**Syllabus of Comprehensive Special Topic Research Course II**

Author：CHEN Xudong Date：2018.3

**I. Course Information**

1. Course Code: A0S183Q
2. Course Category: Pratical Course
3. Course Nature：Compulsory
4. Hours / Credits: Additional Semester,Two Weeks,Ten Weekdays/2
5. Prerequisite:

Theoretical Courses:

C Language Programming Design

Object-oriented Programming and Design

Introduction to Software Engineering

UI Design and Evaluation

Data Structure

Discrete Mathematics

Pratical Courses:

Programming Comprehensive Practice

Object-oriented and Interactive Application Development Comprehensive Practice

Data structure and Algorithm Comprehensive Practice

1. Specialty: Software Engineering

**II. Course Objectives and Students’ Expected Abilities**

The Comprehensive Special Topic Research Course II adopts enterprise training model. Enterprise training refers to the simulation of the companies’real work environment, and implementing a complete project development process with the "Project Team" approach. It trains team collaboration skills. It also requires team members to cooperate and solve difficult applicative problems jointly.

Comprehensive Special Topic Research Course II is a practical course in Software Engineering, aiming to cultivate students' ability to comprehensively apply the basic theory and design and development skills of software engineering, and analyze and solve problems in practice through the introduction of software company real projects, real environment and role. It can also improve students' engineering skills and professionalism in software system design and development.

This course is a comprehensive application of the theoretical knowledge and practical ability involved in pre-requisites. The enterprise project that is adopted should cover as much as possible the existing capabilities of the major knowledge and practical courses that have been completed previously. It focuses on the requirements of software projects, design and implementation of technology knowledge and skills in application and promotion. And it aims to improve students’ comprehensive use of their knowledge and existing practical abilities to solve practical business problems. It comprehensively develops students' soft skills such as project execution ability, teamwork ability, communication ability, lifelong learning, and professional literacy.

Specific Goals:

1. Can analyze and demonstrate the feasibility of actual software engineering problems, and consider the influence of environmental and legal factors;
2. Understand and use software project requirements development and management techniques;
3. Can quickly retrieve, learn and use software development new technologies and corresponding development environment to achieve software functions;
4. Grasp the theory of software testing, use appropriate test methods and tools to complete software testing;
5. Grasp the process of organizing software development according to engineering ideas and implement software project management;
6. Have a good hands-on ability to use professional knowledge to solve practical problems;
7. Experience different roles in team project development, including analysis, design, implementation, testing, documentation, version control, and project leader roles, and assume corresponding work and responsibilities;
8. Master the process of team development and further enhance communication and collaboration capabilities;
9. Through the presentation of the project, the project clearly expresses technical viewpoints and conducts technical exchanges;
10. Understand software engineering ethics, laws and regulations, be able to perform duties, and have a sense of social responsibility;
11. Improve self-directed learning, pioneering and innovative awareness and innovation.

**III. The definition of complicated software engineering problems**

The Comprehensive Special Topic Research Course II adopts enterprise training model. Enterprise training projects should be difficult practical applications. Students need to have a deep understanding of the enterprise specification of software development process in enterprise projects, use the theoretical knowledge of software engineering, analyze and abstract the practical problems facing the enterprise, complete software projects according to software engineering ideas and norms, and use multiple software engineering design environments. And tools, through teamwork, complete project analysis and project establishment, project management, requirements analysis, system design and implementation, functional testing, deployment and delivery, and write relevant documents and complete project defense. In addition, due to the difficulty of the company's training programs and the range of knowledge involved that exceeds the basic requirements of the courses that are usually learnt, students need to learn new knowledge, solve various problems in the project, and reflect innovation.

**IV. Correspondence between Course Objectives and Graduation Requirements**

Table 1 Correspondence between Course Objectives and Graduation Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements for Graduation | Graduation Requirements Index Point | Sustaining Weight | Course Objectives |
| 5.Use Modern Tools | 5.2 Proficient in the use of document search tools to obtain the latest scientific and technological literature on software engineering problems and the latest advances in the field of software engineering. | H(0.3) | 1,3 |
| 5.3 Ability to select and use appropriate technologies, resources, and modern software engineering tools to complete predictions and simulations of software engineering problems, or validation and validation of software engineering projects, and to understand their limitations. | H(0.2) | 2,3,4 |
| 6. Engineering and Society | 6.1 Experience in software engineering practice and social practice. | H(0.3) | 6,8,9 |
| 7. Environment and Sustainable Development | 7.1 Understand the environmental background of software engineering and the connotation and significance of social sustainable development. | H(0.2) | 1 |
| 7.2 Ability to evaluate the impact of engineering practices of complex software engineering issues on the environment and sustainable development. | H(0.2) | 1 |
| 8. Professional Norms | 8.2 In software engineering practice, understanding and observing engineering ethics and norms, with legal awareness and performance of duties. | H(0.2) | 10 |
| 9.Individuals and Team | 9.1 Can independently complete tasks assigned by the team, share information, listen to opinions, and have collaborative spirit and skills. | H(0.2) | 5,7,8,9 |
| 9.2 In a multi-disciplinary team, understand the different roles of team members and assume corresponding responsibilities according to job requirements. | H(0.2) | 7，8 |
| 12. Life-long Learning | 12.2 Grasping the methods and skills of self-directed learning and lifelong learning can be based on the needs of individuals or career development, active learning, and adaptation to development. | H(0.3) | 11 |

**V. Teaching Content and Arrangement**

This course is implemented during the additional semester.It lasts for 2 weeks,10 weekdays, 80 hours in total.

It is suggested that the start-up companies should list the specific teaching content according to the concise workday/study time. The format can refer to the following table.

Table 2 Teaching Content and Arrangement

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Unit | Knowledge point | Requirements | Hours | Graduation Requirements Index Point | Sustaining reasons |
| 1 | Training Preparation  Project Grouping | Enterprise training mobilization | Teachers explain，Grasp | 4 | 9.1  9.2 | The formation of a project team and the division of roles requires the team and its members to carry out, so it supports 9.1 and 9.2. |
| Training process standards and requirements | Teachers explain，Grasp |
| Formation of project team/determination of project objectives/role division | Training,Grasp |
| Training management methods | Training,Grasp |
| Required hardware environment for training | Training,Grasp |
| Team activities: nurturing proactive, open and innovative, responsible for keeping promises, independent collaboration, and sharing successful professionalism. | Training,Participate |
| 2 | Project Analysis | Business scope analysis/identification | Training,Grasp/Familiar | 4 | 5.2  6.1  7.1.  7.2  9.1  9.2 | Analysis of project business scope and issues requires the use of relevant tools for retrieval, thus supporting 5.2; Project analysis is one of the important tasks of internship and practice, and therefore supports 6.1; Analysis of the project requires an in-depth understanding of the background and implementation significance of the project, and evaluate possible impacts, thus supporting 7.1 and 7.2; Formulating project plans and development strategies requires teams to collaborate, thus supporting 9.1 and 9.2. |
| Problem analysis/identification | Training,Grasp/Familiar |
| Project goal determination/SOW | Training,Grasp/Familiar |
| Project plan | Training,Grasp/Familiar |
| Project development strategy | Training,Familiar |
| Review/feasibility analysis | Training,Familiar |
| 3 | Project Management | Project scope management | Training,Grasp/Familiar | 8 | 5.3  6.1  9.1  9.2 | Project management needs to choose the appropriate project management tools, so support 5.3; Project management is one of the important tasks of internship and practice, so it supports 6.1; Project management requires the cooperation of members and members of the team to support 9.1 and 9.2 |
| Software project tracking | Training,Grasp/Familiar |
| Software configuration management | Training,Grasp/Familiar |
| Project management tools | Training,Grasp/Familiar |
| 4 | Requirements Analysis | Demand acquisition/survey studies | Training,Grasp | 8 | 5.3  6.1  9.1  9.2 | The completion of the requirements analysis requires the selection of appropriate requirements analysis tools, thus supporting 5.3; Requirements analysis is one of the important tasks of internship and practice, so it supports 6.1; Requirements analysis needs teamwork to complete, supporting 9.1 and 9.2. |
| Rapid prototyping | Training,Grasp |
| Requirements management | Training,Grasp |
| Requirements analysis support tools | Training,Grasp |
| Requirements assessment | Training,Grasp/Familiar |
| 5 | System Design and Implementation | System architecture design | Training,Grasp/Familiar | 48 | 5.3  9.1  9.2  12.2 | System design and implementation need to select the appropriate technology and design, development tools, functions need test tools to verify, so support 5.3; System design and implementation require teamwork to support 9.1 and 9.2; Problems in the development process, need to learn new technologies to solve them independently, so support 12.2. |
| UI design | Training,Grasp |
| System implementation | Training,Grasp |
| functional test | Training,Grasp/Familiar |
| 6 | System Delivery/Technical Support | Installation and deployment | Training,Grasp | 4 | 5.3  6.1  8.2 | System deployment requires the selection of appropriate software engineering techniques and tools. Acceptance testing is the validation and verification of the project, and therefore supports 5.3; The installation and deployment of the system is one of the important tasks of internship and practice, so it supports 6.1; In the process of installation, deployment and acceptance, it is necessary to abide by the professional ethics and norms of the project. It is legally aware and therefore supports 8.2. |
| User manual/Help | Training,Grasp |
| Acceptance test | Training,Grasp |
| 7 | Project Acceptance and Evaluation | Project conclusion | Training,Grasp | 4 | 6.1  9.1  9.2 | Project defense is one of the regular tasks in the process of internship and social practice, so it supports 6.1; Project conclusion and defense require teamwork to support 9.1 and 9.2 |
| Project team assessment summary | Training,Grasp |
| Project defense | Rated by the teacher |

**VI. Teaching Method**

**Instructional design**:Through external software companies to carry out teaching plan design and teaching activities. The real project of the company reflects the real software project environment and daily assessment model of the software company.

**Teaching method:**Training by gathering together.Ask students to be grouped, set up project teams, complete role assignments and rotations, and students can experience different roles.

The design of training activities needs to be based on specific software project plans, especially at different stages of the software project cycle.

During the training process, teachers need to provide necessary knowledge explanations and deliverables comments for specific project activities.

**VII.** [**Courses Assessment**](javascript:;)

This course is based on a five-level, nine-stage system for final assessment.

A combination of grouping and individual assessment should be used to provide assessments of the processes, results, quality of completion, and responses in accordance with the software life cycle activities of the training program.

Table 3 [Design for Courses Assessment](javascript:;)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Question types | Maximum scores | Graduation Requirements Index Point | Course Objective | Assessment / Evaluation Details | **Rationality statement for assessment way** |
| Project Analysis Review | 15 | 5.2  6.1  7.1  7.2  9.1  9.2  12.2 | 1  5  8  11 | 1）Business scope analysis accuracy  2)Problem Analysis Accuracy  3）Project Goal Clarity  4）Rationality of Project Planning  5）Rationality of Project Development Strategy | This course will be assessed in stages according to the project development cycle. The phased appraisal can timely know the progress of the students' project development, and evaluate the project analysis, requirement analysis, system design and system implementation respectively. It can more accurately reflect the true level of student learning and supervise the students to make reasonable arrangements for the progress of the project. Conduct reasonable management. |
| Reqirements Analysis Review | 15 | 5.2  5.3  7.1  7.2 | 1  2  3 | 1)Rationality of demand acquisition  2)Prototype Display  3)Rationality of Requirement Management  4)The Degree of Application of Requirement analysis tools |
| System Design Review | 15 | 6.1  9.1  9.2 | 7  8 | 1）Rationality of System Architecture Designing  2）Rationality of Module Partitioning  3)Interface Design Accuracy  4）The degree of completion of Detailed Design |
| System Implementation Review | 15 | 5.3  6.1  9.1  9.2  12.2 | 4  5  6  7  8  11 | 1）The degree of completion of the function  2)Code Specification Level  3)Interface Design Specification and Integrity  4)Functional Test Effect |
| Project Defense | 40 | 6.1  8.2  9.1 | 6  9  10 | See details in Table 4 |

Table4. Project Defense Scoring Criteria

|  |  |
| --- | --- |
| Basic Function  (40 Points) | 1. The degree of completion of basic functions 2. The UI design is reasonable and easy to use 3. Functional test results 4. System stability |
| Extended Function  （20 Points） | 1） The degree of completion of extended functions  2） The innovation of extended functions  3） The technical difficulty of extended function |
| Technological Innovation (14 Points) | 1. Difficulties and solutions in project development 2. The key technologies and related modules used |
| Teamwork（8 Points） | 1. The rationality of the division of labor among the team members, the specific task and workload, and the degree of difficulty 2. Communication and cooperation of team members |
| Documents Completion  （18 Points） | Requirements analysis document、System Design Document、System implementation documentation、Testing Report、Project Summary、User Manual |

**VIII. Relation with Other Courses and Respective Responsibility of Each Course**

Theoretical Courses:

C Language Programming Design

Object-oriented Programming and Design

Introduction to Software Engineering

UI Design and Evaluation

Data Structure

Discrete Mathematics

Pratical Courses:

Programming Comprehensive Practice

Object-oriented and Interactive Application Development Comprehensive Practice

Data structure and Algorithm Comprehensive Practice

Relation and Respective Responsibility:

This course is a comprehensive application of the theoretical knowledge and practical ability involved in the above preparatory courses. Selected projects should cover as much as possible the existing capabilities of previously completed theoretical courses major knowledge and practical courses.It aims to improve students’ comprehensive use of their knowledge and existing practical abilities to solve practical business problems.

**IX. Textbooks and Reference Books**

1. Software Engineering: Processes, Methods and Tools.TANG Xiaojun, Tsinghua University Press.2013.12
2. Software Project Management Case Tutorial (3rd Edition).HAN Wanjiang,JIANG Lixin, Machinery Industry Press.2015.6
3. Software Engineering and Computing (Volume Two): The Technical Foundation of Software Development.LUO Bin,DING Eryu,Liu Qin, Machinery Industry Press.2012.12
4. Software Requirements Engineering (Version 2),WU Guoqing,LIANG Zhengping,YUAN Mengting,LI Yonghua, Machinery Industry Press.2013.5
5. UML System Analysis and Design,XUE Junxiao,LI Zhanbo, Machinery Industry Press.2014.9
6. Software Architecture and Model, Joachim Goll,JIA Shan and others translate, Tsinghua University Press.2017.1
7. Software Exorcism: The Art of Debugging and Optimizing Legacy Code,Bill Blunden, Machinery Industry Press.2014.5
8. Software Testing Technology and Use Case Design Training,WEI Nadi,LI Wenbin, Tsinghua University Press.2014.4
9. Software Development Economics,Walker Royce,SU Jingkai and JIN Zhenlin translate, Machinery Industry Press.2010.4