

WHITEPAPER

PROSTEP AS A PARTNER FOR AGILE PLM DEVELOPMENT

Companies in the manufacturing industry face the challenge of responding flexibly to new market and customer requirements. Their IT organizations and PLM landscapes must also meet this challenge. More and more companies are therefore adopting agile approaches and need development partners with the relevant expertise. This white paper describes how PROSTEP supports them in the use of agile methods in the PLM environment.

TRADITIONAL

METHODOLOGY

AGILE

METHODOLOGY





TRADITIONAL

METHODOLOGY

AGILE

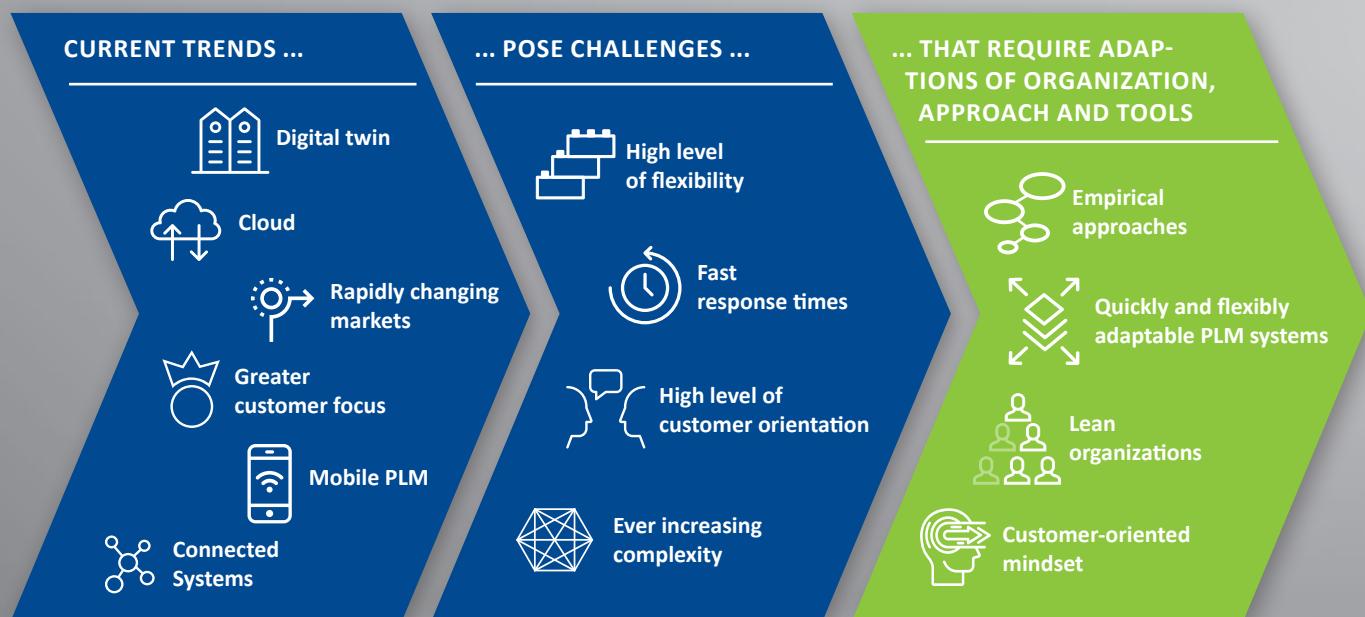
METHODOLOGY

Content

Preamble	2
Increasing dynamics in market requirements	3
Typical challenges on the way to Agile	4
Agile approaches are becoming state-of-the-art.	6
Agile billing models.....	8
The role of PROSTEP	10
Summary.....	11
Our range of solutions	11

Preamble

Rapidly changing markets, greater customer focus, ever-increasing complexity, and increasingly connected systems are creating new challenges for companies and their PLM landscapes. However, the existing IT systems and established development processes that have evolved over time are difficult to adapt to new challenges. Therefore, a growing number of companies are introducing agile process models but are at different levels of introduction. Scrum is the most common approach. For larger projects Scrum of Scrums, Nexus, LeSS and SAFe are well known scaling models. Agile billing models such as Agile Fixed Prices help to redefine the customer/supplier relationship. PROSTEP is a partner with a wealth of experience in using agile methods in PLM environments and is therefore able to adapt flexibly to specific customer's needs.



Increasing dynamics in market requirements

Companies in the manufacturing industry must be ready to quickly respond to changing market and customer requirements. Customer focus is becoming more and more important in market success. In an increasingly digitalized world, customers do not only expect high-quality products but also connected service offerings, with new digital services and service-oriented business models. Today's car owners expect to receive new vehicle features by over-the-air software updates or want to be able to control vehicle functions via their mobile phones. They expect their service repair shops to always have a precise picture of their vehicle's condition. At the same time, the growing importance of software and electronics not only increases the complexity of the connected systems, but also increases the inter-dependencies and dependencies on their environment. Making all these dependencies transparent and manageable is one of the primary goals of product lifecycle management (PLM) and the underlying PLM systems. At the same time, PLM systems must be able to quickly and flexibly adapt to new market requirements to avoid missing new market opportunities.

The requirements from the business- and development-departments are also constantly growing. Customers increasingly expect their PLM solutions to support new approaches, such as model-based systems engineering (MBSE), and the virtual validation of system functions using a combination of simulation techniques. They want to be able to access information from anywhere in the world and where appropriate using mobile devices. They want to be able to visualize this information using the latest technologies such as virtual and augmented reality. Digital information should be available throughout all processes and made available to customers and development partners in a reusable way. PLM systems provide the backbone for doing this, and they must be ready to fulfill future requirements today.

Also, external constraints such as standards and laws are constantly changing. For example, in safety-critical systems, traceability of development steps and deliverables is essential. Requirement specifications must be linked with their verifications and validations, and all this must be captured and documented in compliance with laws and standards. WLTP, GDPR, MDR (Medical Device Regulation), trade, and customs regulations are other examples of ever-changing requirements that constantly challenge PLM systems.

It is also impossible to ignore the trend toward a Digital Twin. Having a complete representation of the developed and manufactured product, and linking it to IoT platforms becomes increasingly important for efficiently operating products, digital services, and continuous optimization.

Therefore, PLM must be ready to respond to constantly changing market requirements and technological trends.

This requires a flexibly adaptable and extensible PLM backbone system that provides all relevant information throughout the entire product and process lifecycle, including the operating phase of the product. To enable this, the entire PLM architecture must be geared towards change. It must be possible to implement changes, adaptations, and extensions within short periods of time to avoid missing market opportunities, or violating legal requirements. "Time to market" is the keyword. However, despite the need for speed and flexibility, the stability of existing PLM functionalities must not be jeopardized at any time.

New market requirements and technological trends also require IT organizations to change. In conventional development projects, implementing new requirements for PLM systems can take up to several years. A growing number of companies are therefore making the decision to introduce agile approaches. Such agile approaches must be embedded in the organization and processes, and they usually require a new way of thinking.

Typical challenges on the way to Agile

Existing IT architectures are often not suitable for quick responses to new or changed markets, or customer requirements. This also applies, albeit not exclusively, to PLM architectures, which play a key role in new business processes and business cases. PLM-Systems are often systems that have evolved over the years and include numerous customer-specific adaptations and dependencies, both within the system and in combination with other systems. They are often very costly to maintain and have a high resistance to change.

Quite often a Waterfall- or V-model is used to develop PLM systems. Long planning cycles and long phases separated by milestones are typical for both models. A major disadvantage of both models is that in complex development projects a great deal of time and effort is spent on requirement specifications. Often, several months pass between definition of requirements and implementation – a long period of time, and typically, requirements change within this time, so they are outdated before being released.

Time-consuming change processes and firm contractual agreements then usually hinder specifications being adapted to accommodate changes once the specification phase is finished. As many stakeholders are aware of this they start “stockpiling” requirements during the specification phase: They tend to define more requirements than needed because they know, later changes are very difficult or near to impossible to make. This increases costs and time needed to implement the project. Also, long feedback loops are typical for both the Waterfall and the V-model. This is because customers and end users are not sufficiently involved in the implementation process once the specification phase has ended. This creates the risk that solutions are developed that miss the real needs of the users. Instead of obtaining feedback from users as early as possible and making necessary adjustments, development sticks to the specifications.

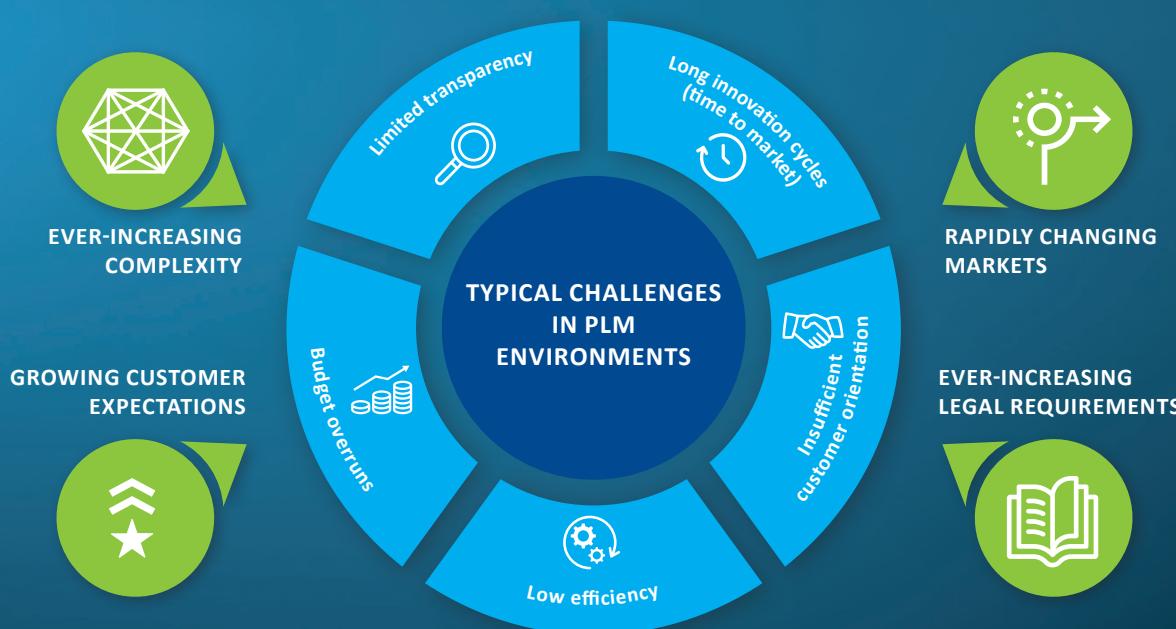
Due to the contractual situation and a commonly observed lack of ownership on the benefits and business values, IT departments often insist on following specifications to the letter – even if this is no longer necessary to achieve the desired business value. This decreases the efficiency in the PLM project significantly and increases the risk of missing market needs.

Additionally, a development process based on the Waterfall or V-model inevitably leads to considerable administrative overhead. A lot of effort is spent on documentation and specifications instead of focusing on usable deliveries such as working software. Partners often get tied up in lengthy contract negotiations instead of optimizing their collaboration. Processes and tools get top priority while human interaction is neglected. They adhere to a rigid plan instead of responding to change.

You can also often observe a blaming culture – a culture in which errors are penalized rather than being considered an opportunity for improvement. Many companies have the tendency to share responsibility over as many people as possible and to keep quiet about mistakes. The traffic lights for a PLM project often remain green even long after the project has run into difficulties. This has a negative impact on transparency in PLM projects and causes risks and delays.

Overall, this can lead to extremely long project durations, which can delay the introduction of innovations into productive PLM operations by months and sometimes even years. Many market opportunities are lost as a result.

Changing this is not easy. Development processes for PLM and IT systems often have a long history and are rarely subject to change or significant improvement. Generally, people responsible in PLM environments have a wealth of technical experience but are reluctant to challenge traditional, well-established software development processes, and familiar ways of working. This inhibits change and improvement.



Agile approaches are becoming state-of-the-art.

In today's world, agile methods are state of the art when it comes to IT project management. Many companies have identified the weaknesses in their existing software development processes and have started introducing agile approaches, or are at least planning to do so. Large corporations use them in large IT projects that they implement together with their partners. Only a few take an approach as radical as the BMW Group, which has mandated 100% agility for its entire IT organization, and is implementing it consequently. In many other cases, companies are just at the beginning of introducing agile methods. They are either still looking for a suitable agile model, or they have found a model and are in the middle of the transformation.

Typical for agile models is an iterative and empirical approach. Instead of specifying everything down to every detail, working software is developed quickly and regularly, and is frequently presented to the customer. Necessary adjustments and optimizations are identified quickly and implemented without long delays. Unclear requirements can be checked using mockups or ad-hoc prototypes. If a solution fails to meet customer expectations, it is discarded and another approach is tried out. This is the "fail fast" principle according to which failures are nothing to be blamed of, but are an opportunity for innovation and improvement instead.

It sometimes happens, that stakeholders realize at an early stage that already 80% of a feature's benefit has been achieved, while only 20% of the efforts have been spent (Pareto principle). In this case, instead of sticking to a previously defined specification, stakeholders should stop further development of the feature and invest the remaining budget in other, more useful features. This optimizes the overall benefit significantly, while keeping the costs the same. Agile approaches encourage this type of response to new requirements or new insights.



The most common agile process model is Scrum, an iterative approach involving teams with six to nine members who implement a backlog of requirements in working software in two or three-week sprints. The requirements are described from the end user's point of view, which is why they are often called "User Stories". Key activities of the Product Owner, who represents the customer or business departments in the team, is capturing and prioritizing the requirements, making them easily understandable for the team. The Product Owner also ensures that the backlog is always filled. Scrum teams operate largely independently and are self-organizing. The Scrum Master supports the team, and ensures that they can work undisturbed and as effectively as possible.

A characteristic of Scrum is that it does not require extensive hierarchies as it was designed for smaller development projects. For larger PLM projects, with thousands of user stories that depend or relate to each other, multiple scrum teams are needed. These teams need to be coordinated. There are several different scaling models with different complexities available for this purpose. They are all used to coordinate multiple scrum teams and manage the dependencies between their tasks. Well-known scaling frameworks are Scrum of Scrums, Nexus, LeSS and SAFe. While Scrum of Scrums and Nexus belong to the simpler scaling models, SAFe (Scaled Agile Framework) is one of the more complex ones. Some companies also use individual agile models.

SAFe is available in different configurations. Essential SAFe is the most basic configuration. It coordinates collaboration between multiple scrum teams using what is referred to as an Agile Release Train (ART). An ART is an organizational unit comprising the participating scrum teams and several coordinating roles such as Product Management and System Architect. In an Agile Release Train, all teams work in sync: the sprint cycles of all teams are in cadence. There is also a higher-level iteration cycle of about 3 to 5 sprints, called Program Increment (PI). This is used for coordination on the program level. In a so-called Program Increment Planning (PIP) event, which takes place shortly before launching a new program increment, the objectives for the next increment are defined, and the teams align on how to achieve them. Dependencies and bottlenecks are identified and resolved during a joint planning process. The resulting plan then provides guidance to the Scrum teams for the next program increment.

The other configurations of SAFe introduce additional levels of coordination for particularly large projects or project portfolios. They will not be covered further in this white paper. SAFe is just one of several scaled process models. When implementing agile methods, companies not only have to decide on a suitable agile model, but also find development partners who are able to go along with their agile approach. Today, a large part of software development is outsourced (nearshoring or offshoring) with the aim of taking advantage of PLM expertise in certain countries, saving costs, or shortening the time to market by parallelizing work. Ideally, partners not only integrate themselves in the customer's agile organization as co-workers, but are also able to actively provide advice for shaping the agile transformation. This requires that they are familiar with agile methods, not only in theory, but also in practice, and have practical experience dealing with the challenges posed by agile software development in globally distributed projects.

Agile billing models

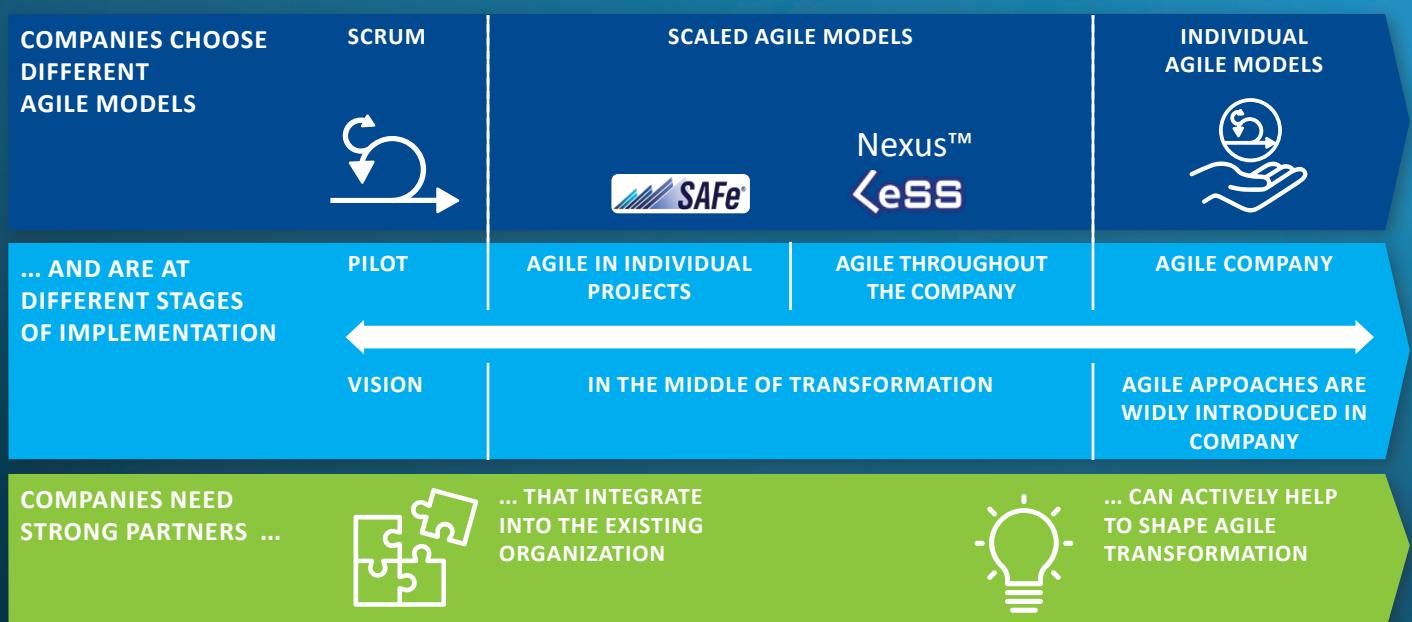
Applying agile methods in large IT projects with suppliers requires to rethink customer-supplier relationships. Companies have to challenge existing (fixed-price) contract models, as in agile approaches, project scope is typically only fuzzily defined when launching a project. So, it is impossible to define a clearly defined scope with clearly defined acceptance criteria at the time of contract negotiations. Typically, there is only a rough target or product vision, a timeframe, and one or more teams. This means that the customer's management must decide on an investment without having a detailed scope and cost estimation. Their decision must be based on a declaration of intent, that defines the goals to be achieved, without knowing the detailed results in advance, and without having a contractual basis for claiming deliveries on a detail level.

This has a direct impact on the billing model, as other criteria to measure supplier performance are needed. There are basically three variants: billing time worked ("Time and Materials"), billing deliverables at requirements level ("Agile Mini-Fixed Price"), or billing deliverables at project level ("Agile Fixed Price"). The Time-and-Materials model is the easiest to implement. It is lean and requires little additional organizational effort. However, it shifts all the project risk to the customer. In the deliverables-based "Agile Fixed Price" model, a fixed price for an overall project goal (product vision) is agreed on. A deliberate decision is made to refrain from creating extremely detailed specifications, arguing that an agile and empirical approach is more efficient. The resulting project risk is shared between both parties by means of appropriate agreements. This model requires a significantly higher level of efforts for controlling, change management, acceptance, and billing. However, it results in better transparency and a more equal risk-share.

The deliverables-based "Agile Mini Fixed Price" model is somewhere between the other two models. In principle it is a fixed-price model with very small billing units (user stories). A framework agreement defines the rules. This model also requires more organizational effort than the Time and Materials model when it comes to acceptance and billing, but much less effort than the "Agile Fixed Price" model. It is often a good compromise between the two models "Time and Materials" and "Agile Fixed Price". However, in both deliverables-based models, conflicts can become problematic when assessing efforts needed for user stories. If the supplier estimates the efforts needed for a user story to be significantly higher than the client is willing to accept, neither side can force the other to accept their view of the situation. This means that such conflicts must be resolved constructively between both partners. A general feature of agile collaboration models becomes apparent here: They require a high level of mutual trust and great willingness to resolve conflicts constructively. An appropriate mindset on both sides is a key to success. This however applies to the successful implementation of IT projects in general. Agile process models merely demonstrate a way of dealing with them professionally and in a result-oriented fashion.

The implementation of PLM solutions in a company's extended IT landscape is a large and complex project. When applying an agile model, there is a risk that a project is launched without having understood the goals and constraints, hoping to clarify those during the project. Agile does not mean you should start a project without a clear vision of where to go. You should not define everything in every detail, but you should know the key outcomes. Unfortunately, agile methods do not provide clear recommendations to what extent expectations should be defined in advance, and how far they can be left open. Therefore, project participants must find the right balance between "creative chaos" and over-specification along the lines of the Waterfall or V-model.

Experience shows that in particular, a lack of knowledge of business processes and inadequate analysis of the data model can result in expensive, abortive developments. This could be avoided with a little more conceptual preparatory work. It can therefore be concluded – and this specifically applies to large PLM projects – that even with an agile approach, sufficient attention to business processes and data model is needed at an early stage, to ensure that the fundamental concepts for the desired functionality are available when needed.



The role of PROSTEP

PROSTEP has been using agile approaches to develop its own software solutions for many years, and as a partner and supplier also brings this experience to bear on customer projects. The PLM consulting and software company is currently involved in agile projects for numerous major customers in the automotive, shipbuilding, and other industries. Development teams from PROSTEP are, for example, helping MEYER Werft restructure its PLM landscape, providing BMW with support for the agile implementation of its iPDM initiative, and helping Daimler transform its central PLM development project to a unified, scaled agile approach. In many cases, PROSTEP assumes overall responsibility for agile projects as general contractor and coordinates subcontractors, be it on site at the customer's premises, or at an offshore partner.

PROSTEP's teams combine PLM expertise and hands-on experience using agile methods. They know the strengths and weaknesses of Scrum, SAFe, and other agile models from practice, and therefore can actively help to shape agile transformation at the customer's site and drive it forward. Together with Daimler, for example, PROSTEP has introduced a unified, SAFe-based project model for the maintenance and further development of Daimler's central PDM system. As part of the "PDM goes SAFe" initiative, different working methods, requirements management processes, billing models, release cycles, etc., were harmonized, and all formerly separated projects merged into one joint, agile project.

Thanks to their PLM expertise, PROSTEP developers are also familiar with the peculiarities of PLM systems, which need to be considered when using agile methods. Many PLM installations are very difficult to maintain by agile methods due to their monolithic software architectures and structures. These then evolve over time to include numerous customer-specific adaptations and the many interfaces to other systems. Therefore, PROSTEP has developed several approaches for handling those limitations in an agile way.

One possible approach, chosen by Daimler but certainly not suitable for every customer, is to build a state-of-the-art software architecture aside the existing legacy architecture. The new architecture provides the basis for fast and agile development while at the same time making it possible to migrate existing functionality step by step in a controlled manner. This requires a clear strategy for replacing one architecture with another and a reliable solution for synchronizing the databases of the old and new architectures. In Daimler's case, this was based on the Graph Database integrated in Oracle.

There are several other measures, that can help to further develop complex, monolithic PLM installations faster and more flexibly, and adapt them to fast changing user requirements:

- Identify modules that have only few interfaces to their environment.
These can often be developed and deployed separately, and therefore be iterated faster.
- Utilizing intelligent branching in configuration management.
- Frequent integration of modules as soon as their development status is stable, even if their development is not yet finished.
- Shortening feedback loops by frequent demonstrations in staging systems without validating all details every time.
- Keeping long release cycles for production releases for security reasons, risk mitigation, and synchronization with external interfaces.

Monolithic PLM architectures often require long test phases to minimize risks involved in the interactions between new and existing functions, especially when it comes to dependencies on other systems. This makes it difficult to shorten release cycles. However, state-of-the-art tools for source code management, automatic testing, automatic analyzes of code quality, and automatic deployment can help to make new PLM functions available to users more quickly. PROSTEP developers have mastered these tools and make effective use of them in agile projects.

As part of its PLM Strategy Consulting offering, PROSTEP provides its customers with advice when it comes to analyzing their existing PLM capabilities and building a sustainable PLM architecture that accommodates the use of agile methods for further development, along with the integration of new technologies. The PLM consultants are well acquainted with the market, with PLM vendors and their solutions, and are therefore able to identify PLM systems that are best suited for an agile approach based on their architecture.

Summary

In order to respond flexibly to dynamically changing market and customer requirements, companies must be able to quickly adapt their PLM solutions and add new technologies. The classic Waterfall or V-Model approaches with long planning cycles, long feedback loops, and long test phases, do not meet these requirements. They also often lead to the inefficient use of scarce development resources. A growing number of companies are therefore starting to implement agile models in their software development process. Experienced development partners, who are familiar with both PLM processes and agile models such as Scrum and SAFe, can provide optimal support. PROSTEP has a wealth of experience gained from its own software development and numerous agile projects carried out for customers in the automotive industry, ship building, and other sectors. This not only makes it possible for the company's developers to go along with the customers' Agile Transformation but to also provide practical advice for shaping it.

Our range of solutions

- When it comes to agile software development, PROSTEP supports as a service provider, or as a general contractor coordinating subcontractors in agile projects.
- Our software developers have hand-on experience with the most commonly used agile process models and a wealth of PLM expertise.
- As part of the customer's team or as a PROSTEP team, we can take over different agile roles: Developer, Product Owner, Scrum Master, Release Train Engineer, or System Architect.
- We provide customers with advice on introduction and continuous improvement of agile processes and methods.
- We offer architecture-specific consulting advice for designing appropriate PLM architectures.
- We provide your users with advice and training in using agile methods, including complex, scaled methods, such as SAFe.
- We provide advice and support for continuous build and deployment, or automated testing, when creating a DevOps infrastructure.
- We provide advice and support setting up and maintaining agile tools.
- We also offer development services for all these points.
- As a service provider, we operate your PLM systems in the context of DevOps.
- We provide advice and support on applying agile methods in PLM environments and overcoming the specific challenges related to it.

COMPONENTS	SOFTWARE DEVELOPMENT BASED ON AGILE PROCESSES AND METHODS				
	AS SERVICE PROVIDER		AS GENERAL CONTRACTOR		
STAFFING AGILE ROLES	DEVELOPER	PRODUCT OWNER	SCRUM MASTER	RELEASE TRAIN ENGINEER	SYSTEM ARCHITECT
	as part of a PROSTEP team			to support a customers team	
	with hands-on experience in agile processes and collaboration in agile teams				
	with a high level of PLM expertise				
DEVELOPMENT SERVICES	SETTING UP AND MAINTAINING AGILE TOOLS		TESTING	CONTINUOUS BUILD & DEVELOPMENT	
OPERATIONAL SERVICES	PLM DEVOPS Operation of PLM systems in the context of DevOps				
CONSULTING SERVICES	ARCHITECTURE CONSULTING	TEST STRATEGIES	DEVOPS	AGILE PROCESSES	AGILE PLM Application of agile methods in PLM environments



Thanks to a wealth of experience, PROSTEP is able to adapt flexibly to a customer's situation and provide customized support





PDF version of the white paper:
www.prostep.com/whitepapers
or scan the QR Code



Do you have any comments or questions?

We look forward to your feedback at
infocenter@prostep.com

PROSTEP AG
Dolivostrasse 11 · 64293 Darmstadt · Germany
Telephone +49 6151 9287-0 · Telefax +49 6151 9287-326 · E-mail info@prostep.com

© 2021 PROSTEP AG. All rights reserved
All the trademarks identified by ® or ™ are the property of their respective owners.

LEGAL NOTICE

Published by
PROSTEP AG

Contact:
Frank Brandstetter
frank.brandstetter@prostep.com

Edition 1, 2021