Je-LKS
Journal of e-Learning and Knowledge Society
The Italian e-Learning Association Journal
PEER REVIEWED PAPERS
BIG DATA, COGNITIVE COMPUTING AND INNOVATIVE TEACHING MODELS
Vol. 12, n.2, 2016
ISSN: 1826-6223 | eISSN: 1971-8829

SOFTWARE ENGINEERING EDUCATION LEARNING PROCESS FOR PROFESSIONAL DEVELOPERS

Ridi Ferdiana

Universitas Gadjah Mada, Indonesia
ridi@ugm.ac.id

Keywords: Software engineering education; Learning process; e-learning; Innovative teaching model.

In this modern era, software projects have been developed by more than one developer that may be on different division or even different organizations. This approach is already known as Join Application Development (JAD). The main problem on JAD is the effectiveness of collaboration and communication between developers. In software engineering, the communication and collaboration can be facilitated through a good project management and IT infrastructure. However, the approach still has challenges in the process of learning and knowledge sharing. For example, how to make sure that the developers have sufficient knowledge on a project. Furthermore, how developers know what they need to learn and what they need to share with the team members. This paper will propose an innovative teaching model for a software engineering education through software engineering education learning process on the organization. Software engineering education learning process will facilitate the developer team to identify, to learn, and

for citations:
Ferdiana R. (2016), *Software engineering education learning process for professional developers,* Journal of e-Learning and Knowledge Society, v.12, n.2, 71-83. ISSN: 1826-6223, e-ISSN:1971-8829

to validate the software engineering education that needed to finish a software project. This model will utilize software development lifecycle (SDLC) as a timeline to integrate between learning process and project execution. The proposed model can be beneficial for project manager to make sure that the developers have sufficient knowledge to finish the project effectively.

1 Introduction

People is the key of the successful and the failure of the project. On software engineering and project management, the composition of the team should be organized well so that it accomplishes the requirements of the project. Consequently, the management is the key how a people behave and productive (Peters, 2008). The productivity is managed by leveraging the knowledge management and continuous improvement on the team project. It means that any knowledge that useful on the project will help the team to work quickly and to decide the best approach to solve the technical problem. On IT projects like software development, the knowledge becomes critical point of the project success especially on competency set within the e-business team (Kollmann, HäSel & Breugst, 2009). Several practical reason are:

- The technical approach on software development grown-up rapidly. For example, the web developer learns about rich internet application like Flash or Silverlight on 2005 but on 2010, the term RIA (Rich Internet Technology) moves to a new technology such as JavaScript and HTML5.
- The requirement complexity of software development is getting higher. In the past, client might only need web application, but then the client needs the mobile solution, web, and web services.
- The higher distraction for the team to learn. The internet connection and digital style activity makes the people have additional activity rather than learning. For example, playing multiplayer game, updating the Facebook status, tweeting the expression makes the learning process is distracted than before.

As a professional for adult career, software developer faces similar problem just like others professional adult learner. Adult learner defines the need by themselves, has a sense of self direction, wide range of personal experiences, learning as necessary for solving problem in the present, and learn because the intrinsic motivation (Mesh, 2010). As a result, it needs more adequate reason to start the learning rather than the non-adult student.

The main problem on the developers are do not have time to learn. Developers are learning by problem, or learning by the need of the tasks. This paper

proposes a model for software engineering education for developers through a specific learning model. The main contributions of this paper can be summarized as follow:

- The proposed learning model will help the developers to plan and to understand the learning curve of the project. It will help them to acquire sufficient knowledge for the project.
- Instead of just using e-learning or personal online strategy (Greyling *et al.*, 2008), The proposed learning model will focus on integrating learning process on software development lifecycle methodology.
- Instead of giving external learning activities for the team project (Bresman, 2010). The proposed learning will deliver contextual learning material based on the team project roles.

2 Literature Review

2.1 Software Engineering Education on a Software Project

The idea of giving proper knowledge on software engineering is started in the term namely Software Engineering Education (Seed). Software engineering education become mainly important during the successful of software industry. The challenge of software engineering education is to limit the gaps between industry and the academic (Almi *et al.*, 2011). For that reason, several researches discuss how to create a software engineering curriculum that reflect the industrial need such as undergraduate student (Mengel, 1998), vocational education (Ji Xueyun & Che Zihui, 2010), all level of education (N. Long, 2008), and even for first graders (Sovic, Jagust & Sersic, 2014).

Although, the previous researches already discuss the software engineering education in all education level. The software engineering education is still evolving through many ways such as a new software engineering method, a new development technology, or even a new API (Application Programming Interface), Therefore, the software engineering education is also needed by the professional developers such as programmers, UX designer, project manager, and many others software engineer roles. On a software project, developers meet several challenges on software engineering education process because:

- Limited time to learn in the project schedule mainframe.
- Limited peers to learn during the project execution because of the difference role on the project.
- Limited facility to learn (no e-learning system, limited documentation, obsolete documentation)
- The mobility of the team (visiting onsite customer, collecting data with the customer)

Based on the situation above, it is hard enough to execute a software engineering education for a software project on a classroom environment. Therefore, the online learning model such as e-learning is proposed as part of this research.

2.2 E-learning for Organization

E-learning becomes one of major important thing on any organization from university, high school, even military. The organization uses e-learning as part of human development process. On the other hand, e-learning is just a tool that need a streamline process to adopt with the organization needs. For example, on military domain a model namely Troop Training Process Improvement (TTPI) is developed to improve and to enhance the e-learning implementation (Chen *et al.*, 2009). The ontology model is also developed to create e-learning that presuming the different situations in the learning process in time so that the learner's emotion could be determined (Zhan, Xu & Mao, 2007). Additionally, the personal context of e-learning is developed to deliver streamline process based on the personal need (Yu & Li, 2009). Based on the several researches, it is shown that:

- Organization should have a specific process to control and to manage an e-learning process.
- Organization should create e-learning environment that helps participants to learn contextually.
- Organization should have a way to identify the learning needs and the requirements for the participant.

Based on the situation above, it is mentioned that the organization needs specific ways and control to execute the e-learning. However, on small organization like a start-up this effort is ignored during the limitation of the organization to setup, to fill the content, and to execute the e-learning system. Small organization prefers to use ready to use MOOC (Massive Online Open Course), third party digital Library, and web forum discussion. Through the rest of this paper, the research will propose a method to utilize 'ready to use' content as software engineering education for the organization. The approach is done to simplify the content creation and the organization can focus to deliver a quality software.

As a result of literature review, the paper proposes several research questions namely:

1. What factor that can be a key consideration before an organization executes a software engineering education process?
2. What kind of learning model that can be implemented on software en-

gineering education process?

3. What steps that needed to implement software engineering education on the organization?
4. How to validate software engineering education process on the organization?

3 Research Method

This paper captures the phenomena how developers learn specific skills that is needed for the software project. After the capturing process, the paper propose the learning process based on the feedback and the lesson learned from the capturing process. At a glance, the research method is based on four-level design on learning model (Sørensen & Levinsen, 2015) as follows:

- Step 1: learning observation. It covers practice and plan for learning process. This step focuses to observe the developers behaviour when they need to learn specific skills for the project. This activity is executed on three months, with two different projects team, and five members on each team. The behaviours observed are:
  - The behaviour when encounters a project that has never been experienced.
  - The behaviour when finding problems.
  - The behaviour when sharing knowledge between members.
  - The behaviour when closing a project and obtain new knowledge.
  - The behaviour when reuse the knowledge.
- Step 2: learning process evaluation. It covers situated and practice-based reflection. This step focuses to get lesson learned after the observation process. This step evaluates several findings such as:
  - How long team members allocates time to learn in each phase of the project?
  - How long team members can resolve the issue through the process of learning?
  - What kind of learning media that is used in the project?
  - What kind of learning techniques that is used by the project member?
- Step 3: proposed learning model. It covers theory-based reflection. This step focuses to propose software engineering education learning process that can be used for the project team. This step will answer 'what' aspect that related with the software engineering education on the project team such as:
  - The process of retrieval knowledge needs in the project.
  - Gap analysis between project requirement and team member

knowledge.
- Learning Design for the team member.
- Knowledge construction to solve problems and issues along the project.
- Documentation of knowledge repository of the project team.

4 Software Engineering Education with Distributed Learning Process

4.1 Learning Observation

As mentioned on the research method, this research is started by doing observation for three months. Two months for project observation and one month for learning evaluation. There are two teams namely team Alpha and team Bravo. Both team attributes are described in the Table 1.

Table 1
TEAM SPECIFICATION

Attributes	Alpha Team	Bravo Team
Organization	Research organization (Academia)	Professional Organization (ISV)
Project Type	Consumer Application (Web)	Consumer Application (Web)
Team Experience	Intermediate (2 years+)	Novice (< 1 year)
Project Experience	New Project Experience	New Project Experience
Project Duration	Two months	Two Months
Team member	5 people	3 people
Member Type	Freelance (20 hours/week)	Full Time (40 hours / week)
Geographical Separated	No	No

The observation provides information about behaviors conducted during the project development. Table 2 shows the behaviors of each team member based on the research method. Table 2 shows the observation attributes from identification the technical gaps, join a formal course, until share knowledge to the others. Both team is powered by online collaboration tools such as Google Docs and Office 365 to share the documentation within the peers. Therefore, it has similar capability to use ICT infrastructure for project productivity. When opening a project, both teams do the repetitive activity to find information through the search engine at the moment of facing problems. When closing a project, the learning documentation is documented as tutorial guide or inside the project source codes. Unfortunately, both team doesn't have an initiative to create knowledge repository to reuse the existing learning documentation for the next project. Table 2 shows the activities that is done by the both team

on the project execution.

Table 2
OBSERVATION RESULT

Observation attribute	Alpha Team	Bravo Team
Project kick off	• Identifying the technical skill gaps • Learning by simulating the requirements through spike solution	• Joining casual course on MOOC • Experimenting by starting the evolutionary prototyping
Facing problem on a project	• Searching on the search engine	• Searching on the search engine
Sharing knowledge between peers	• Sharing knowledge through discussion • Sharing notes through collaboration notes such as OneNote	• Creating a collaboration document to share between peers
Closing a project and gather a new knowledge	• Creating documentation for developers and customers	• Creating step by step guide for the developers
Reuse and documentation the project	• The documentation is integrated with the project source code	• The documentation is integrated with the project source code

4.2 Learning Process Evaluation

There are several models to evaluate the learning process namely: formative & summative evaluation, external & internal evaluation, scientific & illuminative evaluation (Ellington & Earl, 1998). The learning process evaluation on this research follow the external and internal evaluation. The evaluation was chosen because of compatibility with software engineering that requires validation on the external and the internal part. The result of the external evaluation is shown on Table 3.

- External evaluation is evaluation done by the independent expert. The external evaluation will evaluate the result of the software product quality by measuring:
  - The defects of the software on user acceptance test.
  - The architecture review of the software on internal test.
- Internal evaluation is evaluation done by the project manager on the team. It will evaluate the result of the member learning outcome by measuring:
  - The average duration of the team member to find a solution for the problem.
  - The average duration to solve the bugs and redistribute the pro-

posed solution.

Table 3
EXTERNAL EVALUATION

Evaluation Attribute	Alpha Team	Bravo Team
Average Defects	9 bugs / iteration	3 bugs / iteration
Average Defects / Person	1.8 defects	0.6 defects
Architecture Rating	Intermediate (2 years+)	Novice (< 1 year)

Table 3 shows that the alpha team has higher bugs ratio than the bravo team. However, the quality of software architecture on alpha team has higher value than the bravo team. The phenomena that can be taken on the external evaluation are:

- The number of team members is directly proportional to the quality of the architecture. This is due because of the many discussions happening in the team.
- The experienced team members will deliver more software quality on design rather than construction. This is due because the team have experience working on previous projects.
- The fewer team members will give lower software architecture quality rather than the larger team members. This is due because few team members have no time to discuss and to improve software quality during the project.
- The fulltime team members will have fewer software bugs because the time allocation for the project is higher than the freelance team members.

The result of internal evaluation is shown on Table 4. Table 4 is measured by the researcher from the observation that is validated by the project manager. Based on the Table 4 result, it is shown that:

- The number of developers is directly proportional to the numbers of a problem that happen on a project. It can be meant that each team member has sufficient time to explore and to learn the needed knowledge for the project.
- The number of developers is inversely proportional to the velocity of the team to solve a problem. This is due to the ease of coordination on fewer team members.

Based on the three-month observation and monitoring, the research proposes the learning process as discussed in Section 4.3.

Table 4
INTERNAL EVALUATION

Evaluation Attribute	Alpha Team	Bravo Team
Total Problems on Projects	7 issues	13 issues
Total time to solve the problems	21 hours	16 hours
Average time to solve the problem	3 hours / issue	1.2 hours / issue
Average problem / person	1.4 issues / person	4.3 issues / person
Total defects	9 defects	3 defects
Total time that need to redeploy the fix	11 hours	6 hours
Average defects to fix ration	1.2 hours / defect	2 hours / defect
Geographical Separated	No	No

4.3 Proposed Learning Process

Based on the case study result on Section 4.1 and Section 4.2. This research proposes the learning process with two important attributes which are number of the team members and previous project experience. The Figure 1 shows the proposed learning process on software engineering education for professional developers

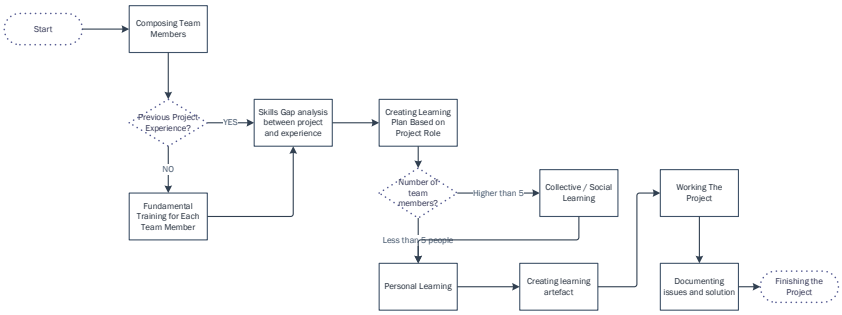

Fig. 1 - Learning process for Professional Developers

4.3.1 Composing Team members

As a first step, the project manager recruits and selects the candidate to join the team. The project manager should measure the complexity of the project and should understand the user or product expectation. Project manager should choose people that has previous experience on a similar project. If not possible, the project manager should make sure that the member understands the basic

competency for the project. The recruitment process should also measure the competency of software engineering knowledge such as project management, software development lifecycle, and software tools competency (i.e. Integrated development environment, computer aided software engineering)

4.3.2 Fundamental training for the team member

This activity is mandatory for the new team member and become optional for the experienced team member. The idea of this activity is to choose the proper training for the team members. The focus of this activity is to fulfill and to review basic needs of the project. The training itself can be done through several ways such as:

- Classroom training. This method is done by inviting an expert to deliver the training. Therefore, the project manager should provide learning timeline and budget for this activity.
- Online learning. This method is done by joining online course such as Microsoft Virtual Academy, Lynda, Pluralsight, or any others MOOC.
- Peers training. This method is done by sharing the knowledge between team members and strengthening the interaction through ad-hoc workshop.

4.3.3 Skill gap analysis between project and experience

In this step, each team member performs problems estimation that may occur and find additional learning material that may be required. The result of this step is a skill gap list that submitted to the project manager. Table 5 shows example of the skill gap Table on a project.

Table 5
SKILL GAP LIST EXAMPLE

Requirements (filled by business analyst)	Skill Gap (filled by the team member)	Recommendation (filled by the project manager)
Implement single sign on between office 365 and Moodle	Never install or use Moodle	• Please follow the course on Pluralsight about Moodle Introduction
Building chat widget for the trainer	Having no knowledge about Widget development on HTML5	• Understanding the basic syntax on W3C school • Review the widget development on library

4.3.4 Creating learning plan based on Project Role

On this step, the project manager structures the learning plan based on the

skill gap list table (Table 5). The learning plan will have:

- What are the topics to be studied?
- Where to learn the topics?
- Why need to learn the topic?
- Who is the person or forum which can be questioned about that topic?
- When the topic needs to be studied?

The learning plan sheet is distributed to team member that need specific skill. Table 6 shows the example of learning sheet for developer that need to learn ASP.NET MVC.

Table 6
LEARNING PLAN SHEET EXAMPLE

Project Name	Point of Sales Web App
Learning Plan Duration	9 hours / 3 days
Learning Plan Course Name	Understanding MVC and data binding
Learning Plan Purpose	• Understanding data binding on ASP.NET MVC • Understanding connecting data source • Understanding CRUD operation with Entity Framework
Learning Media	• Course 2014N – Understanding ASP.NET MVC • Beginning ASP.NET MVC 4.0
Learning Evaluation	• Can develop CRUD features on order module (1d) • Be able to create master detail on order module (3d)
Learning Deliverable	• Online (Self-Paced)
Possible Expert / Forum	• http://forum.msdn.microsoft.com • https://stackoverflow.com/questions/tagged/asp.net-mvc-4

4.3.5 Learning Plan Execution

There are two ways how the learning plan is executed. The first way is to execute as collective learning. Collective learning is executed through mentoring or workshop. The second way is to use personal e-learning. As shown on Figure 1, the e-learning is also used after the team already learn through collective workshop.

The project manager should encourage the team to write the learning reflection on the personal notes. Additionally, project manager can request the team member to write down the issues and solution on a project log. The do-

cumentation result can be stored as knowledge management repository and a retrospective document for the next project.

Conclusion

This research proposes software engineering education learning process for professional developers that work on IT project. This research discusses how to enable software engineering education on professional developers. The conclusions on this research are:

- The number of team members and the experience of the team will reflect on the learning process that will be done by the team.
- Based on the learning observation case study. The structured learning can only happen before the project start, after the project is started the unstructured learning is the only way to learn and to solve the problem. Structured learning can be done by joining an e-learning course, join a webinar, or workshop. Unstructured learning can be done by forming discussion, asking on the forum, or searching on the internet.
- The learning process on a project team can be evaluated through the external and internal evaluation. External evaluation is measuring the product quality such as the number of defects. Internal evaluation is measuring the team productivity to solve the project problem.
- The software engineering education learning process is done through five major steps namely composing team members, planning fundamental training for members, skill gap analysis between project complexity and team competences, creating learning plan, and executing the learning plan.

The future works of this research is to implement and to evaluate the proposed learning process on various organizations structure. Further research should also consider how to monitor the learning process when the team is geographically separated.

REFERENCES

Almi, N. E. A. M., Rahman, N. A., Purusothaman, D., & Sulaiman, S. (2011), *Software engineering education: The gap between industry's requirements and graduates' readiness*. In 2011 IEEE Symposium on Computers & Informatics (pp. 542–547). IEEE. http://doi.org/10.1109/ISCI.2011.5958974

Bresman, H. (2010), *External Learning Activities and Team Performance: A Multimethod Field Study*. Organization Science, 21(1), 81–96. http://doi.

org/10.1287/orsc.1080.0413

Chen, C.-C., Shih, D.-H., Chiang, H.-S., & Lee, C.-J. (2009), *An improvement framework for E-learning processing method development at centralized organization education*. WSEAS Transactions on Information Science and Applications, 6(12), 1872–1881. Retrieved from http://dl.acm.org/citation.cfm?id=1852463.1852466

Ellington, H., & Earl, S. (1998), *Evaluating the Effectiveness of the Teaching/Learning Process*. Retrieved November 14, 2015, from papers2://publication/uuid/9304195C-B636-431E-8F96-F010C985C070

Greyling, F., Kara, M., Makka, A., & Niekerk, S. Van. (2008), *IT worked for us: Online strategies to facilitate learning in large (undergraduate) classes*. The Electronic Journal of E-Learning, 6(3), 179–188. Retrieved from http://www.ejel.org/volume6/issue3

Ji Xueyun, & Che Zihui. (2010), *A new way to software engineering education*. In 2010 International Conference on Educational and Information Technology (Vol. 2, pp. V2–1–V2–3). IEEE. http://doi.org/10.1109/ICEIT.2010.5608454

Kollmann, T., HäSel, M., & Breugst, N. (2009), *Competence of IT Professionals in E-Business Venture Teams: The Effect of Experience and Expertise on Preference Structure*. Journal of Management Information Systems, 25(4), 51–80.

Mengel, S. A. (1998), *Guidelines proposal for undergraduate software engineering education*. In FIE '98. 28th Annual Frontiers in Education Conference. Moving from "Teacher-Centered" to "Learner-Centered" Education. Conference Proceedings (Cat. No.98CH36214) (Vol. 2, pp. 916–919). IEEE. http://doi.org/10.1109/FIE.1998.738862

Mesh, L. J. (2010), *Collaborative Language Learning for Professional Adults*. Electronic Journal of E-Learning, 8(2), 161–172. Retrieved from www.ejel.org

N. Long, L. (2008), *The Critical Need for Software Engineering Education*. The Journal of Defense Software Engineering, (January), 6–8.

Peters, L. J. (2008), *Getting Results from Software Development Teams*. Microsoft Press.

Sørensen, B. H., & Levinsen, K. T. (2015), *Powerful Practices in Digital Learning Processes*. Electronic Journal of E-Learning, 13(4), 291–301.

Sovic, A., Jagust, T., & Sersic, D. (2014), *How to teach basic university-level programming concepts to first graders?* In 2014 IEEE Integrated STEM Education Conference (pp. 1–6). IEEE. http://doi.org/10.1109/ISECon.2014.6891050

Yu, L., & Li, Q. (2009), *Personal media data organization and retrieval in e-learning*. In Proceedings of the first ACM international workshop on Multimedia technologies for distance learning - MTDL '09 (p. 1). New York, New York, USA: ACM Press. http://doi.org/10.1145/1631111.1631113

Zhan, Y., Xu, L., & Mao, Q. (2007), *Ontology based situation analysis and encouragement in e-learning system*, 401–410. Retrieved from http://dl.acm.org/citation.cfm?id=1772177.1772221