Challenges experienced by Medical Device Software Development Organizations while following a Plan-Driven Software Development Life Cycle

Martin McHugh and Fergal McCaffery

Regulated Software Research Group, Department of Computing and Mathematics,
Dundalk Institute of Technology & Lero, Ireland
{martin.mchugh, fergal.mccaffery}@dkit.ie

Abstract

Medical device software organisations face challenges not faced by generic software development organisations. These challenges include the adherence to regulatory controls. Regulatory bodies require medical device software organisations to provide objective evidence that the software they are developing is safe and reliable. To produce this, regulatory bodies require a number of deliverables which must be achieved. However, they do not dictate which Software Development Life Cycle (SDLC) must be followed in order to achieve these deliverables. Despite not dictating which SDLC must be followed when developing medical device software, organisations typically develop their software in accordance with a Plan-Driven software development lifecycle. By conducting semi structured interviews with seven medical device software organisations, we gained a deeper insight into how the challenges experienced impact on the development of medical device software. The interviews also attempted to learn from the participants how they believe the challenges experienced can be overcome. The aim of this paper is to explain the methodology used to perform interviews with medical device software organisations and to present these interviews.

Keywords

Medical Device, FDA, Agile, V-Model, Software Development Life Cycle, Semi-Structured Interview

1 Introduction

Medical device software organisations experience difficulties not experienced by non-regulated software development organisations. Anecdotal evidence suggests the biggest challenge experienced by medical device software organisations is regulatory controls. Medical device software, regardless of the time or money spent on the development of the software, can be deemed useless if it fails to achieve regulatory approval. Medical device software organisations therefore may be reluctant to adopt new techniques to improve efficiencies, fearing that it may hinder their chances of achieving

regulatory approval. Ongoing research will present practices to medical device organisations which aim to achieve efficiencies without hindering the process of securing regulatory approval.

To learn the areas in which difficulties are experienced semi structured interviews were conducted with medical device software organisations. These interviews gave an insight into the real differences between developing medical device software and generic software and also the challenges faced by these organisations. Once this understanding was gained, appropriate recommendations can be made as to how these challenges may be overcome and efficiencies can be successfully introduced.

The remainder of this paper is structured as follows, Section 2 discusses the methodology used to create the interviews, Section 3 presents the participants of the interviews, Section 4 outlines the findings of the interviews, Section 5 discusses the recommendations and in Section 6 the conclusions are presented.

2 Research Methodology

The interviews were conducted in accordance with Wengraf [1]. The interviews were performed on a semi structured basis. This form of interview is known as a Semi Structured Depth Interview (SSDI). SSDIs are characterized by the following features:

- *"The interview is a research interview, designed for the purpose of improving knowledge.*
- *It is a special type of conversational interaction: in some ways it is like other conversations, but it has special features which need to be understood.*
- *It has to be planned and prepared for like other forms of research activity but what is planned is a deliberate half-scripted or quarter-scripted interview: its questions are only partially prepared in advance and will therefore be largely improvised by you as an interviewer. But only largely: the interview as a whole is a joint production, a co-production, by you and your interviewee.*
- *It is to go into matters 'in depth'".* [1]

SSDIs are further categorized into two classifications, Heavily Structured Depth Interviews and Lightly Structured Depth Interviews. The degree of structuring is determined by the degree to which the questions and interventions are pre-prepared by the researcher. Figure 1 shows the relationship between structured and unstructured interviews.

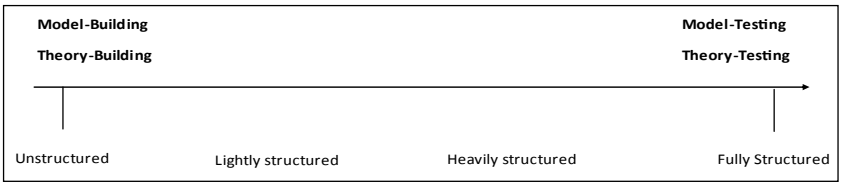

Figure 1 Spectrum from Unstructured to Fully Structured Interviewing, and Possible Relationship to Phases in the Development of a Theory [1]

2.1 Pyramid Model

In accordance with Wengraf, the interview was broken into four elements. These elements are:

- Research Purposes (RP);
- Central Research Question(s) (CRQ);
- Theory Questions (TQ);
- Interview Interventions (II) / Interview Questions (IQ).

The RP is the motivation behind the research being conducted. For this research, the RP is to gain a deeper insight into difficulties experienced when developing medical device software. The CRQ is the primary question(s) to which answers are being sought as a result of the interview being conducted. The TQ are high level questions. These questions are not asked directly to the interview participant. TQ are used to formulate the actual questions that will be asked of the participant. II/IQ is what is actually asked of the participant during the interview. The information gleaned from the responses is compiled to answer the TQ which in turn answer the CRQ which ultimately supports the RP. The relationship between each of these elements is shown in the pyramid model shown in figure 2.

In Dillon [3], the author discusses the various types of questions and non-questions in interview scenarios. He describes that interventions[1] in an interview can be more beneficial than pre-prepared questions. As a result they are included in the pyramid model.

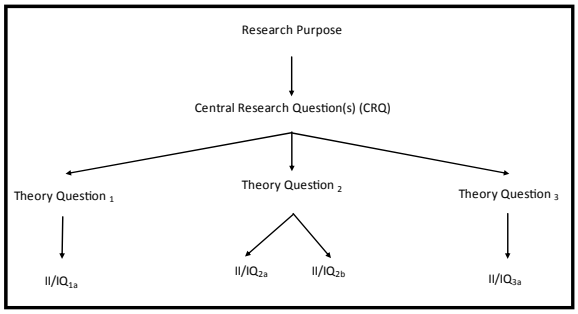

Figure 2 CRQ > TQ > IQ/II: Pyramid Model [1]

3 Interview Participants

A difficulty often associated with SSDI's is the process of achieving representative sampling. Patton [4] presents several different sampling methods used as part of conducting SSDI's each valid to specific methods of research. This is known as Patton's Typology of Randomised and Purposive Sampling [4]. Based upon Patton's Typology of Randomised and Purposive Sampling the most appropriate method of sampling for this research is "Purposeful Sampling" employing "Maximum Variation Sampling". Using this approach, organisations of varying structure and size each of which representing a sector of the medical device software development industry were identified.

Seven medical device software organisations participated in the interviews. Within each organisation, employees best placed to answer questions relating to the development of medical device software were interviewed. The positions which these employees hold within the different organisations varied. Below is the list of roles which the participants perform:

- Chief Technical Officer;
- Head of Development;
- Quality Manager;
- Co-Founder/Director;
- Senior Research and Development;
- Principal Engineer;
- Chief Executive Officer;
- Electronic Design Engineer.

In accordance with Wengraf the following questions i.e. IQ, were established prior to the interviews being conducted. The following tables show how each of the IQ relates to a TQ and in turn how the TQ contributes to answering the CRQ. The CRQ for this research is "*How does the development of medical device software differ to the development of non-regulated software?"*

[1] Interview Interventions are described as questions or statements made during the interview to elicit responses that are not prepared prior to the interview.

Table 1 Research Questions

CRQ	RQ1 What are the issues with developing medical device software?	TQ1	IQ1a.
			IQ1b.
	RQ2 What are the issues with developing medical device software using a traditional software development lifecycle?	TQ2	IQ2a.
			IQ2b.
		TQ3	IQ3a.
			IQ3b.
		TQ4	IQ4a.
			IQ4b.

Table 2 Theory Questions

CRQ	RQ1	TQ1 Does regulatory conformance directly impact the development of medical software?	IQ1a.
			IQ1b.
	RQ2	TQ2 Are you following a Plan-Driven software development lifecycle and if so why?	IQ2a.
			IQ2b.
		TQ3 Does following a Plan-Driven software development lifecycle meet all of the organisation and regulatory requirement?	IQ3a.
			IQ3b.
		TQ4 Would a tailored lifecycle be more appropriate than moving to a different software development lifecycle completely?	IQ4a.
			IQ4b.

Table 3 Interview Questions

CRQ	RQ1	TQ1	IQ1a. How do you believe the development of medical device software differs to that of the development of generic software?
			IQ1b. How do these differences impact on the development of medical device software?
	RQ2	TQ2	IQ2a. For your current software development project which software development lifecycle are you following?
			IQ2b. Why are you following this lifecycle?
		TQ3	IQ3a. What difficulties are you experiencing as part of your current software development project?
			IQ3b. Why do you believe you are having these problems?
		TQ4	IQ4a. Do you believe there is a way to overcome these problems?
			IQ4b. What do you believe are the barriers to moving away from your current software development lifecycle?

4 Interview Findings

The interviews conducted yielded rich qualitative data. However, a method was needed to extract the findings from this qualitative data. The results of the interviews were analyzed in accordance with Wengraf's, Interview Material to Answers to Theory Questions to an Answer to the Central Research Question (IM-ATQ-ACRQ) model [1]. Whilst the CRQ > TQ > IQ/II model utilizes a top down approach, the IM-ATQ-ACRQ model utilizes a bottom up approach to determine the answer to the central research question. This method was used as it complimented the method employed for the creation of the interview questions i.e. RP > CRQ > TQ > IQ/II. The results were also analysed in accordance with Miles and Huberman's [2] method of analyzing qualitative data.

4.1 Qualitative Data Analysis

Miles and Huberman [2] present three stages for qualitative data analysis. The three stages are:

- Data Reduction;
- Data Display;
- Conclusion Drawing and Verification.

The relationship between these stages of qualitative data analysis and data collection is shown in Figure 2.

4.1.1 Data Reduction

SSDIs produce a large amount of data. To navigate all of the data collected the volume must be reduced. Data reduction is a continuous process happening before the data is collected. Before data collection occurs “Anticipatory Reduction” takes place. The interviewer attempts to pre-empt the information being collected and selects questions in an attempt to reduce unnecessary information from being collected.

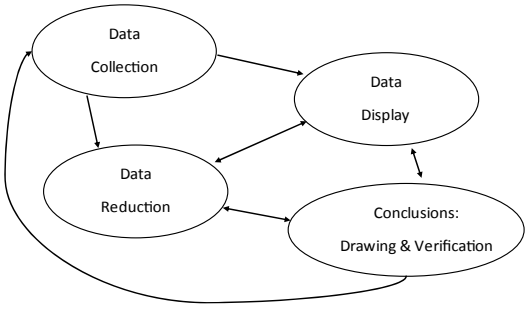

Figure 2 Components of Data Analysis: Interactive Model [2, p.12]

4.1.2 Data Display

As SSDIs can produce large amounts of raw data, in this case over 30 pages of interview transcripts, a method was required in which this data can be displayed in a form easily understood by a person. To achieve this matrices and graphs were employed. A key element of data display is that data display is not separate from data analysis. In fact, it is part of overall process of data analysis [2].

4.1.3 Conclusions Drawing and Verification

The process of drawing conclusions involves examining the collected data and analyzing the implications this data has on the research being conducted. Miles and Huberman [2] discuss that, whilst a final conclusion is created once all of the collected data has been analyzed, conclusions appear very early on in the data collection process and that whilst the conclusions may appear to be established inductively, external influences can have an impact on the development of early conclusions. Conclusions can be derived from analyzing data once. These conclusions are verified by analyzing the data multiple times. The conclusions are deemed to be verified once the results after each analysis are the same.

4.2 Results

After each of the three stages of qualitative data analysis were performed as outlined in the previous section, the results were produced (See table 4). The number sequence after each response correlates to a specific interview and at which point in the interview the response was given.

Table 4 Responses Received during SSDIs

Central Research Question								
RESPONSES RECEIVED	RQ1		RQ2					
	TQ1		TQ2		TQ3		TQ4	
	IQ1a	IQ1b	IQ2a	IQ2b	IQ3a	IQ3b	IQ4a	IQ4b
	There is a level of authorisation surrounding the developing of medical device software which adds a certain amount of complexity [6,015].Production of clinical evidence, being ISO 13485 compliant and following IEC 62304, production of documentation [5,008,012] [1,008] [3,004] [2,002] [7,0034]Risk [4, 002] [1,008] [3,004] Safety [1,008]Prescribed set of deliverables before you start [3,004] Usability [3,004] Process is more strict [2,002]	If you have the right processes in place and you built it into the company it wouldn't take as long as it currently does for us [5,016] The cost associated with making sure the traceability is current all of the way through [4,004] Increased processes infrastructure required and you need to have a disciplined approach [1,020] Large effort put into testing, integration and validation [3,006]Increased time to market, improved quality more expensive to produce [2,006, 008] [7,008]	Waterfall Model [6,021] [1,024] V-Model [5,022] [4,006] [3,014] [2,012] [7,016]	Residue from some of the activities we performed in the automotive industry [4,009] The majority of our customers have asked us to do it [1,028] Auditors are familiar with the V-Model, document outputs are in line with guidance and regulations [2,014] [7,018]	Requirements Changes [6, 031] Validation is probably something and making sure test cases are linked back to the requirements [4,011] Interdependency between stages and impact on other areas [2,015] Requirements or design, retrofitting or filling gaps [1,034] Traceability of requirements [3,0028] [7,022]	We don't have a strong enough structured approach and we cannot formalise the requirements capture [4,013] It wasn't specified up-front [1,040]	Work better up front on capturing requirements precisely [1,042] Modifying the quality management system [3,032] Improve requirements management up front [2,018] combine life cycles i.e. agile and Plan-Driven [7,026]	No barriers at present [6,037] We chose the V-model as it best reflects our processes but we could move away from it [5, 036] Yes, our clients would struggle to understand and it would be difficult to quote on an agile project [4,15] I think the barriers would come from key stakeholders [1,046] We wouldn't have a problem making a change and introducing a new system to get over what difficulties we come across [3,032] Increase cost in retraining staff and redefining new processes [2,020] [7,028]

5 Discussion

Based upon the findings shown in table 4, it can be seen that the organisations involved identified a number of the same problems and challenges as shown by the multiple number sequences after specific responses. The two most cited difficulties are regulatory constraints and managing requirements changes.

5.1 Regulatory Constraints

Software developed for use, in or as a medical device must adhere to strict regulatory controls. These controls are put in place to ensure the safe and reliable functioning of the software. Medical device software organisations must provide objective evidence to regulatory bodies that their device is safe. This evidence is achieved through the production of comprehensive documentation. The production of this documentation can become burdensome for software organisations. One of the interview participants noted that whilst the production of documentation and adherence to regulations can be burdensome, it can also be beneficial. The burden of adherence can act as a barrier to entry into the medical device software development industry potentially reducing the amount of competition within the medical device industry.

To accompany the requirements to produce adequate documentation, medical device manufacturers are advised to adhere to a quality management standard such as ISO 13485 [5] when developing medical device software. Whilst in Europe, it is not mandatory to follow this standard, should a device manufacturer choose not to follow this standard, they must prove to the regulatory bodies that the method which they used to ensure the quality of their device is equally comparable to ISO 13485.

The interviews showed that medical device software organisations typically follow a Plan-Driven Software Development Life Cycle (SDLC) such as the V-Model. Plan-Driven SDLCs produce the necessary deliverables required when seeking regulatory conformance; however, Plan-Driven SDLCs are not seen as efficient [6] and can be difficult to apply to a medical device software development project in a practice. Section 5.2 discusses how these Plan-Driven SDLCs can be modified to become more efficient whilst still producing the necessary regulatory deliverables.

5.2 Managing Requirements

Previous research [7, 8] and the interviews conducted as part of this paper has shown that medical device software is typically developed in accordance with a Plan-Driven SDLC. Plan-Driven SDLCs such as the Waterfall and V-Model are typically performed in a sequential manner with very little scope for revisiting stages. Plan-Driven SDLCs dictate that requirements are gathered up-front prior to any development beginning. However, a medical device software development project can potentially take a number of years to be completed and it can be very difficult to ensure that there will be no change in requirements throughout this period.

Each of the organisations involved in the interviews identified that a major problem they experience is accommodating changes once development has begun. To accommodate changes a number of stages may need to be revisited, having a knock-on effect of increasing rework and therefore increasing cost. When asked in the interviews how to resolve the problems associated with changing requirements a number of responses were given. One organisation suggested the establishment of an incubation period prior to the requirements analysis stage. This incubation period would allow the customer time to consider all potential features they wished to include in the software and ideally removing the need for a change to be implemented once the project has begun. Another organisation suggested placing greater emphasis on up-front planning and again making sure all of the necessary requirements were captured. One organisation suggested “placing manners on the customer” and preventing them from introducing a change once development has begun.

Each of these suggestions has their own merit, however these are proactive steps, none of the organizations were able to suggest a reactive response to when a requirements change was unavoidable. Current Plan-Driven SDLCs are rigid and therefore have difficulty accommodating a change.

Typically, when a change is introduced, a number of stages of development need to be revisited to accommodate the change. This can require a lot of rework therefore increasing cost and development time. Agile practices and methodologies promote the ability to be able to accommodate changes. The agile manifesto states *"welcoming changing requirements, even late in development"* [9]. This would suggest that utilising agile practices in the development of software could offer the *"silver bullet"* to problems associated with late changes in requirements.

However, research [10-12] has shown that it is very difficult to fully adopt a single agile methodology such as Scrum or XP, as no single agile methodology produces the necessary deliverables required when seeking regulatory approval. To overcome this, research suggests that combing agile practices with a Plan-Driven SDLC can reap the most significant rewards as the organisation would still benefit from the structure associated with following a Plan-Driven approach whilst also gaining the efficiencies associated with utilising agile practices.

6 Conclusions

Medical device software organisations face challenges not faced by non-regulated software development organisations. We conducted interviews with seven medical device software organisations to gain a deeper insight into these challenges. We selected organisations of varying size, structure and criticality to act as a broader representation of the medical device software industry as a whole. These organisations included medical device manufacturers, software suppliers to medical device manufacturers and organisations providing design services to medical device software suppliers and manufacturers. Whilst these organisations ranged in maturity, size and software criticality, the challenges experiences by each of them are very similar. The biggest challenge identified is the adherence to regulatory controls. This adherence brings with it the overhead associated with producing large amounts of documentation. It also brings with it the perception that following a Plan-Driven SDLC is required in order produce the necessary deliverables required when seeking regulatory approval[2].

The CRQ of this research is to determine the differences between the development of medical device software and the development of non-regulated software. The interviews revealed that the key difference is the need to adhere to regulatory controls. Regulatory controls appear to restrict medical device software organisations to follow a sequential plan driven SDLC. However, following a plan-driven SDLC can introduce problems such as having difficulties introducing requirements changes.

As medical device software is typically developed in accordance with a Plan-Driven SDLC, medical device software organisations experience the inherent problems associated with following this type of lifecycle. The most identified challenge by the participants of the interviews associated with following a Plan-Driven SDLC is accommodating requirements changes. As Plan-Driven SDLC are completed in a sequential manner, if a stage is completed and development has moved on it can be very difficult to revisit a stage such as *"Requirements Management*".

To overcome this challenge, medical device software organisations are advised to move to a SDLC which can better accommodate requirements changes. Agile methodologies boast the ability to welcome changes throughout the development lifecycle. However, research has shown that it can be very difficult to fully move away from a Plan-Driven SDLC to an agile methodology as no single agile methodology produces the necessary regulatory deliverables. Based on this, future work as part of this research will involve developing and validating a hybrid SDLC which combines a Plan-Driven SDLC with agile practices to introduce efficiencies and to overcome the difficulties associated with requirements management in medical device software development projects.

Acknowledgments

This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero - the Irish Software Engineering Research Centre (*http://www.lero.ie*) grant 10/CE/I1855.

[2] Whilst it is perceived that medical device software regulations and standards require medical device software to be developed in accordance with a Plan-Driven SDLC there is no direct instruction with the regulations or standards dictating the use of a specific SDLC

7 References

[1] T. Wengraf, *Qualitative Research Interviewing*. London: Sage Publications, 2001.

[2] M. Miles and A. M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*. London: Sage, 1994.

[3] J. Dillon, *The Practice of Questioning*. London: Routledge, 1990.

[4] M. Patton, *Qualitative Evaluation and Research Methods (2nd Edn)*. Newbury Park: Sage, 1990.

[5] ISO, "*ISO/IEC 13485:2003 Medical devices -- Quality management systems -- Requirements for regulatory purposes*," ed. International Organisation for Standards - Geneva, Switzerland, 2003.

[6] J. Cadle and D. Yeates, *Project Management for Information Systems*: Pearson Education, 2008.

[7] F. McCaffery, D. McFall, P. Donnelly, and F. G. Wilkie, "Risk Management Process Improvement for the medical device industry," presented at the Conference on Software Development (SWDC-REK-2005) Iceland, 2005.

[8] M. McHugh, F. McCaffery, and V. Casey, "Barriers to Adopting Agile Practices when Developing Medical Device Software," presented at the The 12th International SPICE Conference Process Improvement and Capability dEtermination, Palma, Majorca, 2012.

[9] R. C. Martin, *Agile Software Development - Principles, Patterns and Practices*: Prentice Hall, 2003.

[10] R. Rasmussen, T. Hughes, J. R. Jenks, and J. Skach, "Adopting Agile in an FDA Regulated Environment," presented at the Agile Conference, 2009. AGILE '09. , Chicago, IL 2009.

[11] P. A. Rottier and V. Rodrigues, "Agile Development in a Medical Device Company," presented at the Proceedings of the 11th AGILE Conference. AGILE '08., Girona, Spain, 2008.

[12] H. Mehrfard and A. Hamou-Lhadj, "The Impact of Regulatory Compliance on Agile Software Processes with a Focus on the FDA Guidelines for Medical Device Software," *International Journal of Information System Modeling and Design,* vol. 2, pp. 67-81, 2011.

8 Author CV's

Martin Mc Hugh

Martin received his B.Sc. (Hons.) in Information Technology Management in 2005 and M.Sc. in Computer Science in 2009, from Dundalk Institute of Technology. He is now undertaking research for his Ph.D. in the area of software process improvement for medical devices with emphasis on the usage of agile practices when developing medical device software, as part of the Regulated Software Research Group in Dundalk Institute of Technology.

Fergal Mc Caffery

Dr Fergal Mc Caffery is the leader of the Regulated Software Research Centre in Dundalk Institute of Technology and a member of Lero. He has been awarded Science Foundation Ireland funding through the Stokes Lectureship, Principal Investigator and CSET funding Programmes to research the area of software process improvement for the medical device domain. Additionally, he has received EU FP7 and Enterprise Ireland Commercialisation research funding to improve the effectiveness of embedded software development environments for the medical device industry.