Journal of Health Informatics in Africa

Implications of Software Platform Architecture and Documentation on Developer Productivity: A Case of the Malawi Point of Care EMR Software

Emmanuel Mkambankhani [a], *, Tiwonge Davis Manda [a]

Department of Computing, University of Malawi, Zomba, Malawi

Background and Purpose: Software platform architecture and documentation affect platform customisability, developer productivity, and third-party contributions. This study examines how the Malawi Point of Care Electronic Medical Records Software (PoC-EMRS) architecture and documentation shape the platform's customisation, developer productivity, and third-party contributions. The Malawi PoC-EMRS, as the primary case of analysis, was compared to CommCare and DHIS2 as configurable software platforms. Theoretically, we draw on Generativity and the Boundary Resource Model (BRM). Generativity evaluates the overall capacity of an artefact to produce solutions, for diverse use cases. BRM was used to evaluate how owners of the PoC-EMRS facilitate ecosystem value co-creation through documentation and exposure of APIs.
Methods: Primary and secondary data were collected through interviews, observations and document analysis. Both qualitative and quantitative data were analysed to find common themes.
Results: Malawi PoC-EMRS is less configurable than DHIS2 and CommCare, necessitating more developer effort to support a variety of use cases. Though Malawi PoC-EMRS exposes boundary resources, it lacks incentives to attract third-party developers. Lack of or limited access to detailed documentation also negatively affects internal and third-party development productivity.
Conclusions: We established that CommCare and DHIS2 have the following strengths. First, the platforms feature standardised interfaces that allow third parties to design, configure, and customise the platforms into solutions for a varied range of use cases. Second, CommCare and DHIS2 have comprehensive documentation that is accessible through online repositories, communities of practice and demo sandboxes. Applying the strengths of these platforms in the development of the Malawi PoC-EMRS is likely to boost developer productivity.

Keywords: Health Systems, Configurable Platforms, Software Architecture, Software Documentation, Software Development Productivity.

1 Introduction

Software architecture is defined by ISO/IEC/IEEE 42010:2011 as the fundamental concepts or properties of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution [1]. Software architecture is critical to the success of software platforms, as it manifests not only business and domain requirements but also other attributes like adaptability, maintainability, flexibility, scalability, performance, modifiability, and security [2]. An architecture that is well documented improves communication among stakeholders, serves to restrict design alternatives, channels the creativity of developers, reduces design and system complexity, and acts as a foundation for training new members of a team [3]. Thus, software architecture and its documentation influence the implementation of software platforms and have an impact on software developer productivity and third-party innovation [4]. Currently, software developers are moving towards developing their platform architectures to be more configurable and generic to reach a broader range of use cases beyond the targeted use cases that their platforms were initially designed for, and also to open opportunities for third-party contributions [5].

*Corresponding author address: emmanuel.mkambankhani@gmail.com
 J Health Inform Afr. 2023;10(2):1-9. DOI: 10.12856/JHIA-2023-v10-i2-383

In Malawi, the Ministry of Health and Population (MOH), in coordination with the Digital Health Division (DHD) and the e-Government department, commissions various implementing organisations to develop eHealth solutions, through donor-funded projects. Following are some of the leading organisations in developing eHealth solutions: Baobab Health Trust (BHT), Elizabeth Glaser Paediatric AIDS Foundation (EGPAF), Luke International Norway (LIN), D-Tree, and Lighthouse. Collectively, these organisations have, over the past two decades, provided digital health solutions that have improved the delivery of healthcare services to patients or recipients of care, enrolled in different health programs. Currently, the District Health Information System 2 (DHIS2), the Malawi Point of Care Electronic Medical Record Software (PoC-EMRS), and CommCare are the leading software platforms for HIS in Malawi [6] [7] [8] [9].

Since 2001, Malawi PoC-EMRS has been evolving as an open-source solution for Point of Care (PoC) systems in Malawi, with over 206 health facilities using it as a Point of Care solution and 520 health facilities using it as a retrospective data entry application. Despite its widespread use, the development of the Malawi PoC-EMRS has hardly attracted external contributions. Thus, organisations responsible for its development have had to considerably keep expanding their internal software development teams, to meet growing demands for software products. Efforts to acquire skilled and experienced software developers, as well as bring new developers on board, can be quite costly. For example, new developers may require extensive training and onboarding efforts. In contrast to the Malawi PoC-EMRS, DHIS2 and CommCare solutions development in Malawi has leveraged the contributions of smaller teams from across multiple organisations. The two platforms have a growing international reach and increasing third-party initiatives. According to extant literature, internal and external development productivity and third-party innovation can be attributed to software architecture [3] [4] [10] [11] [12].

In order to improve the Malawi PoC-EMRS architecture and documentation, we sought to investigate the implications of its software's architectural design and documentation and compare it with the configurable platform approaches of CommCare and DHIS2. Following this exercise, our aim was to inform software architectural design revisions and improvement of documentation for the Malawi PoC-EMRS, to facilitate improvements in developer productivity and third-party contributions. The study was guided by the following question: *"How does the software architecture and documentation of the Malawi PoC-EMRS affect internal and external productivity and innovation?"* The study also analysed how the strengths of DHIS2 and CommCare can be leveraged to increase the productivity of internal and third-party contributions in Malawi PoC-EMRS.

2 Materials and Methods

2.1 Case Selection

During the study, the primary unit of inquiry was the Malawi PoC-EMRS platform. However, CommCare and DHIS2 were used as comparison benchmarks. The platforms were selected for two major reasons. First, they are widely deployed in Malawi's health sector. For example, the Malawi PoC-EMRS has the point of care and ART eMastercard modules deployed in 726 sites [7]. CommCare is used for mHealth solutions for Community Health Workers (CHW) and it is the platform upon which ART Back to Care (B2C) applications and a key Logistic Management Information System (LMIS) are based [8]. DHIS2 is also widely used, forming the basis for the National Health Management Information System [9], the Integrated Community Health Information System (iCHIS), and the National Agriculture Management Information System (NAMIS). Further to this, PEPFAR and WHO use DATIM, a DHIS2-based implementation [13]. Second, the study platforms were chosen because they are open-source. In being open-source, the platforms provided installation and code analysis flexibility, allowing analysis of each platform's architecture, maintainability, customization, and possibilities for facilitating third-party contributions.

2.2 Research Paradigm and Approach

Research paradigms, often termed "philosophical worldviews," are fundamental beliefs, assumptions, and values that govern research decisions. The study employed a constructivist view. Researchers may use

 J Health Inform Afr. 2023;10(2):1-9. DOI: 10.12856/JHIA-2023-v10-i2-383

constructivism to develop subjective meanings of research phenomena when interacting with participants or research objects [14] [15]. To triangulate data and control for biases, the study used multiple sources and data collection methods, including questionnaires (for structured questions), participant observations, document analysis, and artefact analysis. Collected data include perceptions of developers for the Malawi PoC-EMRS, DHIS2, and CommCare in terms of ease of mastery, adaptability, and boundary resource use.

2.3 Sampling

Participants were selected using purposive and snowball sampling. First, purposive sampling relies on the researcher's judgement to select the appropriate sources of data to meet study goals. Purposive sampling was used to acquire data from CommCare, DHIS2, and Malawi PoC-EMR platform developers. Key people from organisations and government departments that develop and implement digital health solutions using the platforms were contacted to participate in the research. Second, snowballing was utilised to locate other participants since the research population was specific and more data was needed to qualitatively address the research question. Snowballing is useful when researchers know little about a group or organisation [16]. Specialists from the following five organisations participated in the study: Baobab Health Trust (BHT), Elizabeth Glaser Paediatric AIDS Foundation (EGPAF), Last Mile Health, Luke International Norway (LIN), and the Ministry of Health's Digital Health Division (DHD). By applying the techniques mentioned, 22 individuals were identified to participate in the study but only 19 participants participated in the study. Table 1 depicts participants' positions by the platform they responded to.

Table 1: Participant’s Platform and Position

#	Position	CommCare	DHIS2	Malawi PoC-EMRS
1	Digital Health Specialist	0	1	0
2	DHIS2 Programme Manager	0	1	0
3	Informatics Specialist	1	3	0
4	Software Architect	1	0	0
5	Software Developer	2	2	7
6	Systems Analyst	0	0	1
	Total	**4**	**7**	**8**

2.4 Data Collection

The study used a self-administered questionnaire, participant observations, document analysis, and artefact analysis. A self-administered questionnaire was used in the form of a Google form, sent through email. This was done to allow participants to respond at their own pace. The questionnaire was followed by a virtual and in-person interview, to clarify participants’ responses. Second, the study used participant observations, to gain more insights beyond what participants provided through the questionnaires and interviews. Participant observations have been documented to foreground hidden issues that may be sensitive for people to reply to [17]. Participants were observed through two exercises. The first author attended daily stand-up meetings and sprint review meetings and talked to developers about the study platforms. Furthermore, three of the 19 participants that responded to the questionnaire were chosen for controlled observations, through practical tasks. The participants were first oriented on the platforms after which they were given 30 minutes to develop a simple patient registration form, aimed at collecting demographics.

Document analysis also revealed hidden meanings. The definition of documents goes beyond text data; it also includes audio-visual data (photographs, diagrams, animations, video, and sounds) and electronic data like screenshots [18]. Hence, researchers reviewed technical platform documents, attended platform courses offered as video clips, PowerPoint slides, conferences, and webinars, and generated qualitative data notes. Finally, the researchers examined platform artefacts by executing simple tasks on online platforms. The researchers then checked open-source repositories for GitHub commits, forks, and pull requests from

 J Health Inform Afr. 2023;10(2):1-9. DOI: 10.12856/JHIA-2023-v10-i2-383

internal and external developers. The researchers were able to assess platform contributions and whether internal and third-party contributors followed software programming standards.

2.5 Data Analysis

Thematic Analysis was utilised to analyse data based on the generativity and boundary resources model conceptual frameworks [19] [20] [21] [22] [23]. Thematic analysis was used to discover themes in the questionnaire, participant observations, document analysis, and artefact analysis data on developers' viewpoints and platform issues. The researchers then summarised the data to identify themes. Ultimately, the researchers linked and categorised themes according to the conceptual framework to analyse data based on platform generative capability and constrained resource availability and use [17] [24]. The primary themes derived from the conceptual framework encompassed the analysis of platform architecture's capacity for leverage, adaptability, and documentation, with their impact on the productivity of software developers and contributions from third-party entities.

3 Results

3.1 Platform Architecture on Productivity in Solutions for Different Use Cases

The investigations on how the architectural design of the Malawi PoC-EMRS shapes software developer productivity were based on the theory of Generativity [21]. The focus of this study was the generative technology and generative capacity. A generative technology is described to have five characteristics; the capacity for leverage, adaptability, ease of mastery, accessibility, and transferability [19]. Figure 1 illustrates the responses of participants on each of the study platforms' capacity for leverage. The numbers in the figure indicate a distribution of responses by the 19 participants in the study.

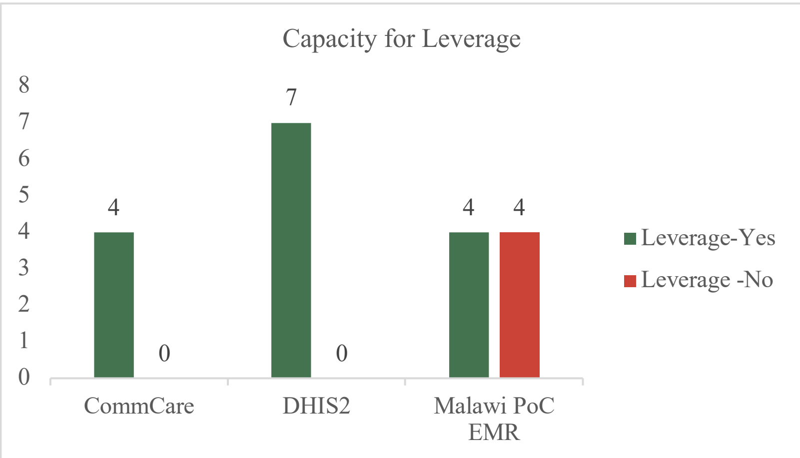

Figure 1: Number of Participants and Capacity for Leverage

From the data, it was established that the leverage of DHIS2 is attributed to its easy-to-use functionalities implemented by drag-and-drop features within the system's user interface. Thus, making it simpler to customise existing or develop new customizable data collection apps, as well as developing dashboards and analytics as one of the participants indicated below.

"DHIS2 has drag and drop functionality and it does not really need one to have the IT background to customise or manage it" (DHIS2 Programme Manager-DHD, 2021).

Similarly, in CommCare, leverage is provided by how simple it is for platform users to develop mobile data collection tools quickly. The CommCare HQ interface facilitates the implementation of data collection apps

 J Health Inform Afr. 2023;10(2):1-9. DOI: 10.12856/JHIA-2023-v10-i2-383

and data management that would otherwise have to be created from scratch. This is clear from the comments that follow:

"The platform provides interfaces that scaffold the heavy lifting e.g. authentication, data storage, etc" (Software Architect EGPF, 2022).

"There are a lot of tools one can use to come up with an app easily" (Software Developer-EGPAF,2022).

In the case of Malawi PoC-EMRS, it was established that the platform does not fully have the capacity for leverage. This is so because a common response was not obtained from participants who are using the platform. The following are some of the expressions that support the preceding statement:

"It is user-friendly and has clear guides on what needs to be done" (Systems Analyst BHT, 2022).

"There are a lot of workarounds required to add specific features to the UI using the touchscreen tool kit. There's no standardisation or reusability of business logic and modularization is terrible." (Software Developer EGPAF, 2022)

In relation to leverage, adaptability is another attribute, which is described as the potential of a platform to be flexible to change, for use in different contexts than the one it was designed for. Figure 2 illustrates the responses of participants to each platform's adaptability.

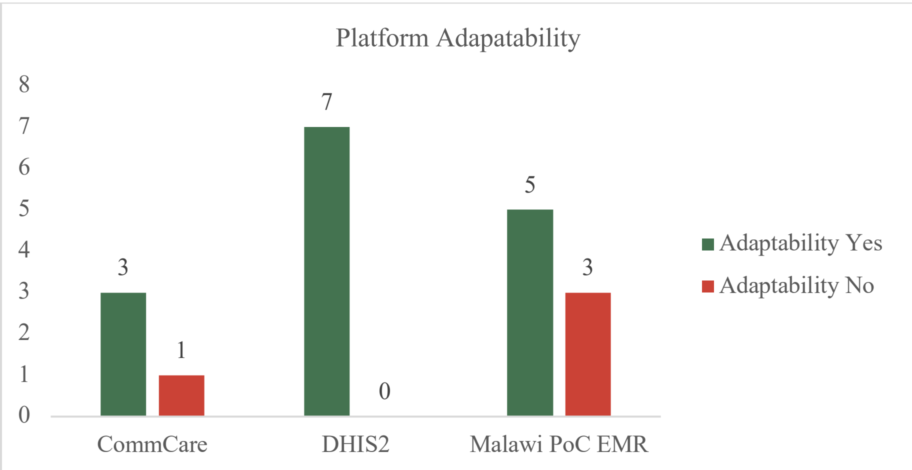

Figure 2: Adaptability of Platforms

From another perspective, adaptability can be expressed in how the platforms respond to multiple end-user devices. From the results in artefact analysis, all three platforms are capable of running on mobile and desktop devices. Front-ends for CommCare HQ, DHIS2, and Malawi PoC-EMRS are all capable of resizing and fitting on multiple device screens for desktops, tablets, and phones.

3.2 Platform Architecture Design and Third-Party Development

Boundary resources are software tools and regulations, such as application programming interfaces (APIs) and software development kits (SDKs), that act as an interface between platform owners and application developers, as well as allow the development of third-party applications. Table 2 shows the existence of boundary resources that each platform is able to expose.

 J Health Inform Afr. 2023;10(2):1-9. DOI: 10.12856/JHIA-2023-v10-i2-383

Table 2: Existence of Boundary Resources in Platforms

Boundary Resource	CommCare	DHIS2	Malawi PoC-EMRS
API endpoints	✓	✓	✓
Open Source code Repository	✓	✓	✓
Debugging Tools	✓	✓	✓
Reusable Libraries		✓	✓
SDK		✓	

GitHub forks are one indicator of the presence of third-party developers. Forking is the process of creating a new software repository by copying an existing one [25]. This enables third-party developers to experiment with their own local repository without affecting the original project [26]. Furthermore, forking allows third-party developers to submit pull requests to original code repositories to which they do not have access rights contribute [25] [27]. Table 3 shows the summary of repository forks that third-party developers made to CommCare, DHIS2, and Malawi PoC-EMRS.

Table 3: Summary of Platform Forks

Platform	Platform module	Number of forks
DHIS2	DHIS2 Core	263
	DHIS2 application platform	7
	DHIS2 UI	4
CommCare	CommCare HQ	196
	CommCare Mobile	22
Malawi PoC-EMRS	EMR-API	3
	Core	3

3.3 Platform Documentation and Productivity

Documentation is a form of boundary resource that gives actors the generative capacity to utilise other boundary resources on a platform [23]. As shown in Table 4, DHIS2 and CommCare platforms have a variety of information available online that enables third-party development via user manuals, online courses, and conferences.

Table 4: Participants able to access platform resources

Boundary Resource	Number of Participants		
	CommCare	DHIS2	Malawi PoC-EMRS
SDK/API documentation	3	7	5
Technical Specification Documents	4	7	3
User Manuals	4	6	3
Training material or tutorials	4	6	3
Demo site/platform for practice	0	1	0
Platform Academy	0	1	0

The availability of documentation contributed to how much time it took the participants in this study to master the platforms. The study established that in CommCare, three of four participants said it took them no more than three months to be proficient using the platform, while four of seven in DHIS2 said the same. This is attributed to the presence of online training resources for CommCare and DHIS2. In the case of Malawi PoC-EMRS, five of eight participants indicated that it took the same time, while three of eight participants indicated that it took 7 - 12 months, thus establishing that CommCare and DHIS2 were easier to master. This was also observed during the participant observation exercise. The task took an average of 16.11 minutes to complete for all participants using CommCare and an average of 22.86 minutes for people

 J Health Inform Afr. 2023;10(2):1-9. DOI: 10.12856/JHIA-2023-v10-i2-383

using DHIS2. All participants were able to complete five of the nine form fields needed for the Malawi PoC-EMRS in an average of 44.6 minutes but they took an average of 82.15 minutes to complete the task.

4 Discussion and Conclusion

Malawi PoC-EMRS was compared with DHIS2 and CommCare architectures to determine the productivity of internal and third-party developers. Using concepts from the theory of generativity like generative technology and generative capacity, it was established that platform architecture affects developer productivity [21] [23] [28]. We also established that software platforms that provide incentives to developers enhance productivity enhance productivity and attract third-party contributions. By incentives, we mean platforms being configurable, flexible enough to be used for more than one use case, and cross-platform. This agrees with Msiska and Nielsen [19] that, in order to leverage a platform's capabilities, the platform needs to provide incentives to actors to use it. The Malawi PoC-EMRS can be used for cross-platform devices, but it lacks the configurability of DHIS2 and CommCare. Thus, the Malawi PoC-EMRS requires more effort by developers, in order to develop solutions for a diverse number of use cases. In this scenario, adaptability is expressed by how the system can be used for multiple business use cases therefore, the platform is not adaptable in this regard, thus concurring with other studies [11] [29] that platforms that allow customisations through the use of configurable templates attract more innovations and contributions, and that when the configurable templates fail, developers can find a workaround.

Software architecture design defines the availability of interfaces that allow internal and external software developers to interact with the platform's core functionality. This study used the Boundary Resource Model (BRM) to examine the three platforms' availability of boundary resources and how they are used. We established that a lack of third-party oriented incentives in the Malawi PoC-EMRS contributes to the absence of third-party contributors. These findings agree with Russpatrick [30] and Chirwa et al. [31] in that the availability of boundary resources alone is insufficient to attract third-party developers; external incentives are required. In addition, according to Msiska and Nielsen [19] third-party development is only possible when there is sufficient external generative capacity, regardless of how good a software ecosystem's boundary resources are. This was obvious because DHIS2 and CommCare have a greater number of third-party developers than the Malawi PoC-EMRS.

Extending from the theory of generativity and the usage of boundary resources, the construct of generative capacity has two perspectives: one that focuses on technology and another that focuses on actors. Documentation of a platform increases the generative capacity of an actor to be able to produce something, using the boundary resources of the platform, thus reducing the gap between technology and actor [32]. In this study, we established that the availability of documentation increases productivity, while its absence limits it and concurs with Duarte [4] where it was identified that effective training and the availability of quality documentation on a platform, increase the productivity of software developers. This is evident as DHIS2 and CommCare have considerable online documentation accessible to developers and third-party innovators, while Malawi PoC-EMRS has limited documentation and access to it is also limited [33], thus affecting productivity.

In this study, we identified three areas where platforms show their strength and increase the productivity of internal and third-party developers. Malawi PoC-EMRS can increase the productivity of internal and third-party developers by improving the availability of boundary resources through providing configurability and standardisation of its APIs, boundary resource use, and platform management. Improving the availability of boundary resources can be achieved by enhancing the Malawi PoC-EMRS to provide configurable interfaces to enable end-user configurability. This concurs with the assessment done by Munthali et al. [34] on BHT, which was the Malawi PoC-EMRS implementer, where five key areas of improvement were identified, one of which was to improve the productivity and quality of the platform to achieve 70% reconfigurability and 30% customizability. Secondly, Malawi PoC-EMRS needs to standardise and open its API endpoints in order to give third-party innovators the capacity and flexibility to improve or customise the platform [35]. In addition, there is a need to enhance the capacity of developers thus, improving boundary resource use [10] [30] [31]. This can be achieved when Malawi PoC-EMRS improves its documentation and provision of training materials, to internal and external developers. The absence of sufficient documentation is another factor that limits productivity, as highlighted by Duarte [4]. Lastly, the

Malawi PoC-EMRS needs to establish a Community of Practice (COP) to allow platform owners and contributors to share knowledge, and establish a governance structure. COPs play a significant role in platform management and governance as they are a major component of generativity in platform ecosystems [21] [30] [31] [36].

Acknowledgements

The authors would like to thank everyone who participated in this study.

Statement on conflicts of interest

The authors declare that there is no conflict of interest

References

[1] A. A. C. da Costa, S. Misra, and M. S. Soares, "A Systematic Mapping Study on Software Architectures Description Based on ISO/IEC/IEEE 42010:2011," in *Computational Science and Its Applications – ICCSA 2019*, S. Misra, O. Gervasi, B. Murgante, E. Stankova, V. Korkhov, C. Torre, A. M. A. C. Rocha, D. Taniar, B. O. Apduhan, and E. Tarantino, Eds., in Lecture Notes in Computer Science, vol. 11623. Cham: Springer International Publishing, 2019, pp. 17–30. doi: 10.1007/978-3-030-24308-1_2.

[2] S. M. A. Khan, *Popular Software Architecture Used In Software Development*. 2023.

[3] N. A. Ernst and M. P. Robillard, "A Study of Documentation for Software Architecture." arXiv, May 26, 2023. Accessed: Aug. 16, 2023. [Online]. Available: http://arxiv.org/abs/2305.17286

[4] C. H. C. Duarte, "Software Productivity in Practice: A Systematic Mapping Study," *Software*, vol. 1, no. 2, pp. 164–214, May 2022, doi: 10.3390/software1020008.

[5] M. Engert, M. Pfaff, and H. Krcmar, "Adoption of Software Platforms: Reviewing Influencing Factors and Outlining Future Research," 2019.

[6] Baoba Health Trust, "BHT 3-Tier Software Architecture." 2018.

[7] EGPAF, "Improving Quality of Care and Health Impact Through Innovative Systems and Technologies in Malawi (Health Information Systems (HIS)), CDC - EGPAF," *Elizabeth Glaser Pediatric AIDS Foundation*, 2022. https://pedaids.org/project/improving-quality-of-care-and-health-impact-through-innovative-systems-and-technologies-in-malawi-health-information-systems-his-cdc/ (accessed Jun. 02, 2022).

[8] A. Chhetri, M. Iversen, J. Kaasbøll, and C. Kanjo, "Evaluating mHealth Apps Using Affordances: Case of CommCare Versus DHIS2 Tracker," in *Information and Communication Technologies for Development. Strengthening Southern-Driven Cooperation as a Catalyst for ICT4D*, P. Nielsen and H. C. Kimaro, Eds., in IFIP Advances in Information and Communication Technology, vol. 551. Cham: Springer International Publishing, 2019, pp. 619–632. doi: 10.1007/978-3-030-18400-1_51.

[9] Ministry of Health of Malawi, "NATIONAL DIGITAL HEALTH STRATEGY." 2020.

[10] D. Ametowobla and S. Kirchner, "The organization of digital platforms," p. 26, 2022.

[11] L. F. Byson and T. D. Manda, "Software Architecture and Software Usability: A Comparison of Data Entry Form Designing Configurable Platforms.," p. 8, 2020.

[12] S. Himayat and Dr. J. Ahmad, "Software Understandability using Software Metrics: An Exhaustive Review," *SSRN Electron. J.*, 2023, doi: 10.2139/ssrn.4447189.

[13] "DATIM | OpenHIM." http://openhim.org/docs/implementations/datim/ (accessed Aug. 17, 2023).

[14] S. Dawadi, S. Shrestha, and R. A. Giri, "Mixed-Methods Research: A Discussion on its Types, Challenges, and Criticisms," *J. Pract. Stud. Educ.*, vol. 2, no. 2, pp. 25–36, Feb. 2021, doi: 10.46809/jpse.v2i2.20.

[15] D. Mertens, "PHILOSOPHICAL ASSUMPTIONS AND PROGRAM EVALUATION," 2015.

[16] D. Makwana, P. Engineer, A. Dabhi, and H. Chudasama, "Sampling Methods in Research A Review," 2023.

[17] J. Creswell, *Research Design*, 4th ed. SAGE Publications, 2014.

[18] D. M. Arya, J. L. C. Guo, and M. P. Robillard, "Information correspondence between types of documentation for APIs," *Empir. Softw. Eng.*, vol. 25, no. 5, pp. 4069–4096, Sep. 2020, doi: 10.1007/s10664-020-09857-0.

[19] B. Msiska and P. Nielsen, "Innovation in the fringes of software ecosystems: the role of socio-technical generativity," 2018.

[20] D. Fürstenau, A. Baiyere, K. Schewina, M. Schulte-Althoff, and H. Rothe, "Extended Generativity Theory on Digital Platforms," *Inf. Syst. Res.*, p. isre.2023.1209, Mar. 2023, doi: 10.1287/isre.2023.1209.

[21] L. D. W. Thomas and R. Tee, "Generativity: A systematic review and conceptual framework," *Int. J. Manag. Rev.*, vol. 24, no. 2, pp. 255–278, Apr. 2022, doi: 10.1111/ijmr.12277.

[22] N. Staub *et al.*, "Governance Mechanisms in Digital Platform Ecosystems: Addressing the Generativity-Control Tension," *Commun. Assoc. Inf. Syst.*, vol. 51, pp. 906–939, 2022, doi: 10.17705/1CAIS.05137.

[23] T. Wulfert, "Boundary resource management in innovation ecosystems: The case of e-commerce," *Electron. Mark.*, vol. 33, no. 1, p. 27, Dec. 2023, doi: 10.1007/s12525-023-00651-6.

[24] J. Caulfield, "How to Do Thematic Analysis | A Step-by-Step Guide & Examples," 2019.

[25] S. Zhou, B. Vasilescu, and C. Kästner, "What the fork: a study of inefficient and efficient forking practices in social coding," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Tallinn Estonia: ACM, Aug. 2019, pp. 350–361. doi: 10.1145/3338906.3338918.

[26] J. Businge, M. Openja, S. Nadi, and T. Berger, "Reuse and maintenance practices among divergent forks in three software ecosystems," *Empir. Softw. Eng.*, vol. 27, no. 2, p. 54, Mar. 2022, doi: 10.1007/s10664-021-10078-2.

[27] B. B. Chua and Y. Zhang, "Applying a Systematic Literature Review and Content Analysis Method to Analyse Open Source Developers' Forking Motivation Interpretation, Categories and Consequences," *Australas. J. Inf. Syst.*, vol. 24, Jun. 2020, doi: 10.3127/ajis.v24i0.1714.

[28] S. D. S. Amorim, J. D. McGregor, E. S. De Almeida, and C. V. F. G. Chavez, "Software Architectural Practices: Influences on the Open Source Ecosystem Health," *J. Softw. Eng. Res. Dev.*, Aug. 2023, doi: 10.5753/jserd.2023.967.

[29] M. Li, "Utilizing space for User Participation : Enabling and constraining factors for User Involvement when Designing a Data Entry Interface in a Generic Health Information Software in Uganda," University of Oslo, 2018.

[30] S. Russpatrick, "Understanding Platform Ecosystems for Development: Enabling Innovation in Digital Global Public Goods Software Platforms," in *The Future of Digital Work: The Challenge of Inequality*, R. K. Bandi, R. C. R., S. Klein, S. Madon, and E. Monteiro, Eds., in IFIP Advances in Information and Communication Technology, vol. 601. Cham: Springer International Publishing, 2020, pp. 148–162. doi: 10.1007/978-3-030-64697-4_12.

[31] E. Chirwa, P. Abbott, and J. Foster, "UNDERSTANDING THE INTERPLAY BETWEEN BOUNDARY RESOURCES AND GOVERNANCE PRACTICES IN INFLUENCING ECOSYSTEM VALUE CO-CREATION FOR DIGITAL PLATFORMS: A CASE FROM THE GLOBAL SOUTH," 2021.

[32] C. Sadowski and T. Zimmermann, Eds., *Rethinking Productivity in Software Engineering*. Berkeley, CA: Apress, 2019. doi: 10.1007/978-1-4842-4221-6.

[33] D. B. Nkhoma and K. Munthali, "Investigation of Key Factors Affecting Quality of Patient Data from National Antiretroviral Therapy Electronic Medical Record System in Malawi," 2021.

[34] A. Munthali, T. D. Manda, and K. Chikadza, "Strategic Plan for Baobab Health Trust (Malawi)." 2019.

[35] I. Miremadi, M. Khoshbash, and M. M. Saeedian, "Fostering generativity in platform ecosystems: How open innovation and complexity interact to influence platform adoption," 2023, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0048733323000653

[36] D. Paparova and M. Aanestad, "Governing Innovation in E-Health Platform Ecosystems – Key Concepts and Future Directions," 2020.