

## Mobile Medical App Development with a Focus on Traceability

Journal:	<i>Journal of Software: Evolution and Process</i>
Manuscript ID	JSME-16-0196
Wiley - Manuscript type:	Special Issue Paper
Date Submitted by the Author:	19-Sep-2016
Complete List of Authors:	Trekte, Kitija; Dundalk Institute of Technology, Computing and Mathematics Regan, Gilbert; DKIT, Computing McCaffery, Fergal; Dundalk Institute of Technology, Computing and Maths Flood, Derek; Dundalk Institute of Technology, Computing Lepmets, Marion; Dundalk Institute of Technology, Regulated Software Research Centre Barry, Grainne; Dundalk Institute of Technology, Computing and Mathematics
Keywords:	Medical Device, Medical Device Software, Mobile Application, Mobile Medical App, Traceability, TAIF model

SCHOLARONE™  
Manuscripts

view

## Mobile Medical App Development with a Focus on Traceability

Kitija Trektere, Gilbert Regan, Fergal Mc Caffery, Derek Flood, Marion Lepmets,  
Grainne Barry {kitija.trektere, gilbert.regan, fergal.mccaffery, derek.flood}@dkit.ie,  
marion.lepmets@gmail.com,  
grainne@salaso.com

**Abstract.** Today the growth of medical devices and mobile medical applications is increasing enormously thanks to the efficiency and enhancement of new technology. When it comes to mobile medical apps, developers need to understand what is required when a mobile application fulfils the definition of a medical device. Such applications have to be developed in compliance with medical device regulations. This can be a challenge for mobile medical application developers as medical device software is normally developed in a manner that will also ensure the production of regulatory documentation that is essential to market such devices. In this journal paper we identify the key criteria for a mobile medical application development framework and describe how the results were collected through performing a Medical Device Software Development workshop. Furthermore, we describe how MDevSPICE<sup>®</sup> together with an agile software development approach can be tailored to support a mobile medical applications development framework. We detail one of the key criteria for mobile medical application development framework – traceability.

**Keywords:** Medical Device, Medical Device Software, Mobile Application, Mobile Medical App, Traceability, TAIF model.

### 1 Introduction

Today, medical device (MD) software systems, including mobile medical apps (MMA), play an important role in our lives. MD software, according to the FDA definition, is a: “software system that has been developed for the purpose of being incorporated into the medical device being developed or that is intended for use as a medical device in its own right” [1]. However, a MMA is “a mobile app that meets the definition of device in section 201(h) of the Federal Food, Drug, and Cosmetic Act (FD&C Act) 4; and either is intended to be used as an accessory to a regulated medical device or to transform a mobile platform into a regulated medical device” [2]. “If a mobile app is intended for use in performing a medical device (MD) function (i.e. for diagnosis of disease or other conditions, or the cure, mitigation, treatment, or prevention of disease) it is a medical device” [2].

Given the safety-critical nature of MDs, standards are in place to ensure the safety of these devices. In order to market a MD, it is mandatory to satisfy the regulatory requirements within that particular region. Previously, the authors developed MDevSPICE<sup>®</sup> which contains all the MD software development best practices [3] and most of the standards in one place. This will be discussed in more detail further in the paper as by tailoring MDevSPICE<sup>®</sup> for MMAs - we envisage developers using it as their main development approach for MMAs.

#### 1.1 Problem Statement

The challenge that MMA software development companies face when they want to market an app is the adherence to the large number of regulatory requirements spec-

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

ified in various international standards. In order to help these companies better prepare for demanding and costly regulatory audits alongside the development process, we propose to develop a MMA development framework. This framework would ease, assist the development process of an app and facilitate the collection of all regulatory requirements by guiding developers from the initial requirements gathering right through to the finished product.

Within this paper we discuss how MMA development with an agile approach may be introduced into MDevSPICE<sup>®</sup> and finally create a MMA development framework that would comply with all the regulations and allow developers to successfully market their application [4].

Section II, looks at the different MD regulations and classifications for MD and MMA. In Section III, we describe MDevSPICE<sup>®</sup>. In Section IV, we describe the methodology adopted to develop the MMA development framework and we describe the workshop that was performed to discuss the different software development approaches for MD software. In Section V, we focus on one of the MMA development framework criteria that emerged from the workshop— traceability. We look at its benefits and barriers, and describe the TAIF model that was developed by some of the authors and validated to assist MD companies. The paper concludes with details of our future work in Section VI.

## 2 Medical Device Regulations

The largest global bodies responsible for issuing and managing MD regulations are the FDA in the US and the European Commission in the EU.

- The FDA issues the regulation through a series of official channels, including the Code of Federal Regulation (CFR) Title 21, Chapter I, Subchapter H, Part 820 [5].
- In the EU, the corresponding regulation is outlined in a number of directives: the general Medical Device Directive (MDD) 93/42/EEC [6]; the Active Implantable MDD (AIMDD) 90/385/EEC [7]; and the In-vitro Diagnostic (IVD) MDD 98/79/EC [8] - all three of which have been amended by 2007/47/EC [9].

### 2.1 Medical Device Classification

The safety classifications for MDs depends upon the particular geographical region, the two that we refer to are the US and Europe. The MD safety classification relates to the clinical safety of the device.

There are three MD safety classifications within the US: Class I, Class II and Class III.

- Class I devices, such as a thermometer are not intended to support or sustain human life, and may not present an unreasonable risk of harm.
- Class II devices, such as a powered wheelchair could cause damage or harm to humans.
- Class III devices, such as an implantable pacemaker are usually those that support or sustain human life, and are of significant importance in the prevention of human health impairment.

In the EU, Class I corresponds to Class I in the US, with Class IIa and IIb corresponding to Class II in the US, and Class III to Class III in the US.

### 2.2 Medical Device Software Regulations

Within the safety-critical domains it is important to have a Quality Management System (QMS) in place. For example, the QMS required to meet the regulatory require-

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

ments of the MD domain in the EU is ISO 13485 [10]. ISO 13485 is a specialization of ISO 9001 [11] for the MD domain. QMS requirements need to be adhered to in order to market a MD anywhere in the world. As ISO 13485 does not target software requirements, it can be used in conjunction with IEC 62304 [12] to support the safe design and maintenance of MD software. IEC 62304 is a critical standard for MD software developers as it is the only standard that provides recommendations for MD software implementations based on the worst case scenario of software failure causing hazards. For general MD risk management, IEC 62304 is used in conjunction with ISO 14971 [13] and IEC 80002-1 [12]. IEC 80002-1 provides guidance on the application of ISO 14971 for software development.

IEC 62304 considers a MD to include software as opposed to the being a complete MD system in its own right. The system or product level requirements are not included within IEC 62304 but instead within the MD product standard IEC 60601-1 [12]. Due to the increasing importance of usability of devices within the MD industry, organizations should also adhere to the MD usability engineering process requirements outlined in IEC 62366 [15].

Standalone software systems (such as MMAs) may be defined as a MD in their own right, due to an amendment to the MDDs [9]. However, until recently international MD standards did not address standalone software as a MD. To address this IEC FDIS 82304-1 [15] will soon be published. IEC FDIS 82304-1 applies to the safety of health software that is designed to operate on general purpose IT platforms intended to be placed on the market without dedicated hardware, e.g., tablet and phone applications. Therefore, IEC FDIS 82304-1 will be applicable to MMAs.

If the company is planning to market a MD in the US, they need to register their product with the FDA. The FDA has issued an overview of their guidance documents for developers of MD software [16]. There are four guidance documents: the FDA Guidance on Premarket Submissions [17]; the FDA Guidance on Off-The-Shelf Software Use in Medical Devices [18]; the FDA General Principles of Software Validation [19]; and Guidance on MMAs [1] to provide clarity for MMA developers in terms of what types of applications will be regulated and how.

Whenever companies developing MD software wish to market a device they must adhere to a large number of regulatory requirements and international standards. In the case of MMAs meeting the definition of a MD, they have to comply with the same regulations as a MD. In order to reduce the demanding and costly overhead associated with preparing for regulatory audits, we developed MDevSPICE<sup>®</sup>. MDevSPICE<sup>®</sup> integrates the software requirements from all the above mentioned international MD standards and guidance documents with generic software development best practices into a single reference source. This reference source may then be used to assess the capability of an organization's MD software processes.

### 2.2.1 Medical Device Software Classifications

IEC 62304:2006 [12] describes MD software lifecycle processes and outlines software safety classification, based on the worst possible consequence in the case of a software failure. In the case of failure of software that is of safety Class A no injury or damage to health of a patient can occur. When software of safety class B fails, injury may occur but it is not serious or life-threatening. Class C is of the highest risk and in the case of failure of such software death or serious injury can happen. Depending on the functionality of software within the MD, the software safety classification may vary from the overall MD safety class. When software is of critical functionality of the

MD, it will carry the same classification as the device, i.e. Class C software in Class III device.

### 2.3 Mobile Medical App Types

Both US and Europe have released guidelines “to support developers and manufacturers ensure public safety” [20].

In September 2013, the FDA in the US issued a Final Guidance on Mobile Medical Applications [1]. The FDA categorized MA into three categories depending on their risk to a human in case of its failure:

- (1) Those apps meeting the definition of a MD and “either is intended to be used as an accessory to a regulated MD; or to transform a mobile platform into a regulated MD” [1], therefore “are the focus of FDA’s regulatory oversight” [1]. These mobile apps pose high risk to a human in case of its failure – i.e. injury or death. An example of an app for this category would be one that regulates infusion pump settings;
- (2) Those apps meeting the definition of a MD, however, through posing a low risk to a human in case of its failure, the FDA intends to exercise enforcement discretion [1]. For example, an app that helps an asthmatic to track their inhaler usage;
- (3) The third category includes all apps that do not meet the definition of a MD, therefore the FDA does not regulate them. For example, an app that educates patient or tracks their fitness progression. To sum it up, if a mobile app cannot in any way harm a human, the FDA will not subject it to regulations.

In Europe, in January 2012, MEDDEV 2.1/6 was published to provide guidelines on stand-alone software development for MMAs [21]. In order to market an app as a MMA, a CE Mark is required indicating that the requirements of either the MDD or the IVDD have been fulfilled.

### 3 MDevSPICE®

MDevSPICE® was designed to assist manufacturers to develop MD software through following a traditional waterfall model or V-model lifecycle model. Such development lifecycles are very rigid, prescribed and sequenced, which makes them both unsuitable and slow for MMAs development.

As IEC 62304 describes only the software lifecycle processes, additional processes are required when the software developed forms the complete MD. In such cases systems development processes need also to be considered. These systems development processes were derived from ISO/IEC 12207:2008 [22]. Quality Management and Risk Management related requirements from ISO 13485 and ISO 14971 were also added to MDevSPICE® as both ISO 13485 and ISO 14971 are de facto standards for MD software development organizations. MDevSPICE® consists of 23 processes, see table 1. Ten are system lifecycle processes, eight are software lifecycle processes and five support both the system and lifecycle processes.

MDevSPICE® was validated through both international expert review and industrial trials. The foundation of MDevSPICE® is IEC TR 80002-3, the Process Reference Model for IEC 62304 [23]. Upon successful completion of international expert reviews, MDevSPICE® was validated through pilot assessments in 10 MD software development organizations over two years.

MDevSPICE® provides full lifecycle coverage in a plan-driven manner describing the requirements of all associated MD standards in one place, integrating software engineering best practices so that safer MD software may be developed.

1  
2  
3  
4  
5  
6  
7 However, the emergence of MMAs presents a new challenge to the development of  
8 MD software is with.

#### 9 **4 Approach used to understand the needs for Mobile Medical App** 10 **Development Framework**

11 The research questions we were exploring are:

- 12 • What are the challenges associated with development of MMA software?
- 13 • What are the criteria/features developers would like to see within MMA develop-  
14 ment software framework?
- 15 • What are the existing software development approaches for developing MMA  
16 software?

17 To answer these questions, we performed a workshop that included the experts from  
18 the MD community. The Oxford dictionary defines a workshop as “a meeting at which  
19 a group of people engage in intensive discussion and activity on a particular subject  
20 or project” [24]. An action research methodology consists of five phases: diagnosing,  
21 action planning, action taking, evaluating and specifying learning [25] and will be de-  
22 scribed in more detail in the following subsections. In the development of the MMA  
23 framework the research methodology adopted is action research and the workshop  
24 formed a key part for the diagnosing stage.

#### 25 **4.1 Mobile Medical App Development Framework Workshop**

26 In order to be able to run a truly scientific workshop we approached 37 organizations  
27 that develop either MD or MMA software. The Regulated Software Research Centre  
28 (RSRC) maintains a repository of contact details of MD software experts within Ire-  
29 land and this was used as the basis for identifying suitable organizations. The selec-  
30 tion process then consisted of inviting representatives (from this repository) from mul-  
31 ternational and SMEs MD software communities within Ireland.

32 Fifteen (41%) of the invited organizations sent at least one representative, resulting in  
33 a total of 23 representatives. We ran two workshops in two different locations, Dun-  
34 dalk (North of Ireland) and Limerick (South of Ireland) in order to cover the geograph-  
35 ical spread, see table 2.

36 A structured invitation template explaining the objective of the workshop and location  
37 was developed along with a separate word document detailing the agenda that ex-  
38 plained the workshop structure. These details were sent to chosen experts approxi-  
39 mately a month prior to the first workshop. After sending invitations, a reminder email  
40 was sent one week prior to the first workshop, followed by an email to the confirmed  
41 representatives detailing location (including a map and the conference room num-  
42 ber). Both workshops were run in academic institutes – Dundalk Institute of Technol-  
43 ogy and the University of Limerick; we discussed the changing face of MD software  
44 and the associated standards, and regulations that organizations need to be aware of  
45 today. We also looked at the current MMA software development approaches and  
46 discussed the challenges when developing mobile medical applications. Then, we  
47 had a session discussing the awareness levels of agile for MMA software develop-  
48 ment and we concluded the workshop with suggestions with what should be included  
49 in a MMA development framework.

#### 50 **4.2 Mobile Medical Application Development Challenges**

51 One of the discussion topics within the workshop was: What are the challenges as-  
52 sociated with developing MMA software? Participants were divided into groups of 4-5  
53 people from different companies and were allocated 20 minutes for discussion. At the  
54

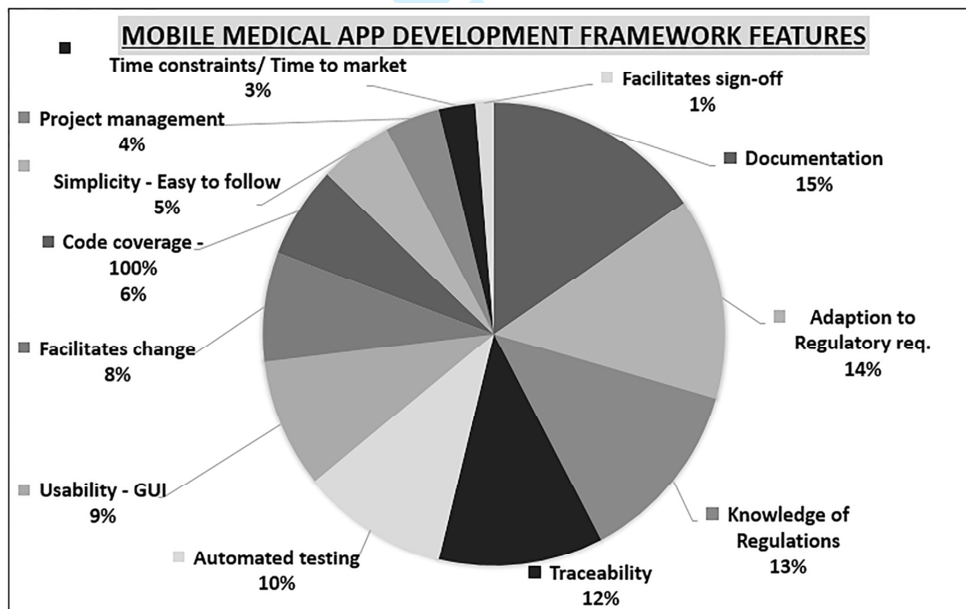


end of this discussion, group leaders were asked to preset the list of MMA dev challenges on a flipchart. Once all challenges were reported back, an overall discussion took place to enable us to arrive at the final agreed list of challenges which are shown in Table 3.

### 4.3 Mobile Medical App Development Framework Features

Our second workshop discussion related to the following question: What are the criteria/features developers would like to see within a MMA development software framework? This discussion was run in the same manner as outlined in Section 4.2.

We discovered that there is a necessity to develop regulatory compliant applications in a shorter timeframe than traditional MD software. Figure 1, represents the features that experts felt should be included into a MMA development framework. As illustrated in figure 1, the most highly desired features were regulatory requirements, the need for documentation to satisfy these requirements through objective evidence and traceability.



**Figure 1 Mobile Medical App Development Framework Features**

In order to be able to market a MMA, it is essential to achieve regulatory compliance. This involves providing fully traceable documentation that forms the objective evidence required by the auditors. Therefore, it is understandable that these features: documentation, adaption to regulatory requirements, knowledge of regulations and traceability were the most sought after from the MD software experts that participated in the workshop. However, time to market is also important and it is a key distinguishing factor between the development of traditional MD software and the development of a MMAs.

1  
2  
3  
4  
5  
6  
7 The workshop outlined that the mentioned features could be achieved through a hybrid development approach when agile practices were integrated into a plan driven software development approach such as the V-model.  
8  
9

#### 10 **4.4 Approaches**

11 Our third workshop discussion related to the following question: What are the existing software development approaches for developing MMA software? This discussion was conducted as outlined in Section 4.2.

12  
13  
14 Generally, in the MD software industry the V-model is adopted. However, due to successes reported within other domains from implementing agile practices, an international working group was set up to investigate if agile practices could fulfil the requirements of IEC 62304. The Association for the Advancement for Medical Instrumentation (AAMI) working group found that “agile practices can be successfully adopted to develop regulatory compliant software” [26] and produced AAMI:TIR 45:2012 [27] which is an internationally accepted technical report called “Guidance on the use of Agile practices in the development of medical device software”. AAMI:TIR 45 provides general guidelines on how agile methods may be used to satisfy IEC 62304. To obtain a suitable development lifecycle for MMAs we have compared various potential MD software development approaches before deciding to develop a MMA framework where agile practices are combined with MDevSPICE<sup>®</sup> while observing the guidelines of AAMI:TIR 45.

15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27 As part of the workshop we presented six possible Software Development Life Cycle (SDLC) approaches that are used for developing MMAs. Based upon the two previous group discussions concerning MMA development challenges and identifying MMA development framework features, it was agreed by participants to categorize each SDLC approach in terms of the following criteria: a) achieving regulatory compliance and producing the required documentation; and b) time to market – does the approach assist the MD to be marketed in a timely manner (see Table 2). In the following section we outline the discussion surrounding each software development approach.  
28  
29  
30  
31  
32  
33  
34  
35

##### 36 **4.4.1 Waterfall or V-model**

37 The intention of plan-driven software development methods is to improve the product quality by reducing the number of errors that are made, while at the same time supporting the achievement of delivery dates, budget constraints and in the case of MD software development the achievement of the primary goals of creating safe and effective MDs. MD software developers typically develop software in accordance with a plan-driven sequential SDLC, such as the V-Model or Waterfall [28], as it is a tried and trusted lifecycle approach within the industry for producing the deliverables demanded by the regulatory authorities [29], [30]. However, participants agreed that this is a time consuming, resource-intensive and rigid plan-driven approach that is not well suited for developing MMAs in a timely manner.

38  
39  
40  
41  
42  
43  
44  
45  
46  
47 The group discussion identified several concerns with adopting either a waterfall or V-Model SDLC adoption, such as: there is no focus upon *iterations*, consequently, making it difficult to deliver frequent releases and to change incorrect decisions in a timely manner; they do not embrace *change* easily, thus, any changes introduced once the project has started can create financial overruns; they do not include a *prototyping* practice, therefore making it more difficult to engage users in the development process; a large amount of time is spent producing and verifying *documentation*, therefore leaving less time for the development and testing; there is a *sign-off*  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60



1  
2  
3  
4  
5  
6  
7 process after every software development phase, hence, bringing an inflexibility in  
8 terms of incorporating changes post sign-off.

9 In traditional MD development there is a heavy emphasis on documentation and sign-  
10 off before proceeding to the next stage. This can lead to delays when changes need  
11 to take place. Such development lifecycles are waterfall in nature with each stage  
12 having to be completed prior to the next commencing. Therefore, the key concern in  
13 relation to this approach is that delivering working software may take a long time.

#### 14 **4.4.2 MDevSPICE®**

15 Similarly, to MD software standards, MDevSPICE® doesn't specify the development  
16 method that should be used in the design and development of MD software. Yet, giv-  
17 en the fact that the following are required for MD regulatory approval: a defined pro-  
18 cess description; sign-off on all tasks and requirements; and documented evidence of  
19 these, this suggests that either a waterfall or V-model lifecycle approach is well suit-  
20 ed.

21 Participants felt, that a significant advantage of MDevSPICE® is that the software  
22 processes necessary to satisfy regulatory requirements are identified from the outset  
23 and implemented accordingly. While this approach involves the production of a lot of  
24 documentation, it streamlines the identification of appropriate software development  
25 practices. It was decided by participants that despite MDevSPICE® streamlining the  
26 procedure for obtaining regulatory compliance and generating the required documen-  
27 tation, a plan driven implementation is still time-consuming and resource-intensive.  
28 Participants felt, that increased agility permits a more rapid adaptation to changing  
29 requirements and allows early detection of errors through prototyping, which fits the  
30 MMA development better than plan-driven methods.

#### 31 **4.4.3 Introducing Agile Practices into the V-model Lifecycle (not based on** 32 **MDevSPICE®)**

33 "Agile development focuses on achieving personal, technical, and organizational  
34 successes" [31]. Agile software development reports "the promise of improved soft-  
35 ware quality and reduced delivery times" [32]. There are several reported benefits for  
36 adopting agile development practices including: more rapid feedback and adjusting  
37 and embracing change accordingly; producing frequent releases of working software;  
38 and reducing formal organisational red tape.

39 An agile approach for software development for MDs goes back to the 2000s, and in  
40 2005 Sutherland evaluated Scrum, which is an agile project management approach,  
41 by using a medical company [33]. Scrum is currently implemented in several MD  
42 software development companies [34], [35], [36], [37]. Scrum is widely used due to its  
43 clear roadmap; its structure is easy to understand and to implement. However, Scrum  
44 is only one part of the solution as it focuses on project management and collabora-  
45 tion by concentrating on business value through demonstration and prioritization.  
46 Therefore, eXtreme Programming (XP) can be used in conjunction with Scrum [2].  
47 Case studies [38], [39], [40], [34] have emphasized that following an agile approach  
48 can overcome the issues associated with plan driven software development [41].

49 Participants felt, that by integrating agile practices into the V-model development  
50 lifecycle it provides the advantage of accommodating changes more easily than the  
51 previous two approaches. However, it was also felt as this approach is not based on  
52 MDevSPICE®, there is an overhead on ensuring that all the regulatory practices with  
53 the necessary documentation are in place. McHugh et al. [26] found that by integrat-  
54 ing agile practices into the V-model to streamline requirements changes, the amount

of rework required is reduced and delivering MD software is achieved in a more timely manner.

#### 4.4.4 Integrating Agile Practices into the V-model (based upon MDevSPICE®)

Alongside with benefits and barriers that an agile approach brings, there are also some challenges, such as, the need for appropriate tool support for generating documented evidence of all iterations; a proper level of traceability, risk mitigation, verification and validation of the processes; and the need to provide detailed documentation (e.g., detailed test steps [40]).

This approach is based upon MDevSPICE® and includes agile practices that may be used to fulfil the requirements defined in the framework. Participants thought that this approach assists companies to ensure that they have all the necessary standards in place including all the documentation gathered during all stages of the lifecycle. However, participants felt, that there is still a challenge in relation to determining how to capture this documentation when integrating agile practices. This approach supports getting the product to the market faster than the previous approach as guidance is provided to assist organizations to put the required standards in place.

#### 4.4.5 Integrating Agile Practices into Reduced V-model (not based on MDevSPICE®)

This approach looks to integrate agile practices into a reduced V-model lifecycle that only includes essential activities for MMA development. The participant group found that by not integrating MDevSPICE®, additional attention is required in order to ensure that adequate coverage of standards is in place within the implemented agile practices. It was felt that as this approach is light on documentation; extra attention should be paid in terms of what documentation is required for regulatory compliance. Participants thought, that whilst agile practices allow working MMA software to be delivered faster, the process of implementing one standard at a time will slow the process down significantly.

#### 4.4.6 Integrating Agile Practices into Reduced V-model (based upon MDevSPICE®)

This approach is similar to the previous approach except it is based upon MDevSPICE®. This streamlines the process overhead of ensuring that implemented agile practices satisfy all the required MD guidelines and also reduces the implementation time requirement. The authors proposed to create a mapping of the most applicable agile practices within the SDLC against the complete MDevSPICE®. Then, populate the reduced V-model that is based on MDevSPICE® with the most suitable agile practices to fulfil the minimum set of requirements for MMA development. This approach is light on documentation, only including the most essential documents to satisfy the regulatory requirements. Crucially, it was felt by participants, that by adopting this approach, MMA software will reach the market faster than the other approaches as developers are only focused on essential activities, with regulatory compliance being achieved seamlessly during development.

The conclusion from this discussion was that “integrating agile practices into reduced V-model” was considered to be the best MMA development approach for the workshop group participants.

The next section is going to discuss one of the key criteria identified in the discussion on features for inclusion in a MMA development framework which is traceability.

## 5 Traceability

Traceability is the ability to establish and use the links (or traces) between source artefacts and target artefacts [42] and it is important as developers need to ensure that their software conforms to customer expectations, functional and regulatory requirements. Furthermore, organizations that produce safety critical software, such as those operating within the MD domain, must ensure their software is safe, reliable and available. Traceability is an important component in ensuring safe and reliable software and is a required component of the approval and certification process [43] which requires that safety critical software is certified before entering service. Moreover, as noted above, traceability is a highly desired feature for MMA development framework.

As in other domains, traceability implementation within the MD domain is challenging and has been highlighted by a recent analysis of the traceability documents submitted to regulators in the US as part of the MD approval process. This analysis has revealed that the traceability data was incomplete, incorrect, and conflicting in many cases [43].

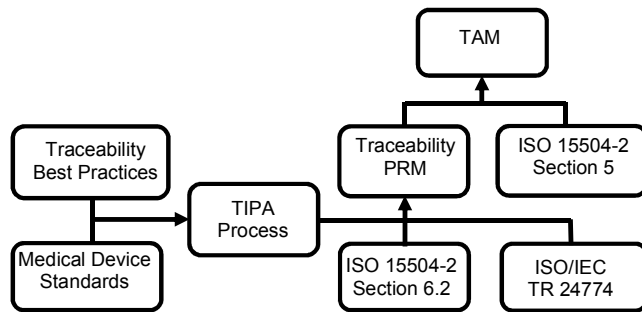
Therefore, to assist MD software organizations implement traceability effectively the authors have developed a traceability assessment and implementation framework (TAIF). The TAIF consists of a traceability assessment model (TAM) and a traceability roadmap (TR). The TAM allows for the identification of any weaknesses in an organizations existing traceability process and the TR provides the pathway for an organization to address those weaknesses. We define a traceability roadmap as a set of steps or activities that, if followed, will provide guidance to an organization implementing traceability, and ensure that it is compliant with the traceability requirements from the MD standards and traceability implementation best practices.

An overview of the approach employed to develop the TAM and its structure is presented in section 5.1. Further detail on the structure of the TAM can be obtained in [44, 45]. Section 5.2 presents an overview of the approach used to develop the TR along with its structure. Further detail on the structure of the roadmap can be obtained in [46]. Section 5.3 provides an outline of the approach employed to validate the TAIF in addition to a summary of the findings from implementing the TAIF within two MD software organizations.

### 5.1 TAM Development and Structure

It was decided to base the development of the TAM on the ISO/IEC 15504 (SPICE) assessment model mainly because ISO/IEC 15504 is used extensively in other safety critical industries such as the automotive industry (Automotive SPICE), space industry (SPICE 4 SPACE) and the MD industry (MDevSPICE).

Having decided to base the TAM on IS/IEC 15504, an approach to its development was formulated as depicted in Figure 2.



**Figure 2 TAM Development Approach**

A Process Assessment Model (PAM), as defined in ISO/IEC 15504, is based on a two dimensional model containing a process dimension and a capability dimension. The process dimension is provided by a Process Reference Model (PRM), which defines a set of processes characterized by statements of process purpose and process outcomes.

The development of the PRM began with an analysis of the MD standards in order to extract the traceability requirements within those standards. The standards analyzed were: ISO 13485; IEC 62304; ISO 14971; IEC 80002-1; IEC 60601-1; and the FDA guidance documents on MD software development which includes the guidance document on the development of MMAs. This analysis resulted in twenty-four traceability requirements. Additionally, as the aim of this study is to develop a TAIF which will assist with the implementation of the standards' requirements for traceability along with the best practices for implementing traceability, the twenty-three best practices identified during a literature review were included in the PRM [47].

To establish a uniformity of description of the processes in this TAM, technical report ISO/IEC TR 24774 (ISO/IEC, 2010) was employed. This report provides guidelines for process description.

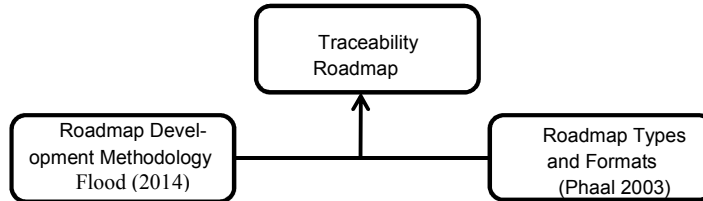
Section 6.2 of ISO/IEC 15504-2 sets out the minimum requirements for a PRM. These requirements state that the process descriptions incorporate a statement of the purpose of the process together with the set of outcomes which demonstrate successful achievement of the process purpose.

While ISO/IEC 15504-2 details the minimum requirements that a PRM and a PAM should meet, it provides no guidance on how to develop the models i.e. it does not tell you how to transform a set of requirements into a PRM or PAM. To address this, issue the Tudor IT Service Management Process Assessment (TIPA) transformation process [48] was employed as it complies with the requirements for PRMs and PAMs as expressed in ISO/IEC 15504-2.

The TAM contains the following four traceability processes: SDLC traceability process; Risk management traceability process; Change management traceability process; and a Traceability implementation best practices process. In accordance with ISO 15504, each of the four processes in this TAM contain a process description and a set of indicators, known as base practices and work products. Base practices provide a definition of the tasks and activities needed to accomplish the process purpose and fulfil the process outcomes. Evidence of performance of the base practices, and the presence of work products provide objective evidence of the achievement of the purpose of the process.

## 5.2 Roadmap Development and Format

The approach employed to develop the TR is depicted in Figure .



**Figure 3: Roadmap Development Approach**

A method proposed in [49] provides a systematic approach to roadmap construction. The first three steps of this method involved *identifying traceability requirements from the standards, grouping the requirements into goals and then separating the goals in line with ISO/IEC 15504 capability levels*. To develop the roadmap, it was not necessary to complete these steps as they had already been completed as part of the development of the TAM. The grouped goals or milestones identified in the above process are equivalent to the grouped goals contained within the TAM and so a milestone equates to a TAM process.

The fourth step involved *ordering the milestones based on the capability level and logical groups*. The four processes within the TAM are all at capability level 1 and so conferred no order priority. Thus the processes were arranged in the order in which they should be implemented within a development project. This order consists of the best practices being implemented before development begins. Then the RM trace process and CM trace process are to be implemented concurrently with the SDLC trace process.

Steps 5 and 6 consisted of *validating the generated roadmap (by industry experts) and identifying activities that can meet the identified goals*. It was considered to be more productive to alternate these two steps as the TAM processes had already been validated by expert review during the development of the TAM. Therefore, activities were identified before the roadmap was referred for initial validation by expert review. These activities are labelled as base practices in the roadmap, with each base practice containing two exemplar methods of how to implement them. For the final step the traceability roadmap underwent industry validation by implementation in two medical device software organisations.

A literature review on roadmaps revealed a taxonomy of eight roadmap types in terms of purpose, and eight roadmap formats in terms of graphical representation [50]. It was considered that the traceability roadmap would better align with the 'Process planning' type of roadmap. This type supports the management of knowledge, focusing on a particular process area. In the case of the traceability roadmap the focus is on knowledge flows that are needed to facilitate effective traceability implementation. In addition to, and perhaps more important than roadmap type, the format of the roadmap was then considered. It was decided that the roadmap should be a hybrid version entailing mainly 'single layer' and 'text' formats, with some pictorial representation. The pictorial representation is presented early in the roadmap to give the user an overview whereas text is used to describe the roadmap in detail. This hybrid version was chosen for the following reasons:

- Combinations of pictorial and text information facilitate significant improvements in understanding [51];
- The single layer roadmap is a subset of the multilayer format and is less complex. A pictorial representation can present a lot of information in a readily understandable format.

### 5.3 TAIF Validation

The approach taken to validating the TAIF is presented in Figure 4. The validation is completed in two stages. The first stage is an initial validation by expert review while the second stage is further validation by implementation of the TAIF in two MD organizations.

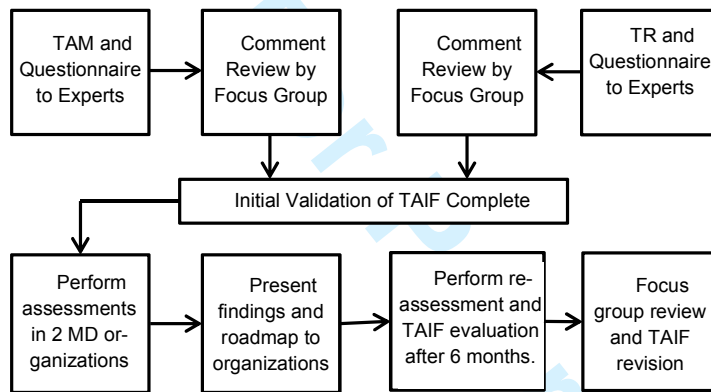


Figure 4: TAIF Validation Approach

#### 5.3.1 Initial Validation

An initial validation of the TAIF was completed through expert review [45, 46]. Thirteen international experts in areas such as ISO/IEC 15504, MD software development and processes, MD software standards, traceability research and implementation, software process improvement, and roadmap development. This validation entailed the experts being asked to review the models and then to complete a questionnaire about the models. The questionnaire was composed of questions relating to the usefulness, usability and content of the models and asked for suggestions for improvement. After the questionnaires were returned, a focus group was formed to consider all the responses and through general consensus decisions were taken as to what changes should be made to the models. After the changes were made the models were now ready for industry trial.

#### 5.3.2 Approach to Industry Validation

To further validate the TAIF a longitudinal study of two MD organizations was performed over a six-month period. An initial assessment was conducted to assess the organisations' implementation of traceability requirements and best practices. One week later a report was presented to both organisations detailing their traceability weaknesses, along with a roadmap which provided recommendations on how they could address those weaknesses. After a six-month period a follow up assessment was conducted. This follow up assessment ascertained if any improvement had been made. Additionally, the follow up assessment ascertained each organisation's perspective of the assessment experience and of the roadmap.



1  
2  
3  
4  
5  
6  
7 To collect the relevant data a mixture of interviews and document reviews were em-  
8 ployed. Interview questions to ascertain the current implementation of traceability re-  
9 quirements and best practices were developed based on the TAM base practices. To  
10 determine the organisation's perspectives of the assessment experience and the  
11 roadmap, questions were formulated which assessed their value, usefulness, and  
12 usability to the organisations. A semi-structured interview was developed based on  
13 the interview guide development strategy championed by [52]. The interview con-  
14 tained a mixture of both open and closed questions. The closed questions ensured  
15 that very specific and required data was collected e.g. the organisations implementa-  
16 tion of specific traceability requirements. The open questions allowed for the inter-  
17 viewees to reflect and give their opinion on the value, usefulness and usability of the  
18 assessment experience and the roadmap. These questions were then independently  
19 reviewed by two persons with extensive experience in developing interview guides.  
20 To introduce and implement the TAIF to both organisations the Adept method [53]  
21 was employed. The Adept method was used as it could provide a comprehensive  
22 assessment of an organizations' traceability compliance without being too onerous in  
23 terms of resources that the organization needs to provide. In addition, the Adept  
24 method also includes a follow up study to evaluate how well the implementation of  
25 the recommended changes was performed allowing for a full evaluation of not only  
26 the assessment model but also the roadmap.

### 27 **5.3.3 Sample Size and Participants**

28 This study was conducted in two MD organizations. The main factors which directly  
29 influenced the sample size for this study were;

- 30 1) Resources in terms of time and cost.
- 31 2) Perceived requirement. The authors considered this study has a narrow topic and  
32 uses in depth interviews which will limit the number of participants needed. Addi-  
33 tionally, the general population of MD software SMEs in Ireland is approximately  
34 fifty which again limits the number of participating organisations required.

35 The two participating organizations were chosen on the basis that the researcher was  
36 able to get personal introductions to management within these organizations. Two  
37 individuals from each organization participated in the interviews. These individuals  
38 best understood each organizations traceability process and were selected by the  
39 organization themselves.

40 The first organization that agreed to participate in the study, Organization A, is a  
41 small MD software company, founded in 2002, based in Ireland. The company has a  
42 total of ten employees which include one programmer, one software tester and one  
43 quality assurance person. The company produces MD software with a software safe-  
44 ty classification of B, meaning non-serious injury is possible.

45 The second company, Organization B is a small Product Development & Design En-  
46 gineering company focused on the MD and Life Science market and is based in Ire-  
47 land. The company, which was formed in 2007, employs 14 individuals with skills in  
48 mechanical, hardware and software engineering. The company are a third party sup-  
49 plier of software to MD companies and have recently been accredited with IEC 62304  
50 certification.

### 51 **5.3.4 Data Analysis**

52 The interviews, which were recorded by Dictaphone, were transcribed into Microsoft  
53 word documents. Additionally, notes were made during the interview. Due to the  
54 open ended nature of part of the interview guide, the data generated consisted of  
55  
56  
57  
58  
59  
60

1  
2  
3  
4  
5  
6  
7 long paragraphs with similar concepts in different locations of the text and with data  
8 unrelated to the study. This data was considered to be unstructured and so qualita-  
9 tive content analysis, using deductive and inductive coding, was used to structure the  
10 data into meaningful and analyzable data from which conclusions could be drawn.  
11 Categories and codes based on the purpose of the interview questions were estab-  
12 lished prior to coding (deductive). Annotating the text involved carefully reading the  
13 text a number of times, each time underlining the text that applied to any category  
14 and annotating the underlined text with the category code. While carrying out this  
15 process a number of new categories emerged (inductive). This process was repeated  
16 until no new text for coding emerged.

### 17 **5.3.5 Implementation Findings**

18 A comparison of the initial and follow-up assessments indicates that both organiza-  
19 tions had significantly improved almost all traceability processes, although none of  
20 the processes improved to a level of 100% adherence. The overriding reason given  
21 for this is resource issues (mostly time) within the organizations. These resource is-  
22 sues are clearly outside the scope of the TAIF.

23 An evaluation of the assessment experience and traceability roadmap clearly indi-  
24 cates that both organizations found them to be very useful, usable and generally  
25 providing an awareness, knowledge and expertise in traceability that they did not  
26 previously have. Additionally, both organizations considered that the assessment  
27 provided them with guidance and real direction for long term traceability improve-  
28 ment.

29 Further detail on the findings can be obtained in [54].

## 30 **6 Conclusion and Future Work**

31 This journal paper described our approach for a MMA development framework. In  
32 order to ensure that the framework addressed the true needs of the MMA  
33 development community we conducted a workshop with the experts within the MD  
34 software domain. Within this workshop there was three discussion areas. The first of  
35 these areas concerned challenges within MMA development. The second area of  
36 discussion concerned the features that should be included within a MMA framework  
37 to overcome the challenges mentioned in first discussion. The final discussion  
38 concerned the actual software development approach for the MMA framework where  
39 we looked at six different software development lifecycles.

40 Whilst the MMA framework will contain many features that were identified within the  
41 workshop by experts, due to space restrictions we have provided an exemplar of one  
42 of these features – traceability. We described the development and validation of the  
43 traceability component of the MMA framework.

44 In the future, we plan to develop the complete MMA framework to include all of the  
45 features identified in Section 4.3. Upon completion of this MMA framework we plan to  
46 validate it within the organizations that participated in the workshop. Another area we  
47 wish to focus upon is the development of an agile approach for MDevSPICE® that  
48 can be used for both MMAs and general MD software development. This work will be  
49 validated both within the agile and MD standards communities and the MD software  
50 industry.

### 51 **Acknowledgments**

52 This research is supported by the Science Foundation Ireland Research Centres  
53 Programme, through Lero - the Irish Software Research Centre (<http://www.lero.ie>)  
54  
55  
56  
57  
58  
59  
60

grant 10/CE/I1855 & 13/RC/2094.

## References

1. Mobile Medical Applications Guidance for Industry and Food and Drug Administration Staff. U.S. FDA, 2013.
2. Abrahamsson P, Warsta J, Siponen MT, and Ronkainen J. New Directions on Agile Methods: A Comparative Analysis. 25 International Software Engineering Proceedings 2003; 244–254. DOI: 10.1109/ICSE.2003.1201204.
3. McCaffery F, Lepmets M, Clarke, P. Medical Device Software as a Subsystem of an Overall Medical Device. Proceedings of The First International Conference on Fundamentals and Advances in Software Systems Integration Medical 2015; 17–22.
4. Trektere K, McCaffery F, Lepmets M, Barry G. Tailoring MDevSPICE® for mobile medical apps. In Proceedings of the International Workshop on Software and Systems Process 2016; 106-110. DOI: 10.1145/2904354.2904361.
5. FDA Chapter I - Food and drug administration, department of health and human services subchapter H - Medical devices, Part 820 - Quality system regulation. FDA, USA.
6. Directive 93/42/EEC of the European Parliament and of the Council concerning medical devices. European Commission, 1993.
7. Council directive 90/385/EEC on active implantable medical devices (AIMDD). Council Directive, 1990.
8. Directive 98/79/EC of the European parliament and of the council of 27 October 1998 on in vitro diagnostic medical devices. European Commission, 1998. Official Journal of the European Communities.
9. Directive 2007/47/EC of the European Parliament and of the Council concerning medical devices. EC, 2007. Official Journal of the European Union. Official Journal of the European Union.
10. ISO 13485: Medical Devices - Quality Management Systems - Requirements for Regulatory Purposes. ISO, 2003.
11. ISO 9001:2000 Quality Management Systems - Requirements. BSI, 2000.
12. IEC 62304: Medical Device Software - Software Life-Cycle Processes. IEC, 2006.
13. ISO 14971 - Medical Devices - Application of Risk Management to Medical Devices. ISO, 2009.
14. IEC TR 80002-1 - Medical Device Software - Part 1: Guidance on the Application of ISO 14971 to Medical Device Software. IEC, 2009.
15. IEC 82304-1: Health Software – Part 1: General requirements for product safety. IEC, 2012.
16. Guidance Document - Medical Devices and Radiation-Emitting Products. FDA, 2015. Center for Devices and Radiological Health.
17. Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. FDA, 2005.
18. Guidance for Industry - FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices. FDA, 1999.
19. General Principles of Software Validation; Final Guidance for Industry and FDA Staff. FDA, 2002.
20. Treacy C, McCaffery F, Finnegan A. Mobile health & medical apps: possible impediments to healthcare adoption. In: eTELEMED 2015, The Seventh International Conference on eHealth proceedings 2015; 199-202.

**Comment [KT1]: Reference style.** References should be cited in the text by number within square brackets and listed at the end of the paper in the order in which they appear in the text. All references must be complete and accurate. If necessary, cite unpublished or personal work in the text but do not include it in the reference list. Where possible the DOI \* for the reference should be included at the end of the reference. **Online citations should include date of access.** References should be listed in the following style:

1. Alonso P, Badia JM, Vidal AM. Solving the block-Toeplitz least-squares problem in parallel. *Concurrency and Computation: Practice and Experience* 2005; 17 (1): 49-67. DOI: 10.1002/cpe.883.
2. Reismann H. *Elastic Plates: Theory and Application*. Wiley: New York, 1988; 1-10.
3. PARAllel Total Energy Code. <http://www.nersc.gov/projects/paratec> [6 November 2004].

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

21. EC, Medical Devices: Guidance Document Qualification and Classification of stand alone software MEDDEV 2.1/6.
22. ISO/IEC 12207:2008 - Systems and Software Engineering - Software life cycle processes. ISO/IEC, 2008.
23. IEC TR 80002-3: Medical device software - Part 3: Process reference model of medical device software life cycle processes (IEC 62304). IEC, 2014.
24. Workshop. <http://www.oxforddictionaries.com/definition/english/workshop> [01/09/2016].
25. Baskerville RL. Investigating Information Systems with Action Research, Communications of the Association of Information Systems 1999.
26. McHugh M, McCaffery F, Fitzgerald B, Stol K, Casey V, Coady G. Balancing Agility and Discipline in a Medical Device Software Organisation. Proceedings in 14th International SPICE Conference 2013; 199–210. DOI 10.1007/978-3-642-38833-0\_18.
27. AAMI TIR 45:2012 - Guidance on the use of Agile practices in the development of medical device software. Technical Report. AAMI, 2012.
28. McCaffery F, Casey V, Sivakumar M, Coleman G, Donnelly P, Burton J. "Medical Device Software Traceability", in Software and Systems Traceability. J. Cleland-Huang, O. Gotel, and A. Zisman, Eds., ed: Springer-Verlag 2012.
29. McCaffery F, McFall D, Donnelly P, Wilkie FG, Sterritt R. A Software Process Improvement Lifecycle Framework for the Medical Device Industry. IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05) Proceedings 2005; 8. DOI:10.1109/ECBS.2005.13.
30. McHugh M, McCaffery F, Casey V, Pikkarainen M. Integrating agile practices with a medical device software development lifecycle. EuroSPI Proceedings 2012; 1–8. DOI: 10.1007/978-3-319-13036-1\_17.
31. Shore J, Warden S. The Art of Agile Development. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
32. Cawley O, Wang X, Richardson I. Lean/agile software development methodologies in regulated environments - State of the art. Lecture Notes in Business Information Processing 2010; 31–36. DOI: 10.1007/978-3-642-16416-3\_4.
33. Sutherland J. Future of scrum: Parallel pipelining of sprints in complex projects. Agile Conference Proceedings 2005; 90–99. DOI: 10.1109/ADC.2005.28.
34. Spence JW. There Has to Be a Better Way! Agile Development Conference Proceedings 2005; 272–278. DOI: 10.1109/ADC.2005.47.
35. Upender B. Staying agile in government software projects. Agile Development Conference (ADC'05) 2005. DOI: 10.1109/ADC.2005.41.
36. Fitzgerald B, Stol KJ, O'Sullivan R, O'Brien D. Scaling agile methods to regulated environments: An industry case study. International Conference on Software Engineering Proceedings 2013; 863–872.
37. Sumrell M. 2007. From Waterfall to Agile - How does a QA Team Transition? Agile Conference Proceedings 2007; 291–295. DOI:10.1109/AGILE.2007.29.
38. Heeager LT, Nielsen PA. Agile Software Development and its Compatibility with a Document-Driven Approach? A Case Study. Australasian Conference on Information Systems Proceedings 2009; 205–214.
39. Rottier PA, Rodrigues V. Agile Development in a Medical Device Company. Agile Conference Proceedings 2008; 218–223.
40. Rasmussen R, Hughes T, Jenks JR, Skach J. Adopting Agile in an FDA Regulated Environment. Agile Conference Proceedings 2009; 151–155. DOI: 10.1109/AGILE.2009.50.

41. McHugh M, McCaffery F, Casey V. Barriers to Adopting Agile Practices When Developing Medical Device Software. *Software Process Improvement and Capability Determination Proceedings 2012*; 141–147. DOI: 10.1007/978-3-642-30439-2\_13.
42. Gotel O, Mader P. Acquiring Tool Support for Traceability. In: Cleland-Huang, J., Gotel O, Zisman A. (eds.) *Software and Systems Traceability*. Springer, New York (2012) DOI: 10.1007/978-1-4471-2239-5\_3.
43. Cleland-Huang J, Gotel O, Hayes J.H, Mäder P, Zisman A. *Software Traceability: Trends and Future Directions*. ICSE, Hyderabad, India (2014) DOI: 10.1145/2593882.2593891.
44. Regan G, Flood D, Caffery F.M, Biro M. Assessing traceability—practical experiences and lessons learned. *Journal of Software: Evolution and Process* (2015) DOI: 10.1002/smr.1728.
45. Regan G, Caffery F.M, Daid K.M, Flood D. The Development and Validation of a Traceability Assessment Model. In: Mitasiunas, A., Rout, T., O'Connor, R., Dorling, A. (eds.) *SPICE*, vol. 447, pp. 72-83. Springer, Vilnius, Lithuania (2014) DOI: 10.1007/978-3-319-13036-1\_7.
46. Regan G, Flood D, McCaffery F. The Development and Validation of a Roadmap for Traceability. *SPICE*. Springer, Gothenberg, Sweden (2015) DOI: 10.1007/978-3-319-19860-6\_5.
47. Regan G, Mc Caffery F, Mc Daid K, Flood D. Implementation of traceability best practices within the medical device domain. In: Barafort, B., O'Connor, R., Poth, A., Messnarz, R. (eds.) *EuroSPI DELTA*, Dundalk, Ireland (2013).
48. Barafort B, Renault A, Picard M, Cortina S. A transformation process for building PRMs and PAMs based on a collection of requirements – Example with ISO/IEC 20000. *SPICE*, Nuremberg, Germany (2008).
49. Flood D, McCaffery F, Regan G, Casey V. A Critical Evaluation of a Methodology for the Generation of Software Process Improvement Roadmaps. *EuroSPI 2014* vol. 425, pp. 36-47. Springer Berlin Heidelberg, Luxembourg (2014) DOI:10.1007/978-3-662-43896-1\_4.
50. Phaal R, Farrukh C, Probert D. Technology roadmapping—A planning framework for evolution and revolution. *Technological Forecasting & Social Change* 71, 5-26 (2003) DOI:10.1016/s0040-1625(03)00072-6 .
51. Pekerti A. Augmentation of information in educational objects: Effectiveness of arrows and pictures as information for actions in instructional objects. *Australasian Journal of Educational Technology* 29, (2013) DOI: 10.14742/ajet.312.
52. Dawson C. *Practical Research Methods*. How To Books Ltd, Oxford (2002)
53. McCaffery F, Richardson I, Coleman G. *Adept – A Software Process Appraisal Method for Small to Medium-sized Software Development Organisations*. EuroSPI06, Finland (2006) DOI: 10344/2080.
54. Regan G, Flood D, McCaffery F. Research findings from an industrial trial of a traceability assessment and implementation framework. *ICSSP '16 Proceedings of the International Workshop on Software and Systems Process*, pp. 91-95. ACM, Austin, Texas (2016) DOI:10.1145/2904354.2904365.



**Table 1. MDevSPICE® Processes**

Medical Device System Lifecycle Processes	Medical Device Software Lifecycle Processes	Medical Device Support Processes
Project Management	Software Development Planning	Configuration Management
Risk Management	Software Requirements Analysis	Software Release
Stakeholder Requirements Definition	Software Architecture Design	Software Problem Resolution
Project Assessment and Control	Software Detailed Design	Software Change Request Management
System Requirements Analysis	Software Risk Management	Software Maintenance
System Architecture Design	Software Unit Implementation and Verification	
System Integration	Software Integration and Integration Testing	
System Qualification Testing	Software System Testing	
Software Installation		
Software Acceptance Support		

**Table 2. Workshop attendance figures**

Location	Attendees	Organizations
Dundalk	13	9
Limerick	10	6
<b>Total</b>	<b>23</b>	<b>15</b>

**Table 3. MMA development challenges in A-Z order**

<ul style="list-style-type: none"> <li>• Adoption/Uptake</li> <li>• Architecture</li> <li>• Automated testing tools</li> <li>• Best available development toolset</li> <li>• Connectivity, security and the validation of them</li> <li>• Data flow</li> <li>• Dependency on OS, hardware and network</li> <li>• Development lifecycle and app distribution</li> <li>• Hardware/Platform eg., Phone/Tablet,</li> </ul>	<ul style="list-style-type: none"> <li>• Overhead of adopting required standards and compliance</li> <li>• Platform selection and validation</li> <li>• Providing essential updates when offline</li> <li>• Release and version management</li> <li>• Synchronisation between different devices for apps</li> <li>• Time to market</li> <li>• Understanding what SOUP is</li> </ul>
--	---



iOS/Android	<ul style="list-style-type: none"> <li>• Usability and Testing on different versions of app hardware and platforms</li> </ul>
<ul style="list-style-type: none"> <li>• Knowing if app is a component or a complete MD</li> </ul>	

**Table 4. Model Comparison**

Software Development Lifecycle	Regulatory Compliance and Documentation	Time to Market
Waterfall or V-model	One standard at a time – very lengthy process	Very slow; Resource intensive
MDevSPICE <sup>®</sup> (waterfall or V-model)	Shorter process – all standards easily put in place	Slow; Resource intensive
V-model with agile practices	One standard at a time; Challenge with documentation	Bit faster; Full lifecycle covered
V-model with agile practices & MDevSPICE <sup>®</sup>	All standards easily put in place; Challenge with documentation	Faster and all standards are covered
Reduced V-Model with agile practices	One standard at a time; Challenge with documentation	Faster; Focused on essential activities
Reduced V-Model with agile practices & MDevSPICE <sup>®</sup>	All standards easily put in place; Challenge with documentation	Faster; Focused on essential activities

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

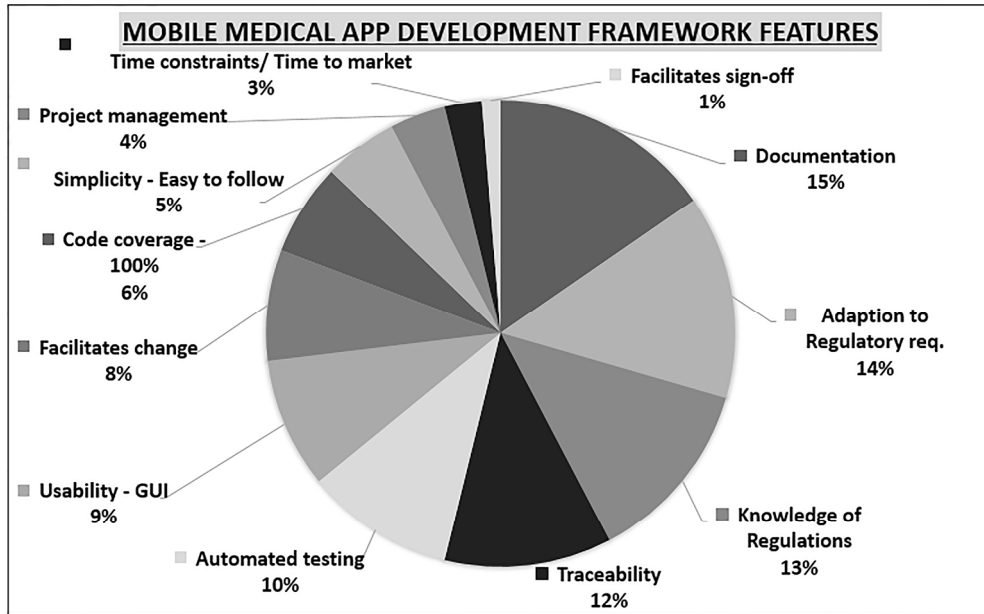


Figure 1 Mobile Medical App Development Framework Features  
Figure 1  
127x79mm (600 x 600 DPI)

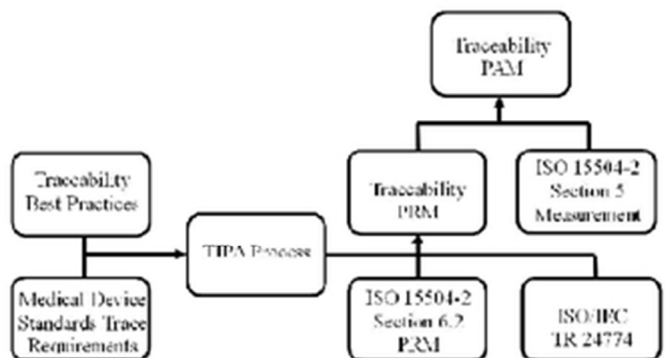


Figure 2 TAM Development Approach

Figure 2

14x8mm (600 x 600 DPI)

Peer Review

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60



Figure 3: Roadmap Development Approach  
Figure 3  
4x1mm (600 x 600 DPI)

For Peer Review

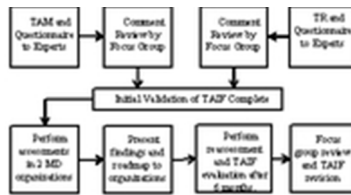


Figure 4: TAIF Validation Approach  
Figure 4  
7x4mm (600 x 600 DPI)

For Peer Review