1. What are the typical states of a Change Request in a maintenance process?
   Answer: receive the CR -> filter/merging CRs -> ranking them -> assign to developers -> implementation (design, code, unit, integration and system testing) -> insert in the next release.
2. What is the basic principle of the Visual approach to GUI testing? What is different with respect to layout-based scripted GUI testing?
   Answer: Visual approach: graphic components (buttons, menus etc) are recognized via image recognition
   Layout approach: graphic components are identified retrieving their ID (used by the graphic library) or by specific unique properties.
3. Provide an example of a static analysis rule from MISRA-C
   Answer: i=i+1 -> i++
4. Describe the 'pipeline' architectural style, and when it can be used
   Answer: Many modules are connected in a sequence. Each module is independent of others. Data from one module flow to the next one. Data is the only communication means to a module.
   Ex: compiler linker. Ex: unix shell
5. In which cases an Oracle can be automatic?
   Answer: If previous, reliable version of software application is available, or if function can be expressed mathematically
6. Describe briefly 'mutation testing' and its purpose **(x2)**
   Answer: A technique to evaluate how good a test suite is. Errors are inserted in the program under test (mutations). The more mutations the test suite finds, the better the test suite.
7. In the context of configuration management, what is the derivation history of a configuration item?
   Answer: History of versions and changes to them
8. What is the core content of the ISO 12207 standard?
   Answer: Hierarchical List of activities in the software process (primary, supporting, organizational).
9. Describe the 'repository' architectural style, its pros and cons **(x4)**
   Answer: Several applications communicate only through exchange of (standardized) data files. No direct interaction between applications, no call to APIs. Example: Eclipse. A core and many plugins attached to it that can communicate only through the core, but can be added at any time, independently from the core development. The repository file is the only point of interaction, centralized model: data evolution is expensive but high level of security, security and efficient backup management.
10. From the point of view of a user of a software application, what is more relevant, a defect or a failure in the software application? Explain why.
    Answer: Failure, because user is exposed to it
11. Describe briefly the problem of interactions and trade offs in non-functional requirements
    Answer: Some NF requirements may be in conflict (ex performance and security, performance and accuracy), so it is unfeasible to achieve both, and trade offs must be accepted
12. Given a software project with 4 team members, what are the risks if no configuration management is used?
    Answer: Concurrent access and inconsistent modifications of CIs, unavailability of past versions of CIs
13. Describe the scrum process, its pros and cons. (**x6)**
    Answer: Sprints iterations of fixed duration (max 4 weeks) producing a working application in increments. Ranking of requirements by end user / customer (requirements backlog), the ranking may change after an iteration. Stand up meeting (15') every day for coordination. Pro: focus on customer and his feedback, daily feedback in a team. Cons: no document, only code. Iteration fixed in time: pros to not loose time, cons for uncompleted work.

~~**14.** Describe the abstract factory design pattern, and when it can be used.~~

15. In the context of verification and validation, describe Weinberg's law **(x3)**

Answer: The creator of a program is unsuitable to test it – for emotional attachment to its code he does not want to find many errors and tends to overlook defects in it.

16. In the context of change control, describe the lock-modify-unlock technique, its pros and cons**(x3)**

Answer: Change control gives a discipline in who can modified what. Each developer does not work directly in the repository, but in his workspace and the two must be synchronized at important changes. In lock modify unlock a developer try to get a lock on a CI using checkout, works on it and after release the lock with check in. Two cannot work at the same time on the same CI and starvation if a developer forgets to unlock.

17. In the context of configuration management, explain what is a baseline and when is it used. **(x3)**

Answer: A baseline is a configuration (== a set of configuration items) that is stable (ex compiles, links, passes all regression tests). Not all configurations are baselines. Used to delivery an application internally / externally, while development continues on next version and to eventually rollback to it.

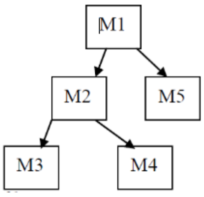
18. Describe the waterfall process, its pros and cons. **(x2)**

Answer: Activities (requirement, design, implementation, unit test, integration test, system test) are done in sequence (activity i+1 starts only after activity i is completed). Document oriented. Pros: easy structure of activities; agreement on design allows to allocate tasks to many, distributed workers/companies. Cons: delivery to customer and validation of requirements and system happen very late. Changes require to restart the process, slowness and lack of flexibility.

~~**19.** Describe the singleton design pattern, and when it can be used. **(x2)**~~

~~Answer: Creational pattern. Ensures that only one instance of the class is created. Ex in an operating system the load balancer must be unique, and could be implemented by a singleton.~~

20. Given this dependency graph propose and justify an integration strategy



If bottom up: test M3, test M4, test M2+M3+M4, test M5, test all

If top down: test M1 +stubM2 + stubM5, test M1+M2+M5+stubM3 + stubM4, test all

Bottom up requires less stubs / drivers

21. In the context of configuration management, explain what versioning is and why it is useful.

Answer: Versioning = keep copy of each instance of a configuration item (CI). This allows to keep the history of all modifications to a CI, and allows to roll back to any past instance of a CI

22. Describe a maintenance software process

Answer: Receive change requests (CR). Filter CR. Rank CR. Assign CR to developer – design, code, unit test, integration test. Merge CR with next release of application release.

23. Function A calls function B, that calls function C. You want to apply bottom up integration. How do you proceed?

Answer: Test C, test B+C, test A+B+C

24. Considering GIT, what are the three project sections that it defines, and how are they used?**(x2)**

Answer: Git directory (which stores metadata and database of versions), working directory (composed of a single checkout of one version of the project under modification), staging area (keep information on what will go on the next commit).

25. What is the typical lifecycle for a change? (draw states and transitions)

26. What measures can be used to evaluate the quality of software? **(x2)**

Answer: Fault density(Number of defects / size), MTBF(Number of defects found over a period of time), user satisfaction (questionnaire)

~~**27.** Describe shortly the Adapter Design pattern**(x2)**~~

28. Describe shortly the Test Driven Development technique

Answer: Write one test case that fails (using requirements), write corresponding code until test case passes, repeat until all requirements are satisfied and all test cases pass.

29. In project management, what are the units of measure for duration and effort? And what is the difference between these measures? **(x2)**

*Answer: Duration: calendar time (hours, days, weeks) – can be relative (two days) or absolute (from 12 7 to 14 7). Effort: person * hours. As by the definition, duration measures the time needed to complete a project, effort the amount of work needed. Given the effort of a project, calendar time depends on how many people work on it (staffing profile).*

30. ~~Describe shortly the Facade Design patter~~ **(x2)**

31. Explain and provide an example of a configuration item **(x4)**

Answer: it is the basic unit of the configuration management system and correspond to a document or piece of code, eg a class, under the configuration control. Each has a name and a version

32. Describe the copy modify merge strategy for controlling changes. List advantages and disadvantages **(x2)**

Answer: It allows many developers to check out and work in parallel on the same files under CM. When a developer check there is the need to merge his work with previous committed ones, manually or automatically.

33. Describe shortly the Pair Programming technique from XP **(x2)**

Answer: Code is written by a pair of people of same experience. One writes and one thinks and they swap periodically. Improvement in quality and duration, most of all with junior developers.

34. ~~Describe shortly the MVC Design pattern~~ **(x4)**

35. Someone asks you to develop a simple web site, a work that requires around 3 person months.
Which milestones would you define?

Answer: Month1, only GUI without logic, month 2 Gui + logic iteration1, month3 delivery

Which deliverables*? Answer: GUI mock up and requirements (M1, design (M1), code (M3)*

36. Describe what is an oracle in testing, the key problems related to it and how can it be implemented **(x2)**

Answer: the expected output in testing is given by an oracle, an ideal concepts that gives always the right prediction (assumption). It can be automatic (generated from software developed by other parties or previous versions of the program) or generated from formal requirement specifications or by an expert in the field.

37. List the types of defects that can be found in a requirement document **(x3)**

Answer: Omission/ incompleteness, Incorrect Fact, Inconsistency/contradiction, Ambiguity, Extraneous Information, Over-specification (design), Redundancy

38. What are the main differences between an iterative and a waterfall process?

Answer: Waterfall only performs one iteration

39. 140 person hours are equivalent to how many person months? Explain your answer

Answer: The conversion rate depends on national and company rules. If one person month == 140 person hours then result is of course 1. If one person month == 160 person hours then result is 140/160. And so on.

40. Describe the key steps in an inspection process of a requirement document. **(x3)**

Answer: planning (moderator choose participants and schedule event); he shows the goal of inspection and techniques to be used, for example perspective-based, scenario-based, defect-based reading and so on; each participant read the document and then they meet up to read the document together and discuss defects found. Defects have to be solved and then there is a follow up to verify fixes.

41. What are Function Points used for? How are they defined? *To estimate the size of code.*

42. What kind of problem 'traceability' identifies?

Answer: Linking requirements to design components (classes) / functions to code components / functions to test cases

43. A defect on a sw product signaled by a customer has been allocated to a developer, who has found the corresponding error. What happens next?

Answer: Fix the defect, rebuild, test, regression test, decide release to attach the patch to, release

44. The size of a project is estimated to be 600 FP. The project is developed in Java, past figures from the company tell that one FP is worth 20 to 30 LOC, and that productivity is 10-15 LOC per person days. How many person days would be required for the project? (show steps followed to compute your answer).

Answer: 600 FP → 12000 to 18000 LOC

Min effort 12.000 / 15 = 800 PD, max effort 18.000 / 10 = 1800 PD

45. A software product is developed custom for a bank. Development takes 18 months, operation and maintenance 10 years. Where do you expect to have the larger part of costs?

Answer: maintenance, because 10 years is way longer than 18 months, evolutive, corrective and enhancement maintenance will likely require more effort than initial development

46. A project is estimated to require 60 person months for development. What could be the calendar duration for development?

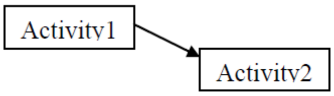
Answer: Depends on number of people working, 3 people, 20 calendar months, 4 people 15 months. But there is a limit on the number of people (ex 20 people is unrealistic, so 3 months unfeasible)

~~**47.** Describe shortly the Function Point method.~~

48. Describe shortly the 'incremental' software process

Answer: like the waterfall but the integration test is split and done incrementally, and each loop produce a part of the system delivered to customers for feedback.

49. A project has the following estimated Gantt.



Planned value of the project is 100 units, 50 on activity1, 50 on activity2. One month after start the project has consumed 60 units, activity1 is finished, activity2 is not. What is the Earned Value of the project?

Answer: EV = 50. Activity2 does not count because is NOT finished

50. "Cost of the project should be measured in Euros" what kind of requirement is this?

Answer: Domain requirement (a kind of NF requirement)

~~**51.** Describe shortly the Delphi estimation method.~~

52. What is the recommend length of an iteration in an agile project?

Answer: 4 -5 weeks at most

~~**53.** Describe shortly the Strategy Design pattern~~

54. What are the possible type of defects in a code module?

Answer: syntax, type and semantic errors; use of undefined or non-initialized variables, not used variable (data-flow), bad smells (too long classes or list of attributes), symbolic execution (translation of mathematical formulas).

55. In the context of project management, give the definition of 'deliverable'

Answer: A Deliverable is a product of the process. It may be either final or intermediate, can be internal (if it must be used only in the producer company) or external (if it is done for the customer or must be *validated by it). Some deliverables (like the requirements management or the design document) may have contractual values between customer and producer.*

56. Describe briefly the ‘layered’ architectural style
57. In the context of project management, give the definition of ‘milestone’ **(x2)**
    Answer: A Milestone is a key event or condition in the project, that can also serve as a synchronization point for its transitions between one phase to another. A milestone has effects on the subsequent activities: whether it is reached or not, the activities to be performed later may have to change.
58. In the context of project management, give the definition of ‘slack time’
    Answer: admissible delay to complete an activity without changing the end time of project
59. Provide an example of two conflicting non-functional properties of a software architecture
    Answer: Very high precision of computation (ex square root precision 10^-10) conflicts with Performance. The higher the precision, the slower the response time.
60. A project is estimated to require 40 person months. Give an estimated range of the required calendar time, and explain how you compute it.
    *Answer: A reasonable team for such a project could be 2 to 5 people. Assuming the team has the same number of people from start to end this gives a calendar range of 20 to 8 calendar months (computed as person * months / persons). A team of 40 people is unreasonable (too much coordination and communication overhead) so 1 month duration would be unfeasible. Similarly for 20, 30 people.*
61. Give the definition of effort in the context of project management, and its unit of measure**(x2)**
    Answer: Effort is the time taken by the staff to complete a task. It is measured in person hours (std ieee 1045), and also person day, person month, person year depend on national and corporation parameters.
62. What is a Configuration? **(x2)**
    Answer: A configuration is a set of Configuration Items (CI), each in a specific version, and it can be seen as a snapshot of the software at certain time. Some CIs may appear in different configurations, and also configuration has a version number
63. Give the definition of calendar time in the context of project management, and its unit of measure
    Answer: Days, weeks, months in calendar, either absolute (july 13) or relative (day1, day2 from ..)

~~64. Describe the CCB (Configuration Control Board) approach for change management~~

65. ~~Describe what are mixed revisions in Subversion?~~
    Answer: Mixed revisions are revisions of a working copy whose files have different revision numbers. Mixed revisions are possible only in working copies and not in the central repository. Example: Suppose you have a working copy entirely at revision 10. You edit the file foo.html and then perform an svn commit, which creates revision 15 in the repository. Therefore the Subversion marks the file foo.html as being at revision 15. The rest of the working copy remains at revision 10. This is a mixed revision.
66. Describe shortly the perspective based inspection technique (**x2)**
    Answer: Inspection where readers use different points of view (ex end user, designer, tester) to read the requirement document.
67. Function A calls function B, that calls function C. You want to apply top down integration. How do you proceed?
    Answer: 1-Test A using a stub for B
    2-Test A + B using a stub for C
    3-Test A+B+C
68. Class A has been developed and you should test it. What is better, do black box testing only, white box only, or both? Why?

Answer: Both. WB and BB have different criteria of writing test cases and, together, higher probability of finding them.

69. The cost of fixing a defect in a requirements document is higher if the project is in the requirement phase, or if the project is in the coding phase? Why?

Answer: Fixing a defect costs less in the requirements phase, essentially because no other artifacts (design document, code, test cases) exist yet at that stage. So fixing a requirements defect in the requirement phase means only changing one document, instead of many (if fixed in subsequent phases). Think to a house: it costs less to add a window in the project plan (= requirements analysis, you just correct tables or diagrams) rather than adding it when the house is in construction (= writing code, you must correct code) or it is already built and people is living there (=software is in use, you must write patches).

70. What are the most important functions of a configuration management tool?

Answer:

- *Identify and manage parts of software*
- *Handle whole history of repository*
  - *Handle branches, configurations and versions of software and eventually revert file/project back to a version.*
  - *Handle accesses and changes to parts (who, what and when)*

71. A defect can be injected, discovered, removed. Of course not injecting defects at all is the best option. Is this feasible? Argument briefly your answer.

Answer: Not injecting defects is not feasible: software is not defect-free for definition, since it's the result of a human activity. Probability of inserting a defect writing or changing code is different from zero (Adams's Law)

72. What is an inspection? How does it work?

Answer: inspection is a static V&V technique, used to verify the requirements/design document, test cases or the developed code. Performing an inspection means reading a document (or code) alone, and then repeating the reading with a group of people (typically more than three) that include the author of the document. The goal of an inspection is to find defects.

73. What is a deadline? Give an example of a deadline in a construction project.

Answer: Calendar date when a task/deliverable has to be completed/issued. Ex: roof ready by may 31 2009.

74. ~~According to the Cocomo model, how are related duration and effort?~~ (**x2)**

Answer: TDEV = 3 PM (0.33 + 0.2 * (B – 1.01)) with B=1 in the simplest model. Writing the formula was not necessary, key point was saying that duration (TDEV) depends on effort (PM) with exponential formula, exponent <1

75. What is a build? What are the related problems?

Answer: Process (including compilation, link, possibly testing and other activities) to produce an executable starting from source code modules and libraries. Often automated with a script. Problems: references and dependencies among modules, finding and using the right modules in term of type and version.

76. In the context of verification and validation, describe a static analysis technique Inspection or dataflow analysis or control flow analysis.

Answer: Control Flow Analysis is based on the Control Flow Graph of the program, and checks the control flow constructs. Data Flow Analysis track variables during execution, to find out anomalies eg unused variables.

77. User Mario commits file A on a subversion server, subversion notifies that the file is at revision 300. User Mario and other users execute 3 more commits of other files. Then John commits again file A. What is the revision number of A?

Answer: 304 on the repository and on John's working copy. (However, Mario's working copy is at 301, unless Mario updates)

78. What are the key differences between testing and debugging? **(x2)**

Answer: Testing tries to find failures, debug tries to discover and fix the correspondent fault(s)

79. Measures. Explains pros and cons of the Lines of code (LOC) measure. **(x2)**

Answer: not well estimable at first ad depends on language and programmer.

80. Software processes. Describe the key points of the evolutionary process

Answer: The evolutionary model (?) solves the problem of re-doing the whole waterfall that is present in the Incremental model: it is similar to the Incremental model, but requirements can change at each iteration. It is similar to prototyping, since at each iteration an incremental prototype is produced and the last prototype developed is the final product.

81. Testing. Describe the difference between bottom up and top down integration testing

Answer: The bottom up incremental integration starts from the dependency graph. First are tested the units that have no dependencies and then one unit at the time is integrated. If some defect is found, it should come from the last integrated unit or from the interaction, but not from the units already tested. The top down incremental integration starts from the highest-level unit and it uses stubs, then integrates the units below one by one (that to be tested used stubs). This testing method fits well if we develop the software in a top down approach and can detect early architectural flaws. This saves the definition of drivers for the bottom classes but require writing stubs for them (the opposite of the bottom-up approach).

82. Define fault and failure, underlining the difference

83. What is the definition of software reliability?

Answer: The best software is the one that has no faults, but a reliable one is a software with no failures, that is different.

84. Highlight the key differences between the incremental and iterative software processes **(x2)**

Answer: In incremental software process, requirement and design document are immutable like in the waterfall but the integration test is split and done incrementally, and each loop produce a part of the system delivered to customers for feedback. The iterative model consists in many iterations of the waterfall model, which of it produce a partial part of the project. The difference from the Evolutionary is that at the first iteration not all the requirements are written, but also those are made incrementally, while Evolutionary have the requirements written at the first loop and only can modify it to adapt to user wills.

~~**85.** List three risks in software engineering projects (from Boehm top 10 list)~~

86. What is a Gantt chart, and when is it used.

Answer: list of activities and for each activities how many people are needed and for how long and it define also scheduling and temporal constraints.