

Current Research Trend on Agile Software Development

Handrie Noprisson*¹

*¹Faculty of Computer Science, Universitas Mercu Buana, Jakarta, Indonesia

handrie.noprisson@mercubuana.ac.id

ABSTRACT

In recent years, the software development methodology evolves from the traditional approach to agile software development. This paper attempted to conduct a systematic literature review (SLR) regarding the improved agile software development to tackle its weakness based on recent research papers. Systematic Reviews and Meta-Analyses (PRISMA) as Systematic Literature Review Method (SLR). SLR is the review method which uses some protocols in order to minimize bias in the reviews. The improved of agile software methodology mostly regarding code reusability, usability, project quality, estimation, software delivery, usability, user responses and requirements delivery, communication between members, usability, practical activities, communication between team and stake holder, usability, workflow (learning), problem identification and effort estimation.

Keywords : Agile Methodology, Software Development, Systematic Literature Review

I. INTRODUCTION

In recent years, the software development methodology evolves from the traditional approach to agile software development (Avison & Fitzgerald, 2002; Fitriyah, Hidayanto, Zen, & Arymurthy, 2015; Sucahyo, Rotinsulu, Hidayanto, Fitriyah, & Phusavat, 2017) That methodology is the iterative, incremental methodology which followed agile manifesto (Schmidt, 2016). Agile software development became one of the most popular methodologies because it can reduce time-consuming in completing the software development project (Read, Arreola, & Briggs, 2011).

All types of agile software development have the similar life cycle. The first step of that methodology is requirements elicitation. In this step, the developer and the stakeholder create the backlog documents. This backlog extracted to achieve the functionality of the system. The next step is the iteration phase,

including project planning, developing, testing, and react and release step (Glennon, 2012).

Agile software development offers quicker development process as its excellence. The never-ending requirements from the user can be handled using this methodology. The requirements directly provided by the user to the developer when undergoing the project, then it delivered in user story (Read et al., 2011).

Despite all those advantages, agile software development also has some disadvantages. The disadvantage of that methodology is the lack of system development documentation (Durrani, Pita, & Richardson, 2014). Another disadvantage of this method is the agile software development focused on the software functionality, not software usability (Larusdottir & Gulliksen, 2017). This condition will be impacted to the low-rate of user usability to use the system.

This paper attempted to conduct a systematic literature review (SLR) regarding the improved agile software development to tackle its weakness based on recent research papers, for example, combining user centered design (UCD) and agile to solve the usability problem and so forth (Fox, Sillito, & Maurer, 2008)(Sfetsos, Angelis, Stamelos, & Raptis, 2016).

II. LITERATURE REVIEW

The first cluster is the sprint process, for example from Larusdotir et al. research (Larusdottir & Gulliksen, 2017). Larusdotir et al. did the same research as Fox (Fox et al., 2008), which also conducted by Sfetsos et al. (Sfetsos et al., 2016). This hybrid method executed inside the sprint process, but not mainly focused on improving the sprint rather in the end product.

The main idea of the improvement is by maximizing the usability of the product. The idea is incorporating the UCD into the sprint of Agile. In this approach, the UCD which focused on user experience (UX) conducted in each sprint.

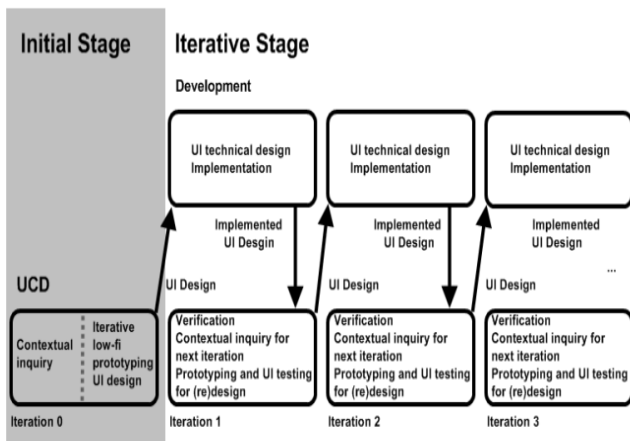


Figure 1. Fox Hybrid (Fox et al., 2008)

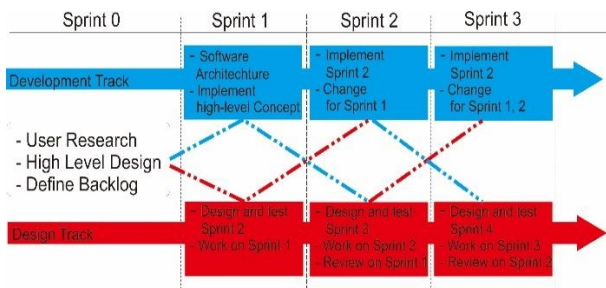


Figure 2. Sfetsos Hybrid (Sfetsos et al., 2016)

Sfetsos and Fox have clear difference, Fox hybrid method conduct the UC step in the initial stage of the process, but in Sfetsos model, the UC conducted iteratively the same way as the Agile Sprint. Larusdotir et al are not pictures any model in their paper but mentioned that the UX Professional who performed the UC have to be participating at least once in the Sprint.

Grapenthin, Poggel, Book, & Gruhn (2015) proposed the method to guarantee better communication between stakeholder and team member in developing process. The proposed method improved the speed of requirement process for each sprint. The proposed method has the different approach compared to the other methods in this cluster. This method mainly focused in the real-time event place.

Dragicevic et al. (Dragicevic, Celar, & Turic, 2017), tried to assimilate Bayesian Network Model into agile software development process. Using Bayesian Network, the team members can predict the outcome and effort for a project. Other research which tried to combine Bayesian Network conducted by Abouelela (Abouelela & Benedicenti, 2010). Abouelela has the similar result with Dragicevic et al. the model proposed by Abouelela can be seen in Fig 5. Korkala et al (Korkala & Maurer, 2014) used waste identification approach to identify if, there were any unneeded communication between team members which stall the development process.

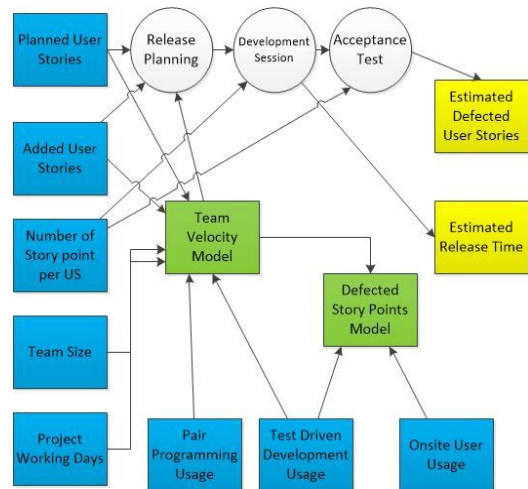


Figure 3. Bayesian Network (Dragicevic et al., 2017)

The research conducted by Matthies (Matthies, Kowark, Richly, Uflacker, & Plattner, 2016). Matthies, proposed the use of ScrumLint. ScrumLint, used to give immediate feedback in development practices. The ScrumLint is a tool which developed using python. The architecture diagram of ScrumLint.

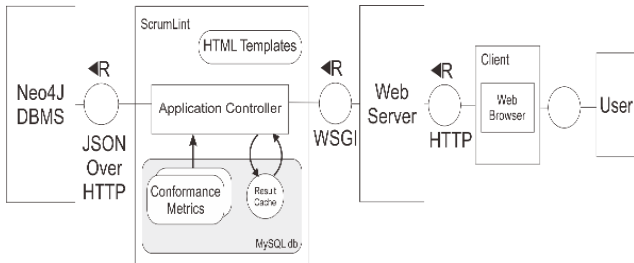


Figure 4. ScrumLint Architecture (Matthies et al., 2016)

ScrumLint used the development artifact as input. Ten conformance metrics used to evaluate the data. The tool is web based.

III. METHODS AND MATERIAL

This paper used the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) as Systematic Literature Review Method (SLR). SLR is the review method which uses some protocols in order to minimize bias in the reviews. This kind of reviews also present some guidance which made the reviews more systematic (Jesson, Matheson, & Lacey, 2011).

In this SLR we used SLR protocol from Jesson et al. (Jesson et al., 2011). The complete protocol comprised of six steps and can be seen in Fig 1. The first step is mapping and determines the review scope, in this step; we define what the scope of this SLR is, made the research question, and design what is the keyword to use in databases. The second step is conducting the research using predetermined research keywords. These keywords will be revised if the keywords were not shown a good result. The excellent result is determined by screening the title and the abstract of the papers.

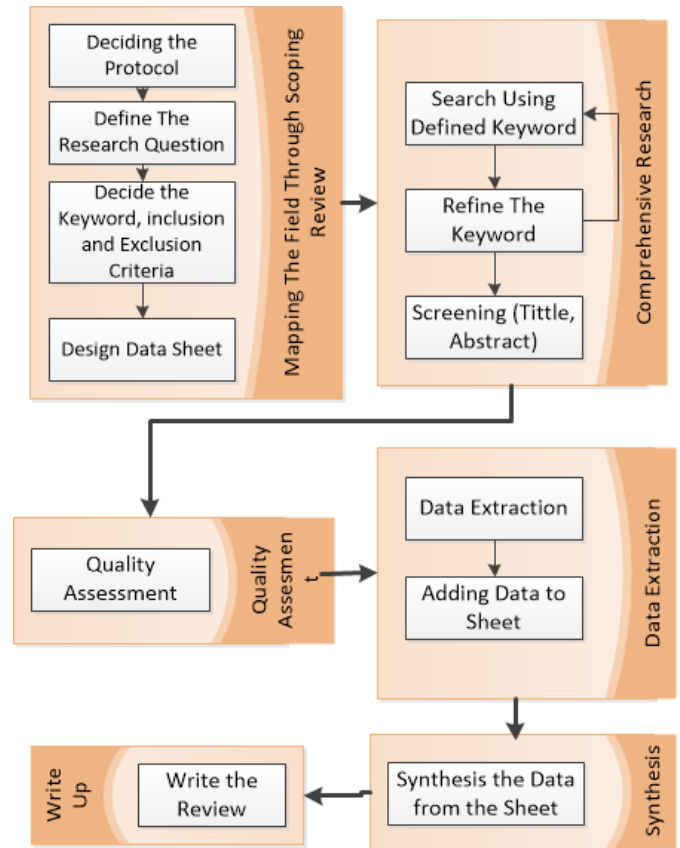


Figure 5. Systematic Literature Review Protocol (Jesson et al., 2011)

A quality assessment conducted after the screening phase. The purposes of this quality assessment are to ensure that the papers are suitable for the reviews scopes. Papers which fulfill the criteria of assessment will be extracted to get the answer to the research question. The next step is synthesizing the extracted data and writes the paper.

TABLE I
DATA COLLECTION

| Author | Title | Source |
|--|---|--------|
| (Singh, Chen, Hunter, Grundy, & Hosking, 2005) | Improving Agile Software development using eXtreme AOCE and Aspect Oriented CVS | IEEE |
| (Fox et al., 2008) | Agile Methods and User-Centered Design: How These Two | IEEE |

| | Methodologies Are Being Successfully Integrated In Industry | |
|---|--|----------------|
| (Agarwal, 2009) | A Control Theory Perspective on Agile Methodology Use and Changing User Requirement | ProQuest |
| (Abouelela & Benedicenti, 2010) | Bayesian Network Based XP Process Modeling | |
| (Darwish, 2011) | Improving the Quality of Applying eXtreme Programming (XP) Approach | IEEE |
| (Butt, Fatimah, & Ahmad, 2013) | Hybrid of Agile Process and Usability Evaluation Method | IEEE |
| (Nazir, 2013) | Cloud Computing ensembles Agile Development Methodologies for Successful Project Development | ProQuest |
| (Korkala & Maurer, 2014) | Waste identification as the means for improving communication in globally distributed agile software development | Science Direct |
| (Larusdottir & Gulliksen, 2017) | A License to Kill- Improving UCSD in Agile Development | Science Direct |
| Choma et al (Choma, Zaina, & Silva, 2015) | Towards an Approach Matching CMD and DSR to Improve the Academia-Industry | IEEE |

| | | |
|---------------------------|---|----------------|
| | Software Development Partnership | |
| (Grapenthin et al., 2015) | Improving task breakdown comprehensiveness in agile projects with an Interaction Room | Science Direct |
| (Sfetsos et al., 2016) | Integrating User-Centered Design Practices into Agile Web Development: A Case Study | IEEE |
| (Matthies et al., 2016) | ScrumLint: Identifying Violations of Agile Practices Using Development Artifacts | IEEE |
| (Lai, 2016) | A Version Control-based Continuous Testing Frame for Improving the IID Process Efficiency and Quality | IEEE |
| (Dragicevic et al., 2017) | Bayesian network model for task effort estimation in agile software development | Science Direct |

IV. RESULTS AND DISCUSSION

The study selected in this paper went through quality assessment. The process result is fifteen papers. These fifteen papers went through next step, which is data extraction. The papers included in the extraction process were the papers which fulfill the inclusion criteria.

There were some papers which excluded from data extraction process. The main reason is the papers do not explicitly show what is the results of the research. Some papers also using other language beside English, which made it impossible to extract the data.

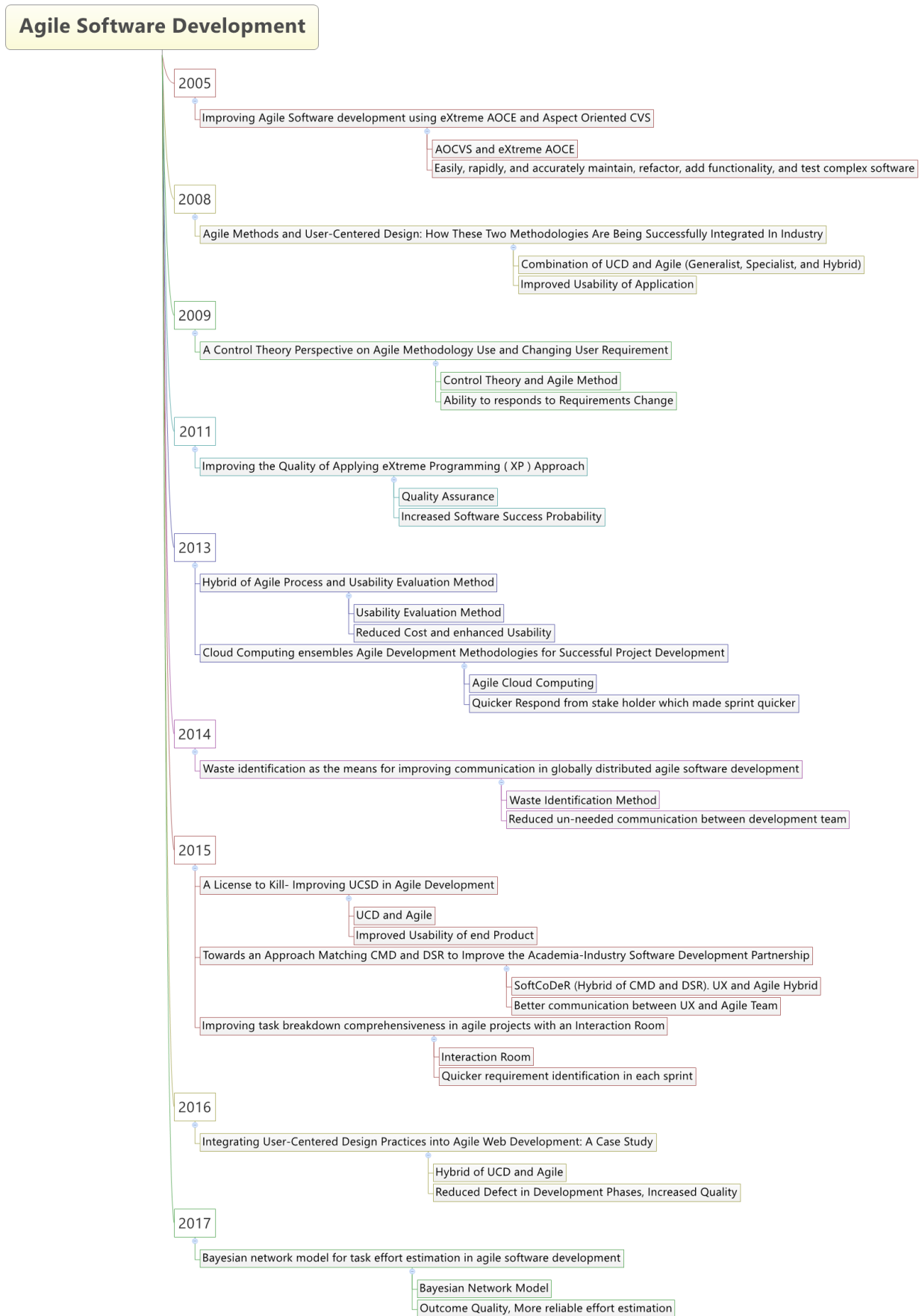


Figure 6. Data Collection

V. CONCLUSION

Many research attempted to refine the Agile Process. Especially the agile life cycle. This research concludes that agile can be improved if some of its neglected artifact reconsidered to be used. The improved of agile software methodology mostly regarding code reusability, usability, project quality, estimation, software delivery, usability, user responses and requirements delivery, communication between members, usability, practical activities, communication between team and stake holder, usability, workflow (learning), problem identification and effort estimation.

VI. REFERENCES

- Abouelela, M., & Benedicenti, L. (2010). Bayesian Network Based XP Process Modelling. *International Journal of Software Engineering & Applications*, 1(3), 1–15. <https://doi.org/10.5121/ijsea.2010.1301>
- Agarwal, R. (2009). *A Control Theory Perspective om Agile Methodology Use and Changing User Requirement*. 20(3), 377–399.
- Avison, D., & Fitzgerald, G. (2002). *Information Systems Development Advances in Methodologies , Components , and Management* (1st ed.).
- Butt, S. M., Fatimah, W., & Ahmad, W. (2013). *Hybrid of Agile Process and Usability Evaluation*. 25(4).
- Choma, J., Zaina, L. A. ., & Silva, T. S. da. (2015). Towards an Approach Matching CMD and DSR to Improve the Academia-Industry Software Development Partnership. *Brazilian Symposium on Software Engineering*, 51–60. <https://doi.org/10.1109/SBES.2015.18>
- Darwish, N. R. (2011). Improving the Quality of Applying eXtreme Programming (XP) Approach. *International Journal of Computer Science and Information Security*, 9(11), 16–23.
- Dragicevic, S., Celar, S., & Turic, M. (2017). The Journal of Systems and Software Bayesian network model for task effort estimation in agile software development. *The Journal of Systems & Software*, 127, 109–119. <https://doi.org/10.1016/j.jss.2017.01.027>
- Durrani, U. K., Pita, Z., & Richardson, J. (2014). *Coexistence of agile and SCM practices An exploratory study of Australian agile software development organizations*. <https://doi.org/10.1108/JSIT-09-2013-0045>
- Fitriana, D., Hidayanto, A. N., Zen, R. A., & Arymurthy, A. M. (2015). APDATI: E-Fishing Logbook for Integrated Tuna Fishing Data Management. *Journal of Theoretical & Applied Information Technology*, 75(2).
- Fox, D., Sillito, J., & Maurer, F. (2008). Agile Methods and User-Centered Design: How These Two Methodologies Are Being Successfully Integrated In Industry. *Agile*, 63–72. <https://doi.org/10.1109/Agile.2008.78>
- Glennon, C. (2012). An Overview of Agile Development Methods for Engineers and Tech Professionals.
- Grapenthin, S., Poggel, S., Book, M., & Gruhn, V. (2015). Improving task breakdown comprehensiveness in agile projects with an Interaction Room. *Information and Software Technology*, 67, 254–264. <https://doi.org/10.1016/j.infsof.2015.07.008>
- Jesson, J. K., Matheson, L., & Lacey, F. M. (2011). *Doing Your Literature Traditional and Systematic Techniques*. London: SAGE.
- Korkala, M., & Maurer, F. (2014). The Journal of Systems and Software Waste identification as the means for improving communication in globally distributed agile software development. *The Journal of Systems & Software*, 95, 122–140. <https://doi.org/10.1016/j.jss.2014.03.080>
- Lai, S. (2016). *A Version Control-based Continuous Testing Frame for Improving the IID Process Efficiency and Quality*. 464–469. <https://doi.org/10.1109/IMIS.2016.113>

- Larusdottir, M., & Gulliksen, J. (2017). A license to kill – Improving UCSD in Agile development. *The Journal of Systems and Software*, 123, 214–222. <https://doi.org/10.1016/j.jss.2016.01.024>
- Matthies, C., Kowark, T., Richly, K., Uflacker, M., & Plattner, H. (2016). *ScrumLint: Identifying Violations of Agile Practices Using Development Artifacts*. 40–43.
- Nazir, A. (2013). *Cloud Computing ensembles Agile Development Methodologies for Successful Project Development*. (November), 28–35.
- Read, A., Arreola, N., & Briggs, R. O. (2011). *The Role of the Story Master: A Case Study of the Cognitive Load of Story Management Tasks*. 1–10.
- Schmidt, C. (2016). *Agile Software Development Teams*.
- Sfetsos, P., Angelis, L., Stamelos, I., & Raptis, P. (2016). Integrating User-Centered Design Practices into Agile Web Development: A Case Study. *Information, Intelligence, Systems & Applications (IISA)*, 13-15 July 2016. <https://doi.org/10.1109/IISA.2016.7785424>
- Singh, S., Chen, H., Hunter, O., Grundy, J., & Hosking, J. (2005). *Improving Agile Software Development using eXtreme AOCE and Aspect-Oriented CVS*.
- Sucahyo, Y. G., Rotinsulu, Y. Y., Hidayanto, A. N., Fitriyah, D., & Phusavat, K. (2017). Software as a service quality factors evaluation using analytic hierarchy process. *International Journal of Business Information Systems*, 24(1), 51–68.

Cite this article as :

Handrie Noprisson, "Current Research Trend on Agile Software Development", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, ISSN : 2456-3307, Volume 5 Issue 5, pp. 99-105, September-October 2019. Available at doi : <https://doi.org/10.32628/CSEIT195511>
Journal URL : <http://ijsrcseit.com/CSEIT195511>