market. From your very first day you are facing arguments from your fellow programmers who feel embarrassed about the unfamiliar concepts you suggest. Below some of their arguments; formulate a convincing response to each one!

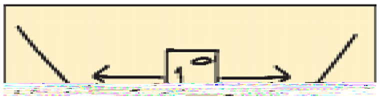
1) “*We can't deviate from our initial project plan – the customer requirements will not be implemented in time, and we will be lost in schedule*.”
2) “*We always do testing in the end when the full plot is visible. This is much more efficient*.”
3) “*A user interface is a user interface – we simply translate our Windows look-and-feel into HTML*”
4) “*incremental delivery? how can we deliver partial, incomplete stuff without being grilled by our customer?”*
5) “*I absolutely cannot accept that a customer can continuously influence our work and even redefine specs. This is anarchy – give me an instrument to handle development!”*

Task 4.3 (3 pts): After some time, however, GrandpaSoft developers get so enthusiastic about the additional freedom perceived that they tend to throw overboard all rules. Again, you have to educate them and formulate a convincing response to each argument below.

1) “*Stuff will change constantly anyway, there is no point in trying to understand requirements*.”
2) “*We don’t need to do any user testing. I’m a user, and I find this easy to use*. “

3) “*Good design is just a matter of cool graphics*.”

Task 4.5 (4 pts): Below you see the sketch of the Web design pattern „process funnel“. Describe its purpose (i.e., the problem/task it addresses), and name 3 concepts it suggests for this purpose.



5 Software Process and Project Management

Task 5.1 (4 pts): What is the difference between a milestone and a deliverable?

Task 5.2 (4 pts): Explain why best programmers do not always make best software managers. You may refer to the typical management activities as discussed in lecture (cf. “What fills a PM’s day”).

Task 5.3 (2+2+2+2 pts): Based on the type of system being developed, suggest (and give reason for) the most appropriate software process model that might be used as a basis for managing the development of the following systems:

1) Anti-lock braking system
2) Virtual reality system to support software maintenance
3) University accounting system that replaces an existing system
4) Interactive system that allows railway passengers to find train times from terminals installed in stations.

6 Software Process Models

Task 6.1 (10*0.5 pts): Kal Toth assesses in his article referenced in lecture the most common software lifecycle processes. Fill in the missing assessment using the attributes *H=High; M=Moderate; L=Low*:

LifecycleProcessCriteria	Waterfall	Incremental	Agile/XP
Maintainability	H		L-M
Uncertain Requirements	L		
Progress Visibility	L		
User Involvement	L		
Requirements Volatility	L		H
Urgency	L		

Task 6.2 (2+2 pts): Name and briefly describe two types of evolutionary development.

Task 6.3 (2 pts): Why is it increasingly irrelevant to distinguish between a software development and software evolution phase?

7 Software Package Management

Task 7.1 (3 pts): Explain what RPM is, and name two advantages of shipping rpm instead of tar or zip files.

– end of exam sheet –