Deloitte.

Next Step for Agile

How medical device manufacturers can improve their agile setup

MAKING AN IMPACT THAT MATTERS since 1845

Going Agile in the Medical Device Industry

How to achieve a value-driven process design and mindset in a regulation-dominated environment

Most medical device manufactures are either experimenting with agile practices or have already adopted them, though playing "agile by the book" doesn't seem to be having the desired effect in a regulated environment. To strike the right balance between agile, efficient engineering and meeting compliance and documentation requirements, the smartest answer may not be building complex, hybrid process landscapes with parallel workstreams for engineering and regulatory. Thanks to our vast experience in this market, we can give you the support you need to adapt existing agile process models to the specific needs of medical device manufacturers.

A value-centric, agile approach to engineering

Modern medical devices are often designed to connect to the Internet, hospital networks and other medical devices. This connectivity has the potential to improve healthcare and help healthcare professionals give their patients the best treatment.

Unfortunately, the engineering practices at most of today's medical device manufacturers do not meet the increasing demand for value-driven and customer-centric solutions. Traditional product development lifecycles (PDLC) focus on the "big bang" delivery of a large, standalone products more focused on compliance and documentation requirements.

In the modern context, engineers use digital applications or supporting products to convert standalone medical devices into smart, integrated solutions that are much better equipped to meet patients' needs. These add-ons are often developed independently of the standalone device, sometimes even using completely different procedures, tools or even teams.

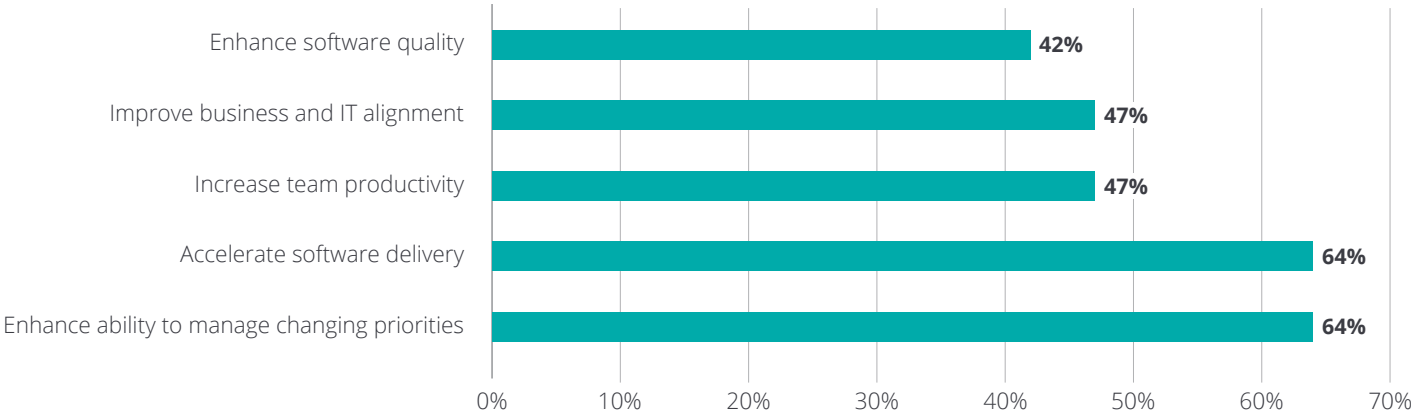
Agile practices – which have had huge success in other industries – promise to make the development lifecycle more value-centric and to improve harmonization and integration of components developed separately to form a single solution.

Seventy-five percent of executives say their agile teams perform better or significantly better than traditional teams[1]. For years, the State of Agile Report[2] has not only cited flexibility but also better performance and quality as the top reasons to implement agile/lean methods.

[1] Bain Healthcare Practice Agile Survey 2019

[2] digital.ai, 15th State of Agile Report (multiple responses to this question possible)

Fig. 1 – Top 5 reasons to adopt Agile

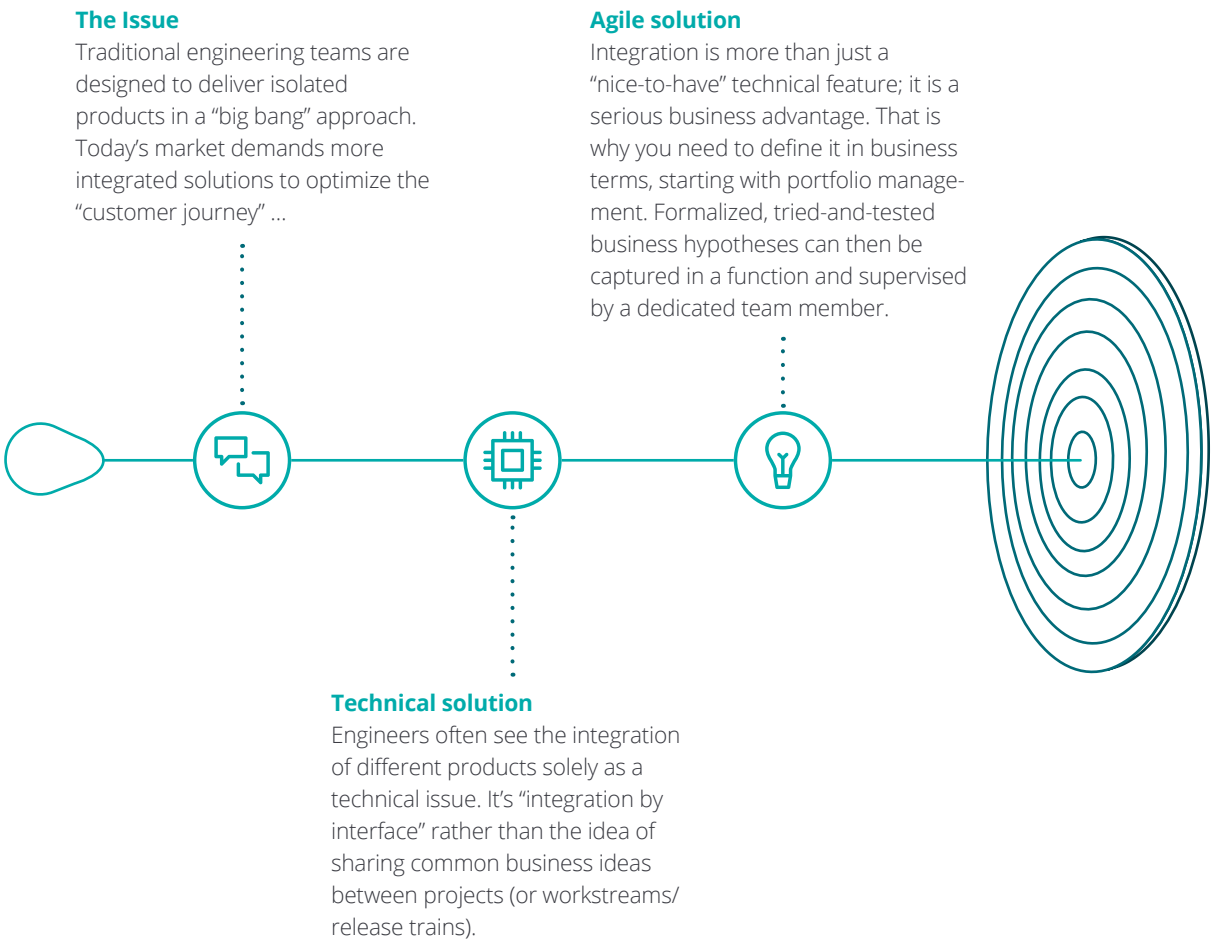


Challenges in the medical device industry

In regulated environments, developers may use the term "agility" to refer to only the technical aspects of the agile universe. Process is king in these environments, and "agile" for some may mean simply following a different development process. The whole concept ends up relegated to just another type of "project management", with a focus on requirements written as user stories and regular daily meetings. Certain regulatory tasks like risk management, cyber security or usability engineering are not even included in the new process landscape but run in parallel.

While there may be some advantages to working this way, it will not allow you to leverage the full power of an agile culture or take your enterprise closer to adopting a broader, more customer-centric solution.

Fig. 2 – Smart approach building medical ecosystems



Implement sound agile processes and events

The standard approach to going agile is to train the team, install the process by the book and start working. It is up to so-called scrum masters to manage the shock of "jumping into the cold water" and use meaningful retrospectives to mitigate the impact. Over time the team gets better, more used to the process and more efficient. Unfortunately, the reality in regulated environments is different from "standard" agile practices. Medical device manufacturers face the following challenges when it comes to implementing the technical aspects of the agile method (roles, activities, events):

Adapt agile to the regulated environment

To establish/maintain a working medical device quality management system with design controls that meet ISO 13484 chapter 7.3.9, manufacturers have to adapt the standard agile process models. Regulations like Patient Risk Management (ISO 14971), Development Life Cycle (IEC 62304 or IEC 81001-5-1) or Usability (IEC 62366) call for specific procedures and specialists. There are no Risk Manager, Validation Manager or Regulatory Affairs roles in a standard agile team, so it is up to them to find the tasks and tools they need to contribute to the product development process. Systems engineering, safety testing, summative evaluation and other activities must also be built into the process. This requires an interim step to adapt existing standard agile practices to the specific needs of a regulated medical device development lifecycle.

Extend agile to the enterprise level

Enterprises that develop different products in isolation without a conclusive, value-generating solution in mind may need to adopt lean portfolio management (LPM), which aligns strategy (expressed as strategic themes) with dedicated engineering tasks (expressed as features). LPM will shift your enterprise towards product-oriented management and allocate funds based on business outputs and incremental results driving the solution rather than the outcome of an isolated project.

Implementing LPM is a good starting point to organize your enterprise along value chains. You can find more information in Deloitte's Point of View on Agile Portfolio Management.

Fig. 3 – Adding medtech specifics to standard agile practices

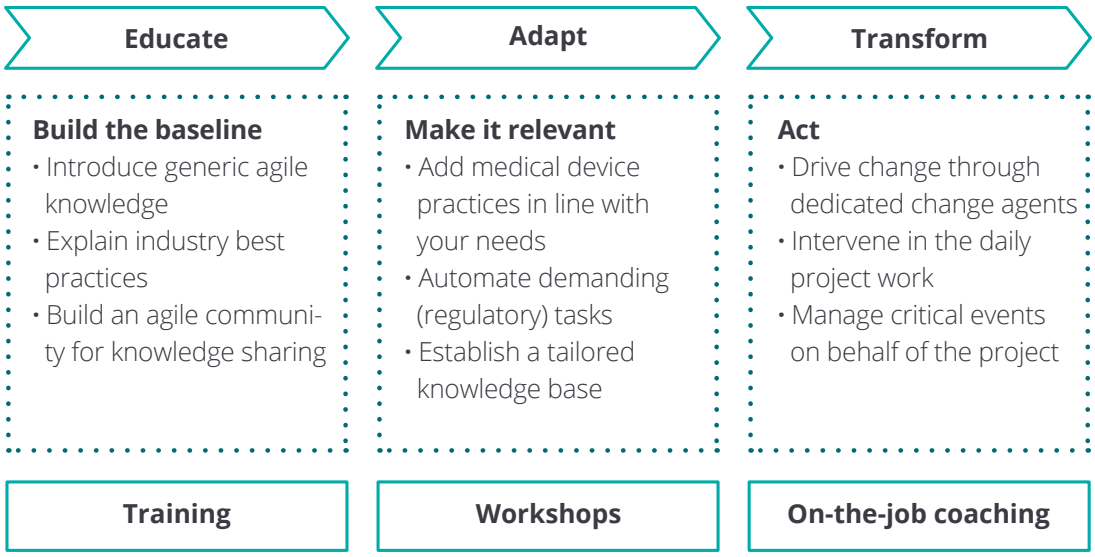
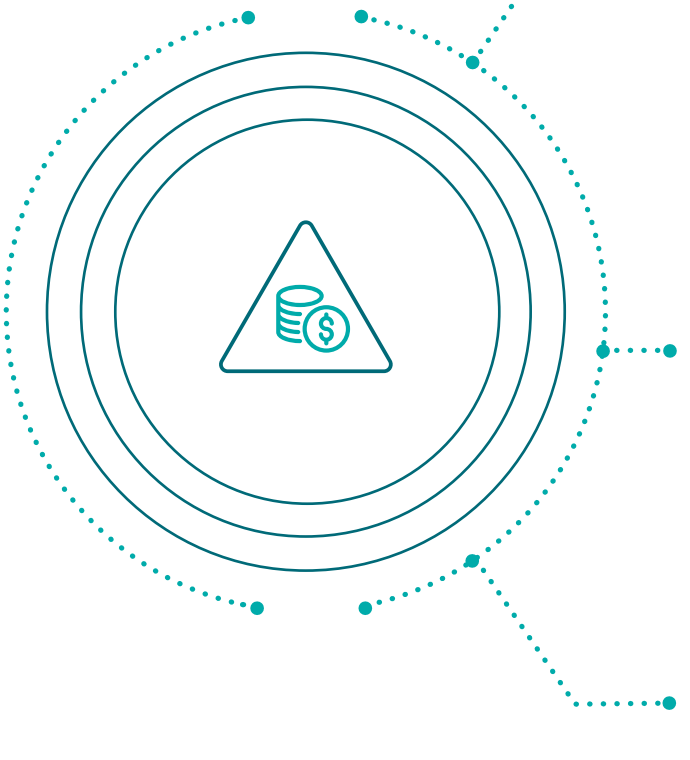


Fig. 4 – Adding the business perspective



Strategy and Investment Funding

- Aligns the portfolio with the broader enterprise strategy
- Allocates funding to the right product features
- Ensures engineering teams deliver on time and on quality.

Agile Portfolio Operations

- Coordinate and support decentralized project execution
- Align different value streams to achieve the overall product vision
- Support execution and enable operational excellence. This approach sees portfolio management as more than a “once in a year” activity that dictates project ideas and cash to the execution teams.

Lean Governance

- Forecasts and manages the budget dynamically using data taken directly from the market and value streams
- Measures portfolio performance and manages compliance on an ongoing basis

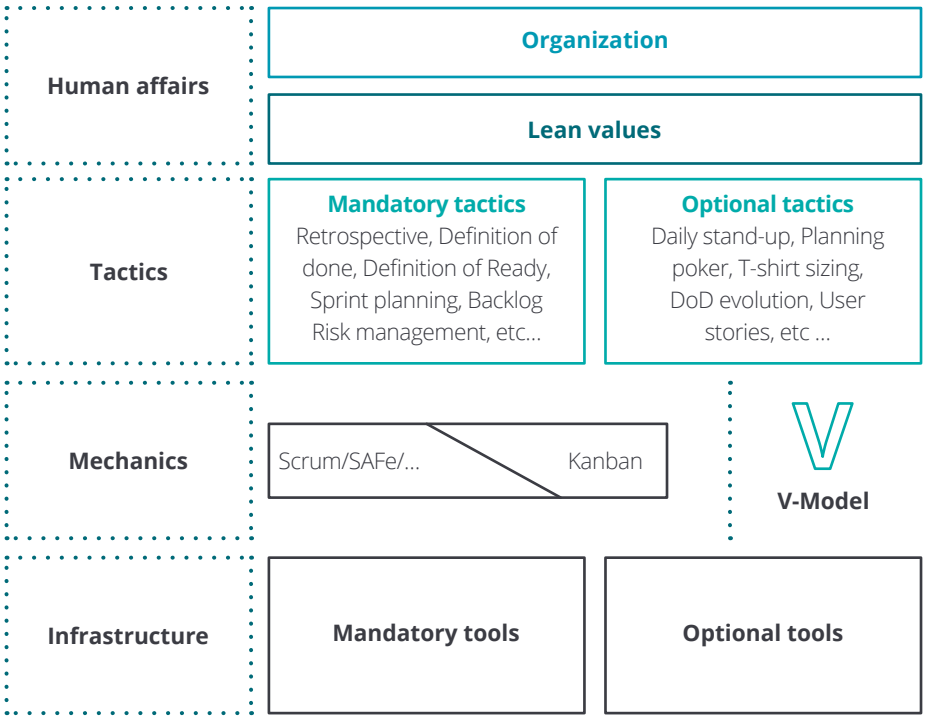
Building interconnected medical devices or even complex medical ecosystems without managing business value and dependencies on portfolio level is impossible. Everything starts with Lean Portfolio Management or a similar technique.

Give the workforce the guidance they need

In regulated environments, many companies only document their procedures (SOPs) to comply with regulations – everything else is left to the engineers. The resulting centralized quality management system lacks detail and guidance for the engineering workforce and is not seen to be the repository documenting the engineering way of working. This lack of a "marketplace" for instructions means the product lifecycle has multiple local implementations and interpretations, some of which are incomplete, while others lack efficiency. Building a community of practice can help, but it is important to have a forum where participants in each community can document and communicate new ideas and how-tos.

Development teams need to promote their tactics – the practices that make the framework run – and make them easily accessible, providing clear explanations of their value as well as training in their use. All of this information should be available in a single repository with the meta-data on purpose and value as well as traditional process elements, which some projects might still be using in a V-model approach – and for good reason.

Fig. 5 – Linking culture to technology



Adopt regulatory practices in a smart way

Even though most regulations, guidelines or standards do not explicitly ask for a waterfall approach, there is a need for a stable, "frozen" information base for some regulatory activities (e.g., a list of requirements, hazards, etc.). Frequent changes in all this information are inevitable in an agile setup and it is important for you to properly prepare your activities to deal with this. The following are examples of essential activities in a lifecycle that need to be adapted to the agile way of working:

Development plan

IEC 62304 calls for a development plan. That may seem like a bit of a contradiction for agile teams that pride themselves in their self-organized, empowered way of working. However, once you have established a repository of agile activities and practices (see above), you can simply point to this and post additional meta-information for the project, such as:

- A timeline and certain major milestones
- Assignment of staff to different roles
- Deviations from the standard development process described in the SOPs

Your plan should establish and document the management roles that require specific standardized work instructions within the agile setup. There should also be a mechanism in place to align the results of the retrospectives with subsequent updates to the project development plan. Ultimately, the plan should function as a living document and a record for your retrospectives.

Design input documentation

An easy way to better understand design input is to look into ISO 13485, which refers to the following types of requirements:

- Functional (7.3.3a)
- Performance (7.3.3a)
- Usability (7.3.3a)
- Safety (7.3.3a)
- Statutory/Regulatory (7.3.3b)
- Outputs of Risk Management (7.3.3c)
- Previous and Similar Designs (7.3.3d)

The "Design Control Guidance for Medical Device Manufactures" is old but still very informative and a good source for the US regulations.

Regardless which regulatory framework governs your processes, it is obvious that the user-centric backlog established for agile practices will not satisfy every regulation. User stories in an agile setup are more for describing specific features from the user perspective than they are for meeting the formal requirements for design input. You need to introduce activities to convert the user stories into meaningful specifications at the right level, available at the right time and stabilized as well as approved in signed documents. The FDA's "General principles of software validation" is still valid and requires manufacturers to perform a review before implementation to determine whether the requirements are complete and appropriate. This is easy to implement as a "definition of ready" statement, but it needs to be formally documented.

Tool support as well as checkpoints (e.g., definition of ready, definition of done) are essential to avoid the costly reverse engineering of formal requirements once the incremental phases are complete.

Non-functional requirements

In an agile world, developers often see non-functional requirements as design constraints and document them in either dedicated user stories, acceptance criteria of "functional" user stories or the definition of done (DoD). Due to the agile way of working, these requirements may undergo frequent changes. The agile method does not know about formal documentation and traceability process nor are requirements subject to a formal, documented testing process. In the medical device development lifecycle, non-functional requirements are part of the design input and may even be linked to patient risk management, so you must manage them appropriately.

It is difficult to manage certain aspects of the definition of done as if they were formal non-functional requirements, particularly those that were improved in group discussions during a retrospective.

Threat modeling

As outlined in a position paper by Team NB[3], threat modelling is a cornerstone of good cybersecurity design. It is important to develop threat models, not only to satisfy the regulatory paperwork but as an integral part of your design activities. While cybersecurity standards like the IEC 81001-5-1 provide for multiple potentially acceptable methods, regulators usually ask for STRIDE because this model makes it easy to check for completeness. Unfortunately, you have to input your data into STRIDE as a list of requirements, interfaces, etc., which are subject to frequent changes in an agile environment. That's why it is so important to have a mechanism in place to handle these changes as well as update and complete the threat model throughout the agile project.

Formal architectural design

The purpose of architectural design software (IEC 62304, 5.3) is to create a design that delivers on the requirements and allows for verification against them. While basic agile methods (e.g., scrum, XP) leave

[3] see https://www.team-nb.org/cyber-security-position-paper/

the architecture to the agile team, scaled agile methods call for a dedicated team. None of these methods ensures the proper documentation and the necessary traceability between software requirements and software design. As soon as you adopt a formally documented architectural design, it is vital to keep it up to date. There needs to be a mechanism in place for agile teams to request ad hoc changes as well as a governance structure to identify and evaluate the impact these changes have on other teams, patient risk analysis, segregation of software units, test automation, etc. While dealing with the impact of changing requirements and refactoring, it is important to keep in mind that there is a mandatory design review at the end of the software design process to ensure that the design is correct and accurate before implementation. This goes into the definition of ready and – again – demonstrates just how important this checkpoint is.

Formal detailed design

Once the agile team has control over the architectural design, it is up to them to decide on the detailed design. The purpose of this step (IEC 62304, 5.4) is to provide a design that is detailed enough to allow for coding and testing. Delegating this step to a centralized architecture team would damage the agile principle of the empowered

nical rules and a governance structure in place to deliver a compliant detail design (at least for class C software). It is important to establish traceability between the detailed design and the specifications and architectural design. Complexity may be an issue if the agile team manages the specifications and architectural design in a scaled agile setup. The team needs to set up interfaces to the people responsible for requirements (product manager) and architecture (solution architect), making sure they strike the right balance between empowering the agile teams and supplying stable specifications and architectural design documents in the required format.

Patient risk management

In a legacy, waterfall-type environment, patient risk management is based on a stable, frozen state of the device documentation. Going agile opens the door for a continuous quality/risk approach. So, it is important to actively manage and update the risk management files according to ISO 14971 throughout the agile development lifecycle. A good place to start is with the software risk class, the definition of the "level of concern" or the identification of software items (including SOUP) that could contribute to hazardous situations. Requirements and architecture are the key inputs for these activities, but as both are

(e.g., definition of done) is key when it comes to managing change. You need to keep updating the risk rating and mitigation actions as well. This will help avoid a "big bang" at the end of the project, when the team tends to review and improve risk mitigation actions in a rush.

Patient risk management often requires agile teams to consult with external experts or interact with other agile teams. To do so, you will need an established automated process and a dedicated team member at the product manager level.

Formative evaluation

Formative tests are any usability tests performed during the development process, and they generally take place during the review process. Unfortunately, there are no formal standards regarding the planning or documentation of these activities. The most important reason for documenting formative usability testing is to identify which user interface designs failed and why, so that future design teams can learn from your mistakes. To fit into the agile way of working, there is also a certain amount of compliance paperwork involved in formative evaluations.

Traceability

In an agile environment, where backlog

Download the full paper here:

Contact

Dr. Carsten Heil
Director
Risk Advisory
Tel: +49 151 19175210
cheil@deloitte.de

Deloitte.

Issue 02/2023