TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 3087- 3093

Characterizing Knowledge Utilization during Software Design Process Using Pkami Model

Salfarina. A[1], Sazly. A[2], Marzanah. A. J[3], Yusmadi. Y. J[4], Hazlina. H[5]
[1,3,4,5]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia,
[2]Universiti Kuala Lumpur Malaysia France
Institute salfarina@upm.edu.my,
sazly@unikl.edu.my

Article Info
Volume 81
Page Number: 3087- 3093
Publication Issue:
November-December 2019

Abstract:

Abundant of studies and intellectual efforts have been invested particularly in determining the influencing and inhibiting factors towards effective knowledge transfer. The intriguing aspects of knowledge transfer have enabled it to be studied and examined from diverse angles and multi perspectives. Many however, tend to overlook the essence of effective knowledge transfer that is, the utilization of knowledge being transferred. For knowledge-intensive environment such as software development project, knowledge integration which encompasses activities of transferring various knowledge across different software processes often takes place. Best case scenario would be a generous opportunity to learn from each other, creating new knowledge and ultimately delivering quality software. But many failed to reap its advantage resultant from its impulsive nature and lack of methods to verify use of knowledge when the transfer happens. Alarmed by this situation, we proposed PKAMI, a model for characterizing knowledge utilization thus enables the verification of effective knowledge transfer. In our previous study we focused into software architecture development. In this paper, we attempt to investigate knowledge transfer a step further, which is in the process of designing the software to be developed for a project, carried out by our students and supervised by our industrial partners. Our aim is to determine the occurrence of effective knowledge transfer by characterizing the knowledge use during software design process. We believe that our effort put forth in this research would be an extremely significant contribution to software engineering as well as knowledge management research.

Article History
Article Received: *5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *14 December 2019*

Keywords*: PKAMI model, software design process, knowledge utilization, knowledge characterization, knowledge transfer.*

INTRODUCTION

A study conducted by Kunttu (2017), revealed that the educational collaboration in the selected university–industry relationships included the following four forms of educational collaboration: 1) *Student projects for groups of undergraduate students*, 2) *Thesis projects*, 3) *Tailored degree courses*, and 4) *Jointly organized courses.* For our case, we chose the first form as our educational collaboration initiative. We are so fortunate to be able to attract several of our industrial partners to participate along in our Software Engineering Project Team course plan. This course with the code of SSE4301 has been carefully tailored to ensure by the end of semester, the students will be able to gain real-client experience throughout the software development processes. They were assigned with a project title and requirements from the industries, regularly held meetings at their clients' workplace and basically worked under their supervision.

In our previous work, we studied knowledge transfer during the process of

November-December 2019
ISSN: 0193-4120 Page No. 3087- 3093

developing the software architecture. It was a delicate experience but had become a great foundation for embarking further research particularly in the area of knowledge transfer for software engineering. This time we decided to focus into the process of software design.

The remainder of this article is structured as follows: In the next section, we describe the software design process and the model we used to characterize the utilization of knowledge namely the PKAMI model. Following this, we present the reports of our findings through some discussions. In the discussion, we then deliberate on implications of our study for practice and research.

SOFTWARE DESIGN PROCESS AND PKAMIMODEL

While software architecture is responsible for the skeleton and the high-level infrastructure of a software, the software design is responsible for the code level design such as, what each module is doing, the classes scope, and the functions purposes. Software design is the process of conceptualizing the software requirements into software implementation. This is the initial phase within the software development life cycle (SDLC), shifting the concentration from the problem to the solution. Figure 2.0 below shows the general model of the software design process.

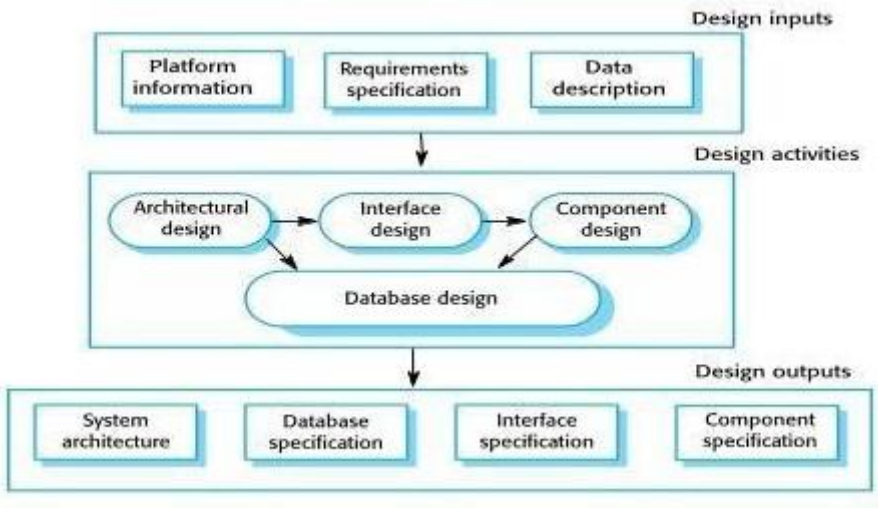

Figure 2.0 The general model of the software design process

We proposed this model initially during our attempt to empirically investigate knowledge transfer in software architecture development process. We found it to be very useful in helping us characterizing the knowledge use during the event of knowledge transfer. Our premise was for knowledge transfer to be effective; it must demonstrate the use of knowledge being transferred. Otherwise, it cannot simply deem as an effective transfer. The model we designed was aiming at one particular purpose that is to determine the extent of knowledge use in order to verify effective knowledge transfer. PKAMI consists of: a) Identify the particular software **P**rocess where KT is expected to occur, b) Determine the general and specific **K**nowledge areas used involved during the software process, c) Determine primary and secondary **A**ctivities involved in the software process, d) Determine the **M**edium used in facilitating the software process (either requires the application of knowledge obtained from previous engagement, or the need to get input and agreement from other people) and finally e) Construct the questionnaire **I**tems according to the knowledge utilization scales and activities in the software process.

November-December 2019
ISSN: 0193-4120 Page No. 3087- 3093

The knowledge utilization scales used in this model were adopted from the Knowledge Utilization Stages (KUS) by Knott and Wildavsky (1980). It has been referred and applied by many researchers in knowledge use area of interest. Table 1.0 below depicts the KUS.

Table 1.0 Knowledge Utilization Stages (KUS)

Stage	Description
1. Reception	Information is received; within reach
2. Cognition	The infromation is read and understood
3. Reference	The information changes the way the person views the topic area or situation
4. Effort	The information influences action
5. Adoption	The information influence outcome
6. Implementation	The information becomes incorporated into practice
7. Impact	The information yields tangible benefits

As shown in Table 1.0, stage 1 and 2 are neither exactly reflecting the action of knowledge use nor explicitly affecting the receiver and its environment. Therefore, they are categorized as no knowledge transfer. Stage 3 onwards on the other hand, implies the position of putting the knowledge into meaningful use thus considered as yes knowledge transfer.

The construction of questionnaire items is the most important part in this model. It relies heavily on the knowledge from the first four steps in PKAMI model. Every item is constructed based on the possible application of related knowledge into each possible step-by-step activity in the software process. The linchpin of the items lies in how it can tell where the participant gains the knowledge from, and how does the knowledge being put into use to accommodate the activities involved (Salfarina et al., 2018). This process is what we refer to as characterizing the knowledge use. Figure 1.0 shows the PKAMI model (Salfarina et al.2018).

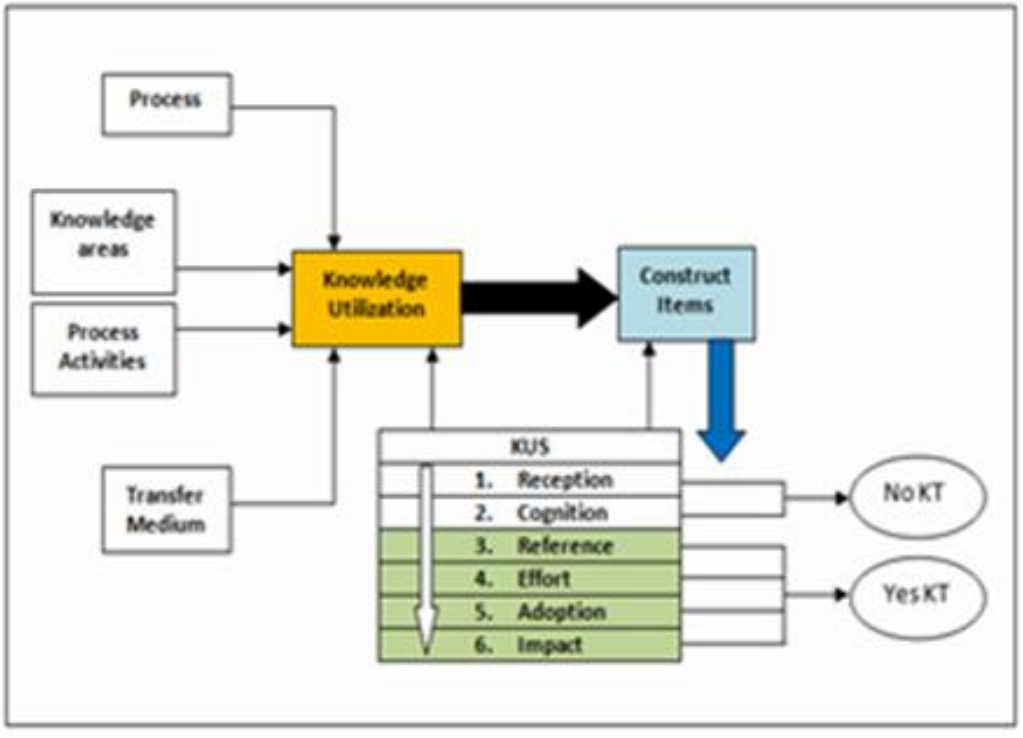

Figure 1.0 PKAMI Model

TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 3087- 3093

METHODOLOGY

As mentioned in earlier section, we managed to gather several of our industrial partners to be on board in the execution of our SSE4301 course plan. It was really a privilege to cooperate with companies such as MIMOS Sdn. Bhd, Petronas IT Division, Viamed, Bit Software, ANSI and School of Graduate Studies, UPM. Our 57 students from two classes were assigned into each of these companies and carried out the project that the companies initiated. Each project consisting of at least 8-10 students. They started off with project planning, gathering requirements, designing, implementing, testing and deploying the complete system. Regular meetings were held throughout the process and close monitoring was done by the industrial supervisors. On our site, we gave the students theoretical foundations and ensure that they produced the necessary deliverables for each phase.

Our methodology is straight forward. We followed the five steps of PKAMI model accordingly and administered interview sessions with the students using the questionnaire we constructed in stage 5 of PKAMI model. Their responses are mainly based on their working experience with their client's projects. İn our attempt to construct the questionnaire items, we interviewed three experts in software design process to ensure all important vitals and information concerning software design knowledge areas and activities are fully captured. These experts are senior lecturers and active researchers in software engineering field for over 20 years. They also possess credible experiences as consultants for several industrial projects.

FINDINGS ANDDISCUSSION

This section presents the findings of PKAMI model applied on the software design process. We will also reveal the findings from our interviews with the students to determine the extent of knowledge use. Table 2.0 shows the findings of PKAMI model.

Table 2.0 Findings of PKAMI model

PKAMI model		
Process		Software design
Knowledge areas:	*General*	System design approach, interface design, architectural design, data/class design, problem solving skills, IT infrastructure, software development methodology
	Specific	Modeling diagrams: scenario-based elements (use case diagram, activity diagram, swim lane diagrams); class-based elements (Class diagram, CRC models, collaboration diagrams); behavioral elements (state diagrams, sequence diagrams) information, requirement specification, data description, operation specification, object interaction, algorithms
Process **A**ctivities:	*Primary* *Secondary*	Translating requirements model to design model - Architectural design, interface design, component design, database design
Transfer **M**edium		Face-to-face meetings, progress reviews, discussions, Software requirement specification (SRS) and other related artifacts

Based on the above information, a list of 12 items were presented and used during the interview sessions with the students. Refer Table 3.0 for details. Unlike the development of software architecture, the flow of activities for software design process is not noticeably defined which means any of them can be done without following any sequence of order.

November-December 2019
ISSN: 0193-4120 Page No. 3087- 3093

Table 3.0 Items

	Items
1	Based on the SRS, we design data/class by transforming analysis classes into implementation classes and data structures.
2	Based on the SRS, we design the software architecture by defining relationships among the major software structural elements
3	Based on the SRS, we design the interface by defining how software elements, hardware elements, and end-users communicate with each other
4	Based on the SRS, we design the software component by transforming structural elements into procedural descriptions of software components
5	We make design decisions based on mutual agreement with the other team members.
6	Using our knowledge in software development methods, we document the design decisions, rationale and diagrams used in modeling the software into software design document (SDD).
7	Through several meetings and progress reviews, we get input on needs to evolve and improve the software design.
8	We evaluate the software design through various means including prototyping, reviews, and assessments.
9	We prepare design documents and deliver presentations to the stakeholders and other development teams.
10	We discuss thoroughly with our team members about the solution concept, algorithms, and alternative approaches while designing the software.
11	Based on our design knowledge, we label all diagrams well at an appropriate level of abstraction so that stakeholders familiar with the problem domain could readily understand them.
12	Based on our design knowledge, we make sure that all notations used in the diagrams are appropriate to the diagram type and are used correctly.

In this research, our study focuses in determining the occurrence of effective KT as well as to find the extent of knowledge utilization during the process of software design. We interviewed 30 respondents whom are our students currently working on different projects initiated by companies from the industry. All interviews were conducted in semi structured form according to the participants' time and venue preferences. Each session took about 10-15 minutes. A list of 12 items were presented and used during the interview sessions. As anticipated, 100% of the participants agreed to have performed all the listed items regarding knowledge utilization in software design process (Refer Table 4.0). This provides evidence that they have engaged in effective KT during the process and at the same time shows software design is indeed a knowledge intensive process. This is also consistent with the prerequisite of effective KT that emphasizes putting the knowledge into action and not merely transferring and receiving knowledge.

Table 4.0 Items used to characterize KT in software design process

Items		Frequency (and percentage %)		
		Somehow agree	Agree	Strongly agree
1	Based on the SRS, we design data/class by transforming analysis classes into implementation classes and data structures.	4 (13%)	18 (60%)	8 (27%)
2	Based on the SRS, we design the software architecture by defining relationships among the major software structural elements.	1 (3%)	22 (73%)	7 (23%)
3	Based on the SRS, we design the interface by defining how software elements, hardware elements and end-users communicate with each other.	2 (7%)	13 (43%)	15 (50%)
4	Based on SRS, we design the software component by transforming structural elements into procedural descriptions of software components.	2 (7%)	16 (53%)	12 (40%)
5	We make design decisions based on mutual agreement with the other team members.	2 (7%)	8 (27%)	20 (67%)

6	Using our knowledge in software development methods, we document the design decisions, rationale and diagrams used in modeling the software into software design document (SDD).	1 (3%)	4 (13%)	25 (83%)
7	Through several meetings and progress reviews, we get input on needs to evolve and improve the software design.	1 (3%)	7 (23%)	22 (73%)
8	We evaluate the design through various means including prototyping, reviews, and assessments.	3 (10%)	9 (30%)	18 (63%)
9	We prepare design documents and deliver presentations to the stakeholders and other development teams	0 (0%)	13 (43%)	17 (57%)
10	We discuss thoroughly with our team members about the solution concept, algorithms, and alternative approaches while designing the software.	0 (0%)	8 (27%)	22 (73%)
11	Based on our design knowledge, we label all diagrams well at an appropriate level of abstraction so that stakeholders familiar with the problem domain could readily understand them.	2 (7%)	18 (60%)	10 (33%)
12	Based on our design knowledge, we make sure that all notations used in the diagrams are appropriate to the diagram type and are used correctly.	1 (3%)	12 (40%)	17 (57%)

Finally, to determine the extent of knowledge utilization from the specified activities in software design process, we mapped every questionnaire item with the six KUS (Refer Table 5.0 below).

Table 5.0 Mapping Items with Stages of Knowledge Utilization

Items	**Stages of Knowledge Utilization**
1. Based on the SRS, we design data/class by transforming analysis classes into implementation classes and data structures.	Effort
2. Based on the SRS, we design the software architecture by defining relationships among the major software structural elements.	Effort
3. Based on the SRS, we design the interface by defining how software elements, hardware elements and end-users communicate witheach other.	Effort
4. Based on SRS, we design the software component by transforming structural elements into procedural descriptions of software components.	Effort
5. We make design decisions based on mutual agreement with the other team members.	Adoption
6. Using our knowledge in software development methods, we document the design decisions, rationale and diagrams used in modeling the software into software design document (SDD).	Adoption
7. Through several meetings and progress reviews, we get input on needs to evolve and improve the software design..	Effort
8. We evaluate the design through various means including prototyping, reviews, and assessments.	Adoption

9. We prepare design documents and deliver presentations to the stakeholders and other development teams.	Impact
10. We discuss thoroughly with our team members about the solution concept, algorithms, and alternative approaches while designing the software.	Adoption
11. Based on our design knowledge, we label all diagrams well at an appropriate level of abstraction so that stakeholders familiar with the problem domain could readily understand them.	Effort
12. Based on our design knowledge, we make sure that all notations used in the diagrams are appropriate to the diagram type and are used correctly.	Effort

According to PKAMI, stage 3 onwards in knowledge utilization indicate the occurrence of effective KT. Hence, we can conclude that all respondents managed to maximize the benefits from KT. Note that every activity in the software design process involves collaboration across different functional teams including system and business analysts, software architects, software designers and also the clients. The task specified for each activity either requires the application of knowledge obtained from previous engagement with other people/team, or necessarily demand for participation from other people/team for their input, view and agreement on certain issues (Salfarina et al.,2018).

CONCLUSION

Software design can be claimed as the most important phase in software development because the success of the whole process will be built upon decisions made in this phase. What we already knew is there is transfer of knowledge among the team members and members across the teams throughout the process. But little that we realize we have yet to fully understand what exactly it takes for KT to be effective. No one seems to care enough to understand the essence of effective KT which is the use of knowledge, going beyond the reach, read and understood. Knowing the extent of knowledge use helps us to verify our engagement in effective KT, which then allows us to learn from each other, creating new knowledge and ultimately delivering quality software. Based on this premise, we are convinced that the capability in delivering quality software can potentially relies on how well we position ourselves in the process, when the transfer happens. For example, an activity that is carried out at implementation level indicates higher level of knowledge use compared to activity that is done at effort level. This signifies the more we make use of knowledge in an activity of a process, the better the outcome would be. This paper presents our intention to study the occurrence of effective KT in software design process by characterizing the knowledge use throughout the process using PKAMI model. We believe our effort could assist software practitioners to better strategize on improving themselves and keep on producing quality software project deliverables. We are also aiming at elevating the essence of knowledge utilization to encourage those involved in development to find ways and opportunities to learn from others'experiences.

ACKNOWLEDGMENT

The authors would like to express appreciation for the support of the sponsor (GP-IPM 9557000), as well as to Dr. Sa'adah Hasan, Dr. Salmi Baharom, and Associate Professor Dr. Marzanah Abdul Jabar for their time and willingness to share their expertise in software design process.

REFERENCES

1. Kunttu, L. 2017. Educational Involvement in Innovative University–Industry Collaboration. Technology Innovation Management Review., 7(12): 14-22.
2. Knott, J. And Willdavsky, A. 1980. If Dissemination Is the Solution, What Is the Problem? Knowledge Creation, Diffusion, Utilization., 1(4): 537-578.
3. Salfarina Abdullah, HazlinaHamdan, Marzanah Abdul Jabar and SazlyAnuar. 2018. A Model for Characterizing Knowledge Utilization in Verifying Successful Knowledge Transfer. In the Proceeding of Knowledge Management International Conference (KMICe) 2018, 25 –27 July 2018, Miri Sarawak, Malaysia.