Examination, ETS200/TFRN 10
Lund University, Department of Computer Science
Time: 2010-03-12 8:00-13:00
Place: MA9a,b
Assessment: total 60 points, at least 30 points is required in order to pass the exam.

1. Define the terms *verification* and *validation*. Give example of at least two test activities for verification and validation, respectively. Discuss whether a test activity may be used for both verification and validation. (5p)

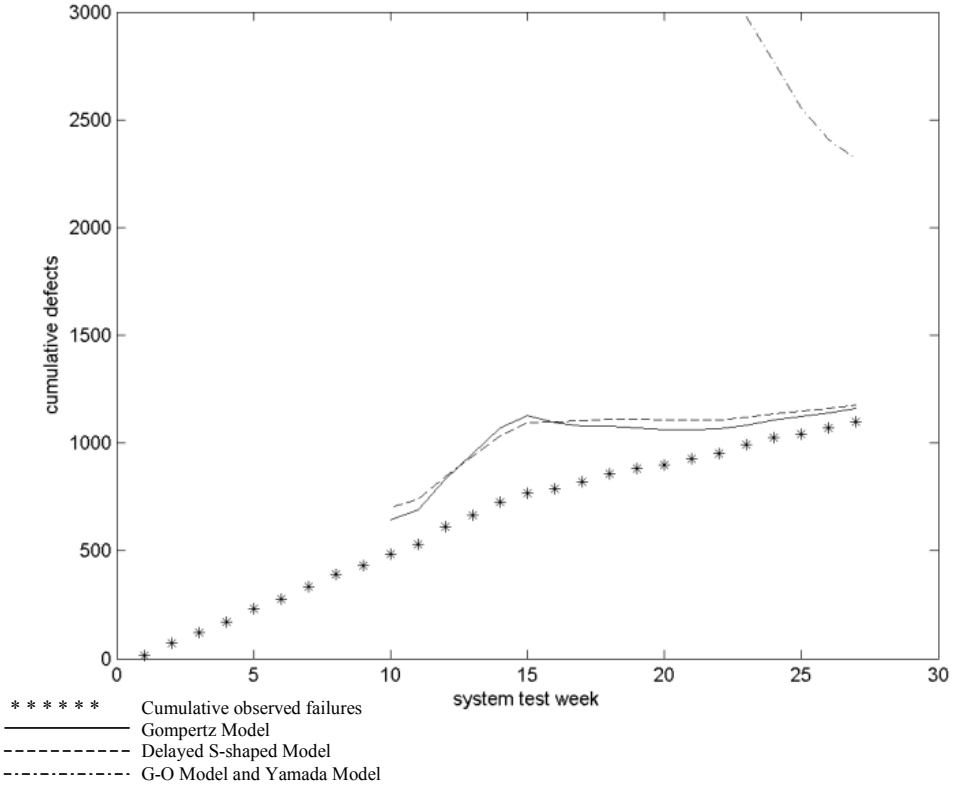
   2 p – a correct definition – verification=is the system right; validation = is it the right system
   2 p – examples: Requirements review validates towards customer expectations; Integration test verifies that the implementation meets interface specifications.
   1p – same activity may be used for both, including motivation. Example answer: system test may verify that system requirements are met and validate that the system behaves better than competitors in a general meaning.

2. Describe the difference between the white box and black box testing strategies. Discuss at which test levels white box and black box testing strategies are most feasible. (4p)

   2 p – white box – test based on internal system information May be structural or data flow based. Black box – test based on external information only, e.g. interface specifications.
   2 p – white box is generally more used for low level testing, e.g unit testing, black box more high level testing, e g system testing

3. How can reliability growth models be applied to make stop-test decisions? The graph below shows a plot of cumulative number of observed failures during system test, and estimated total number of failures for three different software reliability growth models (Gompertz, Delayed S-shaped and a combined G-O and Yamada model). The estimates are calculated in each point of time, e.g. at week 10, the total number of defects is estimated to be around 650 and above 3000, and in week 15 and forward, the estimate is round 1050 and 2400, respectively. What recommendation would you give at weeks 10, 17 and 25 respectively, for the three models? Motivate the recommendations. (7p)

   1 p – stop test decision by estimating number of remaining faults – test until a certain number or percentage is found, alternatively estimate MTBFand test until the estimate passes a pre-defined threshold.
   2 p for each recommendation:
   w10 continue since no estimate is stable;
   w 17 may be a stop recommendation or a continue, depending of what risk you accept.. For G-O definitely continue.
   w 25, stop testing since the estimates are stable. For G-O, either discard the estimate, since it is ”in the blue” or continue testing.



4. Describe inspection as a type of technical review. In order to get full points, you should describe inspections in general, the process, the roles of team members and suggested team size, when and how the methods are used, and its advantages and disadvantages. (6p)

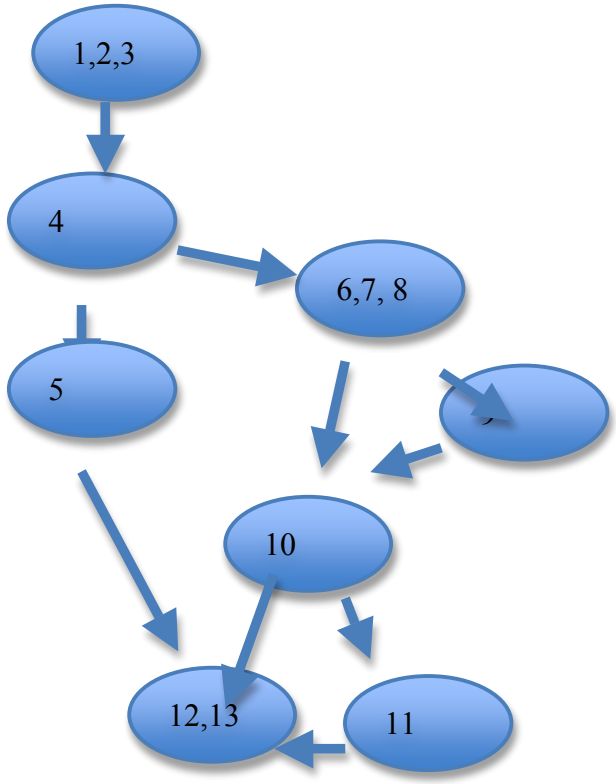
   1 p overview, 1 p process, 1 p roles, 1 p use, 2 p pros&cons
   Overview - Inspection is a formally defined review, with phases and roles
   Process: Planning/distribution of tasks -> individual preparation -> inspection meeting -> rework and follow-up
   Roles: Moderator, Reviewers (Inspectors), Presenter, Author
   Pros: Thorough approach to reviews
   Cons: Costly (both in terms of labor and calendar time)

5. For the procedure below,
   a) draw the control flow graph, calculate the McCabe Cyclomatic number, define the test cases needed to achieve 100% branch coverage. (3p)
      1 p for each of graph, MCC and test cases
   b) Set up def-use tables for all four variables, and define test cases needed to achieve 100% def-use coverage. (4p)
      2 p for tables, 2 p for test cases

   Note that test cases should include both input values and expected output.

```
1 procedure liability (age, gender, married, premium);
2 begin
3   premium := 500;
4   if ((age < 25) and (gender = male) and (not married)) then
5      premium := premium + 1500;
6   else
```

```
7   begin
8       if (married or (gender = female)) then
9            premium := premium – 200;
10       if ((age > 45) and (age < 65)) then
11           premium := premium – 100;
12   end;
13 end;
```



MCC = Edges – Nodes + 2

For MCC coverage:

	Age	*Gender*	*Married*	*premium*
TC1	*20*	*Male*	*Not*	*2000*
TC2	*50*	*Male*	*Not*	*400*
TC3	*40*	*Female*		*300*
TC4	*50*	*Female*		*200*

For branch coverage, TC1-TC3 are sufficient

Def-use analysis

Variable	*Def-use*	*Ex test case*
Age	*1-4; 1-10*	*Any; TC2*
Gender	*1-4, 1-8*	*Any; TC2*
Married	*1-4; 1-8*	*Any; TC2*
Premium	*3-5, 3-9, 3-11, 9-11*	*TC1; TC3; TC2;TC4*

Def-use coverage for Premium covers all def-uses.

6. Develop black box test cases using equivalence class partitioning and boundary value analysis to test a software component of an automated teller system. The module reads

in the amount the user wishes to withdraw from his/her account. The amount must be a multiple of 100 SEK and be less than or equal to 2500 SEK. Be sure to list any assumptions you make and label equivalence classes and boundary values that you use. Each test case should include test case identifier, input values, expected output values, valid and invalid equivalence classes and bounds covered. (9p)

3p for a reasonable partition model (
1 p for definition of assumptions (e.g. data types are checked before and hence no equivalence class is needed for wrong data types)
3 p for test cases

Assumptions: correct data types
Equivalence Partitions:
EP1: input amount less than is on the account, valid
EP2: input amount more than is on the account, invalid
EP3: input amount 1-25 multiples of 100SEK, valid
EP4: input amount less than 100SEK or more than 2500 SEK, invalid
EP5: input amount not a multiple of 100 SEK, invalid

7. Testers may be organized in many different ways. Three of them are a) Each person in the unit is responsible for both programming and testing b) Each unit has both developers and testers in it, and c) Separate units for programming and testing. Discuss pro's and con's with the three approaches. To get full points, you should cover competence provisioning, communication, management, and scale-up to large-size organizations. (8p)

2 p for each of the four items to discuss. See Kit chapter 13 (on the course web among lectures) for more detail.

	A	*B*	*c*
pros	*Natural solution*	*Testing other's software*	*Testing other's software*
cons	*Testing own software*	*Harder to manage with differnt competencies*	*Communication harder*
Comptetence	*Each person need double competencies needed*		*Specialized competencies*
Communciation	*Close*		*Distant*
Management	*Small scale – easy management*	*Manager must manage two types of competencies*	*Management may spcialize*
Scale-up	*Does not scale up to system level*		*Scalable*

8. System test often requires many resources, so many companies focus on only some types of system test, for example performance testing, security testing and recovery testing etc. How will a company do a system test if they focus on
   a) Performance? Give an example and motivate. (3p)
   Need:

Load generators to generate system workload.
Probes to log events,
Real runtime environment (HW) to get realistic behavior.
Tools for data collection and analysis

b) Security? Give an example and motivate. (3p)

Security tests:
Password checking (legal and illegal entries, expiration)
Encryption of data/messages
Browsing/acces – checking system priviledges
Trap doors – entries e.g. for test purposes that are not closed
Virus – try to infect system

2 p description, 1 p motivation

9. A current trend in testing is automation. Elaborate on test automation costs and benefits. Which types of costs are related to automation? What are the gains? Which type of testing is most beneficial to automate? (8p)

4 p for costs (tools, training, developing and maintaining scripts), 3 p for gains (reduced manual work, more execution at the same cost), 1 p for type of testing (repetitive)