1. Stages of software development. Features of various software development processes.
2. Definition of requirements for the program being developed. Types of requirements. Hierarchy of requirements, requirements tracking. Planning game in extreme programming technology. The role of requirements tracking in software quality assurance.
3. Ways of formalizing requirements. Technical specification for the GOST 19.201-78 program
4. Ways Formalization Requirements. Software requirements specification: IEEE 830 standard.
5. Ways of formalizing requirements. Use cases. UML use case diagram. Relations between use cases. Use case description templates.
6. Technical design. The role of design in ensuring the quality and reliability of software. Design solutions(Steps, Stages). The program call tree. Specification of a function (method).
7. Technical design. Explanatory note GOST 19.201-78. Description of the program GOST 19.402-78.
8. Testing the program. Types of testing. Use of different types of testing at different stages of software development. Areas of testing. Non-functional tests. The role of testing in ensuring and controlling the quality and reliability of software. Test-driven software development(Gost 19.301-79 test program).
9. Unit testing. Unit test automation tools. Develop unit tests using the VisualAssert and QTestLib libraries.

10. Software reliability tools built into programming languages. Statements. Exceptions. Debug libraries.

11. Version control of the program. Centralized version control systems: basic concepts, Algorthms, basic commands, typical sequence of operations when working. Advantages and disadvantages of centralized version control systems

12. Version control of the program. Distributed version control systems: basic concepts, basic commands, typical sequence of operations when working. Advantages and disadvantages of distributed version control systems.

13. The process of building the program(Project). The goals of the assembly(build). The role of the preprocessor, compiler, and linker in the build process. Types of build commands. Managing assemblies. Basic settings of the preprocessor, compiler, and linker. User-configurable stages of building the program.

14. Stage of the working project. Connection of program documentation with changes in the program code in the process of development. Automation of the generation of program documentation from the source code of the program using the doxygen program.

15.Program optimization. Optimization goals. Factors affecting program performance. Microoptimization problem. Program profiling to identify code that affects application performance. Modern profiling tools, steps.

16. Assessment of the quality and reliability of the developed software.

Tools for automating the evaluation of the developed software. Static analysis of the program code. Dynamic analysis of the program code.

17. Evaluation of the quality and reliability of the developed software. Determination of the program code coverage by tests.

18.Software metrics. The role of metrics in ensuring the quality of software. Indicators and metrics of software quality defined by the ISO/IEC 9126 standard.

19.Software Metrics. Most Commonly Used Metrics. Physical Metrics: Number of Lines of Source Code and Percentage of Commentary. Functional Units. Software Complexity Measures: Halstead Metrics, McCabe Cyclomatic Complexity, Chapin Metrics, Robert Cecil Martin Batch Metrics.

20. Operational documentation of the program. System Programmer's Manual GOST 19.503-79. Programmer's Manual GOST 19.504-79. GOST 19.105-78.