**SYLLABUS**

Software Engineering Methods

for students training on major program 710400 “Software Engineering.”

Profile: Software Engineering Team Foundation

# Total credits: 6 credits

# including:

# Class work 96 hours

# Independent work of students 84 hours

# Laboratory work 64 hours

Practical work 32 hours

The total complexity 180 hours

**Bishkek-2016**

**SYLLABUS**

**114.Б.2.3: Software Engineering Methods**

**DATA teachers**:

Valeeva Asiya

Associate Professor of Computer systems software,Kyrgyz State Technical University I.Razzakov ,

Bishkek, Kyrgyzstan

**CONTACT INFORMATION**

+996 312 523593

E-mail: vaa54@mail.ru

**Number of credits**: Course " **Software Engineering Methods** " 6 credit - 180 hours. Of these 96 hours of classroom hours.

**DATE**: 2016, 3 Semester (Autumn Semester)

**The purposes of discipline**.

* 1. **The purposes of discipline**.

The purpose of the study discipline " software engineering methods " is studying modern engineering methods to create a reliable, high-quality software that meets its requirements; formation of students' understanding of the need to apply these principles, methods and techniques in software engineering.

The work program agreed on with the work program previously studied disciplines:

1) Computing

2) Programming in a high level language

**1.2. Competence, formed as a result of the mastering of the discipline:**

As a result of the mastering of the discipline formed following the competence of the PLO-

OС - general cultural competence;

GPC - general professional competence;

PC - professional competence;

PAC - professionally applied competence;

OС 1 - the capacity for self-organization and self-education;

GPC 1- Possession of the basic concepts, principles, theories and facts related to computer science;

GPC 3 willing to accept the basics of computer science and programming for the design, construction and testing of software products;

PC- 2 - a willingness to use the methods and tools of research facilities of professional activity;

PC-9- ability to assess the time and complexity of the software;

PAC -1- possession of the skills of reading, understanding, and highlight the main idea of reading the source code, documentation.

**1.3. Purpose and task of the discipline**

**Task of the discipline**  The purpose of the study discipline " software engineering methods " is studying modern engineering methods to create a reliable, high-quality software that meets its requirements; formation of students' understanding of the need to apply these principles, methods and techniques in software engineering.

**Requirements to the level of mastering of the maintenance of discipline**

**Knowledge**

* As a result of study of discipline students should be formed representation-

basic and auxiliary processes of software engineering; the advantages of an engineering approach to software development; the main difficulties encountered in the implementation of such an approach; the history and development of software engineering; connection with software engineering life cycle of software; basic methods and tools of software engineering.

**Skills**:

* independently find information on the subject and present processes and functions in the form of block diagrams, charts and program code using software engineering methods to solve this problem.

**To use**

• methods of building models and project management processes and software, software design methods, tools and techniques of software engineering.

**The list of laboratory and practice tasks.**

The purpose of practical training are: consolidation of knowledge gained during practical training, consolidate skills and competencies in methods of software engineering.

.**LABORATORY CLASSES:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Topic** | **Laborat hour** | **Literature** | **Control Form** | **Marks** | **Notes** |
| 1 | **Models and Profiles software product lifecycle.** Analysis of the requirements of the task, the determination of specifications. Application of program structuring method - decomposition of tasks; construction of the general program of the algorithm. Toolset - MS office Visio, MS office word. | 4 | 1-3 | paper report formed by requirements | 3 | Laboratory work is performed by a student.  Independently. |
| 2 | **Models and processes. Designing with incremental detail**. Analysis of the requirements of the task, the definition of specifications. Design. Application structuring method with incremental detail - decomposition of the problem, building a detailed algorithm and decomposition into modules. Toolset - MS office Visio, MS office word. | 4 | 1-3 | paper report formed by requirements | 5 | Laboratory work is performed by a student.  Independently. |
| 3 | **The method of top-down design.** Top-down programming. Analysis of the requirements of the task, the definition of specifications. Determining requirements for input and output data. Application of the method of top-down design – decomposition of tasks into modules, testing on cohesion of modules. Assembly of problem - "the top down". Coding algorithm, testing inputs. Toolset - MS office Visio, MS office word. Visual C++. | 6 | 3,4,6 | Demonstration by PC, paper report formed by requirements. | 5 | Laboratory work is performed by a student.  Independently. |
| 4 | 1. **Bottom up method of design.** Analysis of the requirements of the task, the definition of specifications. Determining requirements for input and output data. Application of the method of top-down design - decomposition of tasks into modules, testing for connectivity and on cohesion of modules. Assembly of problem - "bottom-up". Coding algorithm, testing inputs. Toolset - MS office Visio, MS office word. Visual C++. | 6 | 3,4,6 | Demonstration by PC, paper report formed by requirements. | 5 | Laboratory work is performed by a student.  Independently. |
| 5 | **The constructive method of design.** Analysis of the requirements of the task, the definition of specifications. Determining requirements for input and output data. Application of structural design - decomposition of tasks into modules, testing for connectivity and on cohesion of modules. Assembly of problem - "bottom-up". Coding algorithm, testing inputs. Toolset - MS office Visio, MS office word. Visual C++. | 4 | 3,4,6 | Demonstration by PC, paper report formed by requirements. | 8 | Laboratory work is performed by a student.  Independently. |
| 6 | **Integration testing.** Strategy of integration modules with using stubs. The strategy by top down. The strategy by the bottom up. Functional strategies (end to end) - connection for one function. Checking of the program for errors determination. Toolset - MS office Visio, MS office word. Visual C++. | 4 | 3,4,6 | Demonstration by PC, paper report formed by requirements. | 8 | Laboratory work is performed by a student.  Independently. |
| 7 | **The method of Jackson.** Notations of Jackson method. DiagramofJackson Toolkit - MS office Visio, MS office word. | 4 | 1,6,10 | Demonstration by PC, paper report formed by requirements | 5 | Laboratory work is performed by a student.  Independently. |
| 8 | **Methods oriented on data structure.** Methods of organizing data structures for an example of linear linked list with the use of diagrams f Jackson.  An identification of links for processing (compliance) between the structures and functions of the data in the task must be to make. Write a program in the structural notation. Toolkit - MS office Visio, MS office word. | 4 | 17,18,19 | Demonstration by PC, paper report formed by requirements. | 8 | Laboratory work is performed by a student.  Independently. |
| 9 | **Methods are oriented on data structure.** Make the identification processing of constraints (compliance) between the structures and functions of the data in the task. Write program structure notation. Toolkit - MS office Visio, MS office word. | 4 | 17,18,19 | paper report formed by requirements | 5 | Laboratory work is performed by a student.  Independently. |
| 10 | 1. **Methodology of Jackson.** Construction of of Jackson map. Selecting objects -naction, structure- Entity Structure Diagram (ESD). Defining relationships. Presentation in the form of diagrams of Jackson. Toolkit - MS office Visio, MS office word. | 4 | 17,18,19 | paper report formed by requirements | 8 | Laboratory work is performed by a student.  Independently. |
| 11 | **Testing.** Functional testing. Checking the the program to an error determination. Scripts of incorrect behavior software. Comparison of test results. Toolkit - MS office Visio, MS office word,Visual C++. | 2 | 3,4,5,11,17 | Demonstration by PC, paper report formed by requirements. | 4 | Laboratory work is performed by a student.  Independently. |
| 12 | **Prototype Methods.** Development of test mockup information system (IS) based on rapid prototyping and sketch. To develop - reasoning for development, purpose of IS and the goal of the project. Build a paper mockup IS . Toolkit - MS office Visio, MS office word | 2 | 3,4,5,11,17 | Demonstration by PC, paper report formed by requirements. | 4 | Laboratory work is performed by a student.  Independently. |
| 13 | **Prototyping Methods**. Development of test mockup information system (IS) based on rapid prototyping and sketch with using Mockup Builder.  Toolkit – MS office Visio, MS office word, Mockup Builder | 2 | 3,4,5,11,17 | Demonstration by PC, paper report formed by requirements. | 4 | Laboratory work is performed by a student.  Independently. |
| 14 | **Prototyping Methods.** Development of of test mockup of information system mock (IS) on the basis of the evolutionary model. Make - a systematic analysis of the subject area; description of the entities and their attributes.  Toolkit – MS office Visio, MS office word | 2 | 3,4,5,11,17 | Demonstration by PC, paper report formed by requirements. | 8 | Laboratory work is performed by a student.  Independently. |
| 15 | **Prototyping Methods.** Development of of test mockup of information system mock (IS) on the basis of the evolutionary model. Define the tasks and functions to be solved by the information system. Information reporting requirements. Reliability requirements. Toolkit – MS office Visio, MS office word | 4 | 3,4,5,11,17 | Demonstration by PC, paper report formed by requirements | 5 | Laboratory work is performed by a student.  Independently. |
| 16 | **Prototyping Methods.** Development of of test mockup of information system mock (IS) on the basis of the evolutionary model. Specify the tasks and functions,functions, the entities and their attributes to be solved by the information system. Develop a conceptual model IS. Develop a diagrams IDEF0. Carry out decomposition. Toolkit – MS office Visio, MS office word , BPWin. | 4 | 3,4,5,11,17 | Demonstration by PC, paper report formed by requirements. | 5 | Laboratory work is performed by a student.  Independently. |
| 17 | **Prototyping Methods.** Development of of test mockup of information system mock (IS) on the basis of the evolutionary model. Develop a functional model. Develop a diagrams DFD. Carry out decomposition. Toolkit – MS office Visio, MS office word , BPWin. | 4 | 3,4,5,11,17 | paper report formed by requirements | 10 | Laboratory work is performed by a student.  Independently. |
|  | **Total** | 64 |  |  | 100 |  |

Notes: *Laboratory works is performed strictly during the classes, because they are performed in groups or internet. At the end of the semester undelivered works are not accepted.*

**The list of sections and topics for individual work under the guidance of a teacher**

|  |  |  |
| --- | --- | --- |
| **№** | **Topic** | **Mark** |
| 1 | **Module 1**  **The task for the project №1 Testing software.** Checking the the program to an error using the method to find the "white box". Analysis of test results. | 15 |
| 2 | **The task for the project №2 Testing software.** Checking the the program to an error using the method to find the "black box". Analysis of test results | 15 |
| 3 | **Module 2**  **The task for the project №3** Evolutionary prototyping method for developing database | 24 |
| 4 | The essay - Quality Software | 8 |
|  | **Total** | 62 |

**The list of sections and topics for independent work**

|  |  |  |
| --- | --- | --- |
| **№** | **The list of sections and topics** | Mark |
| 1 | Profiles of standards lifecycle of systems and software tools in software engineering. | 4 |
| 2 | Culture and ethics of software engineering. | 2 |
| 3 | The basic processes of software engineering | 2 |
| 4 | Quality Software | 2 |
| 5 | Evolution of the Software | 2 |
| 6 | Agile development of the Software | 2 |
| 7 | Economics of Software Engineering | 4 |
| 8 | Prototype-model approach to the software development of complex software systems | 4 |
|  | Total | **22** |

**Rating control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ module** | **Module and scope Hours** | **The rating in marks** | | **Terms monitoring** |
| **min** | **max** |
| module 1 | **Linked data structures**  Lab. - 16 hour.  Pract**.-** 32 hour.  independent work | 8  8  4 | 12  12  6 | 8 week of 3 semester |
| module 2 | **Algorithms of data processing**  Lab. - 16 hour.  Pract**.-** 32 hour.  independent work | 8  8  6 | 12  12  6 | 16 week of 3 semester |
| Results for the current rating |  | 40 | 60 |  |
| Final control |  | 20 | 40 |  |
|  | **Total** | **60** | **100** |  |

**The scale of the transfer points in the evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| **unsatisfactorily** | **satisfactorily** | **Good** | **Excellent** |
| **Up to 60 scores** | **61-73** | **74-86** | **87-100** |

**1.** Implementation of laboratory and practical works 40 points.

2. Intermediate control 10 points

3. Perform independent work of students (0-10) points.

4. The final control (blanks tests, oral questioning) (0-40) points.

**Penalty points are:**

1. Being late to class in an amount equivalent to 1 to 3 times fails to appear (-1)

2. The use of mobile during the exam (-20 points)

3. Late submission of laboratory and practical works (projects) (-2)

4. absenteeism without good reason, ie 1 nb = - 1ball (-1)

5. Plagiarism (- 30 points)

**Note**: The number of points sets the teacher individually, depending on the criterion evaluation.

Law student in case of disagreement with the actions of the teacher evaluation. Given the possibility and procedure of appeal, appeal to the academic advisors to Deputy Dean for Academic Affairs, to appellate commission.

**Control questions for self-examination**

1. What is the purpose the methods of Software Engineering ?
2. What is the purpose of software engineering tools?
3. Classification of methods of Software Engineering
4. What are the methods of systematic programming?
5. What are main ways of implementing the modular method
6. Basic principles of structural methods Software Engineering.
7. Main characteristics of the structural programs?
8. What is the basic principle of structured programming and modes to implement it?
9. What objects can serve as components of the block diagram software package?
10. What are the main elements of the structural design and programming
11. What is a software module
12. Main characteristics of the software module
13. Basic methods development program structure
14. What is the main point of the method of top-down development?
15. What is the main point of the method of bottom-up development?
16. Feature constructive approach
17. Feature architectural approach
18. What are provides methods focused on data structures?
19. What are the four components provide hierarchically – structuredness program according to the method of of Jackson?
20. What is the order of the program structure of the development in accordance with the method of of Jackson?
21. Define the life cycle.
22. The main stages of the software life cycle.
23. List the model lifecycle
24. What is special about the cascade model? Advantages and disadvantages of this model.
25. What is special about the evolutionary model? Advantages and disadvantages of the model.
26. What makes an iterative model? Advantages and disadvantages of the model.
27. What is the peculiarity of the spiral model? Advantages and disadvantages of the model.
28. What is a software prototyping?
29. The classification of protopies.
30. Essential Tools Prototyping
31. What is the feature of rapid prototyping?
32. What is the feature of evolutionary prototyping?
33. What is the feature of horizontal prototyping
34. What is the feature of vertical prototyping?
35. What you are the feature of disposable prototyping?

**The basic literature**

1. С. Орлов, Б. Цилькер. Технологии разработки программного обеспечения4-е
2. издание. 2012 г. 608 с.
3. С. Орлов. Технологии разработки программного обеспечения: — СПб.:

Питер, 2002. — 464 с.: ил. — ISBN 5-94723-145-Х.

1. Douglas Bell. Software Engineering for Students. A Programming Approach
2. Fourth Edition. Addison – Wesley. 2005
3. К. Гецци, М. Джазайери, Д. Мандртоли, Основы инженерии программного обеспечения, 2-е издание, СПб.: БХВ-Петербург, 2005.
4. Соммервиль Иан. Инженерия программного обеспечения, 6-е издание. : Пер. с англ. – М.: Издательский дом “Вильямс”, 2002. – 624 с.
5. SWEBOK) TECHNICAL REPORT ISO/IEC TR 19759 IEEE First edition 2005-09-15
6. CMMI® for Development, Version 1.2, CMU/SEI-2006-TR-008 ESC-TR-2006-008
7. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и
8. программная инженерия. Процессы жизненного цикла программных средств
9. В.В. Липаев Программная инженерия. Методологические основы. Учебник. М.: ТЕИС, 2006.
10. ЛаврищеваЕ.М. , Петрухин В.А.Методы и средства инженерии программного обеспечения. Учебное пособие. Москва 2006

**An additional literature**

1. Software Engineering — Guide to the Software Engineering Body of Knowledge
2. Брукс Ф. Мифический человеко-месяц или как создаются программные
3. системы. СПб: Символ-Плюс, 2006. – 304 с., ил.
4. Рудаков, А.В. Технология разработки программных продуктов [ - 9-е изд.,
5. стер.. - М.: Академия, 2014. - 208 с.
6. Мацяшек Л.А., Лионг Б.Л. Практическая программная инженерия на основе учебного примера. Москва -Бином-лаборатория знаний, 2012г.- 948с.

**Internet resources**

1. <https://files.ttuwiki.ee/SPJackson_2010.pdf> -Подходы к проектированию сложных систем. Методика джексона.
2. Software Engineering Conference (Russia) 2005, 2006, 2007 http://www.secr.ru/