

Using Agile Practices to Address Defects in Medical Device Software Development: A Survey of Medical Device Software Organisations in Ireland

Misheck Nyirenda*

*Regulated Software Research
Centre*

*Dundalk Institute of Technology
Dundalk, Ireland*

misheck.nyirenda@dkit.ie

**Corresponding author*

Martin Mc Hugh

*Regulated Software Research
Centre*

*Dundalk Institute of Technology
Dundalk, Ireland*

martin.mchugh@dkit.ie

Róisín Loughran

*Regulated Software Research
Centre*

*Dundalk Institute of Technology
Dundalk, Ireland*

roisin.loughran@dkit.ie

Fergal McCaffery

*Regulated Software Research
Centre*

*Dundalk Institute of Technology
Dundalk, Ireland*

fergal.mccaffery@dkit.ie

Abstract—Agile Practices are widely used in the noncritical software development domain and are advocated to reduce defects. However, there is limited evidence in the existing literature about their exclusive implementation in Medical Device Software (MDS) development. Prior research has shown that agile practices are often integrated into traditional approaches for specific purposes, but there is little evidence in the existing literature about using agile practices specifically for addressing defects in MDS development. Through a survey of MDS organisations in Ireland, this study investigated the use of agile practices for addressing defects. The findings revealed that organisations implement agile practices for addressing defects, but prioritise identifying defects during coding and testing phases, primarily using code review, unit testing, integration testing, and coding standards. Defect identification usually occurs in the later part of the software development cycle. However, addressing defects late in the software development cycle can lead to costly rework and potential catastrophes when defects slip into the final product. We argue that in MDS development, both defect prevention and identification should be prioritised and implemented to manage defects more effectively and comprehensively. This approach could help to avoid costly rework and potential consequences arising due to defects that occur later. Agile practices can help in preventing defects early in the software development process and identifying those that slip to later stages, thereby ensuring overall software quality.

Keywords—*agile practices, medical device software, defects, software defects, defect prevention, defect identification*

I. INTRODUCTION

Medical Device Software (MDS) has become an integral part of modern healthcare systems and performs numerous functions such as diagnosing illnesses and regulating the amount of medication required by a patient [1]. Despite its growing importance and use, MDS is still released with defects. Defects in MDS have led to patient injuries and deaths and remain a major cause of medical devices recalls [2], [3]. It is important that defects are addressed early in MDS development to avoid these costly consequences.

Agile Practices are commonly used in the noncritical software development domain and are advocated to reduce defects [4]. However, studies reporting the exclusive implementation of an agile methodology in MDS development are scarce [2]. Literature shows that agile practices are often integrated into the Waterfall or V-Model in MDS development for specific purposes such as accommodating changing requirements [5]. Nonetheless, the literature provides scant evidence on the application of agile practices specifically for addressing defects in MDS development [2].

The aim of our ongoing research is to assist MDS organisations in using agile practices for preventing and identifying defects in MDS development. Through a literature review we identified agile practices that can be beneficial for defect prevention and identification in MDS development [2]. Defect prevention is the process of improving software quality and productivity by averting the introduction of defects into the software product [6], [7]. This process usually happens in the early stages of the software development life cycle, including requirements gathering and analysis, system design, architectural design, and module design [8]. While defect prevention is crucial in software development, it is often overlooked or improperly executed using ineffective methods [9]. Defect identification is the process of detecting defects in software modules with the goal of enhancing the quality of a software system [10]. This process occurs in the later stages of the software development life cycle, including coding, testing, release, and maintenance [8]. Identifying defects late in the software development life cycle can lead to costly rework and a higher likelihood of defects slipping into the final software product [11]. This can ultimately increase the risk of software failure, potentially causing catastrophic events [12].

This study investigated the use of agile practices for addressing defects in MDS development through a survey of MDS organisations in Ireland. The aim was to gain insight into which agile practices are used and how they are applied for addressing defects in MDS development. We received 15 responses from 15 different MDS organisations. The rest of the paper is structured as follows. Section II describes the survey process, Section III presents the survey results, Section IV

presents discussion, and Section V presents the conclusion and future work.

II. SURVEY OF MDS ORGANISATIONS IN IRELAND

We conducted a survey of MDS organisations in Ireland to investigate the use of agile practices specifically for addressing defects and to gain insights into their application. The survey targeted developers with experience in MDS development.

The MedTech sector in Ireland has experienced significant growth over the past three decades [13]. The current strategy of the Irish Medical Devices Association (2022-2025) reveals that there are over 450 MedTech organisations in Ireland [14]. However, it does not provide a detailed list of these organisations. In contrast, the 2016-2020 strategy categorises several MedTech organisations including 62 that develop MDS [15]. We used the 2016-2020 strategy document as a starting point in identifying MDS organisations that were contacted for the survey. We also invited MDS organisations with existing working relationships with the [Regulated Software Research Centre](#) to participate in the survey. Through these methods, a total of 65 email addresses were obtained from 65 different organisations. We only required one response from each MDS organisation. The survey link, the information leaflet, and consent form were then sent to 65 respondents from different organisations through email. We received responses from 15 different organisations, comprising 12 Small and Medium-sized Enterprises (SMEs) and 3 large organisations[16].

III. RESULTS OF THE SURVEY

The survey revealed varied experience in MDS development among the organisations ranging from 1 year to over 20 years. The survey also revealed that 60% of organisations, including 7 SMEs and 2 large organisations, use exclusively an agile methodology for MDS development. A small number, 7%, of SMEs use exclusively the V-Model, while another 7% of SMEs use both the V-Model and Waterfall. Lastly, 27% of the organisations, comprising 3 SMEs and 1 large, use a hybrid methodology which combines traditional and agile approaches. The reasons for organisations using their methodology were accommodating changing requirements (80%), addressing defects during development (67%), reducing time to market (60%), and both reducing development costs and meeting regulatory requirements (47%) each. However, two-thirds of the large organisations and a quarter of SMEs did not indicate addressing defects as one of the key aspects in their methodology implementation.

The survey also explored whether MDS organisations prioritise defect prevention or identification when developing MDS, and whether they use agile practices specifically for addressing defects. The results showed that 94% of the organisations that responded to the survey prioritise defect prevention over identification. However, when asked about the phases where they address defects, most organisations indicated Testing (93%), Coding (87%), Release (80%), and Maintenance (73%). This shows that while MDS organisations aim to be proactive in defect management, in practice more emphasis is on defect identification.

The data analysis also showed that 87% of the organisations that responded to the survey use agile practices for addressing

defects. Those using agile practices for addressing defects further indicated the various agile practices as shown in Table I. The Table shows that the commonly used agile practices are Code Review, Unit Testing, Integration Testing, and Coding Standards. These practices are typically applied after code has been written, except Coding Standards which is used during actual writing of code. As shown in Table II, of the organisations using agile practices for addressing defects, 85% implement them during coding and testing phases, indicating a priority for defect identification. This aligns with the earlier findings about the phases where organisations address defects and the agile practices they use.

TABLE I. AGILE PRACTICES USAGE FOR DEFECTS IN MDS DEVELOPMENT

Agile Practice	Responses on Agile Practices MDS Organisations use for Addressing Defects
	<i>Use for defects</i>
Code Review	85%
Unit Testing	77%
Integration Testing	77%
Coding Standards	69%
User Story	62%
User Acceptance Testing	54%
Refactoring	54%
Continuous Integration	54%
Test-Driven Development	46%
Small Releases	38%
Pair Programming	38%
Use Case	31%
Code Ownership	23%
Onsite Customer	23%
Behaviour-Driven Development	15%

When further asked about other agile practices useful for addressing defects, four respondents indicated additional techniques used in their organisations. One respondent from an SME indicated use of Good Automated Manufacturing Practice 5 (GAMP 5), a risk-based approach to compliant Good Practice (GxP) computerised systems [17]. Another respondent from an SME indicated Lean approach, stating, “*Pull the lever from anyone in the company to stop work when an issue is identified*”. The third respondent also from an SME indicated using Dogfooding [18], stating, “*Dogfooding and getting working software and devices into people’s hands as soon as possible. Get people using it, even if they are not the end customer/user*”. The fourth from a large organisation stated, “*We use most of the common useful ones*”, without mentioning any agile practice or technique. While these responses do not directly relate to agile practices, they indicate that organisations implement diverse techniques to address defects.

TABLE II. SOFTWARE DEVELOPMENT PHASES WHERE AGILE PRACTICES ARE USED FOR ADDRESSING DEFECTS IN MDS DEVELOPMENT

Software Development Phase	Responses about Phases where agile practices are used for addressing defects in MDS development
	<i>Agile practices usage</i>
Requirements	38%
System Design	38%
Architectural Design	38%
Module Design	46%
Coding	85%
Testing	85%
Release	38%
Maintenance	46%

When asked to provide comments about using agile practices to address defects, two respondents from SMEs that use agile practices for addressing defects provided contrasting perspectives regarding agile adoption in MDS development. One expressed concern that agile might lack the aspect of feedback from clients, especially those that the organisation has trusted relationship with. The respondent stated as follows, *“Agile misses aspects of feedback from key customers where you have trusted relationships to test new features/Alpha versions, etc and give feedback.”* The other respondent suggested that misconceptions that agile practices are not suitable for MDS development might be hindering their adoption and use for addressing defects, stating: *“I believe that a lack of understanding and alignment on what is meant by agile, particularly outside development teams, leads to a perception that agile is not a good fit for medical device software development. My view is that there are some core principles of agile practices that promote good software development and reduce the number of defects and help ensure the software is delivering on the user needs”*. These contrasting views may potentially explain the slow adoption of agile and the limited use of agile practices particularly for addressing defects in MDS development, as indicated in the existing literature [2].

Two respondents from SMEs that do not use agile practices for addressing defects provided comments on using agile practices for addressing defects. One supported using agile practices for MDS development but mentioned change notification requirement to regulators as a hindrance, commenting as follows: *“It is fine to create software using agile methods but once it becomes a medical device there is a disincentive in medical device software to changes owing to the significant change notification requirement to the notified body and registration with regulators. This overhead leads to doing larger releases which could be agile, but they tend towards waterfall or v models at that scale”*. The other respondent was more sceptical about using agile practices for defects, stating: *“I don't see agile as a method to addressing defects”*.

VI. DISCUSSION

This ongoing research aims to increase the understanding and application of agile practices to prevent and identify defects in MDS development. While the literature reveals that the Waterfall and V-Model are commonly used in MDS development, this study found that a majority (60%) of the organisations that responded to the survey use exclusively an agile methodology. This shows a significant increase from the previous survey which found that 25% of the organisations were using agile approaches for MDS development [19]. This shift towards agile approaches can be attributed to the extensive research in this area over the years that has identified and addressed the challenges of using agile practices for MDS development. However, it is noteworthy that some MDS organisations express scepticism towards both the adoption of agile practices and use of agile practices for addressing defects. This perspective aligns with prior research findings which suggest that MDS organisations do not typically use agile practices to address defects[2]. Interestingly, the same research also found a wide application of agile practices for addressing defects in non-MDS domain. The results in this study also show that despite extensive research addressing the challenges of adopting agile practices in MDS development [19], [20], some MDS organisations still view agile practices as not suitable for satisfying regulatory requirements.

The findings in this study indicate that a significant proportion of MDS organizations that use agile practices for addressing defects, primarily implement them for defect identification. As can be seen in Table II, most organisations carry out this process typically during the coding and testing phases of the development process. However, as discussed in Section I, addressing defects late in the software development process can lead to high costs. Defects that slip through the early stages of software development become more difficult to find later, leading to costly rework and increased risk of software failure. Identifying defects becomes significantly difficult once the software has been released. Conversely, preventing defects early in the software development process helps avoid these costly consequences. The survey findings indicate that while most MDS organisations recognize the significance of defect prevention, it is not extensively implemented in practice as one might expect. Therefore, it is recommended to prioritise and implement defect prevention strategies in MDS development. These strategies should be complemented with defect identification methods to catch any potential defect that might slip through the early stages. This provides a thorough method for defect management.

Previous research identified specific agile practices that can help to prevent and identify defects in MDS development. The survey presented here identified code review, unit testing, integration testing and coding standards as the most frequently used practices for addressing defects in MDS development. However, in non-MDS development domain agile practices such as pair programming, test-driven development, code review, unit testing, and refactoring are commonly used for addressing defects. Clearly, an observable difference exists in the agile practices commonly used for defects between these two domains. Moreover, the survey results reported here indicate that the MDS domain has not yet fully leveraged most agile

practices for addressing defects. This suggests that the MDS development domain could benefit from learning and implementing a broader range of agile practices. Further research is needed to help organisations understand and utilise agile practices to prevent and identify defects in MDS development.

V. CONCLUSION AND FUTURE WORK

This study aimed at gaining insight into the application of agile practices in MDS development organisations to address defects. Through a survey of MDS organisations in Ireland, we found that 87% of the organisations use agile practices for addressing defects. Interestingly, despite the stated priority of defect prevention, the practical focus leans towards defect identification. Most of the organisations employ code review, unit testing, integration testing, and coding standards commonly during coding and testing phases. Identifying defects late in the software development process can lead to costly consequences. On the other hand, preventing defects in the early stages of software development avoids these issues. Although MDS organisations aim to proactively address defects, defect prevention is often overlooked in practice. However, it should be prioritised and complemented with defect identification to address defects effectively and comprehensively. Despite the use of agile practices for addressing defects in MDS development, the focus is primarily on defect identification. Moreover, the MDS domain has not yet fully leveraged most agile practices for addressing defects. Therefore, MDS organisations could greatly benefit from learning and implementing a broader range of agile practices. As part of our future work, we plan to develop a framework that integrates agile practices suitable for preventing and identifying defects. The goal is to provide MDS development organisations with a comprehensive approach to defect management when developing software. This approach can help MDS development organisations reduce the risk of defects in the final software product, ultimately avoiding costly consequences.

ACKNOWLEDGMENT

This research is funded through the HEA Landscape and Technological University Transformation Fund, co-funded by Dundalk Institute of Technology.

REFERENCES

- [1] J. G. Ronquillo and D. M. Zuckerman, "Software-Related Recalls of Health Information Technology and Other Medical Devices: Implications for FDA Regulation of Digital Health: Recalls: Implications for FDA Regulation of Digital Health," *Milbank Q.*, vol. 95, no. 3, pp. 535–553, Sep. 2017, doi: 10.1111/1468-0009.12278.
- [2] M. Nyirenda, R. Loughran, M. McHugh, C. Nugent, and F. McCaffery, "Identifying Agile Practices to Reduce Defects in Medical Device Software Development," in *Systems, Software and Services Process Improvement*, M. Yilmaz, P. Clarke, A. Riel, and R. Messnarz, Eds., in Communications in Computer and Information Science. Cham: Springer Nature Switzerland, 2023, pp. 61–75. doi: 10.1007/978-3-031-42310-9_5.
- [3] N. G. Leveson, "The Therac-25: 30 Years Later," *Computer*, vol. 50, no. 11, pp. 8–11, Nov. 2017, doi: 10.1109/MC.2017.4041349.
- [4] R. Noor and M. Fahad Khan, "Defect Management in Agile Software Development," *Int. J. Mod. Educ. Comput. Sci.*, vol. 6, no. 3, pp. 55–60, Mar. 2014, doi: 10.5815/ijmecs.2014.03.07.
- [5] M. McHugh, O. Cawley, F. McCaffery, I. Richardson, and X. Wang, "An agile v-model for medical device software development to overcome the challenges with plan-driven software development lifecycles," in *Proceedings of the 5th International Workshop on Software Engineering in Health Care*, in SEHC '13. San Francisco, California: IEEE Press, May 2013, pp. 12–19.
- [6] M. A. Memon, M.-U.-R. M. Baloch, M. Memon, and S. H. A. Musavi, "Defects Prediction and Prevention Approaches for Quality Software Development," *Int. J. Adv. Comput. Sci. Appl. Ijacs*, vol. 9, no. 8, Art. no. 8, 49/01 2018, doi: 10.14569/IJACSA.2018.090857.
- [7] F. Huang and B. Liu, "Software defect prevention based on human error theories," *Chin. J. Aeronaut.*, vol. 30, no. 3, pp. 1054–1070, Jun. 2017, doi: 10.1016/j.cja.2017.03.005.
- [8] S. Ergasheva and A. Kruglov, "Software Development Life Cycle early phases and quality metrics: A Systematic Literature Review," *J. Phys. Conf. Ser.*, vol. 1694, no. 1, p. 012007, Dec. 2020, doi: 10.1088/1742-6596/1694/1/012007.
- [9] N. Jamal, M. Zulqarnain, M. W. Boota, S. Khan, and S. M. A. Akber, "Role of Defect Prevention Techniques Vs Defect Detection to Improve Software Quality: Critical Analysis Summary of Defect Preventive Approaches," vol. 2, no. 3, 2015.
- [10] A. N. Mahmoud and V. Santos, "Statistical Analysis for Revealing Defects in Software Projects: Systematic Literature Review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 11, 2021, doi: 10.14569/IJACSA.2021.0121128.
- [11] E. A. Felix and S. P. Lee, "Predicting the number of defects in a new software version," *PLoS ONE*, vol. 15, no. 3, p. e0229131, Mar. 2020, doi: 10.1371/journal.pone.0229131.
- [12] W. E. Wong, X. Li, and P. A. Laplante, "Be more familiar with our enemies and pave the way forward: A review of the roles bugs played in software failures," *J. Syst. Softw.*, vol. 133, pp. 68–94, Nov. 2017, doi: 10.1016/j.jss.2017.06.069.
- [13] D. McKernan and O. McDermott, "The Evolution of Ireland's Medical Device Cluster and Its Future Direction," *Sustainability*, vol. 14, no. 16, Art. no. 16, Jan. 2022, doi: 10.3390/su14160166.
- [14] Ibec, "Medtech Strategy 2025 - IBECE." Accessed: Nov. 27, 2023. [Online]. Available: <https://www.ibec.ie/connect-and-learn/industries/life-sciences-and-healthcare/medtech-strategy-2025>
- [15] Ibec, "Irish Medtech 2020 Strategy The Global Hub." Feb. 2016. [Online]. Available: <https://www.ibec.ie/-/media/documents/connect-and-learn/industries/life-sciences-and-healthcare/irish-medtech-association/irish-medtech-2020-strategy-the-global-hub.pdf>.
- [16] EU, "The new SME definition: User guide and model declaration." 2022. [Online]. Available: <https://www.eusmeccentre.org/cn/wp-content/uploads/2022/12/SME-Definition.pdf>
- [17] F. Pedro, F. Veiga, and F. Mascarenhas-Melo, "Impact of GAMP 5, data integrity and QbD on quality assurance in the pharmaceutical industry: How obvious is it?," *Drug Discov. Today*, vol. 28, no. 11, p. 103759, Nov. 2023, doi: 10.1016/j.drudis.2023.103759.
- [18] E. Silva, E. Tanaka, and G. Tordin, "Dogfooding: 'eating our own dog food' in a large global mobile industry player," in *Proceedings of the 14th International Conference on Global Software Engineering*, in ICGSE '19. Montreal, Quebec, Canada: IEEE Press, May 2019, pp. 52–57. doi: 10.1109/ICGSE.2019.000-6.
- [19] M. McHugh, F. McCaffery, and V. Casey, "Barriers to Adopting Agile Practices When Developing Medical Device Software," in *Software Process Improvement and Capability Determination*, vol. 290, A. Mas, A. Mesquida, T. Rout, R. V. O'Connor, and A. Dorling, Eds., in Communications in Computer and Information Science, vol. 290., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 141–147. doi: 10.1007/978-3-642-30439-2_13.
- [20] M. McHugh, F. McCaffery, and V. Casey, "Barriers to using Agile Software Development Practices within the Medical Device Industry," 2012.