Computer and Information Science; Vol. 12, No. 2; 2019
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Towards Efficient Tracing in Software Product Lines: Research Methodology

Zineb Mcharfi[1] & Bouchra El Asri[1]

[1] IMS Team, ADMIR Lab, ENSIAS, Rabat IT Center, Mohammed V University in Rabat, Morocco

Correspondence: Zineb Mcharfi, IMS Team, ADMIR Lab, ENSIAS, Rabat IT Center, Mohammed V University in Rabat, Morocco. Tel: 212-661-448-917. E-mail: zineb.mcharfi@gmail.com

Received: February 24, 2019 Accepted: March 24, 2019 Online Published: April 30, 2019

doi:10.5539/cis.v12n2p138 URL: https://doi.org/10.5539/cis.v12n2p138

Abstract

Software Product Lines represent a solution for massive development with minimum costs, while assuring product high quality and interesting time to market. In fact, Software Product Lines systems are used for massive productions, and are based on systematic reuse of commun components, while offering the ability to add specific development, in order to satisfy particular users or market needs. However, to maintain such complex and large-scale systems, it is mandatory to adopt a suitable tracing policy that satisfies the system constraints, especially cost and complexity. Unfortunately, tracing is rearly applied in Software Product Lines as it presents several constraints, especially its cost. Through our research work, we tried to come up with elements that would help break this prejudice. Therefore, we worked on a cost and Return on Investment estimation model that helps identify the optimal conditions (phase and policy) for implementing a tracing solution. As a result of our work, we found that implementing specific trace links, in a targeted approach that meets business goals, and starting from the Domain Engineering phase, costs less and presents the most interesting Return on Investment. To conduct this study and reach those findings, we followed the Design Science Research Methodology. In this article, we detail the steps of our research according to this methodology's phases.

Keywords: software product lines, tracing, research methodology, design science, design science research methodology

1. Introduction

In order to cope with the tough competitiveness of markets and their rapid evolution, companies are looking at ways to reduce costs and time to market, while providing customized products with irreproachable quality, in order to gain customer's satisfaction.

From a software engineering point of view, Software Product Lines (SPLs) represent a promising concept, a guarantee of success against these challenges (Northrop, 2002; Pohl et al., 2005).

Software Product Lines are platforms composed of reusable elements, common to all the products to be generated, in addition to specific components allowing the customization of these products depending on the needs of each customer. The reuse of artifacts makes it possible to generate quickly the desired products, massively and of high quality, while having the possibility to personalize them by using variability.

However, in some business environments, adopting an approach based on Software Product Lines may show limitations in terms of responsiveness to market evolution and development. Indeed, the design of a Software Product Line requires the deployment of significant efforts to project into the future of the business and the market, and to provide functionalities that can be used in the medium and long term. In addition, the Return On Investment (ROI) of this considerable effort may be insufficient in some situations, particularly for volatile or rapidly changing markets (Díaz et al., 2011). Therefore, a solution to react quickly to external changes would be a great contribution.

Also, to maintain a system in order to have a durable software solution, by continuously adding new elements, correcting and evolving the existing ones, while ensuring the broadcast of these updates to all concerned products, is a laborious task that requires good organization and mastery of the system. The implementation of a traceability approach can be a solution to these issues.

In fact, the Product Lines are complex and based on the principle of variability. The adoption of a well-documented traceability approach makes it possible to ensure the coherence of the links between the artifacts and to facilitate

any evolution or change in the implementation (Pohl et al., 2005).

The combination of the two concepts, namely the implementation of a traceability approach as part of a Software Product Lines solution, without further complexing the system nor adding unnecessary costs, was the main objective of our work. Indeed, we worked on a model that we named MeTra-SPL (Metrics for Estimating Traceability in SPL), to quantify the Return on Investment of traceability in this context and to detect the true gain of traceability and how to optimize it by defining the ideal combination of when and how to trace in Software Product Lines.

To set up this model and answer the defined research questions, we needed to follow a rigorous research methodology such as Design Science Research Methodology (DSRM). Design Science is a paradigm that helps providing answers to specific business issues (by implementing efficient solutions) (Simon, 1996), and the DSRM (Peffers et al., 2007) is a systematic methodology based on this paradigm, which come in a six steps process, starting from identifying research motivation, and up to evaluating the developed artifact. In this article, we present our research work and its results as conducted according to DSRM.

The remainder of this paper is structured as follow: In Section 2 we introduce our work related to tracing in SPL. In Section 3 we present the Design Science paradigm and the DSRM and how this later was instanciated in the context of our research work. We conclude and present further lines of research in section 4.

2. Background and Motivations

2.1 Software Product Lines

(Northrop, 2002) defines SPL as "a set of software-intensive systems that share a common, managed feature set satisfying a particular market segment's specific needs or mission and that are developed from a common set of core assets in a prescribed way*"*. From a business point of view, Product Lines represent a strategic solution to the need to reuse in order to optimize and achieve gains (cost of deployment, maintenance, time to market ...). From a technical point of view, the Product Lines can be described as a platform for generating products quickly and massively, as the base is already in place, and with high quality (the components are tested several times, with each product) and with the possibility to have customized features. For this, this platform is based on the strategic reuse of elements, the architecture and the design of the solution are thought upstream, and the generated products share a same platform containing the specifications, the architecture, the components, the documentation, the test plans , ... in addition to some features related to the specific features of each product.

Two elementary phases define the SPL set up process: Domain Engineering and Application Engineering. Domain Engineering is "*the process of software product line engineering in which the commonality and the variability of the product line are defined and realised »* (Pohl et al., 2005). It is the phase where the SPL core assets and variation points are created to implement commonalities and variability. It is also during this phase that project plan is established.

The Application Engineering phase is "*the process of software product line engineering in which the applications of the product line are built by reusing domain artefacts and exploiting the product line variability*" (Pohl et al., 2005). It is the instanciation of the Domain Engineering elements, made by choosing a variability scenario.

An other phase can be added to the two previous ones: the Maintenance phase, which is as important as Domain Engineering and Application Engineering phases. This phase handle elements adding, removal, and changes in variation points.

2.2 Traceability in Software Product Lines

Traceability is the ability to interconnect two artifacts. It helps to improve the quality of the system and makes it possible to choose an architecture adapted to the needs and constraints of the project, to identify errors and to facilitate communication between the various stakeholders of the project (Anquetil et al., 2010). Traceability provides significant support for maintenance and system evolution and enables impact analysis of changes on system components (Cavalcanti et al., 2011), which is very important in the context of SPLs.

Tracing in SPLs can be useful in the development phase. It allows in the short-term to check and validate the implementation of all functional requirements, and helps, for the long-term, in the process of maintenance, for a better understanding of the Product Line architecture (how artifacts are interconnected), to drive changes through impact analysis, or for components reuse (Cleland-Huang et al., 2012; Ramesh & Jarke, 2001; Spanoudakis & Zisman, 2005).

However, many difficulties can be faced while implementing traceability in SPLs (Jirapanthong & Zisman, 2005): (i) more documentation, (ii) documents heterogeneity, (iii) the need to link between different products and between

the products and their architecture, (iv) variability, as the impact of this dimension must be understood throughout the phases of implementation of the Product Line.

Software Product Lines are therefore large-scale, proactive and complex systems and their maintenance could be laborious without support and strategy. The implementation of an effective traceability process would be of great help (i) from a technical point of view, to avoid inconsistency errors between different phases and different models, (ii) from a strategic one to optimize response time when updates, and (iii) a financial one to minimize costs. However, and despite the importance given to traceability in research, its implementation in practice is still rare, due to the complexity of its implementation and its costs. It is therefore of great interest to quantify this traceability, hence the idea of measuring the ROI of the implementation of a traceability process in SPLs, and define the elements that might help to optimize it. Such a study is beneficial for the decision making of tracing approach implementation.

3. Our Research Work for Measuring and Optimizing Traceability in Software Product Lines – An application of Design Science Research Methodology

Conducting a research in Information Systems domain, as in many others, needs rigor. Therefore, it is mandatory to follow a well-defined methodology while conducting the study. Two paradigms are mentioned in literature related to Information Systems (Alan et al., 2004): Behavioural science to conduct researches on the impact of implementing an Information System on organization's processes and human resources, and Design Science, which is related to "means" that would help answering specific business issues by implementing efficient Information Systems.

In the case of our research, we adopted a Design Science methodology, as our main goal is to define the optimal "conditions" under which tracing in Software Product Lines would be more profitable. Thus, we are more interested in the implemented system itself than in its environment and involved human resources.

We summarize hereinafter the main elements that characterize the Design Science paradigm and the Design Science Research Methodology (DSRM), and how this methodology was instanciated in the case of our research.

3.1 Design Science for a Rigorous Research Methodology

(Simon, 1996) describes Design Science as a paradigm that "attempts to create things that serve human purposes", compared to "natural sciences and social sciences [that] try to understand reality". We understand through this definition that the aim of Design Science is to help providing answers (by implementing efficient solutions) to specific business issues.

In (Alan et al., 2004) are listed seven guidelines to address for Design Science researches. We summarize those guidelines in figure 1 bellow.

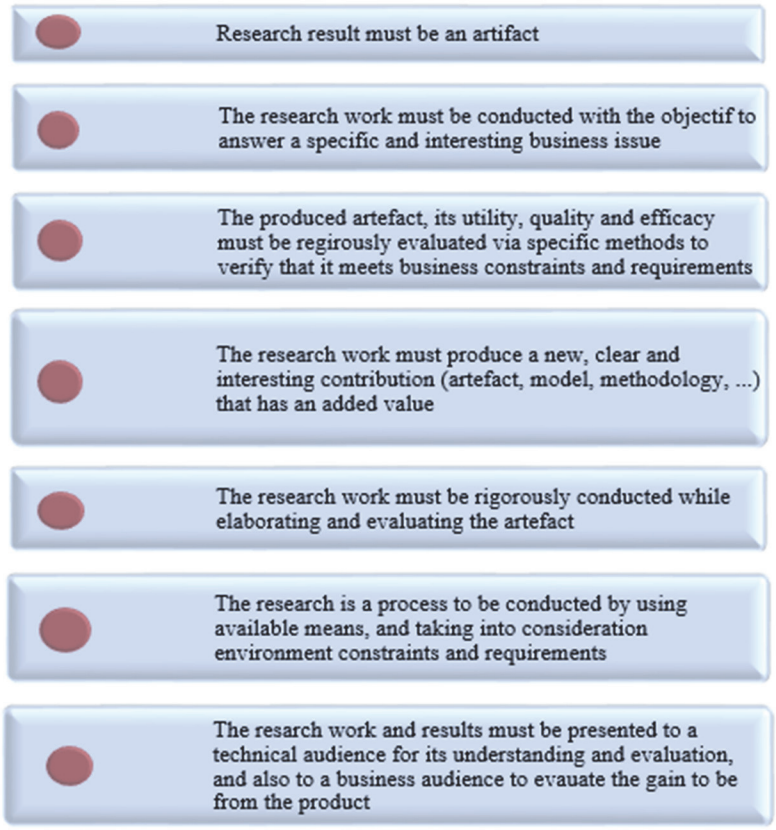

Figure 1. Design Science Guidelines

Based on the guidelines above, (Peffers et al., 2007) proposes a Design Science Research Methodology, as a result of three main elements: (i) Conceptual principals, (ii) practice rules and (iii) a process.

The conceptual principals can be summarized in one definition (Alan et al., 2004): Design science "creates and evaluates IT artifacts intended to solve identified organizational problems". Those IT artifacts can be models, methods, and instantiations (Alan et al., 2004), or even new properties of technical, social, and/or informational resources (Järvinen, 2007).

The practice rules are based on the seven guidelines listed in figure 1.

Considering the process, it is a sequence of six iterative activities described bellow (figure 2) (Peffers et al., 2007):

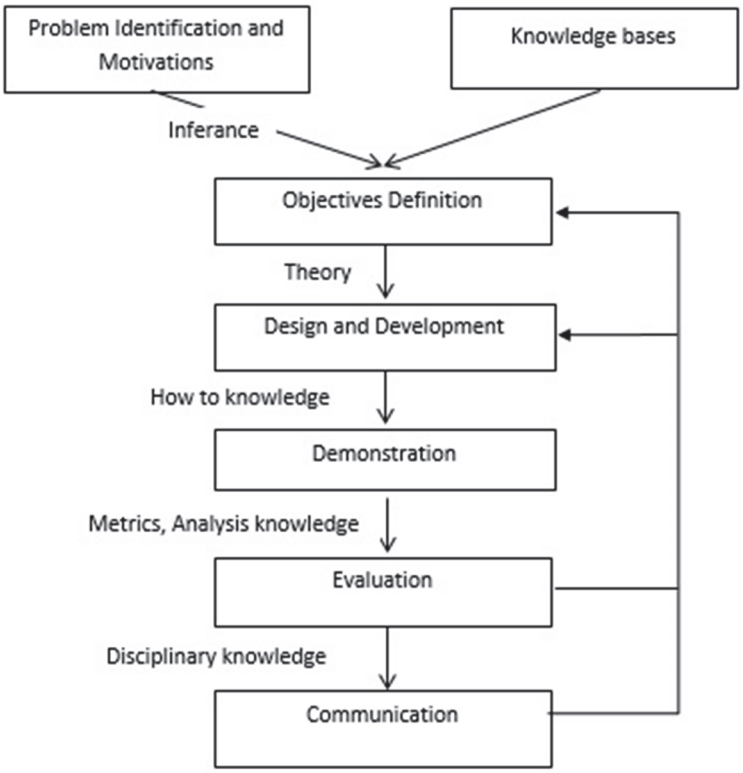

Figure 2. Design Science Research Methodology process

The first step of the DSRM process is to identify the problem and the motivations behind the research. The aim of this step is to present the problem in order to justify the need of a solution and its added value.

Starting from the problem definition, and combined to knowledge bases related to the problem context, research objectives (quantitave and/or qualitative) are defined for a better draw of the problem solution. The knowledge bases include problem context and existing solutions.

Once the solution objectives are defined, the artifact needs to be designed and developed. The research contribution must be part of the artifact design. Some theory elements are welcomed to help create the desired artifact.

After the artifcat is created, its use needs to be demonstrated, through a case study, an experimentation …A "How to" knowledge, which corresponds to the ability to know how to use the created artifact, is needed at this stage.

The fifth step of DSRM process consists on evaluating the created artifact. This evaluation can take different forms depending on the artifact nature. It can be based on quantitative measures (hence the need of some metrics knowledge), or some surveys results, or a comparaison with the solution objectives that have been identified earlier.

The last step of this process consists on communicating the research results. This communication has to be adressesd to a business audience in order to verify the artifact accuracy, efficiency and added value regarding business objectives, as well as to a technical audience for a better comprehension and the evaluation of the artifact (Alan et al., 2004; Peffers et al., 2007).

However, the DSRM process is iterative. Therefore, if evaluation results are not satisfying, or some recommendation are made after communication, the process can be restarted from step two or three.

3.2 Design Science Research Methodology Applied to Our Work

In this paragraphe, we present the result of applying the DSRM to our research work (figure 3).

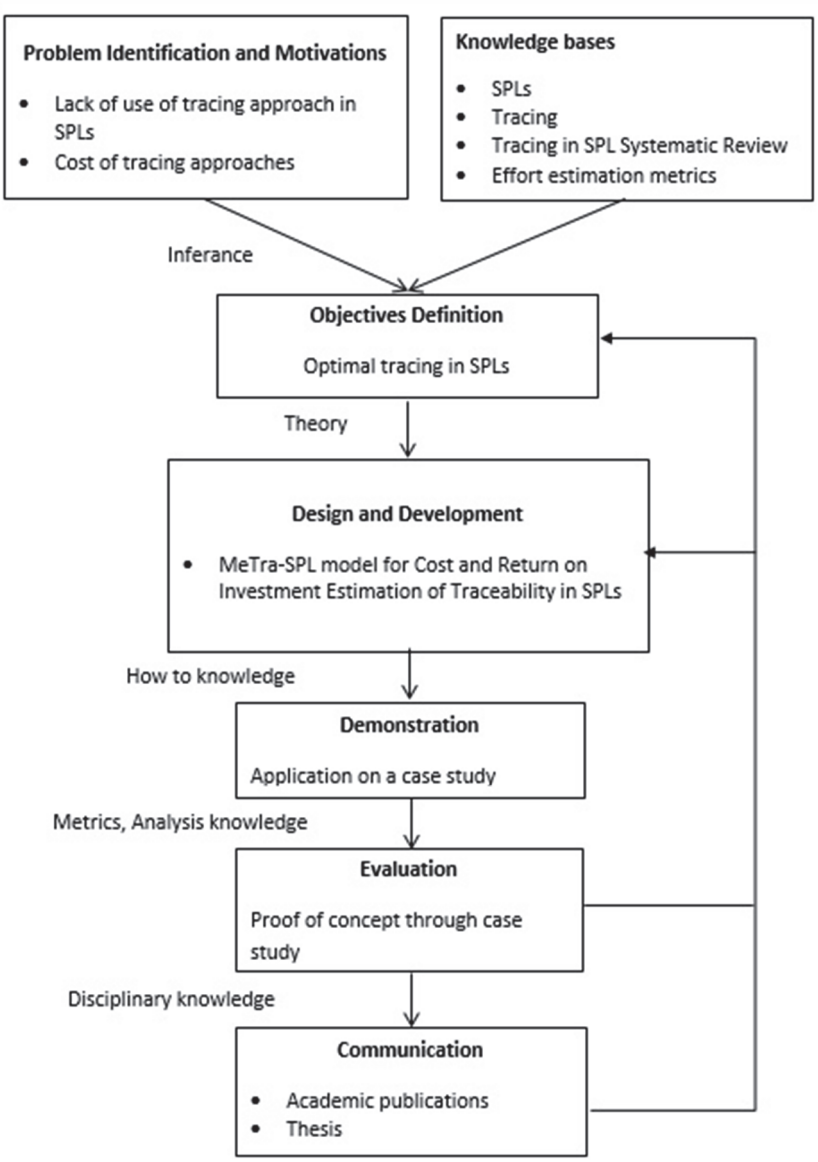

Figure 3. DSRM applied to our research work

- Problem Identification and Motivations

While browsing SPLs literature, we noticed a lack of use of tracing approaches in SPLs, despite the benefices that tracing could provide for SPL maintenance. This is due to the cost of implementing such approaches as they can present several difficulties: large documentation, heterogeneity of documents, the need to link between different products in one side and between the products and the architecture in the other side (Jirapanthong & Zisman, 2005). But what if the cost of maintaining a SPL whithout a rigorouss traceability is larger than the cost of applying a tracing approach? What if there was a way to control traceability costs by limiting superfluous efforts? Those questions were the start and motivation behind our research wok.

- Objectives Definition

After defining the problematic, we set the objective to put in place a solution that helps to adopt an optimal traceability policy, in terms of cost and effort, in the context of Software Product Lines implementation, despite the constraints that may arise because of the complexity and size of the system.

Starting from the knowledge base wich comprises, among others, SPLs and traceability literature, we set a list of research question to define the objectives of our work:

Q1: How to quantify the contribution of a traceability approach in a Software Product Line?

Q2: How to set up an optimal traceability policy in a Software Product Line?

Q2.1: At what stage of the process of setting up a Software Product Line is it interesting to introduce traceability?

Q2.2: What type of traceability (i.e. ad hoc, complete or targeted) is the most profitable?

- Design and Development

To answer previous questions, we set up a cost estimation model, named MeTra-SPL, which helps quantify the cost and Return On Investment of tracing, depending on the traceability approach adopted (ad hoc, complete or targeted), and in which stage of the process it was implemented (Mcharfi et al., 2016).

We set up this artifact starting from an existing cost estimation model, named COPLIMO (Boehm et al. 2004; Yang et al. 2006) that we upgraded to reflect the impact of treaceability in SPLs.

Our study allowed us to analyse how the cost of setting up a traceability policy progresse depending on (i) the phase (i.e. Domain Engineering, Application Engineering or Maintenance), (iii) the type of traceability (targeted, full or ad hoc) and (iv) the number of products to be generated.

We were also able to quantify, through the three phases of SPL life cycle, the ROI if a full traceability, as well as a targeted traceability, is to be deployed.

From this quantification, we could establish that a targeted traceability would engage in the Domain Engineering phase as much effort as a full one, with no immediate ROI. However, it has an advantage in the Maintenance phase where its profitability is palpable as early as first products to generate. Also, and apart from the Domain Engineering phase where no ROI has yet been made, the targeted traceability presents positive results for the two phases: Application Engineering and Maintenance. Therefore, a targeted traceability implemented in the Domain Engineering phase would be more appropriate to adopt, as it is more profitable in the medium and long terms.

- Demonstration

We applied our estimation model on a case study for setting up a tracing approach in a telecom operator SPL that is dedicated to manage offers. We showed, through this case study, how to apply the MeTra-SPL model to estimate cost and ROI for the different tracing approachs, and also how to analyse the results.

- Evaluation

We evaluated our artifact using the same case study. The results obtained reinforced our findings.

- Communication

We shared our findings and research results by publishing papers in journals and conferences proceedings (Mcharfi et al., 2018; Mcharfi, El Asri, Dehmouch, Baya, et al., 2015; Mcharfi et al., 2016; Mcharfi, El Asri, Dehmouch & Kriouile, 2015), which helped us to collect remarks and advises in order to improve our work.

4. Conclusion

Our research work focused on the issue of traceability in Software Product Lines, with the objective of proposing an optimal traceability solution to minimize any additional cost (resources allocation, additional time, documentation management ...) that can be caused by the implementation of a traceability approach, which generally discourages not only engineers but also managers, to undertake such an approach.

To conduct this study, we adopted the DSRM research methodology, and went through its six steps to get consistant findings. We defined the research problem and motivations behind, and the objectives of our study. We proposed a cost and ROI estimation model named MeTra-SPL as artifact, and presented the results of our analysis. We applied our model to a telecom operator case study. It helped us to demonstrate how to apply our model in order to identify the optimal combination of when and how to apply a traicing policy in SPL, and also to evaluate our findings.

To enrich our research methodology, it would be interesting, as future work, to use metrics to evaluate our estimation model. We can also compare the results of our work to other estimation technics to verify their accuracy.

References

Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105.

Anquetil, N., Kulesza, U., Mitschke, R., Moreira, A., Royer, J.-C., Rummler, A., & Sousa, A. (2010). A model-driven traceability framework for software product lines. *Software & Systems Modeling*, *9*, 427–451.

https://doi.org/10.1007/s10270-009-0120-9

Boehm, B., Brown, A. W., Madachy, R., & Yang, Y. (2004, August). A software product line life cycle cost estimation model. In *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04.* (pp. 156-164). IEEE. https://doi.org/ 10.1109/ISESE.2004.1334903

Cavalcanti, Y. C., do Carmo Machado, I., da Mota Silveira Neto, P. A., Lobato, L. L., de Almeida, E. S., & de Lemos Meira, S. R. (2011). Towards metamodel support for variability and traceability in software product lines. In *Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems - VaMoS '11* (pp. 49–57). New York, New York, USA: ACM Press. https://doi.org/10.1145/1944892.1944898

Cleland-Huang, J., Gotel, O., & Zisman, A. (Eds.). (2012). *Software and Systems Traceability*. London: Springer London. https://doi.org/10.1007/978-1-4471-2239-5

Díaz, J., Pérez, J., Alarcón, P. P., & Garbajosa, J. (2011). Agile Product Line Engineering - A Systematic Literature Review. *Software: Practice and Experience*, *41*(8), 921–941. https://doi.org/ 10.1002/spe.1087

Järvinen, P. (2007). Action Research is Similar to Design Science. *Quality & Quantity*, *41*(1), 37–54. https://doi.org/10.1007/s11135-005-5427-1

Jirapanthong, W., & Zisman, A. (2005). Supporting product line development through traceability. In *Proceedings - Asia-Pacific Software Engineering Conference, APSEC* (Vol. 2005, pp. 506–514). https://doi.org/ 10.1109/APSEC.2005.101

McHarfi, Z., El Asri, B., Dehmouch, I., Baya, A., Hammani, F. Z., & Kriouile, A. (2016). MeTra-SPL for an economic analysis of traceability in software product lines. In *Lecture Notes in Electrical Engineering* (Vol. 381, pp. 373–382). https://doi.org/10.1007/978-3-319-30298-0_39

Mcharfi, Z., El Asri, B., Dehmouch, I., Baya, A., & Kriouile, A. (2015). Return on Investment of Software Product Line Traceability in the Short, Mid and Long Term. In *Proceedings of the 17th International Conference on Enterprise Information Systems* (pp. 463–468). SCITEPRESS - Science and and Technology Publications. https://doi.org/10.5220/0005465304630468

Mcharfi, Z., El Asri, B., Dehmouch, I., & Kriouile, A. (2015). Measuring the impact of traceability on the cost of Software Product Lines using COPLIMO. *Proceedings of 2015 International Conference on Electrical and Information Technologies, ICEIT 2015*, 192–197. https://doi.org/10.1109/EITech.2015.7162966

Mcharfi, Z., El, B., & Kriouile, A. (2018). A Systematic Review of Tracing Solutions in Software Product Lines. *American Scientific Research Journal for Engineering, Technology, and Sciences*, *49*(1), 226–247.

Northrop, L. M. (2002). SEI's software product line tenets. *IEEE Software*, *19*(4), 32–40. https://doi.org/10.1109/MS.2002.1020285

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pohl, K., Böckle, G., & Linden, F. Van Der. (2005). *Software product line engineering*.

Ramesh, B., & Jarke, M. (2001). Towards Reference Models for Requirements Traceability. *Software Engineering, IEEE Transactions on*, *27*(1), 58–93. https://doi.org/ 10.1109/32.895989

Simon, H. A. (1996). *The sciences of the artificial*. MIT Press.

Spanoudakis, G., & Zisman, A. (2005). Software Traceability: A Roadmap. In *Handbook Of Software Engineering And Knowledge Engineering* (Vol. III, pp. 395–428). WORLD SCIENTIFIC. https://doi.org/10.1142/9789812775245_0014

Yang, Y., Boehm, B., Baik, J., In, H. P., & Kim, S. S. (2006). A quality-based cost estimation model for the product line life cycle. *Communications of the ACM*, *49*(12), 85-88. https://doi.org/ 10.1145/1183236.1183273