Towards process guidelines for software feature prioritization at different stages of product maturity

Adam Krbusek[1]

[1] *Prague University of Economics and Business, W. Churchill Sq. 1938/4, Prague 3, Czech Republic*

Abstract

Software product management is a strategic discipline that governs a product from its inception to its delivery. This discipline is integrated into the software engineering processes, wherein software product managers prioritize individual requirements and guide development teams on which functionalities to develop next. The prioritization process is supported by established methodologies, such as RICE, MoSCoW, and the Kano Model. However, these methodologies often overlook a crucial aspect – the phase of a product's life cycle. Different sets of functionalities should be prioritized when a product is new compared to when it is well-established. Ignoring this aspect can result in developing the right functionality at the wrong time, wasting resources, and potentially leading to an unsuccessful product. Therefore, this research project aims to introduce process guidelines for software feature prioritization at different stages of product maturity. The Design Science Research Methodology is employed in this research. The first phase of the research introduced a taxonomy of frameworks and methodologies used in software product management. Subsequent phases will explore techniques employed by software product managers, investigate lifecycle-related feature prioritization approaches, challenges and best practices, and ultimately introduce the proposed artifact.

Keywords

Software Product Management, Software Product Development, Feature Prioritization, Methodology

1. Introduction

Over the past few decades, the software industry has emerged as a powerful force in the global economy, transforming the way we live, work, and communicate [1]. Unlike traditional industries such as manufacturing, finance, and healthcare, the software industry is characterized by relatively low barriers to entry [2]. This, combined with rapid technological advancements, creates a hyper-competitive market where product life cycles can be relatively short [3]. In contrast to this, the primary goal of every software company is to develop products that will, for as long as possible, generate revenue or bring other benefits. However, even significant market penetration and a considerable user base cannot ensure the long-term viability of a company or its product, as evidenced by the decline of once-dominant software solutions like Internet Explorer, Yahoo, and MySpace [4, 5, 6].

Ensuring the success of a software product is the responsibility of software product managers [7]. Their primary duty is to develop the right products at the right time and for the right markets [8]. Software product managers (PMs) play a crucial role in companies, with numerous observations indicating that their competence and skills often determine whether a company or its product succeeds or fails [9, 10]. Yet, the job of PMs is becoming increasingly difficult due to the arrival of new technologies that allow customers to easily subscribe or otherwise gain access to software products and just as easily discard them [3, 11]. To ensure that products under their

BIR-WS 2024: BIR 2024 Workshops and Doctoral Consortium, 23rd International Conference on Perspectives in Business Informatics Research (BIR 2024), September 11-13, 2024, Prague, Czech Rep.

adam.krbusek@vse.cz (A. Krbusek)

0009-0003-1871-7384 (A. Krbusek)

supervision do not become irrelevant quickly, PMs need to align strategies according to the product's stage of maturity. For this purpose, the originally marketing concept of the product life cycle (PLC) has been adopted by software product management (SPM) professionals [10].

PLC builds upon the premise that every product typically progresses through 4 key stages during its lifespan – Introduction, Growth, Maturity, and Decline – and at each stage, a different product development strategy needs to be considered [12]. The incorporation of PLC into the SPM is materialized by several comprehensive frameworks. The chief among those is the ISPMA's *Software Product Management Body of Knowledge (SPMBoK)* [10]. Other examples include the *Pragmatic Framework* by Pragmatic Institute [13] and the *SAFe* by Leffingwell [14]. However, the major disadvantage of using the aforementioned frameworks is that they do not give detailed guidance in performing individual SPM activities. One of the most critical of such activities is feature prioritization, which is essential for ensuring that the company builds the right things [15]. There are dozens of specialized methodologies that support this activity [16, 17]; unfortunately, unlike the comprehensive SPM frameworks mentioned above, none of these methodologies considers the product life cycle. This may be problematic because the stage of the product's life cycle is a crucial element influencing which features or functionalities to build next. Not taking PLC into account may lead to wasted resources, missed opportunities, and, ultimately, an unsuccessful product.

Therefore, to address this gap, this research project aims to develop process guidelines for software feature prioritization at different stages of product maturity. The objectives of this research will be achieved by answering the following research questions (RQs):

- **RQ1:** What practices do product managers follow and what challenges do they face when prioritizing features across different stages of software product life cycle?
- **RQ2:** How to design process guidelines for software feature prioritization in different stages of product maturity?
- **RQ3:** How effective are the product management outcomes when employing the proposed process guidelines, in contrast to previous methods?

This paper is organized in the following way. Section 2 describes the current state of knowledge in the subject area. Section 3 focuses on the research approach. Section 4 briefly discusses the initial results, while contributions this research makes are in Section 5.

2. Research background

The existing works can be split into three groups: (i) papers related to PLC in the context of SPM; (ii) papers focusing on feature prioritization methodologies; and (iii) papers addressing PLC in various other contexts.

The first group is composed of comprehensive works focusing on software product management. Geracie and Eppinger's [12] "Product Management and Marketing Body of Knowledge" (ProdBOK) provides a framework and standardized practices for product management and marketing, with a strong emphasis on collaboration and lifecycle management. While it discusses the importance of correct feature prioritization, it lacks actionable, step-by-step guidance on how to achieve this. In contrast, Kittlaus [10], in his SPM handbook, offers guidelines for product management and includes examples of concrete prioritization methodologies such as the *Kano Model*, or the *Analytical Hierarchy Process*. However, no direct link between these methodologies and the broader context of the PLC is offered by the author.

The second group centers on feature prioritization (FP) itself. The most complete work in this area was authored by Bukhsh et al. [16], who conducted a systematic literature review (SLR) to evaluate FP techniques. Similarly, Achimugu et al. [18] identified 49 FP methodologies through an SLR, while Trieflinger et al. [17] performed a grey literature review, categorizing 18 FP

methodologies. Beyond reviews, there are primary research studies where authors propose their own feature prioritization methodologies, such as those by Alrashoud and Abhari [19] and Adhim et al. [20]. Additionally, various papers discuss FP methodologies in different contexts, with example being the work by Wibawa et al. [21]. Nevertheless, like the comprehensive works in the previous paragraph, these studies do not establish a link between FP methodologies and the PLC.

The third group is the most diverse. Wicaksono [22] offers general suggestions on how to monitor for signs of impending product decline. Nikolova [23] argues that a correct and timely approach to quality assurance can extend a product's lifespan. Torres [24] emphasizes that understanding customers' needs and turning them into business opportunities is crucial for building a great product. Fuchs et al. [25] speak about disruptive innovations and their impact on the product life cycle, while Ries [26] discusses pivoting. There are also other works that explore miscellaneous aspects of the PLC. However, none of those found connects the product life cycle with FP methodologies.

In conclusion, while there is extensive research on various aspects of SPM and FP, research linking feature prioritization methodologies directly with the PLC does not exist. All the abovementioned sources may, however, give clues on how to approach the problem.

3. Research approach

This dissertation research employs the Design Science Research (DSR) methodology as defined by Peffers et al. [27]. The DSR was selected for two main reasons. Firstly, it has been successfully applied in similar studies. For example, Bekkers et al. [28] used DSR, complemented by Literature Reviews, Action Research, Focus Groups, and Surveys, to develop the SPM Maturity Matrix. Secondly, DSR's objectives align with the goals of this research project – to solve significant business problems through the creation of innovative artifacts. These artifacts can include algorithms, human/computer interfaces, or methodologies, for example. Furthermore, the strength of the DSR lies in its problem-solving focus, iterative nature, and integration of theory with practice. This approach not only fosters academic contributions but also provides practical solutions. For all the aforementioned reasons, DSR is both applicable and suitable for this work.

The DSR methodology by Peffers et al. [27] consists of the following stages: (1) problem identification and motivation, (2) definition of the solution objectives, (3) design and development of the solution, (4) demonstration, (5) evaluation, and (6) communication about the solution.

The Design Science Research methodology process is structured in nominally sequential order; however, in practice, researchers can initiate at any step and proceed from there [27]. The research process described in this paper started with a problem-centered initiation. The research was divided into three main phases: Preliminary Research, Solution Design, and Solution Validation.

3.1. Preliminary research

The preliminary research phase focuses on the initial two stages of the Design Science Research Methodology. This phase is pivotal as it sets the foundation for the research by identifying the knowledge gap and establishing the solution objectives.

Problem Identification and Motivation: The research commenced with a thorough review of the current state of knowledge in the software product management domain. This involved a detailed examination of the relationship between software product management and the product lifecycle. The aim of this stage was to pinpoint and articulate areas that remain

inadequately covered by existing research, thus identifying the precise gap in the knowledge that the research project aims to bridge. To achieve this goal, a *Literature Review* was conducted.

Definition of the Solution Objectives: Based on the identified knowledge gap, a set of clear and achievable objectives was established. Furthermore, criteria for evaluating the success of the solution were specified.

3.2. Solution design

The solution design phase focuses only on one stage of the Design Science Research Methodology. This stage, however, is the most critical one, necessitating a multifaceted approach to address its complexity.

Design and Development of the Solution: As an initial step, an in-depth examination of the area related to the research gap was conducted. This was done in order to build the theoretical background needed to devise a solution targeting the identified problem space.

Firstly, a *Systematic Mapping Study* (SMS) enhanced by *thematic analysis* was utilized to develop a taxonomy of frameworks and methodologies used in software product management. This helped in building a holistic perspective on the methodological support for various processes and activities conducted by product managers in software companies. Definitions by Petersen et al. [29] for SMS and Braun and Clarke [30] for thematic analysis were followed.

Secondly, a web-based *Survey* will be conducted to identify techniques employed by software product managers. This will add an additional dimension to the knowledge gained from the SMS by understanding the state-of-the-art in the area of software product management. The approach will be grounded in the work of Kitchenham and Pfleeger [31], adhering to the guidelines established by the aforementioned authors.

Thirdly, a *Multivocal Literature Review* will be performed to explore existing approaches to feature prioritization and their relation to the product's lifecycle, thus directly addressing the identified knowledge gap. The multivocal approach will ensure that insights from both academic and professional sources are incorporated. The definition by Garousi et al. [32] will be followed.

Lastly, *Semi-structured Interviews* with software product managers will be performed to understand best practices and challenges associated with feature prioritization at different stages of a product's maturity. These interviews will build upon the results of the Multivocal Literature Review and provide rich real-world insights directly related to the problem space the dissertation research focuses on. Interviews will be conducted according to the guidelines established by Adams [33].

As a next step, based on the theoretical knowledge gained, a solution to the identified problem will be devised. The *Double Diamond design process*, as defined by the British Design Council [34], will be utilized for this purpose. In the "first diamond", through a series of iterations, the problem space will be explored and refined into a precise problem statement. This will be followed by the "second diamond", where, again through a series of iterations, a solution to the aforementioned problem will be developed. During the divergent stage of development, the *Opportunity Solution Tree*, as described by Torres [24], will be used to map all possible opportunities for exploration. In the convergent stage, an interactive feedback loop with feedback from an established *Focus Group*, as described by Krueger and Casey [35], will be utilized to finalize the solution.

3.3. Solution validation

The solution validation phase focuses on the last three stages of the Design Science Research Methodology. The main focus of this phase is the demonstration of the devised solution and the evaluation of its feasibility and usability in real-world settings.

Demonstration & Evaluation: The developed artifact, the "process guidelines for software feature prioritization at different stages of product maturity", will be implemented in a controlled environment, demonstrating its applicability. This will bring forward quantifiable

evidence of the artifact's functionality and its impact on addressing the identified problem. A method of *Action Research*, as described by Staron [36], will be utilized for this purpose.

Communication About the Solution: After all the previous stages are completed, the final findings will be communicated to both scientific and professional audiences. The scientific community will be engaged through software development conferences and peer-reviewed journal publications. The professional audience, on the other hand, will be engaged through articles published on popular platforms like Medium.com. As professional audiences have different expectations, a practical guide on how to implement the process guidelines in software companies will be compiled, enhancing the publication's impact and reach.

4. Initial results

In the first phase of the research, frameworks and methodologies used in software product management were categorized, leading to the introduction of the CoSuCo taxonomy. This taxonomy was developed through a Systematic Mapping Study, enhanced by thematic analysis. In total, 82 research papers were examined in-depth, identifying 122 frameworks and methodologies used in software product management. The taxonomy is organized in a hierarchical tree structure consisting of 3 central categories, which are divided into 9 categories comprising of 22 sub-categories.

Currently, preparations for a web-based survey are underway. This survey, which will materialize as an online questionnaire, will be distributed via LinkedIn to a pre-selected group of SPM professionals. The goal is to identify the techniques and tools used by software product managers in real-world settings.

5. Scientific and practical contribution

Based on the literature review conducted using selected bibliographic databases (Web of Science, Scopus, IEEE Xplore, ACM Digital Library), it was determined that the topic is significantly underserved. In the last few years, several authors have published research papers related to the subject area (see Section 2). None have, however, specifically addressed the relationship between a product's life cycle and feature prioritization.

Therefore, this research will add to the existing body of knowledge by uncovering the nuances of the feature prioritization process and by introducing new process guidelines integrating a product's life cycle. Additionally, the proposed artifact is expected to be immediately applicable in professional settings. There is a significant community of practitioners, as demonstrated by product management-focused conferences like ProductCon[2], where thousands of professionals meet and exchange the most up-to-date knowledge and trends. This indicates that a large group of people could benefit from the results of this research.

Acknowledgment

I would like to acknowledge my supervisors, Prof. Alena Buchalcevova and Dr. Michal Dolezel, from the Department of Information Technologies (FIS), Prague University of Economics and Business, for their guidance and valuable suggestions.

2 ProductCon is a product management conference organized quarterly by Product School (URL: https://productschool.com/productcon)

References

[1] V. Matyushok, V. Krasavina, A. Berezin, J. García, The global economy in technological transformation conditions: A review of modern trends, Economic Research-Ekonomska Istraživanja 34 (2021) 1–41. doi:10.1080/1331677X.2020.1844030.

[2] E. Karp, Why It's So Hard To Fund A Manufacturing Startup, Forbes, 2020. URL: https://www.forbes.com/sites/ethankarp/2020/12/14/why-its-so-hard-to-fund-a-manufacturing-startup/.

[3] A.A. Afifi, Demand Forecasting of Short Life Cycle Products Using Data Mining Techniques, in: Artificial Intelligence Applications and Innovations, Springer International Publishing, Cham, 2020, pp. 151–162. doi:10.1007/978-3-030-49161-1_14.

[4] A.N. Aufa, Critical Analysis of a Technology-Based Enterprise: a Case Study of Yahoo!, TIJAB 2 (2018) 39–49. doi:10.20473/tijab.V2.I1.2018.39-49.

[5] J. Tamary, D.G. Feitelson, The rise of Chrome, PeerJ Computer Science 1 (2015) e28. doi:10.7717/peerj-cs.28.

[6] F. Gillette, The Rise and Inglorious Fall of Myspace, Bloomberg, 2011. URL: https://www.bloomberg.com/news/articles/2011-06-22/the-rise-and-inglorious-fall-of-myspace.

[7] O. Springer, J. Miler, M. Wróbel, Strategies for Dealing With Software Product Management Challenges, IEEE Access 11 (2023) 55797–55813. doi:10.1109/ACCESS.2023.3282605.

[8] C. Ebert, Software Product Management, IEEE Software 31 (2014) 21–24. doi:10.1109/MS.2014.72.

[9] C. Ebert, The impacts of software product management, Journal of Systems and Software 80 (2007) 850–861. doi:10.1016/j.jss.2006.09.017.

[10] H.-B. Kittlaus, Software Product Management: The ISPMA®-Compliant Study Guide and Handbook, Springer, Berlin, Heidelberg, 2022.

[11] G2, G2 Software Buyer Behavior Report 2022, 2023. URL: https://research.g2.com/2022/buyer-behavior-report.

[12] G. Geracie, S.D. Eppinger, The Guide to the Product Management and Marketing Body of Knowledge: ProdBOK, Product Management Educational Institute, Carson City, Nevada, 2013.

[13] Pragmatic Institute, The Pragmatic Framework, 2023. URL: https://www.pragmaticinstitute.com/framework/.

[14] D. Leffingwell, SAFe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises, Addison-Wesley Professional, Boston, MA, 2018.

[15] J. Münch, S. Trieflinger, D. Lang, The Product Roadmap Maturity Model DEEP: Validation of a Method for Assessing the Product Roadmap Capabilities of Organizations, in: Software Business, Springer International Publishing, Cham, 2019, pp. 97–113. doi:10.1007/978-3-030-33742-1_9.

[16] F.A. Bukhsh, Z.A. Bukhsh, M. Daneva, A systematic literature review on requirement prioritization techniques and their empirical evaluation, Computer Standards & Interfaces 69 (2020) 103389. doi:10.1016/j.csi.2019.103389.

[17] S. Trieflinger, J. Münch, E. Bogazköy, P. Eißler, J. Schneider, B. Roling, How to Prioritize Your Product Roadmap When Everything Feels Important: A Grey Literature Review, in: ICE/ITMC 2021, 2021. doi:10.1109/ICE/ITMC52061.2021.9570243.

[18] P. Achimugu, A. Selamat, R. Ibrahim, M.N. Mahrin, A systematic literature review of software requirements prioritization research, Information and Software Technology 56 (2014) 568–585. doi:10.1016/j.infsof.2014.02.001.

[19] M. Alrashoud, A. Abhari, Perception-Based Software Release Planning, Intelligent Automation & Soft Computing 21 (2015) 175–195. doi:10.1080/10798587.2014.960229.

[20] K.K. Adhim, A. Hudaib, B. Al-Shboul, Efficient requirement prioritization based on enhanced multi-verse optimizer, Journal of Theoretical and Applied Information Technology 97 (2019) 4966–4977.
[21] A.S. Wibawa, E.K. Budiardjo, K. Mahatma, Improving the Quality of Requirements Engineering Process in Software Development with Agile Methods: a Case Study Telemedicine Startup XYZ, in: 2021 ICADEIS, 2021, pp. 1–5. doi:10.1109/ICADEIS52521.2021.9701962.
[22] S. Wicaksono, Revisiting Requirement Analysis in Product Life Cycle of Software Product, International Journal of Advances in Social and Economics 1 (2019). doi:10.33122/ijase.v1i3.46.
[23] Z. Nikolova, Testing Strategies in an Agile Context, in: The Future of Software Quality Assurance, Springer, Cham, 2020, pp. 111–121. doi:10.1007/978-3-030-29509-7_9.
[24] T. Torres, Continuous Discovery Habits: Discover Products that Create Customer Value and Business Value, Product Talk LLC, Talc, 2021.
[25] C. Fuchs, F.J. Golenhofen, Mastering Disruption and Innovation in Product Management: Connecting the Dots, Springer International Publishing, Cham, 2019.
[26] E. Ries, The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Currency, 2011.
[27] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems 24 (2007) 45–77. doi:10.2753/MIS0742-1222240302.
[28] W. Bekkers, I. van de Weerd, M. Spruit, S. Brinkkemper, A Framework for Process Improvement in Software Product Management, in: Systems, Software and Services Process Improvement, Springer, Berlin, Heidelberg, 2010, pp. 1–12. doi:10.1007/978-3-642-15666-3_1.
[29] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology 64 (2015) 1–18. doi:10.1016/j.infsof.2015.03.007.
[30] V. Braun, V. Clarke, Thematic analysis, in: APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological, American Psychological Association, Washington, DC, US, 2012, pp. 57–71. doi:10.1037/13620-004.
[31] B.A. Kitchenham, S.L. Pfleeger, Personal Opinion Surveys, in: F. Shull, J. Singer, D.I.K. Sjøberg (eds.), Guide to Advanced Empirical Software Engineering, Springer, London, 2008, pp. 63–92. doi:10.1007/978-1-84800-044-5_3.
[32] V. Garousi, M. Felderer, M.V. Mäntylä, Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, Information and Software Technology 106 (2019) 101–121. doi:10.1016/j.infsof.2018.09.006.
[33] W.C. Adams, Conducting Semi-Structured Interviews, in: Handbook of Practical Program Evaluation, John Wiley & Sons, Ltd, 2015, pp. 492–505. doi:10.1002/9781119171386.ch19.
[34] Design Council, Eleven lessons: A study of the design process, 2007. URL: https://www.designcouncil.org.uk/fileadmin/uploads/dc/Documents/ElevenLessons_Design_Council%2520%25282%2529.pdf.
[35] R.A. Krueger, M.A. Casey, Focus Groups: A Practical Guide for Applied Research, SAGE Publications, 2014.
[36] M. Staron, Action Research in Software Engineering: Theory and Applications, Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-32610-4.