

FDA Digital Health – Software Precertification Pilot Program

ROLE OF AGILE QMS BUILT AROUND SDLC

MAY 29, 2018

BY TARANJIT SAMRA

Introduction

As a former pilot participant for FDA Digital Health Pre-Check Program, I applaud FDA's recent efforts to streamline how it regulates software through the exploration of new paradigms, as various industries (medical devices, biotech, hi-tech/software) converge to develop a wide-range of healthcare software products such as SaMD (Software as a Medical Device), mobile (medical) apps, medtech applications, CDS (Clinical Decision Support), HealthIT and bioinformatics, and technologies such as artificial intelligence, machine learning and cloud. It's one-of-a-kind (if not once-in-a-lifetime) opportunity, both for the FDA and industry alike, to work together in developing a framework which promotes innovation while ensuring regulatory compliance.

Agile QMS (Quality Management System)

Since the launch of the program in 2017 (including [*Developing a Software Precertification Program: A Working Model*](#) more recently in April 2018), FDA hasn't mentioned the term QMS (yet), but virtually everything it has indicated (excellence principles, organizational excellence, culture, processes, systems, maturity models, KPIs, metrics etc), requires a binding framework. How else are we going to execute and demonstrate so many attributes as a working model? It may be deliberate (and rightly so), since the use of the term QMS may shift the focus back on the traditional principles of medical device industry, rather than exploring other methodologies, models, benchmarks and best practices, especially for the new entrants from outside the industry.

For example, in the very first question (1.1) of Component 1 (Excellence appraisal and precertification) of [FDA's working model](#), it asks: *How might an existing excellence or maturity appraisal framework used by a SaMD manufacturer be leveraged to demonstrate an organization's performance and success as outlined by the five excellence principles?* Then in 1.2 it mentions: *organization's objective Key Performance Indicators (KPIs)*. This framework could be anything, and terminology is not important in this exploratory phase of the program (as FDA notes several times in the document: *This is an important first step to help us explore and evaluate the program model to inform how we establish the Precertification Program. Once we determine the elements for a future Precertification Program, we will then consider the appropriate mechanisms for establishing the program, including FDA's current statutory and regulatory authorities. While the FDA has not yet determined the appropriate method*). Back in the industry, it's not that traditional medical device industry has done a fine job with its own QMS, especially around anything software. Perhaps the reason why QMS remains unmentioned in the discussion, so far. Whatever the reasons may be, for the sake of the discussion here, we'll refer to this binding framework as "QMS".

The need for an **agile QMS**, to match the agility of today's software development and deployment, is equally critical for medical device companies and new entrants, but for somewhat different reasons. The traditional medical device companies have to do something different about their QMS as it talks to the software, while new entrants have to start thinking about this concept (and potentially upcoming regulatory/QMS requirements), around their current excellence principles and implemented methodologies. A number of comments by the member from the industry in the [FDA docket](#) has noted various disadvantages small, start-up companies face, especially those from the other industries. But implementing an agile QMS around their existing best practices may actually be an advantage for them (if they understand and interpret the concept correctly), as compared to large medical device corporates. Even latter may consider piloting such an agile QMS at their respective business "units" (as FDA noted in question 1.8), where such software technologies are being developed.

Collaboration vs Controls

In breaking away from its traditions, it's quite noticeable that FDA hasn't used the word "controls", which is at the core of traditional QMS. The traditional medical device industry, went after the controls so much that they forgot "collaboration", especially around the agility at which the software is developed and deployed today. An agile QMS must support the collaboration among those who develop, test, support and use software throughout its lifecycle. Controls will come, may be not just yet.

Data vs Documentation

Also very noticeable is the lack of the word "documentation" by the FDA in this program thus far, and it is about the "data". The traditional medical device companies have operated the other way around. The focus has always been on the documents (e.g. DHF – Design History Files) which most often doesn't contain any actionable data, to make real-time and meaningful decisions. And it's anyone's guess if the documentation itself contains its most current form.

Medical device companies are left with the documentation which lacks collaboration, and a QMS completely detached from the realities of SDLC (Software Development Life Cycle). We've failed the idea of QMS (for software anyway), and it's time to try something a little different.

Implementing agile QMS?

So where do we start implementing an agile QMS? We start exactly where the business takes place, i.e. around software development methodologies and SDLC tools. And the essence of agile QMS is "collaboration" (to which controls can be applied, later) and "data" (from which documents could be retrieved/created, on-demand). That way everyone can be happy and we keep the innovation going while keeping regulatory compliance very much in our minds.

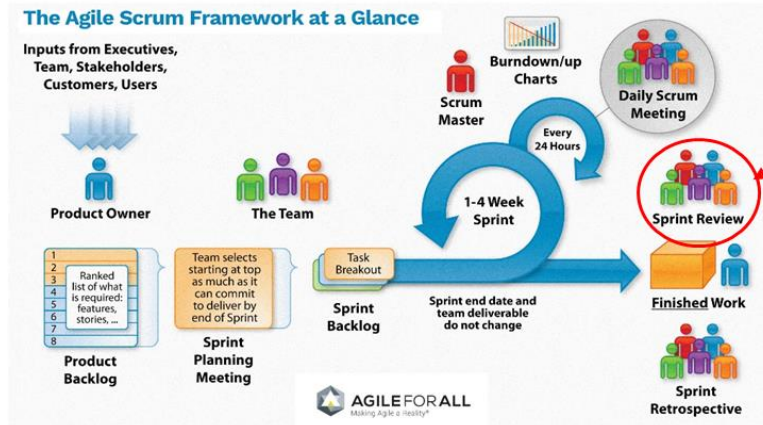
Step 1: Using Agile/Scrum methodology

I've personally worked with [agile/scrum methodology as integrated with SDLC tools for use in medical device software](#) since 2008. The device industry has come a long way since, and what

seemed like a possibility then, is a reality now. But if you're still contemplating whether or not to use agile/scrum/iterative method for SaMD (and other software products and technologies in healthcare), it may itself be a good first step, to inherit this method as the DNA of an agile QMS. If we notice closely, agile/scrum method actually supports a wide-variety of activities and features indicated by FDA in their precertification working model. And previously, FDA had worked closely with the industry during the development of AAMI TIR 45:2012 which provides the guidance on the use of AGILE practices in the development of medical device software. For example, agile method focuses on quality by ensuring correctness of the product, customer needs via product effectiveness through customer collaboration, productivity through improved efficiency, speed (and lower costs), and predictability through improved estimation and planning. Furthermore, Agile principles promote the importance of delivering customer value through frequent customer interaction/validation regarding usability and safety of the product, ability to manage (business and) safety risk through team collaboration by focusing on a robust and safe design, doneness criteria to ensure items related to safety risk management are completed, and process to reflect and adapt to constantly improving the quality (and efficiency) of work at regular intervals.

Now only time will tell, if the industry and FDA would reach a consensus in the precertification program where FDA also becomes a stakeholder (or customer) in this agile software development (in some meaningful way), but this feature is available at the very core of this method. Some possibility like that could answer several key questions in the FDA working model, for example in Component 3: Stream Streamlined premarket review process, FDA imagines an initial, automated part of the review (3.4), or technologies to support bi-directional communication (3.5), or a premarket review without requiring a premarket submission by accessing and interactively reviewing information internal to the precertified organization about the SaMD (3.7). Such avenues could be real-time, proactive opportunities (e.g. during the sprint review/demo before the initial release of SaMD, and subsequent major iterations), or off-line, reactive review of dashboard and metrics (see SDLC section for details) e.g. for minor releases. However, this level of involvement could be seen too intrusive and

unnecessary/frequent by the industry, but can be mitigated by sharing the information or having FDA's involvement at mutually comfortable levels. It's also quite practical that the SaMD iterations (that are externally deployed) wouldn't be as frequent (question 4.13) as in a pure agile/scrum method (e.g. 3-6 months instead of 2-4 weeks).



Is FDA a stakeholder/customer for initial launch, and subsequent major iterations?

Diagram 1: Pure Scrum (courtesy: [Agile for All](#))

(Note: Requires customizations for SaMD; for demonstration purposes only)

Step 2: SDLC Tools – enabling agility and automation of design controls

Once the agile/scrum/iterative method is in place, with needed customizations for SaMD (and other software products as appropriate), the focus of agile QMS construction shifts to the automation of as many of the underlying workflows (e.g. data monitoring/collection/analysis, design controls process, and DHF generation) as possible. Quality and regulatory professionals can work together with software developers in their organizations, and use their expertise to develop/customize SDLC tools (or deploy those available OTS – Off-The-Shelf) for this purpose.

There are agile SDLC tools in the marketplace, for example [Atlassian](#) and its numerous plug-ins, that support agile methodology throughout SDLC (planning, collaboration, user stories, requirements, coding, code-reviews, unit/regression/automated tests, test management, traceability, build management, continuous integration etc). SDLC tools like these enable agile/scrum software development method, and can be used as a foundation of agile QMS to semi-automate data, documentation and various related workflows, as the industry engages

with FDA in the precertification program, and beyond (i.e. to fulfill FDA QSR requirements for marketed SaMD). These SDLC tools, while enabling software teams to deliver an agile software product, also provide insight into the status of the project by permitting the collection of quantitative data related to quality, reliability, maintainability and performance of software.

The built-in data monitoring/collection/analysis views, and pre-programmed templates (e.g. in Atlassian/[Confluence](#)) for various deliverables and documentation, can supply on-demand (e.g. on a push of button) dashboards with desired KPIs and metrics, and DHF documentation, which may be shared internally and externally as needed. This type of implementation will be key in the FDA software precertification program, as the working model (Component 3: Streamlined premarket review process) reflects FDA's current thinking as: *In a streamlined review, the FDA interactively reviews supporting information. For instance, this review may be conducted through screensharing, access to development environment, and testing logs – using freeform audit of test results.* Furthermore, the logistics around major or minor releases (FDA questions 3.9, 2.5, 2.6, 2.7) can also be addressed by employing SDLC, as the tools are automated to capture certain outputs (data and documentation) regardless of whether the release is major or minor.

Workflows that can't be automated using SDLC tools alone, can be implemented through other collaboration tools. For example google docs (or MS Word with Office 365) can be used for design docs or requirements gathering. Working up this way, the design controls basis of our agile QMS can be largely instituted in place. Then the other elements of agile QMS (doc controls and post-market to name a few) can be seamlessly integrated.

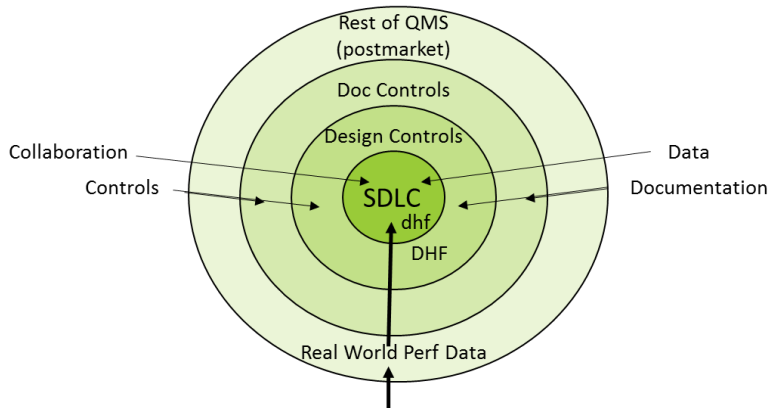


Diagram 2a: Agile QMS: Built around SDLC

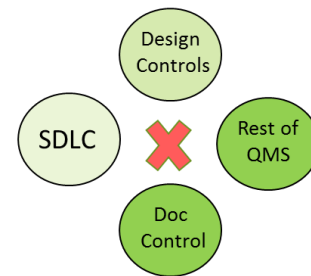


Diagram 2b: Traditional QMS: Built without SDLC

Step 3: Document Controls: Are we controlling quality content today without the collaboration?

While document controls may or may not be interesting at this point in the precertification program, eventually doc control will become the back-bone of our QMS. The traditional medical device industry has completely disjointed interfaces between design lifecycle (including DHF documentation therein) and doc control, the effect of which is compounded many-folds in the agile software development world. In fact, if a key reason is cited as to why agile/scrum method won't work for SaMD, it's because the "documentation" can't be produced (as well as reviewed and approved) at that velocity. On the contrary, this reasoning actually makes a case for an agile QMS to be built around the SDLC tools.

Now, if the DHF documents can be generated within SDLC tools (as discussed earlier), why can't they also be edited, reviewed and approved within the same ecosystem (i.e. SDLC)? Can we build a doc control function within SDLC tools? Perhaps. For example, Atlassian provides plugins like [Comala Workflows](#) which can be used for reviews and e-signature approvals. But wait. Just when we think we may be trying to solve an age-old problem related to doc control systems, the topic of "validation" of such tools is another deal breaker in regulatory circles. Well, if a workflows works for you, by all means go ahead and validate if necessary. In the current least burdensome environment supported by FDA Pre-Check Pilot team, such issues are not noted as a concern.

If for any reason, doc control functionality can't be built within SDLC tools, it's more the reason that a stand-alone doc control system provides an agile and seamless interface to the SDLC, so

that the native deliverables/documents from SDLC environment flow seamlessly from collaboration to controls. Note that any breakpoints in this workflow will cause ripple-effects (like a fender-bender on a highway during rush-hour), thereby making the viability of an agile/scrum development methodology for SaMD impractical. So as we implement doc control functionality for agile QMS for SaMD, let's build it around/within the SDLC tools.

Step 4: Extend Agile QMS into Postmarket Surveillance

Ultimately, post-market requirements such as complaint handling, CAPAs, and reporting requirements may pertain to us (if not already), but even now there are a number of key interfaces that we could incorporate in our agile QMS for the precertification program success. Later on, we can continue to build on the same by adding more controls, but never disjointed from the SDLC tools.

The Component 4 of FDA Software Precertification Program (Real world performance) expects us to integrate the following types of Real World Performance Data (RWPD) into the agile QMS (and SDLC tools): Real World Health Data (RWHD), User Experience Data (UXD), and Product Performance Data (PPD). As discussed earlier, agile/scrum methodology supports such type of data on product effectiveness, usability, safety risk management and post-launch monitoring, through the collaboration with customer and various stakeholders, which when automated using SDLC tools, makes this requirement very plausible. Furthermore, as the program matures, the RWPD can be captured within DHF and risk management files, supporting design and risk management activities throughout the lifecycle of the product. Lastly, it would be mutually beneficial for all industry participants if some of their Real World Health Data (RWHD) can be used in the public domain to the extent feasible, for various diseases and modalities.

A phased or preliminary market authorization (question 3.1.6) also makes a lot of sense, especially for new entrant start-ups. A limited pilot/beta launch (e.g. 5-10 sites or implementations), depending on the risk and type of device, will allow much needed RWPD gathering sooner, to determine whether or not the conditional market authorization (or

precertification status) is confirmed or withdrawn. It may also help determine whether premarket clinical performance is necessary to assess SaMD safety and effectiveness (3.8), or to help develop scientific rigor in analytic methods (4.6). Similarly an upgrade/downgrade criteria can be established between Level 1 and Level 2 pre-certified manufacturers.

Conclusions

As FDA Software Precertification Program gets under way, it's the right time for the industry to think how to best implement an agile QMS, a foundation which not only supports their needs of today but also scales-up for the upcoming needs of tomorrow. Such an agile QMS is built around the preferred (by software industry) SDLC methodologies (e.g. agile/scrum/iterative), and using agile SDLC tools (to semi-automated workflows including data monitoring/collection/analysis, and documentation in the future). The doc control and postmarket QMS are built around the SDLC tools (not the other way). An agile QMS is implemented today for collaboration, to which controls can be applied in the coming months as needed, to formalize a full-QMS for the likes of SaMD (FDA QSR/ISO 13485).

Next topics: *Continual* Integration (CI) for SaMD

As FDA Software Precertification Program proceed further and seeks ongoing public comments in the coming weeks, we'll discuss whether continual (if not continuous) integration is possible for SaMD, and other such software products and technologies.