PLMadvisors
Better Processes. Better Technology. Better Results.

White Paper

10 Best Practices for Successful PLM Implementations

Who hasn't heard stories about nightmarish PLM implementation projects that experienced unforeseen problems and went way over schedule and budget? Some of these stories have even gone public, like the projects at Airbus and Ericsson. And there are many more that nobody other than the people who were involved know about. I mean, who wants to broadcast a failure? But wouldn't it be great to understand what went wrong in these projects and what you can do to avoid the problems these companies went through and ensure a successful outcome of the PLM project at your company?

Having gone through dozens of PLM implementations in various roles, including program and project manager, process consultant, business and solution architect, PLM advisor and subject matter expert, as well as key user and also talking with colleagues in the PLM industry who have implemented PLM I have learned many valuable lessons from the successful implementations, but also from a myriad of problems, obstacles, challenges and issues encountered during the various projects.

The 10 best practices described herein are what I wished I had known before starting each project and I recommend following to ensure a successful PLM implementation. These practices are not listed in order of importance or priority, and some will be more important in certain projects than others. I have tried to indicate for each of the practices in what instance they may be of particular importance.

1. Use a Proven Implementation Methodology

There are thousands of tasks that must be performed during the implementation project by a variety of internal and external team members, and many of these tasks have long lead and execution times as well as critical dependencies, so having a proven implementation methodology is in my experience the biggest guarantor to completing a project within the originally planned schedule and budget.

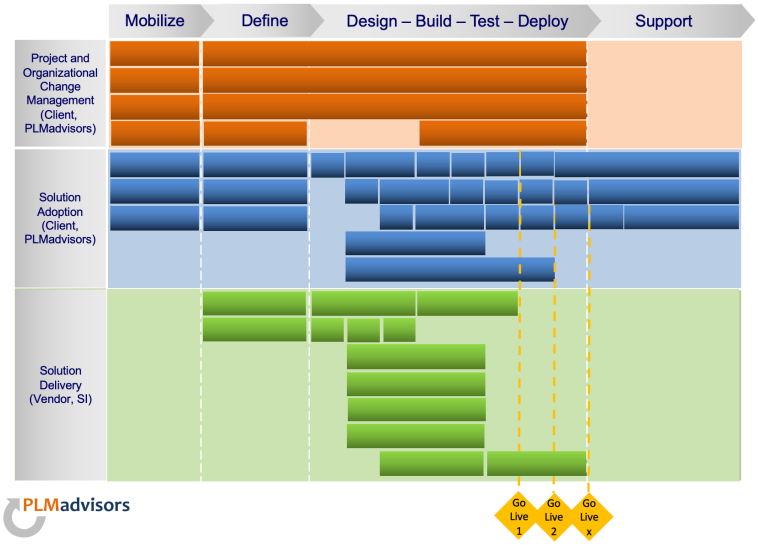
An implementation methodology is more than just an Agile or Waterfall approach. It is having all the important high-level activities and dependencies defined that need to be executed

A good implementation methodology is like having a good recipe when preparing a meal. It ensures nothing is forgotten and all activities are done correctly and in the right order.

It is the biggest guarantor to completing a project within the originally planned schedule and budget.

throughout the project, from the mobilization of the core team all the way to post-go-live support or hypercare so that the individual tasks can be planned, coordinated and started and finished on time and nothing gets overlooked. It is like having a recipe when preparing a meal that ensures no ingredients are forgotten and everything is done correctly and in the right order.

Similarly, a good implementation methodology ensures that important activities are done in the proper sequence and tasks are completed when dependent tasks need to be started. For example, all the business processes, use cases and requirements are defined before the system design starts, that the various environments are ready when the sprints to configure and test the new system need to be performed, and that the data that needs to be migrated is identified, cleaned and ready before the first data migration rehearsal needs to be done.



Each one of the activities mentioned and shown in the simplified Implementation Methodology illustration above can take weeks if not months to complete – as in the case of process definition and data preparation, so if in the initial excitement and rush to get the project started these activities are not given the proper attention, priority and resources, a project delay is almost certainly already guaranteed from the beginning.

Needless to say, a proven implementation methodology is the result of extensive practical experience and expertise in

implementing PLM solutions and of meticulously identifying and documenting what works and what could be done better during dozens of PLM implementation projects.

2. Own the Project Management

It is not advisable to delegate the overall project management responsibility to the implementer. It is your project and you ultimately are responsible for meeting the objectives and staying within the allocated time and budget.

Most service providers that implement PLM software will propose that they staff a project manager as part of the implementation team who will manage the overall project. They will reason that because they are responsible for the implementation their project manager also needs to manage everything.

It is not advisable however to delegate the overall project management responsibility to the implementer. It is your project and you ultimately are responsible for meeting the objectives and staying within the allocated time and budget.

Having the implementation provider assume the overall project management responsibility often creates a conflict of interest for them between maximizing revenue and finishing the project on time and on budget. My experience is most service providers tend to favor the former, that is maximizing their revenue whenever there is an opportunity. Few implementers will push back hard if the customer discovers something nice to have and considers increasing the scope during the project. But that will unfailingly lead to a project change order, higher cost and usually also a delayed go-live.

Ask the implementor to provide a technical project manager who is responsible for managing their own technical resources and tasks. This technical project manager should report to the overall project manager provided by your company. If you do not have an internal resource who has the necessary experience and expertise, hire an independent external PLM project manager who reports to you and has no financial benefit from expanding the scope of the project.

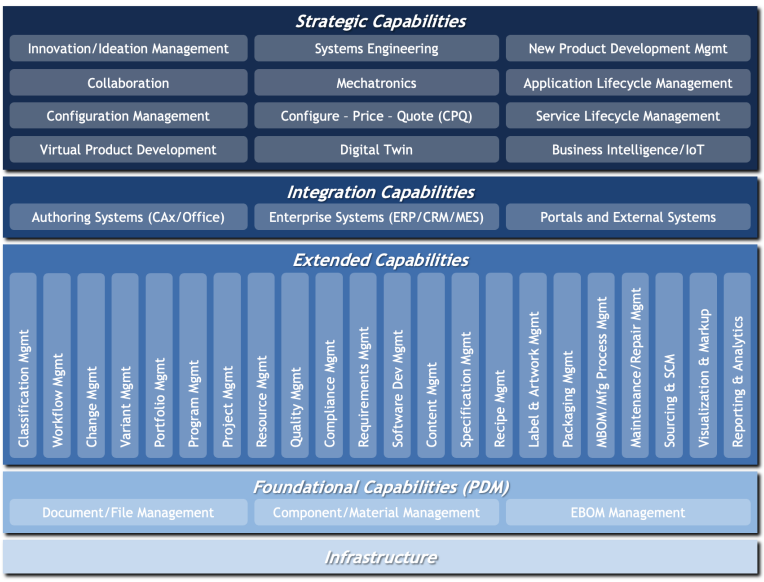
3. Start Small, Grow Incrementally

Before engaging an implementation service provider there is some homework that needs to be done. Defining the overall scope and a plan for a phased implementation, a PLM roadmap, is an important step.

PLM is a very comprehensive solution with 40 different modules or capability areas. It is not realistic, and not even beneficial, to attempt to implement everything at once.

The better approach is to implement PLM in phases. Start with a limited scope in the first phase and then add more functionality incrementally in subsequent phases.

PLM is a very comprehensive solution with dozens of modules or capability areas. At PLMadvisors we distinguish 40 different capability areas, as illustrated in the graphic below. It is not realistic, and not even beneficial, to attempt to implement everything at once, as it would greatly burden the company and take years to complete.



The better approach is to implement PLM in phases, start with a limited scope in the first phase and then add more functionality incrementally in subsequent phases. A PLM Capability & Maturity Assessment can help to determine the most beneficial implementation roadmap and sequence for your company. A good starting point is to begin with the foundational capabilities and maybe two or at most three extended capabilities in the first phase.

Phases should be scoped so that the initial phase can be implemented in no more than 5 to 6 months and any subsequent phases in 3 to 4 months. This will ensure that your company gets quick results and return on investment from PLM.

4. Follow a Systems Engineering Approach

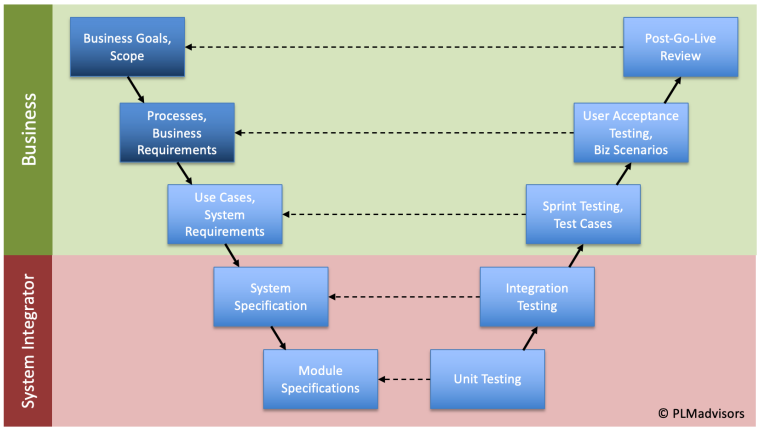
A PLM solution is a very complex system, particularly if the scope goes beyond just the foundational capabilities. All the processes, practices and functional elements of the tool that need to be implemented must work together seamlessly. And

A PLM solution is a very complex system. All the processes, practices and functional elements of the tool that need to be implemented must work together seamlessly.

The V-Model is a well established approach for systems engineering that works very well also for PLM implementations.

that requires a very thorough definition, design, configuration, maybe even customization, and testing of the solution to ensure it ultimately meets all the business and system requirements and works as intended.

A well established and proven approach for systems engineering is the V-Model, also called the Validation and Verification Model. It has been used in the development of complex products in the aerospace & defense, the automotive and machinery & plant equipment industries for decades, and it has been successfully adopted also for software development. At PLMadvisors we have found this approach also works very well for PLM implementations, and we successfully use it in all PLM projects that we lead and manage.



The left side of the "V" represents the definition or validation of requirements and creation of system and module specifications. It starts with the definition of business goals for PLM and the scope required to meet those goals, then the definition of processes and business requirements followed by the definition of use cases and system requirements. These first two activities – colored in a darker blue in the above illustration - can and should be done before engaging an implementation service provider or system integrator.

The next steps on the left are the creation of the system design specification as part of the overall solution design followed by the creation more detailed module specifications, which should be done as part of the sprint in which the respective functionality is configured.

The right side of the "V" represents the verification of the previously defined requirements in various stages. This starts with the unit testing, where the individual functional modules are tested against the module specification. In the second stage, the integration testing, the different functional modules are tested together against the system specification. These two activities should be performed by the implementor or system integrator. In the next stage, the sprint testing, the business develops test cases and tests all implemented functional modules against the use cases and system requirements and verifies that the implemented performs as desired.

Once the entire system has been configured and tested in a series of sprints, the business should then create business scenarios and verify that the entire system works as specified as part of an all-encompassing user acceptance testing or UAT.

5. Define Business Processes Before Starting the System Implementation

Business processes, practices and procedures need to be defined before engaging an implementation service provider.

It can be very beneficial to engage an external party that specializes in business process design and optimization to help guide the business and ensure best practices are used and followed as much as possible.

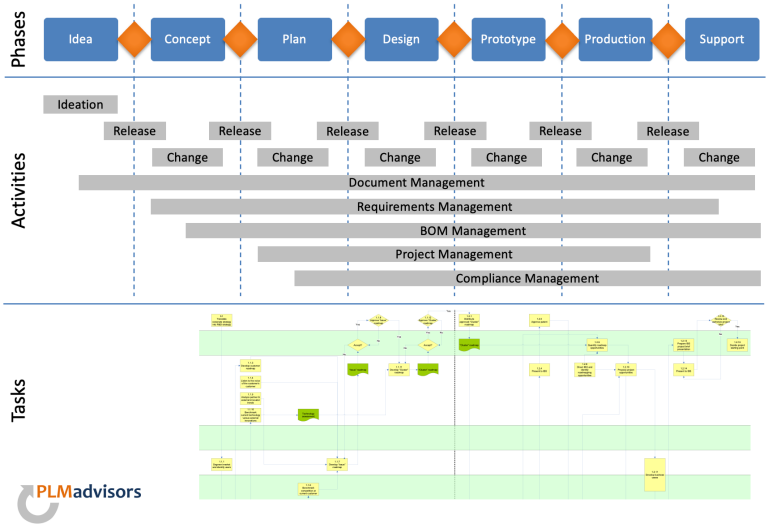
Another part of the homework that needs to be done before engaging an implementation service provider is to identify and define all business practices, processes and procedures on a high level that should be enabled by the future PLM system.

These typically include at a minimum a part creation process, a document creation process, an EBOM creation process, an engineering change (ECR/ECO) process for parts, documents and BOMs, a new product development (NPD) process, and a customer project delivery (CPD) process. Depending on the scope also included could be a product configuration process, an MBOM creation and manufacturing planning process, and quality processes, such as PPAP, NCR, CAPA, deviation, stop-ship, etc.

Although the NPD and CPD processes may initially not be fully implemented in PLM, they will provide the necessary framework for the life cycle or release stages your products go through when they are being developed and/or delivered. And those in turn will determine the access privileges for different users and functions to parts, documents and BOMs in each life cycle or release stage as well as the steps and rigor of the engineering change process in each stage.

At this point it will not be necessary or even helpful to define these processes to the lowest level of detail. It is sufficient to define the phase and activity levels, as shown in the illustration below. The lowest level should be done together with the system implementor during the solution design activity, as the individual tasks will be dependent on the system functionality and configuration.

However, it is important to decide before engaging an implementation firm what processes should be implemented and how complex these processes need to be. Should for example the engineering change process consist of one, two or three parts; issue capture and disposition, change request and change order? Or do we not need all three parts? And should the change process be the same in all stages of the product development and delivery processes or is it better to have less complexity in the earlier stages and more rigor in the later stages?



These are all business related questions an implementation service provider cannot help you to determine and decide. So why pay them while your business leads and subject matter experts go through the effort of determining what your company needs?

What is important is to utilize industry best practices and standards, such as ISO and ASME standards, configuration management best practices, the stage-gate methodology, etc as much as possible. Most PLM systems support these best

practices and often even have preconfigured workflows based on these standards, so adopting them as closely as possible will greatly reduce the time and cost to configure the system.

It can be very beneficial to engage an external party that specializes in business process design and optimization at this stage of the PLM project to help guide the business team and ensure best practices are used and followed as much as possible.

6. Define Realistic Use Cases and Detailed System Requirements

The goal is to define a comprehensive set of use cases necessary so that users can execute all business processes and perform all related activities in the system, followed by detailed system requirements for each use case.

This is the foundation for a good system design as well as the subsequent configuration, possible customization and testing of the system.

Another important step is defining realistic use cases and detailed system requirements. This will be done based on the business processes that were defined previously.

Use cases are activities users have to be able to perform in the system to execute all business process, basically a step by step decomposition of business processes. Using the engineering change process for example, activities, or use cases, users would have to be able to perform are to create an engineering change request, to review the engineering change request, to approve the engineering change request, to attach affected and reference documents to the change request, to view documents attached to the engineering change request, to mark up those documents, etc. These are just some of the use cases related to the engineering change request or ECR process. In addition use cases may include activities that are not directly related to the execution of processes, such as to upload a document to the system, print documents, etc.

The goal is to define a comprehensive set of use cases necessary so that users can execute all business processes and perform all related activities in the system.

Once all the use cases are defined, the next step is to define detailed system requirements for each use case. Using the above example of the engineering change request, related system requirements may be that an ECR has to have a unique number, it has to have certain attributes, it has to be possible to attach documents of certain types, it has to be possible to markup documents without altering the original document, and so on.

Ultimately each use case must have one or several system requirements. First these system requirements are the basis for designing the system and also for determining whether the system can be configured with out-of-the-box functionality or possibly needs to be customized.

If for example a requirement is defined such that part numbers have to be significant, then it could mean that the system will have to be customized to support that requirement. So it is important to understand the functionality and capabilities of the system when defining the system requirements, which is why it is recommended to involve the implementation service provider or system integrator in this activity and ask them to let you know for each requirement if it can be met with configuration or will need customization.

After the design, configuration and possibly customization of the specified functionality sprint testing will then be performed by the business to verify that the system works as desired and all system requirements are met.

7. Configure, Don't Customize

Avoid customization whenever possible. Customization is expensive to implement and even more costly to maintain.

Only if customization provides a true competitive advantage is it worth it.

To borrow from Shakespeare's Hamlet: "To customize, or not to customize, that is the question". In the context of a PLM implementation this question is fortunately not about life and death, but the way it is answered can still have severe and long-lasting consequences. So it pays to carefully weigh the pros and cons before going down the path of customization.

This topic usually comes up during the definition of company specific requirements when the system implementor – hopefully – points out that meeting a requirement the way it is defined will require customization. If the implementor doesn't voluntarily disclose that information, I recommend during the design of the system to explicitly ask for each system requirement that is defined how the system integrator plans to implement it: Does the system support it with out-of-the box functionality, does it require configuration or does the system indeed need to be customized?

If the answer is customization, ask yourself if it is indeed worth it. And ask the implementor – or even better an independent business architect who has no interest in increasing the complexity and cost of the implementation – if there are other

ways to meet the business need. Is it for example worth it to customize the system to support a company specific, significant part numbering scheme or naming convention? Or could maybe the part numbering system be changed to a simple sequential one that is supported out of the box and other available system functionality, for example attributes or classification, be used to find and identify a part and with that avoid the need for a significant part number?

In general my recommendation is to avoid customization whenever possible. Only if customization provides a true competitive advantage is it worth it. In most instances though it is easier and cheaper to change a company practice or process so that the system can be configured to support the requirement rather than to insist that the system must work exactly the way the company currently operates.

8. Start Data Migration Early

Proper preparation and execution of the data migration is crucially important in order to protect your company's intellectual property and ensure only current, unique and valid data is migrated into the new PLM system.

The identification and preparation of the data that needs to be migrated can be very time consuming and resource intensive and should be started as early as possible.

One of the most important but usually also most underestimated activities of a PLM implementation is the data migration.

The data that needs to be migrated during a PLM implementation typically represents the core intellectual property of a company. CAD models and assemblies, engineering drawings, engineering and manufacturing bills of material or EBOMs and MBOMs, material specifications, manufacturing work instructions, and quality documents is just some of the data that will usually live and be managed in a PLM system.

And much of that data has complex relationships. CAD assemblies, and the related BOMs, often consist of hundreds if not thousands of parts, each with its own revision. Many of those parts have a corresponding CAD drawing, and both models and drawings often also have a file in a neutral format, such as JT or PDF, associated to it for viewing purposes.

In addition, many parts and assemblies have reference documents attached, such as material specifications, quality documents, and manufacturing work instructions. All these relationships must be preserved during a migration, or in many instances, created systematically. This can be complicated by the fact that much of this data resides in different repositories prior to migrating it into PLM, and sometimes only people know the relationship between different data sets.

It is also quite common in a file-system based environment that identical or almost identical documents are stored in multiple locations or file folders. But only one, the latest and correct one, should be migrated. Hence it is important to determine the most current version of each file prior to loading data into PLM.

And the data migration gets even more complex if part of the implementation scope is an integration to another system, such as ERP or MES. Then, in many instances, some of the data in the ERP system, such as part master information and BOMs, will have to be migrated to the PLM system so that there is a baseline in the PLM that then can be maintained there and kept in sync between the two systems through the integration after go-live.

Proper preparation and execution of the data migration is crucially important in order to protect your company's intellectual property and ensure only current, unique and valid data is migrated into the new PLM system. One of the objectives most companies have for PLM is to establish a single source of truth for product data. Migrating bad, dated, inaccurate, or incomplete data can quickly undermine or destroy the users' confidence and with that the acceptance and adoption of the new PLM system.

There are eleven steps to a well-executed data migration: 1) Planning, 2) Identification, 3) Analysis, 4) Training, 5) Clean Up, 6) Tool Configuration & Development, 7) Extraction, 8) Staging, 9) Transformation, 10) Loading, and 11) Verification.

These eleven steps do not have to happen in this exact sequence. Depending on the type of data and the source system some activities have to be executed together, and certain activities are dependent on others and have to be performed sequentially.

Our White Paper "11 Proven Steps for Successful Data Migrations" explains these eleven activities in extensive detail.

In the context of this White Paper it is essential to understand that data migration is a critically important part of most if not all PLM implementations. Many of the activities listed above, specifically the data analysis and clean up, can require significant time and effort to complete, many times weeks or even months, so the overall data migration is often on the critical path. Consequently it has to be carefully planned, started early and

executed as an integral part of the overall implementation project. In some instances it may even be necessary to start some of the data migration activities, specifically steps 1) through 5) already before the technical system implementation so that the data is ready for the data migration rehearsals and then of course the production migration.

9. Don't Forget the People

The ultimate success of a PLM implementation project stands and falls with the acceptance and adoption of the system by the users and the larger organization in general.

An organizational change management (OCM) plan should be created and subsequently executed that identifies and addresses all areas of the organization that will be affected by the PLM project.

Implementing a new PLM system usually means a significant change for an organization in how it operates. People will use a new tool and have to do things differently. Some people may not even need to do the same job anymore. For example someone who manually enters BOMs from an Excel spreadsheet into the company's ERP system may not have to do this at all anymore if the implementation includes an integration from PLM to ERP.

And the implementation itself is usually an additional burden on many employees. Some may be involved in the project core team as subject matter experts, business leads and key users and participate in requirements definition and design sessions as well as in the testing of the system, others may just have to attend training to learn the new tool and processes.

Whatever their involvement may be, the impending change and the related uncertainty of what may be coming and how they may be affected is scary for many employees. And fear usually means resistance, in this case resistance to implementing and using a new system and processes.

This resistance may initially manifest in talking badly about the new system, spreading rumors, refusing to actively participate and contribute in design sessions, testing and training, finding problems with how the new system operates and ultimately refusing to use the new system and processes and continuing to work the old, familiar way and insisting that the old tools are better and easier.

The result is that adoption of the new system and processes suffer, that old methods persist, that the anticipated benefits are not realized and that the project ultimately is only a partial success or even a complete failure even though the new system technically works exactly as planned and designed.

How can such an undesirable outcome be avoided? Change is only scary if the outcome is unknown or uncertain. If people know from the outset that the coming change will be beneficial – or at a minimum not be detrimental to them, they will support and embrace it. And that's the key to a successful adoption and outcome of the project.

It is important to inform and involve employees from early on and as much as possible about the nature, scope and benefits of the project and how they might be affected. The employee who thus far manually entered BOMs into the ERP system needs to know that they will still have a job after the new system goes live and what their future job will look like.

However, this of course requires identifying potentially affected employees across the entire organization, analyzing the impact the project will have on their future work and creating and executing a plan to address those changes. Too often this human aspect of a PLM project is largely neglected until shortly before go-live when all of a sudden the project team realizes that people need to be informed, trained on the new system and processes and possibly on entirely new job activities.

To manage all these aspects an organizational change management (OCM) plan should be created that identifies and addresses all areas of the organization that will be affected by the PLM project, including people, processes, systems and procedures. It should include regular information of the core team, management, stakeholders and users, selection of a cross-functional core team, definition of new processes and procedures, testing of the system, training of employees on the new system and processes, short-term hyper-care and long-term post-go-live user support.

This OCM plan should be created very early during the mobilization phase of the project and then executed and managed throughout the project by an OCM lead. Depending on the size and complexity of the project the project manager may take on that role or, in larger and more complex projects, it may require a dedicated resource.

It is important to remember that the ultimate success of a project stands and falls with the acceptance and adoption of the system by the users and the larger organization in general. The system can be perfectly implemented and work flawlessly, if people

reject it and don't use it, the project will not achieve the projected and anticipated benefits.

10. Get Expert Guidance and Advice

A PLM implementation project has multiple workstreams, dozens of resources, hundreds of activities and countless dependencies that need to be planned, coordinated and executed timely and in the correct sequence.

An experienced PLM consultant and advisor can guide the implementation, help avoid common issues and problems and ensure a successful outcome.

Implementing a PLM system is not a core competency of most companies. And neither should it become one. Why not? Because most companies implement a PLM system maybe once in 10 years.

So while most companies have great engineers, project managers and IT specialists, the majority of them lack the experience and expertise to successfully plan, lead and guide the implementation of an enterprise-wide PLM system.

A PLM implementation project has multiple workstreams, dozens of resources, hundreds of activities and countless dependencies that need to be planned, coordinated, executed and considered at the right time and in the correct sequence to ultimately finish the project successfully on schedule and on budget.

A cross-functional core team consisting of executives, business leads, subject matter experts, key users and data migration experts, all with relevant experience, has to be assembled, processes have to be defined based on industry best practices, practical use cases and relevant requirements have to be created, the system has to be designed requiring no or minimal customization, sprints have to planned and coordinated to configure and test the system as well as rehearse the data migration, stakeholders have to be kept up to date on the objectives and progress of the project, users have to be trained, environments have to be provisioned on time to support the development, testing, training and lastly the production go-live, and post-go-live support procedures and staff have to be set up to ensure quick assistance of users in case of the unavoidable issues.

An experienced PLM consultant and advisor can assist in planning the overall project in practical, realistic and digestible phases and subsequently guide or lead the implementation using a proven methodology, help anticipate and avoid the common issues and problems and therefore ensure a successful outcome.

About the Author

Andreas Lindenthal is the founder and principal of **PLMadvisors**. He brings over 25 years of professional, international experience in innovation, new product development and PLM to his clients. He serves clients in the aerospace & defense, automotive, consumer goods, high tech & electronics, medical device and machinery & plant equipment industries. Andreas has successfully assisted over 50 small and large companies in the automotive, aerospace, consumer goods, high-tech/electronics, machinery equipment, and medical device industries in the development of strategies, optimization of business processes and the implementation of new technologies to enable their growth strategies and drive innovation, increase productivity, shorten time-to-market, reduce costs, ensure compliance, and improve product quality.

Andreas is a former partner of Kalypso and co-founder and president of Metafore, both consulting firms focused on innovation, new product development and PLM. He also served as an executive of UGS Corp. (today Siemens PLM Software), a leading provider of product lifecycle management software, both in Switzerland and the United States.

Prior to UGS he worked as head of product development at Sulzer AG, a Swiss technology corporation, where he was responsible for new product development (NPDI). In this role he also led the evaluation, implementation, and operation of an enterprise-wide PLM system with the objective of lowering time-to-market and cost of goods sold for new products. From Sulzer he received the "Innovation Award" for his pioneering ideas and work in establishing and sustaining a corporate culture of innovation and productivity.

Andreas holds an MBA degree from the Graziadio School of Business and Management at Pepperdine University in Malibu, California and a bachelor degree in Mechanical Engineering with emphasis in Computer Integrated Manufacturing (CIM) from the Zurich University of Applied Sciences in Switzerland.

Contact Andreas Lindenthal at andreas.lindenthal@plmadvisors.com or at 888.800.4PLM.

About PLMadvisors

We help our clients to drive innovation, accelerate revenues, increase productivity, reduce costs, improve quality, shorten time-to-market and ensure compliance through the resourceful utilization of best-in-class innovation, new product development and product lifecycle management practices, processes and technologies.

For more information call us at 888.800.4PLM, send an e-mail to info@plmadvisors.com, or go to our website at www.plmadvisors.com.