

A Multivocal Literature Review on Technical Debt in Startups

MERCY NJIMA, University of Antwerp, Belgium

SERGE DEMEYER, University of Antwerp and Flanders Make, Belgium

Startups often operate under limited resources and time, and may intentionally accumulate technical debt to meet immediate market needs. However, this intentional accumulation requires careful management to avoid long-term negative consequences for the startup. We conducted a Multivocal Literature Review of the existing body of knowledge in order to understand the state of the art and the practice of technical debt in startups and its implications. The key outcome of this study is the creation of a framework that provides a comprehensive view of technical debt in startups including its dimensions and attributes as well as a taxonomy that describes the different forms it takes and directions for future work.

ACM Reference Format:

Mercy Njima and Serge Demeyer. 2023. A Multivocal Literature Review on Technical Debt in Startups. 1, 1 (November 2023), 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

In software development, technical debt is a term that is used to describe the compromises that allow a team to deliver a project quickly at the cost of increased maintenance in the future [5]. These compromises create software "debt" that must be repaid in the future to avoid "interest" in the form of reduced maintainability and evolution [6]. Most research on technical debt has been focused on mature software teams say in large organizations who may have less pressure and, therefore, reason about technical debt very differently than software startups [1, 4].

Startups are organizations that create scalable, high-tech innovative products and services under conditions of extreme uncertainty, limited resources, multiple influences, new technologies, and evolving markets [7, 8]. Startups often operate under resource constraints and time pressure, have inexperienced developers, lack a defined development process, and lack autonomy of decision-making, which lead to the intentional or unintentional accumulation of technical debt [1]. The situation is further exacerbated by the need for rapid product development and iteration to gain a competitive advantage in the market [?].

While technical debt is as important to software startups as it is to mature companies, the kind of decisions to be made and the consequences of making the wrong decisions are not the same, justifying further research on technical debt in software startups [8]. In the recent past, there has been a few research efforts to investigate the technical debt phenomenon in startups. However, no structured investigation of the area has been performed and none of the systematic literature reviews or mapping studies on technical debt address technical debt in startups. The goal of this work is to identify and understand the main contributions of the state-of-art and practice of technical debt in startups.

Authors' addresses: Mercy Njima, University of Antwerp, Antwerp, Belgium, mercy.njima@uantwerpen.be; Serge Demeyer, University of Antwerp and Flanders Make, Antwerp, Belgium, serge.demeyer@uantwerpen.be.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

Manuscript submitted to ACM

Given the industry-oriented nature of startups and the limited number of formally-published literature in this area, we conducted a Multivocal Literature Review (MLR) [?]. We chose this method because a significant amount of information related to the research topic is available in grey literature, including technical reports, blogs, and standards that are not typically published in academic sources. MLRs recognize the need for grey literature rather than constructing evidence from only the knowledge reported in academic settings. This work was conducted with the goal of answering the following research questions:

- (1) What is the current state-of-the-art research and practice of technical debt in startups including activities, approaches, and tools?
- (2) What factors influence the accumulation and management of technical debt in software startups?
- (3) What are the challenges and benefits of technical debt for software startups?
- (4) How do startups manage technical debt and when does that become a priority?
- (5) What are the reported mitigating strategies of technical debt in startups?
- (6) What are the possible future directions of technical debt research and practice in startups?

The rest of this paper is structured as follows: The design of our investigation is explained in Section 2. Section 3 discusses the findings and Section 4 presents a discussion of the findings. We present the conclusion and examine threats to validity in Section 5. Finally, Section 6 presents future work.

2 RESEARCH DESIGN

In this section, we outline the Multivocal Literature Review (MLR) technique we adopted in this work. We followed the established procedures and guidelines defined by Garousi et al. [?].

2.1 Motivation

The purpose of this MLR is to provide a comprehensive and inclusive understanding of technical debt in startups by incorporating a wide range of sources and perspectives from both published and non-published literature [? ?]. It is a form of Systematic Literature Review (SLR) that includes grey literature such as blog posts, videos, and white papers, in addition to the formal literature like journal and conference papers. This approach is particularly valuable in software engineering in startups, where it is essential to capture both the "state of the practice" and research and therefore, provide a more holistic view of the field [?].

The MLR process is illustrated in Fig. 1. It consists of three phases: planning, conducting and reporting. We discuss each phase in the rest of this section. This MLR search was conducted in October 2023 and the analysis and reporting was completed in December 2023.

2.2 Conducting the MLR

The MLR search was conducted in three stages.

- (1) Data sources and search strategy: For the formal literature, we selected the list of relevant bibliographic sources following the suggestions of [3], since these sources are recognized as the most representative in the software engineering domain. The list includes: ACM Digital Library, IEEEExplore Digital Library, Science Direct, Scopus, Google Scholar, CiteSeer library, Inspec, Springer link. Thereafter, we applied the snowballing method in order not to miss any potentially relevant publications as outlined in [?]. On the other hand, we used Google's web search engine (<http://www.google.com>) to source for the grey literature. A preliminary search helped to narrow

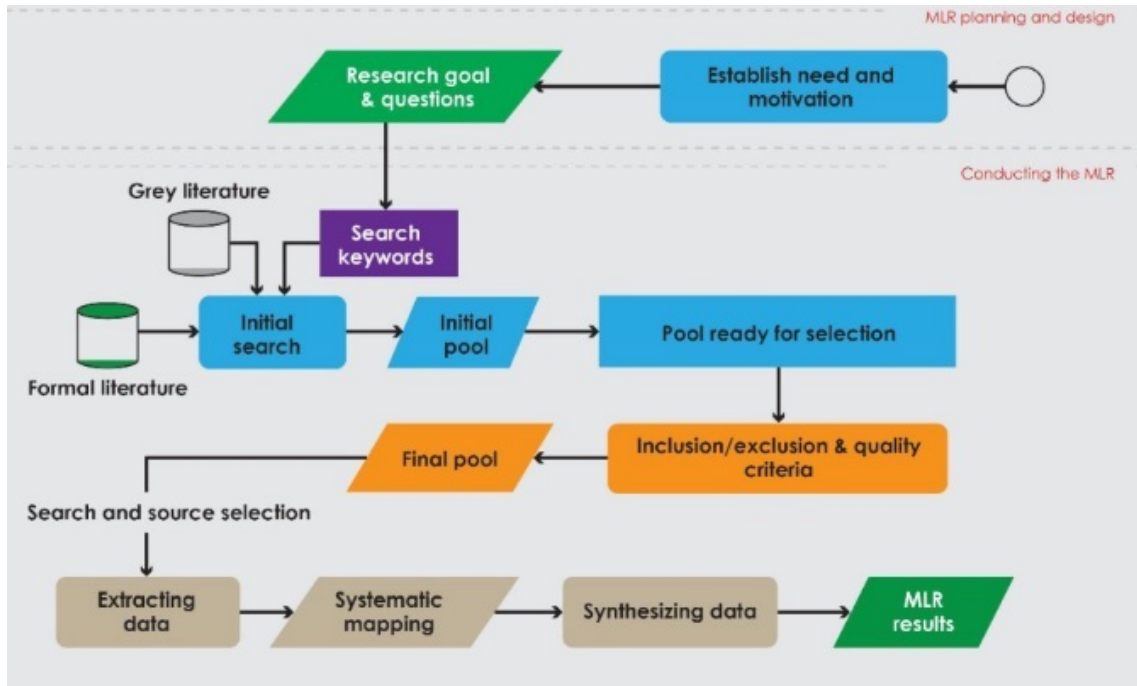


Fig. 1. Multivocal literature review (MLR) steps adopted in this research(Work in Progress)

down the keywords for a suitable search query. We used the terms "technical debt" AND "startup" for both the formal and the grey literature searches.

- (2) Inclusion and exclusion criteria: The results were reviewed against a set of inclusion and exclusion criteria. Search results were included if (a) the main objective of the result which may be a primary or secondary study was either to discuss or investigate technical debt in startups. This inclusion criterion defines the basic scope of our study. (b) must have a software engineering context. To keep our study to manageable levels, the scope of our study was restricted to SE related papers. Search results were excluded if they were (a) a duplicated record (b) only mention technical debt in an introductory manner and do not fully or partly focus on its occurrence in startup context.
- (3) Quality assessment: We adopted the quality assessment checklist of grey literature from Garousi et al [?].

3 FINDINGS AND DISCUSSION

3.1 State of the art

- Publication trend

3.2 Demographics

- Classification / types of literature

3.3 Technical Debt Management Activities

The technical debt management process includes activities used to control and reduce the technical debt in a software project. Table 1 present the various technical debt management activities and the corresponding reference to the work that presents them.

Table 1. Technical Debt Management Activities and the works that report on them

TDM Activity	White Literature	Grey Literature
TD Repayment		
TD Identification	[?]	
TD Measurement		
TD monitoring	[1]	
TD prioritization		
TD communication		[?]
TD prevention	[?]	
TD representation/documentation		

4 CHALLENGES AND FUTURE DIRECTIONS

5 CONCLUSION

REFERENCES

- [1] T. Besker, A. Martini, R. Edirisooriya Lokuge, K. Blincoe, and J. Bosch. 2018. Embracing Technical Debt, from a Startup Company Perspective. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 415–425. <https://doi.org/10.1109/ICSME.2018.00051>
- [2] Barbara Kitchenham and Pearl Brereton. 2013. A systematic review of systematic review process research in software engineering. *Information and Software Technology* 55, 12 (2013), 2049–2075. <https://doi.org/10.1016/j.infsof.2013.07.010>
- [3] Barbara Kitchenham, Stuart Charters, et al. 2007. Guidelines for performing systematic literature reviews in software engineering.
- [4] Zengyang Li, Paris Avgeriou, and Peng Liang. 2015. A Systematic Mapping Study on Technical Debt and Its Management. *J. Syst. Softw.* 101, C (March 2015), 193–220. <https://doi.org/10.1016/j.jss.2014.12.027>
- [5] Everton da S. Maldonado and Emad Shihab. 2015. Detecting and quantifying different types of self-admitted technical Debt. In *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*. 9–15. <https://doi.org/10.1109/MTD.2015.7332619>
- [6] Carolyn Seaman, Yuepu Guo, Nico Zazworka, Forrest Shull, Clemente Izurieta, Yuanfang Cai, and Antonio Vetrò. 2012. Using technical debt data in decision making: Potential decision approaches. In *2012 Third International Workshop on Managing Technical Debt (MTD)*. 45–48. <https://doi.org/10.1109/MTD.2012.6225999>
- [7] S.M. Sutton. 2000. The role of process in software start-up. *IEEE Software* 17, 4 (2000), 33–39. <https://doi.org/10.1109/52.854066>
- [8] Michael Unterkalmsteiner, Pekka Abrahamsson, Xiaofeng Wang, Anh Nguyen-Duc, Syed M. Ali Shah, Sohaib Shahid Bajwa, Guido H. Baltes, Kieran Conboy, Eoin Cullina, Denis Dennehy, Henry Edison, Carlos Fernández-Sánchez, Juan Garbajosa, Tony Gorschek, Eriks Klotins, Laura Hokkanen, Fabio Kon, Ilaria Lunesu, Michele Marchesi, Lorraine Morgan, Markku Oivo, Christoph Selig, Pertti Seppänen, Roger Sweetman, Pasi Tyrväinen, Christina Ungerer, and Agustín Yagüe. 2016. Software Startups - A Research Agenda. *e-Informatica* 10, 1 (2016), 89–124. <https://doi.org/10.5277/e-Inf160105>
- [9] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 1–10.