2018 International Conference on Multidisciplinary Research
Track 1: STEM (Science, Technology, Engineering & Mathematics)
https://doi.org/10.26803/MyRes.2018.02

Implementation of Kanban Practices with a Medical Device Software Development Lifecycle

Juergen Wieland
University of Gloucestershire, School of business and management
Cheltenham GL50 2RH, United Kingdom

Abstract
Objective. The objective of this study is to provide a better understanding of the challenges faced by software developers when developing safety-critical medical device software by the use of a plan-driven SDLC and to resolve these shortcomings by the proposition of a mixed method SDLC (KV-model) to the development of software as medical device.

Methods. The proposed mixed method SDLC was developed on the foundation of a formal development process (v-model) and adds the benefits of lean-Kanban practices. The proposed KV-model includes the principles and recommendations of IEC 60601-1 (v-model) to accomplish regulatory requirements for the development of safe and reliable high quality medical software according to IEC 62344. To provide a better understanding of the challenges faced by software developers, a survey of a sample of 18 German medical software developers was performed. Surveying at least two members of each company, five central development stages with ten Kanban practices were identified. The lean-Kanban practices were then assessed for consistency using IEC 62304 and unified to a hybrid approach.

Results. The following results were discovered: 1.) 72 % of the surveyed companies use v-modelling as SDLC; 2.) 28 % of the survey companies already implement lean-Kanban practices within their SDLC; 3) challenges faced by software developers can be solved by integrating lean-Kanban practices as an iterative and continuously improving process within a plan-driven SDLC 4.) identification of ten lean-Kanban practices to be implemented within a plan-driven SDLC; 5.) the development of a hybrid SDLC combining lean Kanban practices within a plan-driven v-model (KV-model).

Keywords
Kanban practices, ISO 62304, software development lifecycle (SDLC), software as medical device (SaMD).

Introduction

Lean management practices have become highly prevalent since the lean-Kanban manifesto emerged in early 2011 as a reaction to the ineffectiveness of current development methods in a fast-changing software development environment. Kanban software development practices apparently solve the problems of a plan-driven development lifecycle. The Kanban approach is claimed to be "the next wave of software process"1, offering improved software quality at lower development costs and saving resources. Regardless of the shift from plan software development methods to Kanban software development, only a low rate of publicly available information is available which suggest a widespread implementation of Kanban practices for the development of software as medical device. Using Kanban methods, the software developer is directed in a way that enables to resolve programming errors in minimum time frame possible, but on the other hand ensures the use of standardized practices when developing quality- and safety-critical medical device software. It is apparent that medical device software development projects might benefit from using Kanban practices, whilst still keeping the discipline associated with the following plan-driven development lifecycle (Pekar et al., 2016). The purposes of this paper, consequently, are:

- To provide a better understanding of the challenges faced by software developers when developing safety-critical medical device software by the use of a plan-driven SDLC.
- And to show how shortcomings in both Kanban and plan-driven approaches are resolved by the use of a mixed method approach to the development of software as medical device.

To answer these research objectives, a survey with 18 Germany-based manufacturers of medical software has been performed. As a study result, this paper proposes a hybrid SDLC combining both lean-Kanban and plan-driven practices to develop regulatory-compliant medical software. In examining current software development practices, the v-model appears to be the SDLC most widely used by medical device software organisations (Polgár & Kazinci, 2014). This study identifies Kanban practices to be implemented into the v-model and proposes the Kanban v-model (KV-model) as a SDLC for use in the medical device industry. The paper aims to provide developers of medical software with the structure of a plan-driven SDLC whilst reaping the benefits available through the use of Kanban practices.

International regulation

Developers of medical device software must comply with region-specific quality and regulatory requirements, depending on the country where the software as medical device is intended to be marketed. A study carried out by Markets & Markets (2016) see USA and Europe as the major markets for medical device software until the year 2022. Accordingly, medical device software must comply with FDA requirements for marketization within the USA and with the medical device directive (MDR) and the in-vitro diagnostic directive (IVDR) for marketization in Europe.

[1] http://atlanta2010.leanssc.org/

USA

Software as medical device intended to be marketed in the United States of America must comply with regulations specified by the FDA. The software developer must ensure that the software has been designed, developed, coded, validated and verified consistent with these regulations in order to be both safe and reliable according to its intended use. The software developer can use guidance documents issued by the FDA, helping in the interpretation of these regulations:

- General Principles of Software Validation (GPSV) – Final Guidance for Industry and FDA Staff
- Guidance for OTS Software Use in Medical Devices – FDA
- Cybersecurity for Networked Medical Devices Containing Off-the-Shelf (OTS) Software – Guidance for Industry
- Mobile Medical Applications – Guidance for Industry and Food and Drug Administration Staff
- Postmarket Management of Cybersecurity in Medical Devices – Guidance for Industry
- Software as a Medical Device (SAMD): Clinical Evaluation – Guidance for Industry
- Medical Device Data Systems – Guidance for Industry

The FDA's general principles of software validation enforce the software developer to document the requirements, based on the intended use, prior to implementation and testing. Documentation of requirements appears to be a challenge when introducing Kanban practices into the software development lifecycle, as one of the main principles of the lean-Kanban manifesto is "no documentation, ever, which can't make sense". The "guidance for content of premarket submission for software contained in medical devices" forces the manufacturer to use a plan-driven software development lifecycle. This appears to be another challenge for the adoption of Kanban practices within the software development process, clearly favouring the implementation of requirements by the use of a fluid set of practices and in small iterative steps through intuitive abilities. Unclearly defined requirements before initiating a software development process are difficult to trace, as required by the FDA. Furthermore, the software developer must analyse the potential risk of each defined requirement for criticality and probability, placing the intended user at risk. Pekar et al. (2016) emphasise that a unified risk management approach, as required by the FDA, represents another challenge in the adoption of Kanban practices. Guidance on the implementation of risk management activities is almost absent in current development methods based on Kanban practices.

Another hindrance in the adoption of Kanban practices is that developers may perceive software development based on Kanban practices to be of lower quality compared to a plan-driven software development lifecycle (Özcan-Top & McCaffery, 2017). The Kanban manifesto clearly underlines quality as the first principle to be followed, with a clear focus on "quality systems […] and principles that guide the understanding of practices in which we [the software developer] can trust". Nevertheless, the linkage to quality systems when using Kanban practices with a

medical device software development lifecycle is almost absent in academic literature.

Europe

The European parliament amended the existing medical- and in-vitro diagnostic device regulation in 2017 with a transition period of three and five years, respectively. Both directives, the MDR and the IVDR, are mandatory for software as medical device to be marketed within the European Union (Clarke et al., 2017; EU, 2017). These regulations introduce further rules, including specific requirements addressing the classification of software. Accordingly, a notified body certifies conformance with the MDR and the IVDR and then the software as medical device can be marketed within the European Union. For software as medical device, the latest amendment to the MDR and the IVDR introduced several changes, including:

- specific requirements addressing software classification into diagnostic, therapeutic and physiological devices
- the classification of software as active therapeutic medical device, prior classified solely as active medical devices
- identification of eHealth and mHealth software as mobile medical application. Once a mobile medical application is classified as a medical device, the MDR requirements must be considered.

With these changes, the European Parliament placed great emphasis on the international standards to be followed when developing software as medical device. For Europe, the programmer is recommended to develop the software according to IEC 62304 and IEC 60601-1 and its aligned guidance documents. The standard for IEC 62304:2006 Medical device software – Software life-cycle processes emphasises the management of software development to be realised by one independent person having the overall responsibility. Kanban practices in contrast favour the self-organisation of development teams. Software development through self-organising teams hinders a plan lifecycle, removes centrality and decision-making powers from one independent person. This appears to be a challenge, resulting in a loss of management control. Further organisational support might be necessary for Kanban practices to successfully develop medical device software (McHugh, McCaffery, & Casey, 2014).

Lean-Kanban software development

The terms “lean” and “lean-Kanban” are poorly defined in literature on software development (Anand, Chandrashekar, & Narayanamurthy, 2014). The authors Wang, Conboy, and Cawley (2012, p. 3) emphasise that the terms are often “poorly considered, multi-dimensional, ambiguous and inconsistent”. The first step in developing a mixed method approach to the development of software as medical device is a clear concept and explicit corresponding Kanban practices. Corresponding to established lean-Kanban concepts and practices, this paper considers that the development of medical software also involves two main elements: the basic lean-Kanban concept with its corresponding practices:

Lean-Kanban concept

The Kanban concept is not a term uniquely used in software development. The roots of the lean-Kanban concept can be traced back to other disciplines, with most literature dating the origin back to the 1940s and the production system implemented by Toyota (Müller, Tolujew, & Kienzle, 2014). The lean-Kanban practice itself has not been well-known in academic literature since a study carried out by the Massachusetts Institute of Technology (MIT), researching huge differences in quality and productivity between car manufacturers in the USA and Japan. A study carried out by Sugimori, Kusunoki, Cho, and Uchikawa (1977), former Vice Presidents of Toyota Motor Company, developing a guidance system for implementing a continuous improvement system for high quality products in the right quantity, with developers displaying their full capabilities through active participation in improving their own work. The lean-Kanban concept is a development method enabling a company to programme high quality software, create knowledge, defer commitment, deliver fast, respect people, optimize the process, and only create the documentation needed (Poppendieck & Poppendieck, 2007). The lean-Kanban concept interlinks five inherent values underling the Kanban approach:

- Quality systems: Implementing principles that guide the understanding of development practices which the developer trusts;
- Value stream: Mapping the steps of the development process identifying the value it adds;
- Flow: Implementing a continuously flowing development process;
- Perfection: Striving for perfection in the development process by identifying and removing waste in a continuous way;
- Small steps: Implementing changes through small steps driven by continuous improvement.

Lean-Kanban practices

The primary focus and overall practice of lean-Kanban is the identification and implementation of what to develop, when to develop, and in which way to develop (Münch, 2016) with respect to product quality and customer value. Kanban practice is Japanese for "visual sign" and, through a visual framework, classifies work into categories: "work to do", "work being done", "work done". By mapping the SDLC into manageable chunks, "quality adding" and "risk eliminating" activities can be identified and prioritised accordingly. Another main focus of lean-Kanban is implementation of practices in small, iterative steps into a current SDLC. Münch (2016) emphasises that the implementation of Kanban practices can be done by adding "Kanban on top of other existing workflows", such as a plan-driven SDLC.

The contemporary understanding of lean-Kanban development is primarily driven by practitioners' writings (Cawley, Wang, & Richardson, 2012; Klespitz, Bíró, & Kovács, 2015; Mc Hugh, McCaffery, & Casey, 2017; Müller et al., 2014; Münch, 2016; Wang et al., 2012). To maintain the core intent of lean-Kanban, several lean-Kanban

practices have been established and are steadily evolving. For the categorisation of the empirical data on lean-Kanban practices, Table 1 outlines what is considered to be lean-Kanban specific practices necessary for the development of medical software. As certain lean-Kanban practices are overlapping with a plan-driven SDLC and agile practices, this paper limits lean-Kanban practices to those practices which were found to be less represented in academic writing on Kanban software development but were recurring issues in terms of a lean-Kanban context. It is noted that the identification and categorisation of lean-Kanban practices is somewhat subjective. Nevertheless, the list is agreed upon by consensus among software developers and research, offering a perfect starting point for a mixed method approach to the development of software as medical device.

Table 1: Lean-Kanban practices for medical device software development

Lean-Kanban practice	**Reference**
Address bottlenecks	(Mc Hugh et al., 2017; Wang et al., 2012)
Automatic verification and validation	(Reinertsten, 2009)
Avoid waste	(Münch, 2016)
Clear conceptualization of software development process	(Münch, 2016)
Continuous improvement through iterative steps to establish a smoother flow	(Hibbs, Jewett, & Sullivan, 2009)
Create only necessary documentation	(Graziotin & Jedlitschka, 2013; Shalloway, 2018)
Establish software documentation planning to deliver necessary documentation to notified bodies	(Stellman & Greene, 2014)
Establish group performance, no heroic individuals	(Shalloway, 2018)
Implementation of Kanban approach within quality system	(Münch, 2016; Shalloway, 2018)
Implementation of virtual Kanban board	(Münch, 2016)
Link voice of customers to user and usability requirements	(Cawley et al., 2012; Raffo, Mehta, Anderson, & Harmon, 2010)
Manage pair programming	(Al-Baik & Miller, 2015; Cawley et al., 2012; Mc Hugh et al., 2017)
Optimise the whole process and provide feedback	(Cawley et al., 2012; Poppendieck & Poppendieck, 2007)
PDCA (plan do check act) cycle	(Deming, 1986)
Radical improvement within a limited time	(Lester, 2008; Wang et al., 2012)
Visualise the workflow	(Poppendieck & Poppendieck, 2007)

Mixed method Kanban SDLC

American and European guidelines clearly outline the tailoring of a software development lifecycle in order to standardize development practices and to achieve maximum impact (Zeegers, 2018). This paper focuses on the implementation of Kanban practices within a plan-driven medical device software development

lifecycle. Implementing Kanban practices within a plan-driven SDLC has to primarily aim at developing medical device software with the highest quality possible in order to gain regulatory approval, while still reaping the benefits associated with utilising Kanban practices (Mitasiunas, Rout, O'Connor, & Dorling, 2017). Accordingly, the foundation of such a mixed method approach is provided by a plan-driven SDLC, providing a comprehensive approach guiding through all necessary development stages mandatory for the development of safety-critical medical software (EU, 2017; Kim, Park, Lee, & Lee, 2017). Özcan-Top and McCaffery (2017) emphasise the development of safe and reliable medical software particularly through a combination of Kanban practices with plan-driven methods. For the development of a mixed method Kanban SDLC, the v-model was chosen as a foundation, being a widely accepted approach for medical software developers (Mc Hugh et al., 2017). The mixed method approach particularly benefits from choosing a v-model as foundation in that it provides guidance for all necessary stages for developing medical device software with the highest quality possible in order to gain regulatory approval (Trektere, McCaffery, & Lepmets, 2016). Academic research on the implementation of Kanban practices with a plan-driven SDLC clearly emphasise the use of the v-model as foundation, as it provides guidance on how to perform the necessary development activities (Kropp & Meier, 2014; Pekar et al., 2016; Roßik & Schwarcz, 2015). Several studies outline the v-model as a widely accepted approach to medical software (Cicotti, 2017; Mc Hugh et al., 2017; Polgár & Kazinci, 2014). Furthermore, regulations such as IEC 60601-1 propose the use of a v-model when developing medical devices. In selecting the v-model as foundation for a mixed method SDLC, the v-model for certification, as proposed by Ge, Paige, and McDermid (2010) is used. This v-model differs from the typical v-model, as it also includes a development stage for "usability specification and risk analysis" and the requirements outlined by IEC 60601-1 to obtain "regulatory certification". This model consists of five development phases, integrating the 1.) user requirements, 2.) usability specification and risk analysis, 3.) architectural design, 4.) module design specification, and 5.) module coding. These phases are consistent with the main software development activities specified in the EN 62304:2006: understand and specify the intended use and user requirements, identify possible risks and specify usability requirements, identify design solutions to meet user requirements, code and test designs against requirements. This mixed method approach was used as a basis for the survey, fulfilling both American and European requirements in order to gain regulatory approval when filing a software as medical device to the regulatory agency.

Research methodology

In order to gain an understanding of how shortcomings in both Kanban and plan-driven approaches are resolved by the use of a mixed method Kanban SDLC, a questionnaire-based survey with organisations developing medical device software within Germany was carried out. A literature review was performed, examining the current SDLCs used for the development of safety-critical software within the in-vitro-diagnostics and medical device industry. The results of the literature review,

particularly examining the use of a mixed method Kanban SDLC are outlined in section 1. Several papers on the use of lean-Kanban practices when developing safety-critical software were identified, as outlined in Table 2.

In the year 2016, approximately 1200 organisations produced medical devices in Germany, with only 12 % developing medical device software (Flemming, 2016). Sufficient sample size was determined as eighteen companies, using sample size equation. The organisations surveyed ranged from small to medium-sized companies to large organisations, producing Class I to Class III medical device software.2 The goal of the survey was to provide a better understanding of the challenges faced by software developers when developing safety-critical medical device software by the use of a plan-driven SDLC. The insights gained through the surveys then helped in the development of a mixed method Kanban SDLC approach. Participants surveyed included all members of a development team, regulatory affairs and senior management, whereas at least two persons from each of the eighteen companies identified had have been surveyed.

The survey was pursued through open-ended questions, being in several areas supplemented by closed-ended questions. The closed-ended questions were also designed through simple multiple-choice answers, in order to gain an understanding of the challenges faced by software developers when developing safety-critical medical software. Open-ended questions where used to permit respondents to freely choose how to answer (Patten, 2002). The open-ended questions encourage a continued discussion to provide a richer source of information than closed-ended questions, necessary to answer the second research purposes (Özcan-Top & McCaffery, 2017; Patten, 2002).

In a first part, the responded was asked about the actual role in the medical device industry, followed by the years of experience with medical device software. The quality of the responses were validated through asking several persons within the same company.

The second part of the survey, revealing significantly relevant information, was concerned with the question which SDLC the organisation actually follows. The questionnaire then was structured according to the steps of the v-model for certification as proposed by Ge et al. (2010). The third part of the survey was structured according to the five development phases proposed by Ge et al. (2010), asking the respondents to provide recommendations as to how the adoption of Kanban practices can resolve the challenges associated with the current SDLC applied. Based on the five development phases proposed by Ge et al. (2010), the respondents were provided with a set of activities mandatory to be fulfilled when developing safety-critical medical software. For each activity, the responded rated the significance of the activity pertaining to one of the five phases. In particular, the respondents were asked to provide further insights and proposals for the

[2] The safety classification of the software as medical device is based on the European MDR classification system 2017/745/EU. This regulation defines Class I medical device software, which possesses a relatively small risk to the intended user, and Class III medical device software, possessing a potentially life-threatening risk to the intended user.

implementation of these activities. The objective of this questions was to establish which Kanban practices would be merged with the v-model l. The respondents then were asked how they would rate the effectiveness of their organisation in performing these activities. The objective of this question was to provide an understanding which area of the development of safety-critical medical software are performed most efficiently. For each of the five phases pertaining to the v-model of Ge et al. (2010), information is collected identifying which phases of the SDLC are the ones most challenging to software developers. This information is used to answer the first purpose.

In the last part of the survey the respondents were asked to provide information pertaining to the implementation of Kanban practices within a SDLC. The respondents further were asked in which way the adoption of Kanban practices has to occur. The aim of these questions was to evaluate how to implement the Kanban practices within the v-model.

Results of the survey

One of the objectives of the survey was to provide a better understanding of the challenges faced by medical device software developers. The insights provided by the respondents were also used to analyse the findings of the literature review in this context. All of the organisations surveyed develop and market medical software for Europe. Furthermore, 58 % of these organisations develop software as medical device for use in the USA. 72 % of the eighteen companies surveyed develop their medical software products based on the v-model. Furthermore, 28 % of these companies use lean-Kanban practices for the development of their medical software products or combine the v-model with lean-Kanban practices. As part of the survey, the respondents were asked what, in their view, was a challenge to the adoption of Kanban practices. The survey showed that 35 % of the respondents see "regulatory certification" as a challenge to the adoption of Kanban practices. Furthermore, 20 % of the respondents' report "traceability", whilst 18 % of the respondents report "implementation of risk management activities", 18 % report "quality compliance issues", and 9 % report "documentation" as challenges when implementing Kanban practices. The challenge of "regulatory implementation" is consistent with the findings of the literature review, while all other challenges were not reported in the literature reviewed. Finally, the respondents were asked to report on the implementation of Kanban practices within a v-model. The respondents provided ten essential Kanban practices to be implemented within the v-model. The respondents mentioned that "use cases", "user stories", "Kanban cards" and "collocation of team members" are important practices to be implemented within the first stage, that is, the user requirements stage, of the v-model. For the second stage, the respondents did not mention any further practice to be included. For the third stage of the v-model, the respondents mentioned "limit work in progress" and "use of stories" as important Kanban practices to be implemented. For the fourth stage, the module detailed design, no Kanban practices were proposed by the respondents. The respondents mentioned that in the fifth stage, "pair programming", "continuous coding and integration", "continuous improvement through automated testing",

and “continuous feedback” are important Kanban practices to be implemented. Two respondents added that the Kanban practices mentioned in the fifth stage also have to take place when integration and integration testing occur. When asking the respondents how to implement the Kanban practices within the v-model l, 84 % of them answered that an iterative development approach should be used for stages three to five.

Integration of Kanban practices within a plan-driven SDLC

As part of the survey, the respondents were asked on their organisations SDLC actually used and implemented, to develop medical device software. The survey revealed that most companies use the v-model for the development of software of use in the USA and Europe. This result is consistent with the survey carried out by Ryan (2016), emphasising that software development in the medical device industry is traditionally reflected in a v-model standard. The answers of the respondents lead to the decision to select the v-model as foundation for a plan-driven SDLC for the integration of Kanban practices. The answers show that most companies are very familiar with the use and application of the v-model. The respondents further stated that based on a familiar foundation, the later integration of new practices seems to be much easier. In applying structures and phases already known by the German medical device companies, they are much more willing to adopt lean-Kanban practices within the SDLC they use. An insight not yet described in academic literature is that six companies see the implementation of regulatory requirements as a hindrance in adapting a completely new approach, rather than applying new structures to an existent v-model already implemented. In fact, applying a new SDLC would force companies to re-apply for regulatory approval (Mc Hugh et al., 2017). Application for regulatory approval because of the implementation and application of an approach seems to be challenging and therefore not applicable in the introduction of lean-Kanban practices for medical software development. This is another reason for the use of a v-model as foundation, where lean-Kanban practices are implemented, to create a hybrid SDLC.

Another new insight, not yet described in the literature, is that 28 % of these German medical device organisations use lean-Kanban practices within their SDLC. Actual standards and requirements do not force medical device software developers to use a v-model. Nevertheless, it appears to guide the companies through the development process. Furthermore, the iterative nature of a Kanban approach seems to be one of the main reasons for its implementation, particularly because of the iterative nature of software development projects with several prototypes, as noted by three respondents. It was pointed out by the respondents that the structures of the v-model help to produce the necessary documents for gaining regulatory approval. Two respondents commented that this is the main reason for trying to adopt lean-Kanban practices within their SDLC. The respondents further commented that for the organisations it is unclear at which stages iterative lean-Kanban practices are best implemented.

In answering this question, the integration of lean-Kanban practices occurred on the basis of a plan-driven SDLC, the v-model as proposed by Ge et al. (2010). The survey

revealed that, when integrating Kanban practices, iterative steps in the development of medical software are particularly the stages "architectural design", "module design specification" and "module coding". Accordingly, within these development stages, the lean-Kanban practices are implemented as an iterative approach. This is also confirmed by literature, as Graziotin and Jedlitschka (2013) point out that this iterative approach addresses "the issue of managing a constant flow of requirements". The integration of the iterative approach within the v-model is illustrated in Figure 1.

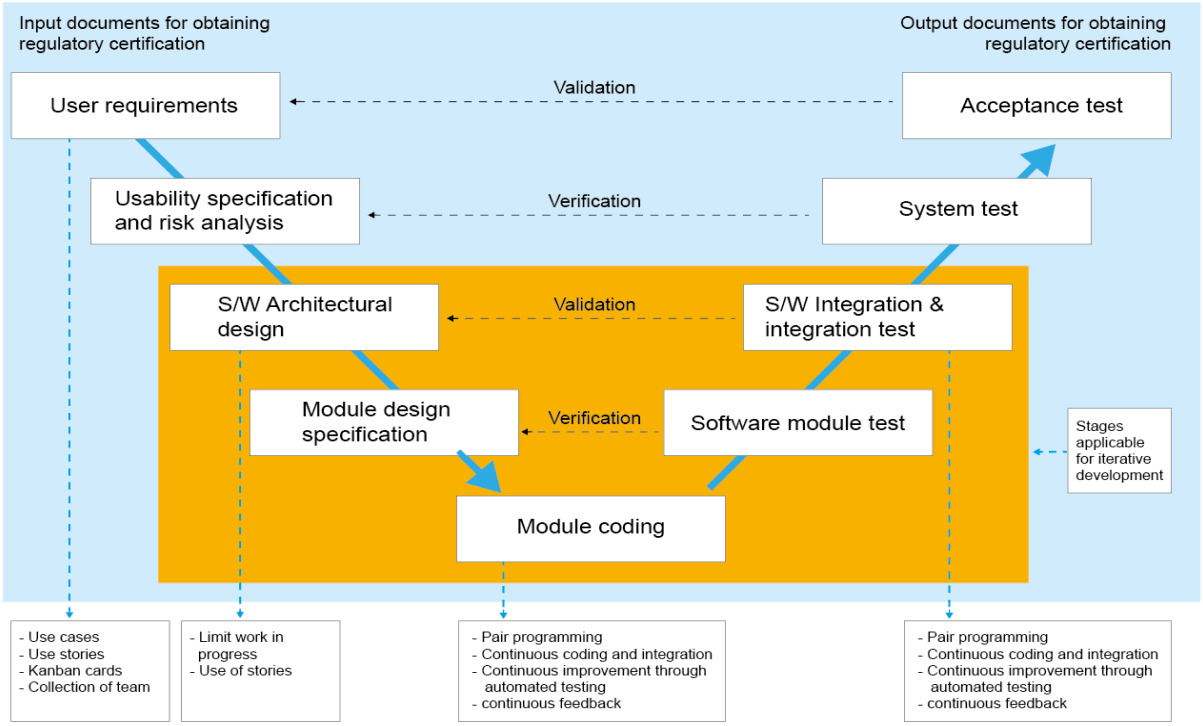

Figure 1: The Kanban V-model (KV-model)

As part of the survey, the respondents were asked what they regard as challenges to the adoption of Kanban practices. In particular, "regulatory implementation" and "traceability" were reported as challenges to the adoption of Kanban practices. The aspect of the implementation of regulatory practices is the challenge consistent with the findings of the literature review (Mc Hugh et al., 2017). An important aspect emerging from the research is the way in which these challenges can be successfully overcome and managed when using Kanban practices. The American, European and Canadian regulatory environment requires the developers of medical software, on the one hand, to document the requirements before beginning with the software development. Afterwards, the requirements are then translated into architectural and modular specifications and required to be traceable during the entire SDLC. Kanban practices require a continuous and fluid set of actions and improvements throughout the entire software project. This does not seem to be contradicting, as a fluid set of actions and continuous improvements is to be implemented in an iterative way within the SDLC, being traceable at any time (Pekar et al., 2016). Much more, the FDA outlines in its guideline for industry on the "General Principles of Software Validation" that:

> "Most software development models will be iterative. This is likely to result in several versions of both the software requirement specification and the software design specification. All approved versions should be archived and controlled in accordance with established configuration management procedures".

Accordingly, after identifying and documenting the user requirements, each of them is broken down into manageable chunks (Kim et al., 2017) and traced during the entire development project. Thus, the requirements to, on the one hand, enable "regulatory implementation", and to ensure, on the other hand, "documentation" right from the beginning, are also reported as being challenges in the implementation of Kanban practices. "Traceability" is another example of how challenges to the implementation of Kanban practices into a plan-driven SDLC can be overcome.

Furthermore, a set of Kanban practices were identified to be implemented within each of the five stages pertaining to the v-model proposed by Ge et al. (2010). This set of Kanban practices is based on the comments provided by the respondents and based on the practices identified during the literature review. The literature revealed several activities consistent with the insights provided by the respondents. Nevertheless, nowhere did the literature provide information as to at which stage of the development the Kanban practices should be incorporated. The Kanban practices proposed by the respondents were implemented to the stages proposed. Furthermore, a cross-checking with the activities defined by IEC 62304:2006 was performed to proof consistency of the Kanban practices proposed by the respondents, as tabulated in Table 2.

Table 2: Proposed Kanban activities and cross-reference to IEC 62304:2006

Development stage	**Respondents' insights**	**IEC 62304:2006**
User requirements	- Use cases - Use stories - Kanban cards - Collection of team members	- No reference
Usability specification and risk analysis	- No propositions made	- No reference
Architectural design	- Limit work in progress - Use of stories	Section 5.3: - Stories
Module design specification	- No propositions made	- No reference
Module coding	- Pair programming - Continuous coding and integration - Continuous improvement through automated testing - Continuous feedback	Section 5.5.5: - Pair programming - Continuous integration - Continuous automated testing

Conclusions

A number of challenges to the adoption of lean-Kanban practices when developing software as medical device has been identified during the literature review. The identified challenges have been confirmed by the respondents during the survey, and challenges not yet described in literature were identified. Key requirement is that the software developer must develop the software according to international regulations and regulatory requirements, in order to obtain regulatory approval. Regulatory bodies do not dictate to the software developer the use of a specific development approach, but generally recommend to the developer the implementation of iterative life cycling. Streamlining the SDLC to gain regulatory conformance, implementing the required stages with design control activities to produce the necessary documentation, and implementation of iterative lifecycle management are perceived as challenges when implementing a lean-Kanban SDLC. The proposed hybrid SDLC (KV-model) is categorised into five development activities, each producing the regulatory deliverables which device manufacturers must deliver to the regulatory bodies. The KV-model proposed in this research is based on a mixed method Kanban SDLC and identifies Kanban practices pertaining to each of the five development stages. The lean-Kanban practices are founded on the comments provided by German software as medical device developers. Selectively choosing appropriate lean-Kanban practices and integrating them within a plan-driven lifecycle reaps the advantages of employing Kanban practices whilst still producing the mandatory regulatory deliverables.

References

Al-Baik, O., & Miller, J. (2015). The kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering, 20*(6), 1861-1897.

Anand, G., Chandrashekar, A., & Narayanamurthy, G. (2014). Business process reengineering through lean thinking: a case study. *Journal of Enterprise Transformation, 4*(2), 123-150.

Cawley, O., Wang, X., & Richardson, I. (2012). *Regulated software development–an onerous transformation.* Paper presented at the International Symposium on Foundations of Health Informatics Engineering and Systems.

Cicotti, G. (2017). An evidence-based risk-oriented V-model methodology to develop ambient intelligent medical software. *Journal of Reliable Intelligent Environments, 3*(1), 41-53.

Clarke, P., Lepmets, M., McCaffery, F., Finnegan, A., Dorling, A., & Eagles, S. (2017). Characteristics of a medical device software development framework.

Deming, W. (1986). Out of the crisisMIT Center for Advanced Engineering. *Cambridge, MA*.

EU. (2017). Medical device regulation. *Official Journal of the European Union*(2017/745).

Flemming, G. (2016). The medical technology industry in Germany. In G. t. a. invest (Ed.). Nürnberg.

Ge, X., Paige, R. F., & McDermid, J. A. (2010). *An iterative approach for development of safety-critical software and safety arguments.* Paper presented at the AGILE Conference, 2010.

Graziotin, D., & Jedlitschka, A. (2013). Recent developments in product-focused software process improvement: PROFES 2013 conference report. *ACM SIGSOFT Software Engineering Notes, 38*(6), 29-34.

Hibbs, C., Jewett, S., & Sullivan, M. (2009). *The art of lean software development: a practical and incremental approach*: " O'Reilly Media, Inc.".

Kim, D., Park, Y.-S., Lee, B., & Lee, J.-W. (2017). Association-Based Process Integration for Compliance with Core Standards in Development of Medical Software *Advances in Computer Science and Ubiquitous Computing* (pp. 1220-1226): Springer.

Klespitz, J., Bíró, M., & Kovács, L. (2015). *Aspects of improvement of software development lifecycle management.* Paper presented at the Computational Intelligence and Informatics (CINTI), 2015 16th IEEE International Symposium on.

Kropp, M., & Meier, A. (2014). Swiss Agile Study 2012. *Agile Software-Entwicklung in der Schweiz. Zürcher Hochschule für Angewandte Wissenschaften*.

Lester, R. I. (2008). Lean Thinking: Banish Waste and Create Wealth in Your Corporation. *Air & Space Power Journal, 22*(2), 103-105.

Mc Hugh, M., McCaffery, F., & Casey, V. (2017). Barriers to using agile software development practices within the medical device industry.

McHugh, M., McCaffery, F., & Casey, V. (2014). Adopting agile practices when developing software for use in the medical domain. *Journal of Software: Evolution and Process, 26*(5), 504-512.

Mitasiunas, A., Rout, T., O'Connor, R. V., & Dorling, A. (2017). Software Process Improvement and Capability Determination. *Communications in Computer and Information Science, 477*.

Müller, E., Tolujew, J., & Kienzle, F. (2014). Push-Kanban–a kanban-based production control concept for job shops. *Production Planning & Control, 25*(5), 401-413.

Münch, P. (2016). Using Smartsheet with Kanban. Retrieved 24.08.2018, 2018, from https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban

Özcan-Top, Ö., & McCaffery, F. (2017). A hybrid assessment approach for medical device software development companies. *Journal of Software: Evolution and Process*.

Patten, D. M. (2002). The relation between environmental performance and environmental disclosure: a research note. *Accounting, organizations and Society, 27*(8), 763-773.

Pekar, V., Felderer, M., Breu, R., Nickl, F., Roßik, C., & Schwarcz, F. (2016). *Integrating a lightweight risk assessment approach into an industrial development process.* Paper presented at the International Conference on Software Quality.

Polgár, P. B., & Kazinci, F. (2014). Process communication in medical software development. *Journal of Software: Evolution and Process, 26*(3), 313-320.

Poppendieck, M., & Poppendieck, T. (2007). *Implementing lean software development: From concept to cash*: Pearson Education.

Raffo, D., Mehta, M., Anderson, D. J., & Harmon, R. (2010). *Integrating Lean principles with value based software engineering*. Paper presented at the Technology Management for Global Economic Growth (PICMET), 2010 Proceedings of PICMET'10:.

Reinertsten, D. G. (2009). *The principles of product development flow: second generation lean product development*: Celeritas.

Roßik, C., & Schwarcz, F. (2015). *Integrating a Lightweight Risk Assessment Approach into an Industrial Development Process.* Paper presented at the Software Quality. The Future of Systems-and Software Development: 8th International Conference, SWQD 2016, Vienna, Austria, January 18-21, 2016, Proceedings.

Ryan, K. T. (2016). Software processes for a changing world. *Journal of Software: Evolution and Process, 28*(4), 236-240.

Shalloway, A. (2018). Lean-Kanban manifesto. Retrieved 24.03.2018, 2018, from http://www.netobjectives.com/blogs/lean-kanban-manifesto

Stellman, A., & Greene, J. (2014). *Learning agile: Understanding scrum, XP, lean, and kanban*: " O'Reilly Media, Inc.".

Sugimori, Y., Kusunoki, K., Cho, F., & Uchikawa, S. (1977). Toyota production system and kanban system materialization of just-in-time and respect-for-human system. *The International Journal of Production Research, 15*(6), 553-564.

Trektere, K., McCaffery, F., & Lepmets, M. (2016). *Development of a software process framework to assist organizations developing mobile medical apps.* Paper presented at the Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on.

Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software, 85*(6), 1287-1299.

Zeegers, R. (2018). *Information Security Management Professional based on ISO/IEC 27001 Courseware–English*: Van Haren.