**The Federal State Autonomous Institution of Higher Education**

**"National Research University Higher School of Economics"**

 **Faculty of Computer Science**

**School of Software Engineering**

**Рабочая программа дисциплины**

Методология программной инженерии

(На английском языке)

Software Engineering Methodology

для образовательной программы «Системная и программная инженерия»

направления подготовки 09.04.04 «Программная инженерия»

уровень - магистр

Разработчики программы

Салех Х.М., к.т.н., доцент, hsalekh@hse.ru

Александров Д.В., д.т.н., профессор, dvalexandrov@hse.ru

Одобрена на заседании департамента программной инженерии «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Руководитель департамента Авдошин С.М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждена Академическим советом образовательной программы

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018 г., № протокола\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Академический руководитель образовательной программы

Дегтярев К.Ю. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2018

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.*

## Course Description

**a. Title of the Course**

The title of the course is Software Engineering Methodology.

**b. Pre-requisites**

Studying of the “Software Engineering Methodology” discipline is based on the following courses:

* Programming,
* Introduction to Software Engineering,
* Functional and Logic Programming,
* Data Bases,
* Program Verification,
* Quality Provision and Testing,
* Operating systems,
* Software Constructing,
* Algorithms and Data Structures,
* Foreign language (English),
* Basics of software Application Development

**c. Course Type**

The course is compulsory.

**d. Abstract**

The course is offered to the students of the Master’s Program «System and Software Engineering (SSE) » at Faculty of Computer Science of the National Research University - Higher School of Economics.

This compulsory course belongs to special subject curricula unit (М.2 unit/ Base module [Special subject disciplines M.2] of 2017-2018 academic year's working syllabus) covered by the list of training courses of master's program (1st year of studies).

The course ‘Software engineering methodology’ presents the engineering discipline which is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.

The main objective of the training course is to examine and discuss with students fundamentals and principles of Software Engineering (SE), get familiar with common methods and standards of SE, development and evolution of complex multiversion and replicated software solutions.

The course is aimed to help students to develop skills that will enable them to construct software of high quality, reliable, and that is reasonably easy to understand, modify and maintain.

It is a double module course, which is delivered in modules #1 and #2 of the first academic year. Number of credits is 5. Total course length is 190 academic hours including 64 auditory hours (24 Lecture (L) hours and 40 Seminar (S) hours) and 126 Self-study (SS) hours. Academic control forms are one home assignment, one test, and one written exam after module #2.

## Learning Objectives

The main objectives of mastering the «Software engineering methodology» discipline are to:

1. Study the place and role of software engineering, place it in a broader systems context and present the concept of software engineering processes and management.
2. Learn the processes, techniques and deliverables that are associated with requirements engineering, system modeling, formal specification
3. Acquire skills in software design, module development (including agile methods, software reuse, CBSE and critical systems development), and integration.
4. Focus on techniques for software verification, testing, critical systems validation and further documentation
5. Discover the importance of management topics in software engineering: managing people, cost estimation, quality management, process improvement and configuration management.
6. Design and experiment with software prototypes, Select and use software metrics.
7. Communicate effectively through oral and written reports, and software documentation
8. Demonstrate professionalism including continued learning and professional activities.
9. Successfully assume a variety of roles in teams of diverse membership.
10. Have an ability to work with other people in a team, communicating computing ideas effectively in speech and in writing;
11. Build solutions using different technologies, architectures and life-cycle approaches in the context of different organizational structures.

The students will learn how to analyze, design, program and test software projects, gather requirements and write user stories, use cases and use case diagrams, draft user interfaces and create functional system prototypes, translate application requirements into working code, simplify complex systems using modern object-oriented analysis and design techniques and ensure software quality with both manual and automated testing techniques.

## Learning Outcomes

Upon mastering the discipline, the successful students will:

*Know and understand:*

* major lifecycle models and methodologies) used for software development of the system development.
* a wide range of principles and tools available to the software developer, such as software process methodologies, choice of algorithm, language, software libraries and user interface technique;
* the principles of object-oriented software construction;
* the software-development process, including requirements analysis, design, programming, testing and maintenance;
* the range of situations in which computer systems are used, the ways in which people interact with them;
* communication issues in large, complex software projects.

*Be able to:*

* Define software engineering and explain its importance;
* Discuss the concepts of software products and software development processes;
* Develop and write a software project proposal;
* Develop and write a Software Requirements Specification;
* Design a software system;
* Document the design of the software system;
* Document the implementation  of a software system;
* Define CASE tools and software development environments;
* Write a Test plan;
* Explain the importance of process visibility and software design techniques
* Conduct a software inspection and review;
* Document software testing;
* Investigate software testing, documentation and maintenance techniques for developing larg;e software systems

*Acquire skills/experience in:*

* effectively participate in team-based activities;
* structure and communicate ideas effectively, both orally, in writing, and in cases involving a quantitative dimension;
* use IT skills and display mature computer literacy, work independently and with others;
* manage learning and self-development, including time management and the development of organizational skills;
* develop and apply testing strategies for software applications;
* develop software applications in a development environment that makes use of commonly supported tools;
* display personal responsibility by working to multiple deadlines in complex activities
* identify some of the main risks of software development and use;
* use network information services;
* Prepare and deliver coherent and structured verbal and written technical reports;
* use the scientific literature effectively and make discriminating use of Web resources;
* analysis of system requirements and the production of system specifications;
* use appropriate computer-based design support tools.

## Course Plan

The course is within the block of basic disciplines. Number of credits: 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Topic Name | Course Hours, Total | Audience Hours | Self Study |
| Lectures | PracticalStudies |
| Module 1 (32 hrs).  |
| 1 | Software engineering fundamentals  | 13 | 2 | 2 | 9 |
| 2 | Software process models | 14 | 2 | 3 | 9 |
| 3 | Project management activities | 14 | 2 | 3 | 9 |
| 4 | Software requirements | 14 | 2 | 3 | 9 |
| 5 | Requirements engineering processes | 14 | 2 | 3 | 9 |
| 6 | System models and architecture  | 13 | 1 | 3 | 9 |
| 7 | System design | 13 | 1 | 3 | 9 |
|  | Module 1, totally: | 95 | 12 | 20 | 63 |
| Module 2 (32 hrs).  |
| 8 | Software development | 13 | 2 | 3 | 8 |
| 9 | Software reuse and evolution | 13 | 2 | 3 | 8 |
| 10 | Verification and validation | 11 | 1 | 2 | 8 |
| 11 | Software testing | 10 | 1 | 2 | 7 |
| 12 | Managing people and cost estimation | 11 | 1 | 2 | 8 |
| 13 | Software risk management | 11 | 1 | 2 | 8 |
| 14 | Quality management, process improvement | 13 | 2 | 3 | 8 |
| 15 | Configuration management | 13 | 2 | 3 | 8 |
|  | Module 2, totally: | 95 | 12 | 20 | 63 |
|  | TOTAL: | 190 | 24 | 40 | 126 |

***4.1 Course Contents***

**Topic 1. Software engineering fundamentals**

♦ Topic outline:

* Introduction to the “Software engineering” course. Definition of the key questions about software engineering;
* Organizations, people and computer systems. Explanation the difference between software engineering and computer science;
* Software costs structure and software engineering methods introduction;
* Introduction of the ethical and professional issues that concern software engineers.
* Software engineering main tools and techniques observation

**Topic 2. Software process models**

♦ Topic outline:

* Introduction of a software process models. Generic process models and when they may be used;
* Process main activities for requirements engineering, software development, testing and evolution;
* Process iteration: incremental delivery and spiral development. Advantages and risks.
* The Rational Unified Process model introduction;
* CASE technology for software process activities;
* Summary of software process models: benefits and risks of each key process model.

**Topic 3. Project management activities**

♦ Topic outline:

* The main tasks undertaken by project managers, roles and responsibilities in managing software development process;
* The planning process and the project planning activities. Graphical schedule representations;
* Project staffing and scheduling process definition. Project costing;
* Project monitoring and reviews. Report writing and presentations;
* Critical system definition and key properties (system dependability, availability, reliability, safety, security) ;

**Topic 4. Software Requirements**

♦ Topic outline:

* Definition of a requirement;
* Different types of requirments:
	+ Concepts of user and system requirements;
	+ Functional and non-functional requirements;
	+ User and system requirements;
* The software requirements document;

**Topic 5. Requirements engineering processes**

♦ Topic outline:

* Requirements engineering processes activities and their relationships;
* How to conduct feasibility studies and how to measure the results;
* Techniques for requirements elicitation and analysis;
* Requirements validation and the role of requirements reviews;
* The role of requirements management in support of other requirements engineering processes;
* Problems of requirements analysis and requirements change management.

**Topic 6. System models and architecture**

♦ Topic outline:

* Model types observation: behavioural modelling, data modelling and object modelling;
* Unified Modelling Language (UML) and CASE workbenches for system modelling;
* Specification in the software process: formal, sub-system, behavioural specification;
* Introduction to system architecture;
* Architectural design process and its importance:
	+ Distributed systems, application and object architecture
	+ Multiprocessor architectures
	+ Client-server architectures

**Topic 7. System design**

♦ Topic outline:

* Definition of a system design process;
* Introduction in to design process activities;
* Design process models: object-oriented design, real-time design, interface design, etc.;
* UML design process brief introduction;
* System design evolution and development.

**Topic 8. Software development**

♦ Topic outline:

* Introduction to a software development process;
* Iterative and incremental development key features;
* Software development methods overview:
	+ Agile development methods
	+ Principles and practices of extreme programming
	+ Rapid application development
* Documenting of software development process, system development planning and reporting;
* Roles of prototyping in the software process.

**Topic 9. Software reuse and evolution**

♦ Topic outline:

* Software evolution processes and ways of software reuse implementation;
* Benefits of software reuse and some reuse problems;
* Development of software product lines;
* Software maintenance and maintenance cost factors;
* Program evolution processes and dynamics.

**Topic 10. Verification and validation**

♦ Topic outline:

* Verification and validation process, what’s the difference;
* Program inspection process, planning of software inspections;
* Verification and formal methods;
* Critical Systems Validation: reliability validation, safety assurance, security assessment;
* Key principles of automated static analysis;
* Cleanroom software development process.

**Topic 11. Software Testing**

♦ Topic outline:

* Software testing guidelines and main tools and techniques;
* Main principles of system and component testing, validation testing and defect testing;
* Strategies for generating system test cases, test case design;
* Essential characteristics of tool used for test automation;
* The results of software testing, how to measure test results.

**Topic 12. Managing People and cost estimation**

♦ Topic outline:

* Staffing principles: selecting and retaining staff;
* Factors that influence individual motivation;
* Key issues of team working including composition, cohesiveness and communications;
* The people capability maturity model (P-CMM);
* Software cost estimation techniques, algorithmic cost modeling.

**Topic 13. Software risk management**

♦ Topic outline:

* The risk management process and objectives;
* Main tasks undertaken by risk managers;
* Risk management planning and the planning process;
* Risk identification tools and techniques;
* Risk analysis: qualitative and quantitive analysis;
* Risk monitoring and control.

**Topic 14. Quality management and process improvement**

♦ Topic outline:

* Quality management process and key quality management activities;
* Process and product quality, the role of standards in quality management;
* Software metrics (predictor metrics and control metrics), limitations of software measurement;
* Quality assurance and standards:
	+ Quality planning
	+ Quality monitoring and control
* Process and product quality improvement. CMMI process improvement framework.

♦ **Topic 15. Configuration management**

♦ Topic outline:

* Software configuration management process;
* Configuration management planning, change management, version management and system building;
* CASE tools to support configuration management processes;
* Configuration management documenting and measuring.

## Reading List

* 1. ***Required***
* [Sommerville, I.](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/98851/source:default) Software engineering / [I. Sommerville](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/98851/source:default). – 10th ed. – Harlow [etc.]: Pearson, 2016. – 810 с. – (Always learning) . – На англ. яз. - ISBN 9781292096131.
* Juan Garbajosa, Xiaofeng Wang, Ademar Aguiar. Agile Processes in Software Engineering and Extreme Programming /19th International Conference, XP 2018, Porto, Portugal, May 21–25, 2018, Proceedings. – URL: https://proxylibrary.hse.ru:2184/book/10.1007/978-3-319-91602-6
* Alessandra Russo, Andy Schürr. Fundamental Approaches to Software Engineering. 21st International Conference, FASE 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings. URL: https://proxylibrary.hse.ru:2184/book/10.1007/978-3-319-89363-1
* Christel Baier, Ugo Dal Lago. Foundations of Software Science and Computation Structures. 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14–20, 2018. Proceedings URL: <https://proxylibrary.hse.ru:2184/book/10.1007/978-3-319-89366-2>
* Amal Ahmed, Programming Languages and Systems. 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings URL: https://proxylibrary.hse.ru:2184/book/10.1007/978-3-319-89884-1
* Hans-Bernd Kittlaus, Samuel A. Fricker. Software Product Management. The ISPMA-Compliant Study Guide and Handbook/ URL: https://proxylibrary.hse.ru:2184/book/10.1007/978-3-642-55140-6
* Jezreel Mejia, Mirna Muñoz, Álvaro Rocha, Tomas San Feliu, Adriana Peña Trends and Applications in Software Engineering.Proceedings of CIMPS 2016/ URL: https://proxylibrary.hse.ru:2184/book/10.1007/978-3-319-48523-2
* Виссер, Д., Разработка обслуживаемых программ на языке Java: десять рекомендаций по оформлению современного кода / Д. Виссер. – М.: ДМК Пресс, 2017. – 181 с. - ISBN 978-5-9706044-7-2: 449.00.
* [Manovich, L.](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/138848/source:default) Software takes command / [L. Manovich](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/138848/source:default). – New York [etc.]: Bloomsbury Academic, 2014. – 357 с.: ил. – (International texts in critical media aesthetics; Vol. 5) . – На англ. яз. - ISBN 978-1-623-56745-3.
* [Вигерс, К.](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/166163/source:default) Разработка требований к программному обеспечению: пер. с англ. / [К. Вигерс](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/166163/source:default), [Дж. Битти](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/166164/source:default). – Изд. 3-е, доп. – М.: Русская Редакция; СПб.: БХВ-Петербург, 2018. – 718 с. - ISBN 978-5-7502-0433-5. - ISBN 9785977533485: 967.20.
* [Cann van, R.](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/149679/source:default) Software business start-up memories: key decisions in success stories / [R. Cann van](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/149679/source:default), [S. Jansen](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/149680/source:default), [S. Brinkkemper](http://opac.hse.ru/absopac/index.php?url=/auteurs/view/149681/source:default). – Basingstoke; New York: Palgrave Macmillan, 2013. – 247 с. – На англ. яз. - ISBN 978-1-13-728046-6.
* Rex Black. Advanced Software Testing - Vol. 1, 2nd Edition: Guide to the ISTQB Advanced Certification as an Advanced Test Analyst 2nd Edition. Rocky Nook.2015
* Hwa Young Jeong. Advances in Computer Science and its Applications: CSA 2013 (Lecture Notes in Electrical Engineering). Springer; Softcover reprint of the original 1st ed. 2014.
* Norman Fenton.Software Metrics: A Rigorous and Practical Approach, Third Edition (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series). CRC Press; 3 edition. 2014
* App Quality: Secrets for Agile App Teams [Text] / Jason Joseph Arbon. - Jason Arbon, 2014.- ISBN: 9781499751277.
* High Performance Android Apps: Improve Ratings with Speed, Optimizations, and Testing - 1st Edition [Text] / Doug Sillars. - AT&T Services, Inc., 2015. - ISBN-13: 978- 1491912515. - ISBN-10: 1491912510.
	1. ***Optional***
* Sommerville I. Software Engineering (9ed.), Addison-Wesley, 2011.
* [ \* original - in Russian] Липаев В.В. Программная инженерия. Методологические основы. Учебник. М.: ТЕИС, 2006.
* Schach S.R.: Object-Oriented and Classical Software Engineering (8 ed.) McGraw-Hill, 2011, 688 pp., pp.35-36
* Avdoshin S., Pesotskaya E., Business informatization. Managing risks, Moscow: DMK Press, 2011, 176 p. [in Russian].
* Bernstein Michael E., Braude Eric J. Software Engineering: Modern Approaches. Wiley; 2 edition, 2010.
* Hisashi Kobayashi, Brian L. Mark. System Modeling and Analysis: Foundations of System Performance Evaluation. Prentice Hall, 2009.
* Ian Sommerville’s web page with available training materials: http://www.comp.lancs.ac.uk/computing/resources/IanS/;
* Course «Applied Systems Analysis» outline: prepared in September-October 2010 [ 2010-2011 academic year ] --- page 8;
* Shari Lawrence Pfleeger and Joanne M. Atlee, Software Engineering: Theory and Practice (4th Edition), Prentice Hall, 2009;
* [Martin L. Abbott; Michael T. Fisher](http://www.informit.com/authors/author_bio.aspx?ISBN=9780137030422), The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise, Addison-Wesley Professional, 2009
* Sommerville I. Software Engineering (8th Edition). Pearson Education Ltd., 2008, 850 pp. ;
* Guidelines for Improving the Software Process. SEI, 2008 Jackson M.C. Systems;
* Software Engineering Inst. Carnegie Mellon Univ. The Capability Maturity Model: Guidelines for Improving the Software Process. SEI, 2008;
* Roger Pressman, Software Engineering: A Practitioner's Approach. McGraw-Hill Science/Engineering/Math; 7 edition (January 20, 2009).
* Eclipse IDE Site [Electronic Resource] / Developers of the site. 2016 - URL: http://www.eclipse.org/
* Android Studio IDE Site [Electronic Resource] / Google Inc., 2016. - URL: https://developer.android.com/develop/index.html
* Java SE Site [Electronic Resource] / Oracle Corporation, 2016. - URL: http://www.oracle.com/technetwork/java/javase/overview/index.html
* Google Developer Console Website [Electronic Resource] / Google Inc., 2016. - URL: https://console.developers.google.com.
* Official Google developers blog [Electronic Resource] / various authors and Google Inc., 2016. - URL: http://googleblog.blogspot.com/search/label/Android.
* Official Android Blog [Electronic Resource] / various authors and Google Inc., 2016. - URL: https://android.googleblog.com/
* Thinking in Java (4th ed.) [Text] // Bruce Eckel. - Prentice Hall, 2006.

## Grading System. Guidelines for Knowledge Assessment

|  |  |  |  |
| --- | --- | --- | --- |
| **Control type** | **Assesment** | **1 year** | **Parameters** |
| 1 | 2 | 3 | 4 |
| Intermediate  | Practice Activity  | \* | \* |  |  | Exercise performance  |
| Homework  |  | \* |  |  | Group project (3-5 student per group)  |
| Test | \* |  |  |  | Written  |
| Overall  |  |  | \* |  |  | Written  |

1. *Practice Activity*

Practice activity during practice hours is assessed by evaluating of student involvement into discussions as well as quality of exercise performance during seminars. Value of practice activity component of final grade formula (PA) is an integer value from interval [0,10].

1. *Homework*

The course ‘Software engineering methodology’ includes a group project (homework), compulsory to all students. Students will work in groups of from 2 up to 5 students on either one of the suggested topics, or on a subject of their own choice. Projects can be of two types: build an application using the techniques studied in the course or study some of the topics more in-depth. Results of group project of first type should be presented in form of report that consists of design document, implementation description, results of testing. Mandatory appendixes are source code for application and database creation script. Results of second type project should be presented in a form of survey or research paper.

Project should be presented and demonstrated by all group members. Each group member should demonstrate complete understanding of all project details and give correct answers to at least two questions of instructor. Value of group project component of final grade formula (P) is an integer value from interval [0, 10] consists of the common score for the report and presentation (from 0 to 5; same score to all group members) and individual student score for the answers to the questions (from 0 to 5). If a student misses the project presentation because of some valid reason, s/he receives «absence» grade. If a student misses the project presentation because of any other reason, s/he receives grade based on individual score set to 0.

1. *Test*

The test is to assess the practical core of course content. It is done during the practical studies. The test (M) is assessed on usual ten-point scale. If a student misses the test because of some valid reason, s/he receives «absence» grade.

1. *Exam*

Exam at the end of the second module (module 2 of 1st year of study) implies arrangement of the oral examination for all students enrolled to the course. Topics covered by the test embraces all course material. If a student misses the exam because of some valid reason, s/he receives «absence» grade. The exam (E) is assessed on usual ten-point scale.

The overall and accumulated course grades Go and Ga (10-point scale each) are calculated as follows:

Ga = 0.5P + 0.3PA + 0.2M;

Go = 0.6Ga + 0.4E.

The overall and accumulated course grades Go and Ga (10-point scale each) include results achieved by students in their practice activities PA, group project P, test of practical studies M and exam E; it is rounded up to an integer number of points. The rounding procedure accounts for students' practice activities during seminars. Intermediate assessment retakes are not allowed. Conversion of the concluding rounded grade (FE) to five-point scale grade is done in accordance with the following table:

Summary Table: Correspondence of ten-point to five-point system’s marks

|  |  |
| --- | --- |
| **Ten-point scale [10]** | **Five-point scale [5]** |
| 1 - unsatisfactory2 - very bad3 - bad | Unsatisfactory- 2 |
| 4 – satisfactory5 - quite satisfactory | Satisfactory- 3 |
| 6 - good 7 - very good | Good - 4 |
| 8 - nearly excellent9 - excellent10 - brilliantly | Excellent- 5 |

## Sample questions for the final control

* What are the fundamental ideas of software engineering?
* Give a definition of terms «system engineering», «software engineering », «subsystem», «software process», «web-based systems», etc. (the matter is related to main concepts introduced in the training course);
* Provide examples of different application types and give its mail characteristics?
* What is the difference between software engineering and computer science? What is the difference between software engineering and system engineering?
* What are the incremental development problems and benefits?
* What is a software process model? Provide examples
* Describe different types of system and their requirements to appropriate software engineering tools and techniques for the development?
* What are the key challenges facing software engineering? Provide examples and give short explanations?
* What is meant by verification and validation? What is the difference between verification and validation?
* What includes system testing? Describe V-model? What types of testing do you know? Provide examples?
* Describe essential attributes of good software (product characteristic)
* What are the main software process activities?
* What is the need of software prototyping? Where it can be used? What are the benefits and problems of prototyping?
* Software reuse and maintenance. Problems and benefits. Provide examples and give short explanations?
* What is the purpose of project management in software engineering? Describe the mail areas to be managed?

## Sample questions for the test control

* What are JUnit classes? List some of them.
* What are annotations and how are they useful in JUnit?
* What is software?
* What are the attributes of good software?
* What is software engineering?
* What are the fundamental software engineering activities?
* What is the difference between software engineering and computer science?
* What is the difference between software engineering and system engineering?
* What are the key challenges facing software engineering?
* What are the costs of software engineering?
* What are the best software engineering techniques and methods?
* What differences has the web made to software engineering?

## Topics for assignments

Home assignment for group project:

* Android-application for 3D indoor positioning of objects in wi-fi-networks
* Editor app of the indoor plans for positioning of mobile objects
* Multi-agent system for control of logistics terminal based on NFC technology
* Cloud mobile app for urban parking
* Auto attendant mobile app for handling the calls and messages
* System for planning and optimization of the enterprise expenses
* Mobile app for automation of service centers
* Android app for organizing the conferences
* App for handling of the orders of flower shops
* The system of emergency alert users of mobile devices on Android platform
* The search service for Android smartphones
* Program to Search Missing People
* Program for health monitoring of outpatient
* Information system for organization of the conference based on mobile devices
* Mining and processing data from twitter accounts
* Unity 3D Game Development
* Text Analog Search Online System Development
* Mail Bot Development
* Search Bot Development
* Mobile Application Development For Personal Accounting
* Mobile Application Development For Lesson Schedule Informing

## Methods of Instruction

The course combines direct and indirect teaching techniques and contains: lectures, collaborative learning, problem based learning, blended learning, reports, feedback and formative assessment. Direct teaching is carried out by a lecture method. The preferred informal lectures (lectures with discussion), where students play an active role. The main purpose of lectures is to introduce course’s topics, to overview basic information and to discuss and form the directions of further course activities. Collaborative and problem based learning is implemented through home assignments and inclass activity. Self-studies with the help of the lecture presentations, software needed, internet, literature sources. Seminars/master classes/code sessions of the computer presentation format. Discussing various questions rising during the lectures, discussing them via email. Home assignment means that students work in small collaborative teams on a project development.

## 7.1. Using MOOCs

As an additional material for self-study, we recommend students to use the following MOOC(s):

|  |  |  |
| --- | --- | --- |
| Course Title | Affiliation | Link |
| Mirosoft Virtual Academy |
| Software Development Fundamentals | *Gerry O'Brien*Microsoft Senior Program Manager; Paul Pardi - Microsoft Senior Content Publishing Manager | https://mva.microsoft.com/en-us/training-courses/software-development-fundamentals-8248?l=xiawPHKy\_5104984382 |
| Software Testing Fundamentals | *Erik Downing*Microsoft Senior Software Development Engineer; Thomas Dawkins - Microsoft Senior Certification Product Manager | https://mva.microsoft.com/en-us/training-courses/software-testing-fundamentals-8305?l=mjfPZiYy\_8604984382 |
| Top 10 DevOps Questions Answered | *Steven Borg*Cofounder and Strategist at Northwest Cadence. Passionate fan of improving software development practices and processes. | https://mva.microsoft.com/en-US/training-courses/top-10-devops-questions-answered-17637?l=bsDibGIuD\_7006218965 |
| Microsoft Azure IaaS Deep Dive Jump Start | *Cheryl McGuire*Microsoft Technical Writer; Ronald Beekelaar - Founder, Virsoft Solutions | https://mva.microsoft.com/en-US/training-courses/microsoft-azure-iaas-deep-dive-jump-start-8287?l=S1oXqFXy\_2804984382 |

## Special Equipment and Software Support

***8.1 Software***

* Browsers: the latest versions of the Chrome, Opera, Firefox with HTML5+JavaScript support.
* Microsoft Power Point 2010 or upper.
* Android studio (<https://developer.android.com/studio/index.html>);
* MySql (<https://www.mysql.com/>).
* Microsoft Visual Studio.
* Eclipse IDE.
* NetBeans.
* Microsoft Project.

***8.2 Remote support***

Yandex Disk is used to remote course support (will be announced at the first lecture).

***8.3 Material and technical resources***

* Projector VGA/HDMI not less than 1024x768 for lectures and practical studies.
* Computer classes with Internet & good quality Wi-Fi and section 8.1 software - for providing seminars.

 The authors of the program: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Hadi Saleh

The authors of the program: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dmitry Alexandrov