

Technical University of Dresden
Software Evolution in the Summer Semester 2013
Docent: Harry M. Sneed (MPA)

Exam Questions

1. Questions on the Nature of Software Maintenance:

1.) How can software maintenance be defined?

- As all work performed on a software product after the initial production release.

2.) How can software evolution be distinguished from agile software development?

- By the degree to which the product is used in production.
- By the degree of product change.
- By the length of the release intervals.

3.) How do software products age?

- The product falls behind the continually changing requirements and fails to keep step with the evolving environment in which it operates.

4.) What are the five phases of a software product life cycle according to Bennett and Rajlik ?

- Conceptual phase
- Development phase
- Evolution phase
- Conservation (Maintenance) phase
- Redemption phase (perhaps Resurrection)

5.) What are the four types of software maintenance according to Lientz and Swanson ?

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Functional enhancement

6.) What does a software product consist of ?

- Requirement documentation
- Design documentation
- Source code

- Testware
- User documentation

7.) What distinguishes adaptive maintenance from corrective maintenance?

- Corrective maintenance is the changing of the code to make it the way it should have been from the beginning.
- Adaptive maintenance is the changing of the code to make it perform the same functionality in a different way or with different data.

8.) What distinguishes adaptive maintenance from functional enhancement?

- Functional enhancement is the insertion of additional code to perform new functions that have not been performed before.
- Adaptive maintenance is the changing of the code to make it perform the same functionality in a different way or with different data.

9.) What is the purpose of perfective maintenance?

- To improve the quality of the product, for instance thru reengineering, refactoring or tuning activities.

10.) What does a Software Maintenance Engineer need to know? List out five items.

- How to comprehend a software system.
- How to conduct an impact analysis.
- How to estimate a maintenance task.
- How to correct an error in the software.
- How to change an existing software.
- How to add functions to an existing software.
- How to restructure procedural software.
- How to refactor object-oriented software.
- How to measure the size, complexity and quality of existing software.
- How to establish a metric database.
- How to redocument an existing software system.
- How to use a software repository.
- How to retest an altered software module.
- How to retest an altered software system.

2. Questions on Software Change Techniques:

11.) Which are the typical software maintenance tasks to be performed ?

- Error Correction tasks (Debugging)
- Change tasks (changing functions and data)
- Functional enhancement tasks (adding new functions and data)
- Reengineering tasks (improving the changeability of the code and data)
- Optimization tasks (improving the performance of the code)

- 12.) How are software maintenance tasks triggered ?
- By a maintenance request (change request or error report)
- 13.) What is the first step to be performed when a maintenance request is processed ?
- An impact analysis
- 14.) What is the purpose of an impact analysis ?
- To determine which source code members, which documents and which test cases are affected by the change ?
- 15.) What may be affected if a database table is changed ?
- All of the components which access that table
- 16.) What may be affected if a class is changed ?
- Any classes which inherit from that class and methods in other classes which invoke methods of that class
- 17.) What are the four approaches to changing a component ?
- Top-Down according to the calling procedures
 - Bottom-Up according to the called procedures
 - Forward from the input arguments to the output results
 - Backward from the results to the input arguments
- 18.) What are the four W Questions in the diagnosis of an error ?
- What occurs
 - Where does it occur
 - When does it occur
 - What effect does it have
- 19.) How are changes to the code validated ?
- By testing the new code against the data of the old code and comparing the new results with the previous results
- 20.) What is the purpose of maintenance auditing ?
- To ensure that the quality of the software product does not regress as a result of the maintenance tasks.

Questions on Procedural Software Maintenance Techniques:

21.) Which are the five semantic levels of software that need to be maintained?

- System requirements
- System design documentation
- Component design documentation
- Source code
- Test cases and test data

22.) Where is the starting point of a bottom-up approach to making a change?

- The source code

23.) Where is the starting point of a top-down approach to making a change?

- The system requirements

24.) How are change requests usually formulated?

- In natural language

25.) What is usually used to document a procedural system design?

- Data flow diagrams
- Tree diagrams

26.) What is usually used to document a procedural program design?

- Control flow diagrams
- Structograms (Nassi-Sheiderman Diagrams)

27.) What are the biggest dangers in changing procedural source code?

- Changing global data structures used by many procedures
- Changing sub routine interfaces

28.) Why do the test procedures have to be changed too?

- To be able to test the changed code against the new data

29.) What documents should be compared with each other ?

- Requirements specification against the Design documentation
- Design documentation against the source code

30.) What is compared in a procedural regression test ?

- The new control flow paths should be compared with the old ones
- The new data results should be compared against the old ones

Questions on Object-oriented Software Maintenance Techniques:

31.) Which are the four semantic levels of software that need to be maintained ?

- System requirements
- Object Design (UML)
- Source code
- Test cases and test data

32.) Where is the starting point of a parallel change process?

- System requirements
- Object Design (UML)
- Source code
- Test cases and test data

33.) Who changes what in a parallel change process ?

- Analyst changes the system requirements
- Designer changes the object Design (UML)
- Programmer changes the Source code
- Tester changes the test cases and test data

34.) Who makes the first change in a top down change process ?

- Analyst

35.) What has to be changed in the static object design ?

- Class diagrams
- Component diagrams
- Distribution diagrams
- Collaboration diagrams

36.) What has to be changed in the dynamic object design ?

- Sequence diagrams
- State transition diagrams
- Activity diagrams

37.) What are the biggest dangers in changing object-oriented source code ?

- Changing super or base classes
- Changing class interfaces

38.) What has to be tested when a class is changed?

- All classes that inherit from that class
- All classes whose methods use the methods of that class

39.) What is the wrong way to change a class ?

- By simply adding more attributes and methods to it

40.) What is the right way to change a class?

- By breaking it up into sub classes and by adding additional methods, i.e. refactoring it

Questions on Evolution Modelling:

41.) What is meant by top-down development ?

- Stepwise refinement of abstract concepts to more and more detailed concepts until one arrives at the elementary operations

42.) What are the main criticisms of top-down, i.e model-driven development ?

- Developers are not able to foresee the consequences of their design decisions
- Requirements are not stable
- Specification and Implementation are intertwined

43.) What is the bottom-up or test driven approach to software development ?

- First create a test
- Then code a solution to pass that test
- Repeat the test until the solution passes the test
- Then document the solution

44.) What is the meant by iterative incremental software development ?

- Develop a software in small packages or subsystems, one after the other
- Design, code, test, document and integrate one subsystem at a time
- Repeat until all subsystems have been implemented, then release the whole system.

45.) What is meant by evolutionary software development ?

- To develop the system as a whole in successive versions, one version at a time, whereby each new version is larger and better than the last one.

46.) What are the five approaches to software evolution ?

- Top-Down model-driven approach
- Bottom-Up code-driven approach
- Parallel approach

- Requirement driven approach
- Agile test-driven approach

47.) What are the three alternatives to describing a software ?

- with text, by means of the written word (code, artificial language, natural language)
- with pictures, by means of drawn diagrams (models)
- with speech, by means of the spoken word

48.) What are the two main types of models and how are they depicted?

- Data models with E/R diagrams
- Program models with UML and OCL

49.) What is the role of software?

- As communication between the human and the machine, but also between humans – man to machine and man to man.

50.) What is the purpose of maintaining a description other than the code?

- To have something to test against – an oracle, to be able to validate the code against the model or the requirements.

Questions on Reverse Engineering:

51.) Which are the two sides of software engineering ?

- Forward engineering
- Reverse engineering

52.) What is the purpose of Reverse Engineering ?

- To recover knowledge about an IT-System
- To discover what it does and how it does it

53.) What are four sources of information about IT-Systems?

- Software
- Databases
- Documents
- Human experts

54.) What are four types of Reverse Engineering tools based on their inputs ?

- Data reverse engineering tools
- Interface reverse engineering tools
- Program reverse engineering tools
- Process reverse engineering tools

- 55.) What is the role of reverse engineering in incremental software development ?
- To repeatedly recreate the design documentation from the latest version of the code
- 56.) What is the role of reverse engineering in software maintenance ?
- To provide a current design documentation of the system under maintenance
- 57.) What is the difference between static and dynamic program analysis?
- In static analysis the source code is processed
 - In dynamic analysis the machine code is executed
- 58.) What is the difference between reverse engineering and reengineering?
- In reverse engineering the code remains unchanged
 - Reverse engineering derives a plan from the product
 - Reengineering creates new code from old code
- 59.) What is meant by software recycling?
- To extract segments of code from existing systems in order to reuse them in another system, e.g. to create web services from existing components
- 60.) Why is it absolutely necessary to automate reverse engineering ?
- Because it is too costly to reverse engineer manually
 - Because manual reverse engineering is too error prone
 - Because reverse engineering has to be repeated every time the code is changed

Questions on Software Redocumentation:

- 61.) What are the four approaches to documenting programs ?
- Manually drawing or writing documents based on the code
 - Having a pre-processor which generates the code to also produce the documents
 - To build the documentation into the code (self documenting programs)
 - To automatically extract documents from the code
- 62.) Who do Programmers hate documenting their programs ?
- Because compared to programming documenting can be terribly boring.
 - Because they do not know how detailed the documentation should be.
 - Because there are no standards for the depth and extent of the documentation.
 - Because they are unwilling to share their knowledge.
- 63.) What is a call or perform graph good for ?

- To depict what modules call what other modules
 - To depict what procedures perform what other
- 64.) What are the data reference graphs good for ?
- To depict what data structures and variables are used by what procedures
 - To depict what data structures and variables are created/changed by what procedures
- 65.) What is a structogram good for ?
- To depict the control flow of a procedure or method
 - To depict the decision logic of a procedure or method
- 66.) What do the sequence diagrams describe ?
- The interactions between classes or methods
 - What methods are invoked by what other methods in a single process
- 67.) What do state diagrams describe ?
- The state transitions of objects
 - The sequence in which the objects are changed
- 68.) What does a data flow table describe ?
- The input and output variables of individual procedures
 - The arguments and results of individual methods
- 69.) How can processing rules or business rules be recaptured ?
- By collecting all of the operations which lead to their result and by picking up the conditions under which they are executed
- 70.) What is the technology called which produces graphic descriptions of existing programs ?
- Software Visualization

Questions on Software Repositories:

- 71.) What is a software repository?
- A database of software artefacts – code, data structures, requirements, test cases, etc. and their relationships to one another
- 72.) What are some of the functions provided by a repository?
- Impact analysis
 - Status reporting

- Document generation
- Code generation
- Query answering
- Quality control
- Consistency and completeness checking
- Project cost calculation

73.) What are the five basic components of a repository?

- Loader
- Viewer
- Reporter
- Generator
- Query Facility

74.) What are the extended components of a repository?

- Impact Analyzer
- Cost calculator
- Consistency and Completeness Checker
- Complexity measurement tool
- Quality measurement tool

75.) How does a repository integrate software tools ?

- It provides a common database with an individual interface to each tool

76.) What semantic levels does a repository have?

- Requirements Specification level
- Architecture design level
- Code level
- Test level
- Operations level

77.) What is the cardinality of the relationship between requirements and code modules ?

- M:N or many to many

78.) What is the cardinality of the relationship between code modules and test cases ?

- M:N or many to many

79.) What is the prerequisite to building up a software repository ?

- A metamodel in which all the stored elements, their attributes and their relationships are defined

80.) What are the main reasons for an impact analysis ?

- To identify those software artifacts affected by a change
- To estimate how much effort and risk is involved in a change

Questions on Technical Debt:

81.) What is technical debt?

- Technical Debt is the sum of all those things you should have done and never did!

82.) How do you measure technical debt?

- Technical Debt is measured as a relation of the costs of putting the software in the proper state relative to the costs of developing it.
- $\text{Technical Debt} = \text{Repair Costs} / \text{Development Costs}$

83.) What is the purpose of Code checking ?

- To discover flaws or deficiencies in the code
- To uncover rule violations
- To find errors in the code

84.) What are four methods of code checking ?

- Manual inspection
- Automated formal checking against the coding convention
- Checking against the specification
- Checking against a previous version

85.) What are code metrics used for?

- To measure the size, complexity and quality of the code
- To compare programs with one another

86.) What is the purpose of a coding convention ?

- To ensure a unified coding style
- To avoid dangerous coding constructs

87.) What is the purpose of a naming convention ?

- To ensure that all data attributes and methods are named everywhere in the same way
- To ensure that all data and method names are understandable

88.) Why should the size of methods and classes be limited ?

- To make it easier to understand them
- To make it easier to change them

89.) When was the first code auditor developed?

- The RXVP Fortran Auditor in the year 1975

90.) How can technical debt be avoided ?

- By continually monitoring the quality of the software – code, test cases, design and requirements documents.

Questions on Software Reuse:

91.) What is the biggest problem involved in reusing existing code ?

- The many dependencies the code has to the other code around it.
- It is similar to an organ transplantation

92.) At least what percentage of the code is reusable according to Capers Jones?

- At least 75%.

93.) What are four types of reusable software artifacts?

- Requirements texts
- Design models.
- Code units
- Test cases

94.) How is the economic benefit of software reuse computed ?

- The costs of developing and adapting the reused code relative to the costs of developing that code each time from scratch.
- The more times a piece of code is reused the cheaper it becomes.

95.) What are the four reuse techniques according to Endres?

- Porting
- Adaptation
- Patterns
- Components

96.) What is the purpose of a software framework?

- To provide all the technical functionality that an application needs to run in a specific environment.

97.) Give two samples of software frameworks?

- J2EE for Java
- DotNet for C#

98.) What distinguishes the include technique from the common module technique?

- With the include technique code members are inserted at compile time.
- With the module technique code modules are separately compiled and linked at build time.

99.) How are reusable code blocks recognized in legacy code?

- By tracing selected results back through the code and marking the statements traversed on the way to that result – data slicing.

100.) How is the reusability of a code block measured?

- $\text{Reusability} = 1 - [\text{Number external Dependencies} / \text{Number Statements}]$

Questions on Software Restructuring:

101.) What are the reasons for dealing with old procedural application systems in Fortran, PL1, COBOL, Natural and ABAP ?

- Because these systems make up more than 50% of the productive software systems
- Because these systems have to be maintained and extended
- Because these systems can provide reusable parts for new applications

102.) What are the four objectives of restructuring ?

- To increase the maintainability of procedural software systems
- To increase the testability of procedural software systems
- To prepare the procedural software for migration to object-orientation
- To enable the reuse of the procedural code in new applications

103.) What was the original goal of program restructuring ?

- To remove the GOTO statements
- To produce pure structured code from unstructured spaghetti code

104.) What was the classic approach to restructuring code ?

- To first recreate an intermediate design language description
- To then generate new source code from the design language description

105.) What are the five steps of the classical restructuring approach ?

- Transformation of code to graphs
- Modularization of the graphs
- Restructuring of the graphs
- Refinement of the graphs
- Generation of code from the graphs

106.) What are the five steps of the SoftRedo restructuring approach ?

- Realignment of the code
- Reassignment of hard coded variables
- Relocation of the IO operations
- Refinement of the control logic
- Restructuring of the control logic

107.) Why is it important to remove hard coded data ?

- Because this data cannot be readily changed
- Because this data cannot be changed without side effects

108.) Why is it necessary to break up large procedures into smaller independent procedures?

- So that the smaller procedures can be relocated
- So that the smaller procedures can be reused

109.) What should be the result of a reengineering project ?

- A better structured program with a three level architecture of control logic, processing logic and data access logic

110.) What is the major prerequisite to migrating old programs?

- First reengineer them, then migrate them

Questions on Software Refactoring:

111.) What are the reasons for refactoring object-oriented code ?

- To increase the maintainability of object-oriented software systems
- To increase the testability of object-oriented software systems
- To enable the reuse of the object-oriented code in other applications

112.) When should a method be split up into sub methods ?

- When the nesting level of the control logic is too deep
- When the method is too large
- When the method is performing different logical functions

113.) When should a local variable become a class attribute ?

- When the same variable is used in two or more methods

114.) What should be the maximum limits of a method ?

- Circa 20 statements

- Nesting level of 4

115.) What can be done with complex case statements ?

- Replace the cases with polymorphic methods

116.) When should a super class be created ?

- When several sub classes have similar methods ?

117.) When is meant by pulling up a method ?

- Transferring a local method of two or more sub classes to a super class

118.) Why should parameter objects be defined ?

- To reduce the number of individual parameters

119.) Why should assertions be inserted in the methods ?

- To ensure that they are producing a correct result

120.) What should be the deepest nesting level of a method?

- Three

Questions on Software Regression Testing:

121.) What is the meaning of regression testing ?

- To retest an existing piece of software which has in some way been altered, i.e it could have been corrected, adapted, perfected or enhanced

122.) Why is it necessary to isolate the software parts from one another?

- So they can be retested independently

123.) What percentage of total maintenance costs is caused by regression testing ?

- Circa 60 %

124.) Why is the cost of regression testing so high ?

- Because it is seldom possible to test only certain test cases
- All or most all of the test cases must be repeated

125.) How can one see what procedures or methods are affected by a test case ?

- By instrumenting the code and monitoring the execution
- 126.) How can the execution of methods and procedures be linked to a particular test case ?
- By comparing the start and end times of the test cases with the execution time of the methods and procedures
- 127.) What is the most common method of validating the correctness of changed software ?
- By comparing the results of the new version against the results of the old version
- 128.) What is meant by selective regression testing ?
- Only selected results are checked rather than all of them
- 129.) What types of tools are required for regression testing ?
- Instrumentor and Tracing tool
 - Capture/Replay Tool
 - Database Comparator tool
 - Interface Comparator tool
- 130.) What is the biggest unresolved problem in software engineering?
- Proving that the software does what it is supposed to do

Questions on Unit Regression Testing Techniques:

- 131.) What is the meant by input/output domain comparison?
- Statically comparing the new input variable types against the old input variables types
 - Statically comparing the new output variable types against the old output variables types
- 132.) What is the meant by data usage comparison?
- Statically comparing the usages of a variable in the new code with the usages of the same variable in the old code.
- 133.) What does a data flow analysis compare?
- It compares the data flow path of a selected result in the new code with the data flow path of the same variable in the old code.
- 134.) What is the purpose of a database or file comparison?
- To compare the content of a newly migrated or updated file or database with the content of the same file or database before the migration or update in order to recognize non-matching data.
- 135.) From where are test cases derived for making a unit regression test?

- From the original source code by means of static analysis.

136.) What is meant by functional equivalence?

- That a converted or revised version of a software component behaves in the exact same way as the original version, i.e., that it executes the code in the same sequence as before to produce the same results as before.

137.) What is the underlying theory for verifying the behaviour of a converted or reengineered code unit?

- No matter how you change the static order of the code blocks – paragraphs, procedures, method – the dynamic execution sequence must remain the same as before.

138.) What is meant by the term test data mutation?

- Making slight changes to the data before each test to see what effect that has on the code.

139.) Why is it necessary to know what test cases traverse which blocks of code, i.e. which methods?

- In order to know which test cases to test when a block of code is changed.

140.) Which two methods are used to link test cases with code blocks?

- Static analysis by extracting the test cases from the code.
- Dynamic analysis by tracing test cases through the code.

Questions on Regression Test Automation:

141.) What are the two main functions of a data testing tool?

- Test Data Generation
- Test Result Validation

142.) What are the three sources of generated test data?

- Existing Test Data
- Production Data
- Assertion Test Scripts

143.) What are the inputs to a test data generator?

- Existing Test Databases
- Existing Test Data Files
- Assertion Test Scripts
- User Parameters

144.) What are the outputs from a test data generator?

- New Test Databases
- New Test Data Files
- Test Data Reports

145.) What are test results validated against?

- Previous Database Output States
- Previous Output Files
- Result Assertions

146.) What are the outputs of a test result validator?

- Data Deviation Reports
- Data Metric Reports

147.) What was the approach to testing the new WKO systems?

- Selective Regression Testing

148.) Why is it necessary to automate the comparison of the test results?

- Because of the large number of data outputs which have to be controlled.

149.) How are test records compared?

- By matching their keys and comparing their attributes by name.

150.) Why is it necessary to trace execution paths through the system?

- To be able to compare the new execution paths with the previous ones.

Questions on the Laws of Software Evolution:

151.) Which are the main component types of a software product?

- User Interfaces
- Databases
- Programs
- Control Procedures

152.) What are the five levels in the evolution of databases?

- Hierarchical Databases
- Networked Databases
- Relational Databases
- Object Databases

- Object-Relational Databases

153.) What are the five levels in the evolution of programming languages?

- Assembler Level
- Procedural Level
- 4GL Level
- Object-oriented level
- Markup Level

154.) What is software evolution all about?

- Continually revising the product to cope with changes in the environment

155.) What are the main differences between software evolution and agile development?

- The degree of change
- The length of the release intervals
- The fact that software evolution is performed upon a productive product.

156.) What are the three types of software systems defined by Belady & Lehman?

- Throw away systems = S type
- Static systems = P type
- Dynamic systems = E type

157.) What are the five laws of software evolution defined by Belady & Lehman?

- Law of continual change
- Law of increasing complexity
- Law of decreasing quality
- Law of diminishing productivity
- Law of restricted growth

158.) What are the major consequences of the laws of evolution?

- You should not change too much at one time. No release should affect more than 10% of the code.
- Software must evolve gradually. Change must be made in small incremental steps.
- Management must control the rate of evolution.

159.) What are the reasons for causing software to die?

- Childhood mortality - the software is unsuited from the very beginning.
- The software is unable to adapt to the changing environment.
- The software becomes too defective.
- The software becomes inadequate to the problem.
- The software becomes obsolete.
- The software is dependent on persons who leave.

160.) How can the life of a software system be prolonged?

- By periodic renovation – fresh cell cure.
- By restricting external dependencies.
- By adhering to international standards.
- By keeping backup personnel.
- By continual quality assurance.

Questions on Software Product Line Management:

161.) What is a software product line?

- A family of related software applications with a common base line.

162.) What is the reason for developing product lines?

- To offer maximize functionality and variability while minimizing costs and time to develop, maintain and evolve.

163.) What is the difference between application and domain engineering?

- Domain engineering is concerned with the development of general purpose components related to a specific domain of knowledge.
- Application engineering is concerned with the construction of customized products for a specific application.

164.) Which are the 6 steps of a software product management process?

- Analyzing the domain.
- Defining the objectives
- Selecting a product strategy
- Devising a business plan
- Specifying the product requirements
- Implementing the product

165.) What is meant by product variability?

- There can be many instances of individual components, each of which behaves differently.

166.) Which are the three semantic levels of a product at which variability can be build in?

- At the requirement level.
- At the design or architectural level
- At the code or implementation level

167.) Which is the best form of organization for product line development and evolution?

- The matrix organization

168.) What is meant by a product scenario?

- The process through which the components are selected and assembled to create a product version.

169.) How are components of a product selected?

- They are rated by the degree to which they fit to the current product requirements.

170.) What purpose does the component evaluation card serve?

- To evaluate individual components based on their generality, i.e. their ability to fit into many products.

Questions on Software Maintenance Processes:

171.) How was software maintenance originally viewed?

- As the last step in the software development process.

172.) Who introduced the spiral life cycle model and when?

- Barry Boehm in 1987.

173.) Which was the first official software maintenance process published and by whom?

- By the US-Bureau of Standards in 1983.

174.) Which was the first ANSI/IEEE standard for software maintenance processes and when did it appear?

- ANSI/IEEE Standard 1219 in 1992.

175.) Which are the four steps of processing a maintenance order according to the ANSI/IEEE standard?

- Analysis
- Planning
- Execution
- Validation

176.) What is a project time box in software evolution?

- The time frame in which the next release has to be prepared and delivered.

177.) Why is there no one unified process for software maintenance?

- Because the process depends on the type of task to be performed.

178.) What are the four main types of maintenance tasks?

- Service tasks.
- Change tasks.
- Renovation tasks.
- Enhancement tasks.

179.) Why is software migration not considered a maintenance process?

- Because it is not a reoccurring process.
- It is a one time process to transport the software product to another environment.

180.) What is the goal of a software integration process?

- To link two or more software systems together.

Questions on Software Maintenance Organization:

181.) What distinguishes project from product management?

- Project management is limited by time and scope
- Product management is continuous.

182.) What is a project in software maintenance and evolution?

- A project produces a new release of a product.
- A project produces an altered product.
- A project is a product state transition.

183.) What are the key business entities of an enterprise model?

- Business resources.
- Business objects.
- Business processes.
- Business interfaces
- Business requirements

184.) What is the purpose of a project in an enterprise model?

- To implement one or more outstanding requirements.

185.) What is the result of a software project in an enterprise model?

- A new or altered software product.

186.) What are some of the tasks performed upon a software product?

- The product can be changed.

- The product can be enhanced.
- The product can be corrected.
- The product can be renovated
- The product can be migrated.
- The product can be integrated.
- The product can be tested.

187.) What kinds of teams are required to maintain and evolve a software product?

- A service team for trouble shooting.
- An evolution team for creating new releases.
- A test team for testing fixes and new releases.
- A configuration management team
- A product support team – product office.

188.) How many source statements can be maintained by one person?

- Up to 100,000 source statements.

189.) What is the minimum level of product support?

- A service team for correcting errors and making environmental changes.

190.) What is the maximum level of product support?

- A service team for correcting errors and making environmental changes.
- A maintenance team for producing new releases.
- A test team for testing corrections and new releases.
- A configuration management team for administering the product.
- A product support team for maintaining the tools and the repository.

Questions on Software Maintenance Economics:

191.) What causes the greatest costs in software maintenance according to the study of Lientz and Swanson?

- System enhancement – adding new functionality.

192.) According to the IBM study how many maintenance requests were there per year for each maintenance engineer in a large mainframe environment?

- Circa 80 a year.

193.) What was the average release interval according to the Nosek and Palavia study in 1990?

- 3 months.

194.) What was the average effort per maintenance request in the Norwegian Telecom Study?

- 5 person days.

195.) How many lines of code were being maintained on average by a maintenance programmer at the Norwegian Telecom?

- 92,000 LOCS.

196.) What was the conclusion of the Krogstie's study at the Norwegian Telecom?

- Organizations, which have modern development methods and tools in service, do not have less maintenance effort than others.

197.) What are the three main factors which influence software maintenance costs?

- Environment
- Personnel
- Product.

198.) How can you compute the annual maintenance productivity?

- $\text{Annual Change Rate} \times \text{System Size} / \text{Annual Effort}$.

199.) What is the COCOMO equation for computing annual maintenance effort?

- $\text{Annual Maintenance Effort} = 1.2 \times [(\text{Annual Change Rate} \times \text{Development Effort}) \times (1.5 - \text{Quality Rating})]$

200.) What is the cost of a standard software system per year relative to the cost of maintaining one's own system?

- < 50%.

Questions on Value-Driven Software Maintenance:

201.) Who is responsible for steering the evolution of a software product?

- Product Manager

202.) What is meant by value-driven Maintenance?

- To select the corrections, changes and enhancements to a software product based on the added value they provide. It should be computable in Euros and Cents.

203.) What is the name of the Austrian Economist who discovered the value of maintaining capital goods?

- Friedrich Hayek
- 204.) How is the value of a capital good determined?
- Relative to the alternative goods offered for the same purpose?
- 205.) Why is it economical to invest in the maintenance of an existing system?
- Because it saves the costs of having to replace that system which are much higher.
- 206.) What is meant by complementarity in regard to software products?
- That the value of interacting software products sinks in proportion to the loss of value of a single product.
 - If systems B and C are dependent on system A then a loss of value to system A will result in a loss of value to systems B and C.
- 207.) What is the purpose of a cost/benefit analysis in software evolution ?
- To determine if the benefits of a maintenance request (correction, change, enhancement or renovation) outweigh the costs
- 208.) How do you compute the Return on Investment of a maintenance project (ROI)?
- $ROI = (\text{benefit} - \text{costs}) / (\text{costs} * \text{risk})$
- 209.) How do you compute the value of a reengineering or refactoring project?
- $Added\ Value = (1 - (NewSize/OldSize))$
 - $+ (OldComplexity - NewComplexity)$
 - $+ (NewQuality - OldQuality)$
- 210.) What are the alternate solutions to the Maintenance Problem?
- Maintain Status Quo
 - Replace with a standard product
 - Renovate/reengineer
 - Migrate
 - Wrap
 - Outsource

Questions on Software Product Metrics:

- 211.) What are the three dimensions of a software product?
- Quantity
 - Complexity
 - Quality.

212.) What four size metrics can be used to measure the quantity of a software product?

- Statements or LOCs
- Function-Points
- Data-Points
- Object-Points.

213.) How is software complexity measured?

- As the relation of system elements to their relationships, i.e. nodes to edges
- $\text{Complexity} = 1 - (\text{Elements} / \text{Relationships})$

214.) What kinds of complexity can be measured?

- Structural Complexity
- Control Flow Complexity
- Data Flow Complexity
- Interface Complexity
- Data Complexity
- Access Complexity
- Language Complexity

215.) Give an example of a metric for control flow complexity?

- McCabe Cyclomatic Complexity = Edges – Nodes x (2 x Number of Subgraphs)

216.) How is software quality measured?

- As the relation of an actual (Ist) measure to an ideal (Soll) measure
- $\text{Quality} = \text{Ist-Measure} / \text{Soll-Measure}$

217.) Why is it necessary to define an arbitrary upper and lower bound for software quality?

- Because quality is relative.

218.) What is the most appropriate scale for measuring software quality?

- Rational Scale.

219.) Which ISO Standard defines software product quality?

- ISO-9126.

220.) Which rating scale is recommended by that standard for classifying software quality?

- 0.8 to 1 = Excellent Quality
- 0.6 to 0.8 = Good Quality
- 0.4 to 0.6 = Satisfactory Quality
- 0.0 to 0.4 = Poor Quality

Questions on Software Maintenance Cost Estimation:

221.) What are the three classes of maintenance costs to be calculated?

- Costs of correcting the current release.
- Costs of producing the next release.
- Overhead costs.

222.) Why is it so hard to predict the costs of correcting the current release?

- Because you do not know how many errors will come up.
- Because you can not know how much effort will go into correcting an error.

223.) What are the arguments to calculating the costs of correcting a release?

- Size of last release
- Complexity of last release
- Quality of last release
- Number of defects in last release
- Test coverage of last release
- Effort required to correct defects of last release
- Size of current release
- Complexity of current release
- Quality of current release
- Test coverage of current release

224.) How is the system size adjusted?

- Multiplying the size by the complexity and the quality ratios.

225.) How should the size of an existing system be measured?

- In statements, function-points or object-points.

226.) What is a prerequisite to estimating the costs of a new release?

- Impact analysis of the change requests.

227.) What does an impact analysis reveal?

- Those software artifacts affected by the change requests.

228.) How is productivity measured in software evolution?

- Number of statements changed or added / effort in person days
- Inverse of number of errors corrected / effort in person days

229.) What are the major overhead costs in software evolution?

- Costs of product and project management
- Configuration management
- Quality assurance
- Change management
- Problem management

230.) What is the prerequisite for an automated maintenance cost estimation?

- A metric database in which the size, complexity and quality metrics of all maintained software artifacts are stored.

Questions on Software Quality Management:

231.) What is product quality?

- The sum of all characteristics possessed by a given product enabling it to fulfil a set of given requirements with given resources within a given context within a given time. The requirements and constraints given are derived from the purpose of the product.

232.) Which are the three dimensions of software quality?

- Revisional = ease of adaption and enhancement
- Transitional = reusable, portable, interoperationable
- Operationable = performance, efficiency, security, etc.

233.) What are levels of the software quality hierarchy?

- Quality Goals
- Quality Characteristics
- Quality metrics

234.) Why is it necessary to have an independent quality control of software at all levels?

- Because the software developers and maintainers cannot be trusted to work properly

235.) Which are the main quality control measures to be taken before testing begins?

- Requirement Analysis
- Design Reviews
- Code Inspections
- Code Audits

236.) Which are the four ways to check a software result – document, code or test?

- Check against itself (completeness & consistency)
- Check against standards and conventions (conformity)
- Check against the specifying or proceeding document (verify correctness)
- Check against human assessment (compatibility with requirements)

237.) What is the role of metrics in software quality assurance?

- Metrics is an attempt to place quality assurance on an objective basis
- In order to objectively assess quality it must be measurable, i.e. it must be quantified
- Metrics is the quantification of quality.

238.) Who should check the quality of the software?

- The developers themselves
- The project leader
- Independent testers
- QA Deputies

239.) What are the objectives of quality assurance in software evolution?

- To monitor increasing complexity and decreasing quality
- To ensure continual conformity to conventions and standards

240.) Which are the five basic approaches to assuring software quality?

- Psychological approaches
- Constructive approaches
- Analytical approaches
- Empirical approaches
- Reengineering Approaches

Questions on Software Error Projection:

241.) Why do the costs of error correction remain relatively stable?

- Because the sinking number of errors is offset by the increasing costs of error correction.

242.) How much more does an error correction in production cost relative to the correction at coding time according to the first TRW study?

- 10-20 times more.

243.) What was the difference between correcting errors at test time to correcting errors in production in the GEOS project?

- 6 hours at test time versus 25 hours in production.

244.) What is the primary metric for measuring the degree of correctness?

- Defect density.

245.) How is defect density measured?

- Number defects per thousand statements.
- Defects / Statements

246.) How is the defect insertion rate measured?

- Number of defects per person day
- Defects / Person days

247.) What type of application has the highest defect rate?

- Web applications.

248.) How is the defect removal rate computed?

- Defects removed before release / total defects removed.

249.) What is the U.S. average defect removal rate?

- Circa 75%.

250.) Which are the three variables for computing the remaining error probability?

- Number of statements.
- Number of errors discovered
- Number of statements tested

Questions on Software Product Measurement:

251.) What are the main criteria for metric selection?

- Effectiveness
- Feature coverage
- Low effort to measure
- Empirical Evidence

252.) What types of product metrics can be collected?

- Requirement metrics
- Design metrics
- Code metrics
- Test metrics
- Performance metrics

253.) How are software product metrics obtained?

- By analysing the software artifacts – requirements, design models, code and test cases

254.) How is the reliability of a software product measured?

- By comparing the weighted error rate with the product size in statements or lines

255.) How is the maintainability of a software product measured?

- By comparing the effort required to make changes with the size of the changes

256.) What is the IEEE standard for software product metrics?

- ANSI-IEEE Standard 982

257.) What does the GQM Method mean?

- Goal, Question, Metric method

258.) How is the GQM Method applied?

- First, set the goals to be measured
- Second, ask questions on how the goals are to be reached
- Third, define measurements to answer those questions

259.) How can metric data be transferred?

- As XML files

260.) How can metric data be stored?

- In relational database tables

Questions on Software Product Evaluation:

261.) What are the two main reasons for a software product evaluation?

- To determine what to do with the product
- To determine the value of the product
- To collect data for planning a reengineering or migration project

262.) What are the four sizes of a software product?

- Functional size or quantity of functions fulfilled
- Data size or quantity (number of tables, files, records, attributes, keys, etc.)
- Procedural size or quantity (number of statements, branches, references, calls, etc.)
- Architectural size or quantity (number of components, modules, classes, etc.)

263.) What kinds of complexities does a software product have?

- data complexity
- data flow complexity

- processing complexity
- access complexity
- interface complexity
- structural, ie. architectural complexity

264.) What are the two main categories of product quality attributes?

- internal quality attributes
- external quality attributes

265.) What are examples of external quality attributes?

- Reliability
- Security
- Performance
- Usability

266.) What are examples of internal quality attributes?

- Modularity
- Testability
- Reusability
- Portability
- Convertibility

267.) Which are the three main quality goals according to McCall?

- Functionality
- Rentability
- Reusability

268.) How is the achievement of a quality goal measured?

- Soll / Ist
- Quality achieved / Quality planned

269.) What are the four steps in preparing a software product evaluation according to the ISO standard?

- Quality Requirement Definition
- Metric Selection
- Setting up a metric evaluation scale
- Defining the evaluation criteria

270.) What kind of metric scale is recommended by the ISO standard?

- a rational scale from 0 to 1 with 0.5 as the median

Questions on Software Problem Management:

271.) What is the difference between an incident and a problem?

- An incident is any event which disturbs the user.
- A problem is a flaw in the product.

272.) Which are the standard three levels of support?

- Level 1 = Telephone or Email consultation, i.e. Hotline or Helpdesk.
- Level 2 = Personal intervention of the customer support or trouble shooting team.
- Level 3 = Responsible maintenance group is called upon to correct error.

273.) Which are the five classes of errors according to the IEEE Standard P-1044?

- fatal,
- severe,
- problematic,
- disturbing
- unknown

274.) Why should errors be weighted?

- Because the effects of the errors are quite different depending on their severity.

275.) Which are the seven steps in a error tracking process?

- Opening the error report
- Classifying the error
- Analyzing the error
- Recording the error
- Initiating error correction
- Validating the error correction
- Closing the error report

276.) What attributes should a problem report contain? Name at least five.

- Reporter of the problem
- Date and Time of the incident
- Version the user was working with
- Severity
- Priority
- Category
- Effect
- Description

277.) What is the purpose of an error database?

- Tracking the status of the errors

- Enabling an error statistic or defect analysis
- 278.) How is error density computed?
- Weighted errors / system size (Locs, Statements, Funct-Points, Object-Points)
- 279.) Where do most errors originate?
- In the requirements specification.
- 280.) When does the error frequency reach its peak?
- Right after the first release when the customers begin to work with the system.

Questions on Software Change Management:

281.) What is the prerequisite for an orderly change management ?

- Product line development strategy

282.) What is the primary goal of Change Management?

- To sustain the continuity of the service.

283.) Which are the major constraints to Change Management?

- Limited personnel capacity,
- Product inflexibility,
- Lack of product knowledge, documentation
- Organizational resistance to change

284.) How should a change process begin?

- With an analysis of the change request

285.) How should a change process end?

- With a regression test of the changed system

286.) What are three alternate approaches to implementing a change?

- Top-Down,
- Bottom-Up,
- Parallel.

287.) Which approach is used most in practice ?

- Bottom-Up

288.) What are the reasons for not updating the requirement and design documentation?

- The programmer has no time
- Programmers hate dealing with documentation
- There is no one responsible for the documentation

289.) What happens when the specifications and designs are not updated?

- The system becomes less comprehensible
- Knowledge of the system is lost
- Dependence on key persons increases
- System quality decreases

290.) How can one avoid system deterioration, i.e. it becomes less and less changeable?

- By continually updating the documentation
- By periodically renovating, i.e. refactoring, the code
- By controlling the quality after every release

Questions on Software Configuration Management:

291.) What does Software Configuration Management entail?

- Storing, administrating and securing all artefacts belonging to a software product

292.) What types of software artifacts belong to a software product?

- Documents,
- Diagrams,
- Tables,
- Source code
- Test cases

293.) What should a requirement documentation consist of?

- Context diagrams,
- Process modelling diagrams,
- Data modelling diagrams,
- User interface designs,
- Business rules,
- Use case diagrams
- Usecase specifications
- Acceptance criteria

294.) What should an object-oriented system design consist of?

- Class diagrams,

- State diagrams,
- Timing diagrams,
- Interaction or sequence diagrams,
- Activity diagrams
- Component diagrams,
- Distribution diagrams
- OCL Notations

295.) What should the test documentation consist of?

- Test plans,
- Test designs,
- Test cases,
- Test procedures, i.e. test scripts,
- Test logs,
- Test reports.

296.) Which IEEE Standard defines Software Configuration Management ?

- ANSI-IEEE-828

297.) What is a Check-In/Check-Out procedure intended to prevent?

- That unauthorized persons access system configuration items
- That no more than one person at a time can alter a configuration item.

298.) Which states may a software configuration item be in?

- planned,
- in work,
- completed
- validated

299.) Which steps are involved in a Release/Rollout process?

- Version Extraction,
- Compile & build
- Distribute

300.) What advantages does a centralized software configuration management offer?

- Universal access to the configuration items, i.e. artifacts,
- Better control over the state of the product
- Centralized reporting.