Practical Contents Design for the Curriculum Software Project Management

Hong He[a,*], Sumei Li[b], Changlong Jin[c], Jie Sun[d]

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, Shandong, China

[a]hehong@sdu.edu.cn, [b]lisumei@sdu.edu.cn, [c]cljin@sdu.edu.cn, [d]sunjie@sdu.edu.cn

*Corresponding author

Keywords: SPM; Practice Content Design, IEEE SWEBOK, Document Template for the Software Engineering

Abstract: Software Project Management (SPM) is a compulsory curriculum for the major of software engineering, a limited curriculum for computer-related majors and a general curriculum for the other majors. It is one of the important courses for cultivating high-end IT talents and innovative and entrepreneurial ability. There are many knowledge points in this course, but not many teaching hours. At the same time, practice credits account for a large proportion. Therefore, the quality of the practical contents has a great impact on the quality of the course and whether the training goal can be achieved. As an example, the practice content design method of SPM is introduced with 1 credit of 18 teaching hours and 1 credit of 36 practice hours, emphasizing the use of project management software tools and international and domestic design standards. Let the students complete the given case requirements analysis report and design report using the IEEE SWEBOK document template. Some problems and solutions found in the practical guidance are list to raise concern.

1. Introduction

In the era of artificial intelligence (AI), software is everywhere because that computing is everywhere. Software is the soul of devices and data processing tool. Innovation and entrepreneurship related to the Internet and AI are inseparable from the large or small software projects [1]. Learning how to manage software projects is one of the important links of innovation and entrepreneurship ability training. SPM is both art and technology. Compared with other engineering project management, the use of international standard engineering documents template [2] and management tools in SPM can effectively reduce the risks posed by the stakeholders.

Following contents take the required courses SPM of 18 teaching hours and 36 practice hours as an example. The course is taught through school-enterprise cooperation and uses case teaching. The main teaching contents of this curriculum include the basic concepts of SPM, the management objects and methods of each stage of the software life cycle, and the quality management, cost management, risk management, resource management and other relevant contents [3,4]. The score of the course is 60% for test papers and practice reports are 40%.

Since this course is offered in senior grades of software engineering major and has a high proportion of practical scores, therefore, it has the conditions for students to achieve mastery through a comprehensive study of the basic knowledge and basic tools of software development. We attach great importance to the design of practical contents, try to make students really improve their understanding and management ability of software projects through the links of course practice, and help them smoothly switching from the software engineer to the project manager when watching the software engineering. It effectively helps to students' career development.

In order to inspire and learn from each other, the practically practical content and design goal of SPM course are described. The purpose is throwing a sprat to catch a whale and improving the practice design level of SPM.

  DOI: 10.25236/iche.2022.064

2. Practice Contents Design of the Curriculum SPM

The SPM course is of 36 practice hours and 1 credit. It doesn't need to apply to a laboratory, can allow students to finish their practice reports with their own computers. How to arrange the practice contents in the order of the course contents are following described.

2.1. Face to face with entrepreneurs (Practice 1: Four credit hours)

This is the practice of the basic concept of SPM chapter 1. The CEO of a famous software enterprise is invited to give a 2-hour speech and on-site communicating with the students online or offline. The communication topics include:

- How do companies select a project manager?
- How to build a project team?
- What help can the company provide to the team during the project?
- Under what circumstances do the project has to be delayed?
- How to calculate the project cost and quotation?
- How to manage a project team in which members come and go?
- ……

If you don't get the right entrepreneur to talk to the students face to face during this time period, you can ask him to record the above contents directly into MOOCs, and ask the students to watch videos and organize a discussion. The discussion results are arranged as the first practice report. The aim of this report is to deepen the students' impression of these problems through rumination, so to avoid ignoring the problems in the future.

2.2. Use of SPM Tools (Practice 2: Eight credit hours)

Innovators and entrepreneurs must get used to embracing open-source software [5]. Software engineering management tools of open source can be divided into three categories:

- Project planning and tracking tools: They are used for software projects to measure project workload, cost estimation and project progress, such as Primavera, MS Project, etc.
- Risk management tools: For detecting, assessing, and monitoring risks.
- Evaluation tool: For performing activities related to a software evaluation project.

Every person entering the project team should attend the training at first and learn to use the uniform project management software tools designated by the project team. It is the duty of the project manager to arrange the entire team to train the project management tools. This practice content is arranged as the second practice report, which is that students independently download, install and learn to use one of the most commonly used tools software at home and abroad, and deepen the impression through learning and comparison.

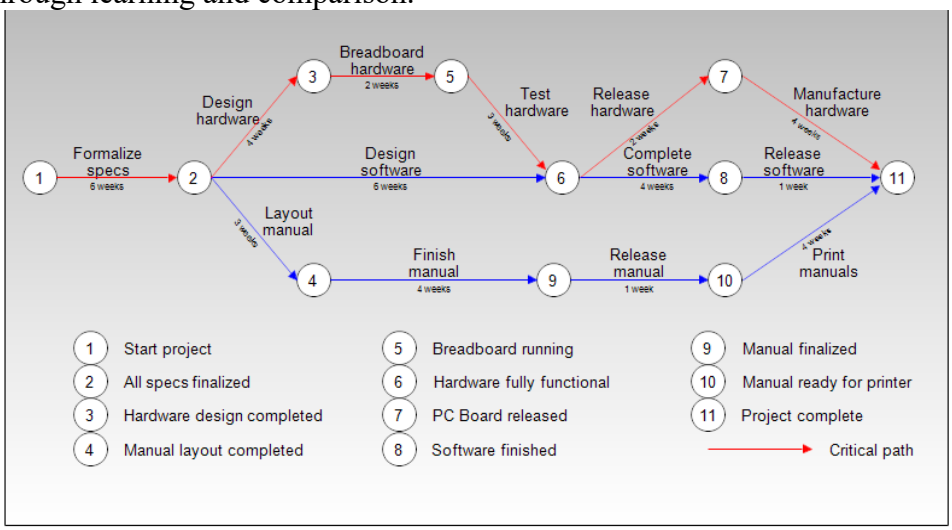

Figure 1 The PERT/CPM chart.

To ensure that students can complete the report carefully, some students were randomly selected

to demonstrate various functions of various project management software tools online. Since students will use different project management software tools in the future, teachers should be responsible for commenting on the characteristics of the tools. Considering most of the students will work in China, the selected software tools are also localized. Common project management charts, such as Milestone Method, Critical Path Method (CPM), Program Evaluation and Review Technique (PERT), Gantt Chart, and Full-wall Scheduling Method are also need to be explained. Different charts have different roles [2], as shown in Figure 1.

2.3. Learn to Use International Standardized SPM Document Templates

2.3.1. IEEE Recommended Practice for Software Requirements Specifications (SRS) (Practice 3: Eight credit hours)

(1)Character of good SRS	(2)Structure of SRS
① The nature of SRS ■ Definition The SRS is a speciTcation for a particular software product, program, or set of programs that performs certain functions in a speciTc environment ■ Author of SRS Supplier Customer ■ Contents of SRS Functions peripheral interface Performance (Quality) attributes → design constraint → Methodology of development → Tools adopted → Standard to follow → 。。。 ② Features of SRS ■ All of the requirements should be included ■ The principle or scope of design or implementation should be specified, but no details of design or implementation should be specified ③ Characteristics of a good SRS ④ Correct ■ Object of reference → Superior specification (SyRS) → Standard → Agreement... ■ No ambiguity With the help of UML ■ Integrity Contains the content: TBD(TO be determined**):** To define what circumstances and conditions can eliminate the TBD ■ Internal consistency The properties and behavior descriptions of objects conflict The same object has different names in different locations ■ Priority Division basis: degree of stability → The number of possible changes → Degree of importance Essential, Conditional, Optional ■ Validable ■ Method with validation ■ Revisability ■ **W**ell-formed → Quote → tabulation ■ No redundancy → A requirement cannot appear in multiple locations → Requirements should be expressed independently and cannot be mixed with other needs ■ Trackable ⑤ Constraints on design ⑥ Does not include information about project management Time Cost Management method person	⑦ Introduction ⑧ OverallDescription Product perspective System Interface UI Hardware interface Software interface Communication interface Product function Overview of functions Use a chart User characteristics ■ Constrains Management policy, hardware restriction, safety requirements, 。。。 ■ Hypothesis and dependency Specific needs Writing principle Each requirement that meets high quality has a unique number ■ The contents of each requirement are contained External interface definition(**input**, **output**) → Data item definition: format, unit, source, valid range, and display format → Functional description → The operation performed, including: input validation, operation procedure, exception handling (response) Performance Logic database requirements Design restriction → Is a supplement to overall description? Quality attribute → Reliability → Serviceability → Safety → Maintainability → Portability Organization of requirements(Structure) System model Users'category Business object Service Feature Stimulus/Response → Stimulus → Response Compages

Figure 2 The SRS Template.

The practice of the following sections requires a practical software project. The teacher plays the

party A of the project and propose the project requirements to the students who play the party B of the project. The IEEE recommended document template of software requirements specifications is introduced to students. A good SRS needs to simultaneously satisfy many conditions such as correctness, unambiguous, complete, consistent, rank for importance, verifiable, modifiable, traceable and so on. The contents of the SRS template from IEEE SWEBOK is shown in Figure2. Students must use the given template to complete the SRS of the given project.

2.3.2. IEEE Recommended Practice for Software Architecture Design (Practice 4: Eight credit hours)

This practice project is to complete SAD report based on the previous SAS, using the SAD template and Rose model from IEEE SWEBOK. See Figure3.

① Architecture
- Definition of architecture

It reflecforms the basic organization of a system, the relationship between components, the relationship with the environment, and the principles guiding the design and development; the system: the organized components that complete a specific function or set of functions; the environment (or context): determines the development, operation, policy and the environment and settings that will cause other effects on the system.
- Architecture defines structure
- Architecture defines action
- Architecture focuses on important elements

  Performance

  quality attributes
- Architecture should balance the needs of all stakeholders
- Architecture based on reasonable evidence is decision concrete (definition of architecture)
- Architectures will often follow an architectural style
- The architecture is influenced by its environment

  The system exists in the environment

  Internal technical constraints

  External technical constraints

  Business requirements

② Architect
- Technical leadership
- May be performed by a team

  Data Architect
- Skills needed
- Business knowledge

  Design

  Communicate

③ Architecture Design
- Activities for architecture definition, documentation, improvement, maintenance, and validation.
- It's science
- It's art
- It's a gradual activity
- Often to compromise

④ Description of SAD(How to write architecture document)

⑤ Steps
- Identify the stakeholders
- Select viewpoint

Different stakeholders have different viewpoints
- Different viewpoints require to define different working products (model)
- Create Work Product (View from ViewPoint)
- function model
- Deployment model
- Schema description package

Form the SAD

⑥ Architecture meta-model
- Viewpoints

It's a template for building a view (including the goal of the view, users, and their analysis, and creation techniques)

Stakeholders have a viewpoint

Basic viewpoint

Requirements viewpoint

Function viewpoint

The view point of Deployment

The view point of validation
- View

Set it up from the point of view

Made up of models
- Model

⑦Architecture description framework
- 4+1view model

⑧Software architecture Document
- SAD
- caArray architecture document
-

Figure 3 The SAD Template.

The teacher should remind students to use the 4 + 1 RUP description, and the architecture of the software system and the global design elements and algorithms will be defined in the SAD document that will serve as the basis for the following detailed design and database design. Detailed design is not used as a specialized practical content in this course.

2.3.3. System security design(Practice 5: Eight credit hours)

Professional designers have to consider the security design of the software, and strive to make the software to the ideal security height [6]. The designer should adopt appropriate safety design criteria and the safety design shall be testable. The SAD document shall clearly identify all the security-critical elements, and modularize the security-critical design to meet the needs of the practical application.

This practice report follows the previous SRS and SAD report and the content of the safety analysis and design are supplemental added in appropriate locations. In this practice report, security interactivity design is a difficulty for students. In order to impress the students, the teachers should systematically explain the principles of safety design in class.

2.4. Documents Collation of Software Engineering Project Acceptance (Practice 6: Four credit hours)

According to IEEE610.12-90, SPM can be defined as application management activities (planning, coordination, evaluation, monitoring, control, reporting) to ensure that software development and maintenance is systematic, disciplined and quantitative. The project plan determines project deliverability through the following (but not limited to the following) projects:

- Runable software
- Customer requirements
- Functional standards
- Design criteria
- Design documentation
- Source code
- Test report
- User's manual
- Operation principle
- Installation guide
- Maintenance process
- Training materials

The last practice is a list template for organizing the above documents one by one against the above list. Students have not seen some of the above information. The teacher should suggest that they consult the various documents and then organize a simple delivery list based on the previous projects.

3. Conclusion

The well organization of the practice contents is critical to improving the course quality of SPM. The contents arrangement and requirements to the six practice reports are proposed to help the student transition from a software engineer to a project administrator and improve students' understanding and control ability of SPM. The goal is to enable students understand the basic process, methods and technology of SPM, pay attention to the role of international standards in design and management, learn to teamwork, improve entrepreneurial awareness and professional awareness.

Acknowledgements

Supported by the Education Reform Surface Project (M2020285) of Shandong Province, and the Key Project “New Liberal Arts, New Engineering Research and Reform Practice Project” (Z2021012)of Shandong University, Weihai.

References

[1] Li Guojie, Several cognitive issues about artificial intelligence, CCCF Vol17,No.7, 2021.7

[2] (USA)Project Management Institute IEEE. The Project Management Knowledge System Guide (PMBOK Guide)(V5)— Software volume.

[3] Zhang Jiahao, Software Project Management, China & Machinery Industry Press, 2005.2

[4] (India) Ramesh, Global Software Project Management, China & Machinery Industry Press, 2003

[5] CCF C3 7th, How do startups embrace open source, Communications of the CCF, Vol 8, 2021.8.

[6] Birhanu Eshete, Making machine learning trustworthy, Science, 2021,373(6556): 743-744. https://www.science.org/doi/10.1126/science. abi5052 . Translated by the CCF special guest staff members. Communications of the CCF, Vol 11, 2021.11.