

IEEE Copyright Notice

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Proposal of a Support System for Visualization of Acquired Skills Using iCD in Project-Based Learning for Software Development and How to Acquire its Data

1st Hane Kondo
Graduate School of Education
Tokyo Gakugei University
Tokyo, Japan
m228110m@st.u-gakugei.ac.jp

2nd Hiroaki Hashiura
Faculty of Advanced Engineering
Nippon Institute of Technology
Saitama, Japan
hashiura@nit.ac.jp

3rd Atsuo Hazeyama
Dept. Information Science
Tokyo Gakugei University
Tokyo, Japan
hazeyama@u-gakugei.ac.jp

Abstract—There have been many practices using software development as a subject for Project-Based Learning (PBL). While PBL for software development has high educational effects, it is difficult for learners to realize many of its educational effects because they overly focus on the activities themselves. Therefore, this paper proposes a support system for visualization of acquired skills in PBL for software development using i-Competency Dictionary. We also describe the data acquisition method for the realization of the system.

Index Terms—software engineering education, software development, Project-Based Learning, acquired skills, visualization, i-Competency Dictionary

I. INTRODUCTION

In light of the social demand for the training of software engineers and the fact that Project-Based Learning (PBL) is attracting attention, there have been many studies on the subject of PBL for software development (e.g., [1]).

There are two educational effects of PBL for software development: generic skills and technical skills related to software development [2]. Although PBL for software development is expected to have higher educational effects, there are some issues in its implementation. Learners overly focus on the activities themselves and lack awareness of the learning objectives and self-transformation that the activities aim to achieve, making it difficult for them to gain a real feeling of learning [2].

Therefore, in this paper, we propose a support system that visualizes the skills acquired by learners through PBL for software development. This system is expected to help learners gain awareness of their own learning based on objective facts.

II. RELATED RESEARCH

Raza et al. developed a tool that indicates potential performance problems and their root causes for individual developers and that supports performance analysis and recommendations

for improvement [3]. This allows learners to visually monitor their own performance, which is significant for ascertaining the learner's situation. However, it is an exercise in software quality and testing, and is not intended for PBL for software development. In addition, what is visualized is the time taken to solve the problem, and does not cover the educational effects that can be expected from PBL for software development.

III. SYSTEM OVERVIEW

There is “i-competency dictionary (iCD)” that systematizes the abilities and knowledge of IT personnel [4]. iCD associates tasks and skills in software development. This study uses iCD because it can cover the educational effects of PBL for software development and because it can relate tasks and skills.

Figure 1 shows an overview of the proposed system. First, it acquires data on artifacts, such as requirements specification, source code, communication messages, and so on, created in PBL for software development. Next, we associate these data with iCD tasks and extract iCD skills associated with the tasks. Finally, the skills are visualized for learners.

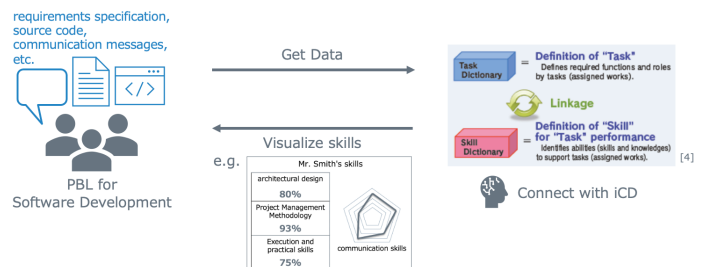


Fig. 1. System Overview

TABLE I
TARGET ICD

Skill Classification Task Subcategories	(Planning) Requirements Analysis Methodology	(Planning) Non-functional Requirements Design Methodology	(Implementation) Software Engineering Methodology	(Support Activities) Information Security	(Systems) Basic software technology	(Systems) Software construction techniques	(Systems) Software usage technology	(Systems) Basic Web system technology	(System) Web system usage technology	(System) Network usage technology	(Systems) Fundamental Technologies of Cloud Computing	(Development) System Architecting Technology	(non-functional requirements) Non-functional requirements (availability, performance and scalability)	Laws, Standards, and Norms
Define functional requirements	⊙		⊙		⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	
Define non-functional requirements	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	
Documentation and review of systematization requirements	⊙	⊙									⊙	⊙	⊙	
Define security requirements	⊙	⊙		⊙						⊙		⊙		⊙

TABLE II
ICD TASKS AND THEIR EVALUATION ITEMS

Task Subcategories	evaluation items
Define functional requirements	(a) Embody function-related requirements as functional requirements to be realized as a system (b) Extract constraints on functional requirements
Define non-functional requirements	(c) Embody requirements related to availability, performance/scalability, operation/maintainability, migration, security, and system environment/ecology as non-functional requirements to be realized as a system (d) Evaluate the results of reviews from stakeholders and revise the systematization requirements definition document
Documentation and review of systematization requirements	
Define security requirements	(e) Identify requirement items such as authentication, usage restrictions, data confidentiality, fraud tracking and monitoring, network countermeasures, malware countermeasures, and web countermeasures to avoid security risks (f) Define and document security measures that will achieve the determined security requirements

IV. SYSTEM IMPLEMENTATION POLICY

A. Targeted PBL for Software Development and iCD

To implement the system, the targeted PBL for software development is defined as follows.

- The software development process is based on the waterfall model.
- Artifacts are created at each phase (requirements specification, class diagrams, source code, etc.).
- All artifacts are text-based and can be analyzed.
- Review and acceptance testing are conducted by a teacher and teaching assistants (TAs).

The iCD has an extensive and exhaustive structure so that it can be utilized by any type of organizations. For this reason, we revised iCD by selecting tasks according to the aforementioned target PBL for software development (in this paper, only tasks that can be read from the requirements specification are discussed). The correspondence between the tasks and skills in the revised iCD is shown in Table I. For example, if the task “Define Functional Requirements” is extracted, we regard the creator has skills such as “(Planning) Requirements Analysis Methodology” and “(System) Basic Software Technology.” In addition, for each task in iCD, there are “evaluation items” that indicate specific implementation details (actions) and can be used as a guide when judging whether the task has been extracted or not [4]. The evaluation items corresponding to the revised iCD are shown in Table II.

B. Data Acquisition Method

The key to this system is how to acquire tasks data (input) and visualize acquired skills (output). The key to the data acquisition method is how to determine whether or not the iCD tasks have been extracted from the artifacts of PBL for software development. Two methods for determining the tasks are described below.

The first method is to check the status of accomplishment of each item at the time of review by a teacher and TAs. For example, if the functional requirements are described in detail, it can be judged that the task of “defining functional

requirements” has been extracted, because it corresponds to the evaluation item (a). If the specification, which is the artifacts of PBL for software development, is reviewed and the revision history is described, it corresponds to the evaluation item (d), and therefore it can be judged that the task of “documentation and review of systematization requirements” has been extracted.

The second method is to automate these operations using Artificial Intelligence. For evaluation item (a), it is possible to determine whether a sentence is concrete or not by using a lexicon of vocabulary evaluated for concreteness and abstractness and a technique using the neural network. For evaluation items (c), (d), (e), and (f), it is possible to determine whether or not there are keywords related to security using techniques such as algorithms based on machine learning and unsupervised learning algorithms.

V. CONCLUSIONS AND FUTURE PROSPECTS

This paper has proposed a support system for visualization of acquired skills using iCD in PBL for software development, and described the data acquisition method for its realization. In the future, we will establish the data acquisition method, develop the system, and evaluate it.

REFERENCES

- [1] Y. Sunaga, H. Washizaki, K. Kakehi, Y. Fukazawa, S. Yamato, and M. Okubo, “Relation between combinations of personal characteristic types and educational effectiveness for a controlled project-based learning course,” *IEEE Transactions on Emerging Topics in Computing*, vol.5 no.1, 2016, pp.69-76.
- [2] H. Kondo and A. Hazeyama, “Systematic literature review on educational effectiveness of project-based learning for software development,” *Proceedings of the 29th Asia-Pacific Software Engineering Conference (APSEC)*, 2022, pp.584-585.
- [3] M. Raza, J. P. Faria and R. Salazar, “Assisting software engineering students in analyzing their performance in software development,” *Software Quality Journal* 27, 2019, pp.1209-1237.
- [4] IPA (Information-technology Promotion Agency, Japan), “i Competency Dictionary,” <https://www.ipa.go.jp/en/it-talents/skill-standard/icd.html>, (2023-05-08).