

Role of PLM in the software lifecycle

A whole product approach for enabling high tech and electronics companies to manage the entire software lifecycle

www.siemens.com/teamcenter

white paper



- ▶ To address the rising role of embedded software in today's product offerings, best-in-class high tech and electronics companies want to more effectively manage their software development process. Teamcenter® software drives this strategic objective by enabling you to manage your software lifecycle in the context of a whole product lifecycle that includes your mechanical, electronic, software and control system components. This whole product, mechatronics approach enables your company to accelerate product introduction, lower cost and improve quality.

PLM Software

Answers for industry.

SIEMENS

Table of contents

Executive summary	1
Business challenges	2
Using Teamcenter to manage your software cycle	7
PLM – a key component of the software lifecycle	16

▶ Executive summary

As product complexity increases and software-driven electronics make their way into more and more product offerings, today's high tech and electronics companies need to find a way to break down the barriers that have isolated their software development process from the rest of the product lifecycle.

More specifically, best-in-class product makers are looking to integrate their software development domain with their mechanical, electronic and control system design domains and intelligently manage their various interdependencies. In essence, they want to be able to manage the software lifecycle in the context of a total and truly integrated product development environment. Three key capabilities are required to drive this initiative, including the ability to manage:

- Software requirements as part of a whole product requirement and specification
- Software as a component or “part” of the total product
- All of the tools and processes used across the software lifecycle

Software as a part. The ability to track, manage and configure software as an integrated part of the product lifecycle requires a single source of product and process knowledge that links each software “part” to the rest of the product. By treating software as a part, product makers can tie software features to the product requirements that determine how the software will interact with other parts of the system.

By treating software as a part in the product configuration and establishing the appropriate dependencies and compatibility during product development, product teams can ensure that:

- Software identification, auditing, accounting and configuration management are enabled
- Various software modules are compatible with one another
- Software is compatible with the electronics hardware
- Changes to software and hardware modules are managed from inception to end-of-life
- Impacts of software changes are identified before changes are incorporated
- Software development implements features based on an requirements-driven process

Tool and process management. To effectively manage the entirety of today's software development lifecycle, product makers need to manage core software development tools and processes – as well as the activities of dozens of globally dispersed software development team members, including software developers, project managers, QA teams and hardware engineers.

To manage the entire product development process, product makers need to leverage software configuration management (SCM) or application lifecycle management (ALM) systems in conjunction with product lifecycle management (PLM) solutions. In a best practice scenario, PLM is used for requirements management, system design and hardware development while SCM/ALM is primarily used during the software development phase.

As development nears completion, PLM plays a dominant role in the configuration, change management, production, support, maintenance and end-of-life phases of the product lifecycle. From a purely functional point of view, PLM and SCM complement each other by enabling product makers to address the entire software lifecycle and integrate the software development process into a whole product lifecycle.

Role of PLM. PLM solutions like Siemens PLM Software's Teamcenter software play a crucial role in ensuring that high tech and electronics companies are able to capture, implement, develop and maintain a complex product that the market wants and will actually buy.

In practical terms, Teamcenter provides the PLM backbone and mechatronics data model that addresses the needs of the software lifecycle. Teamcenter facilitates system-level integration, enabling all of the domains in an electromechanical design and product development effort to share and manage data created by a wide variety of teams or internally developed and third party applications. By addressing the software lifecycle as a piece of the whole product, Teamcenter accelerates product introduction, lowers cost and improves quality.

Increasing use of embedded software

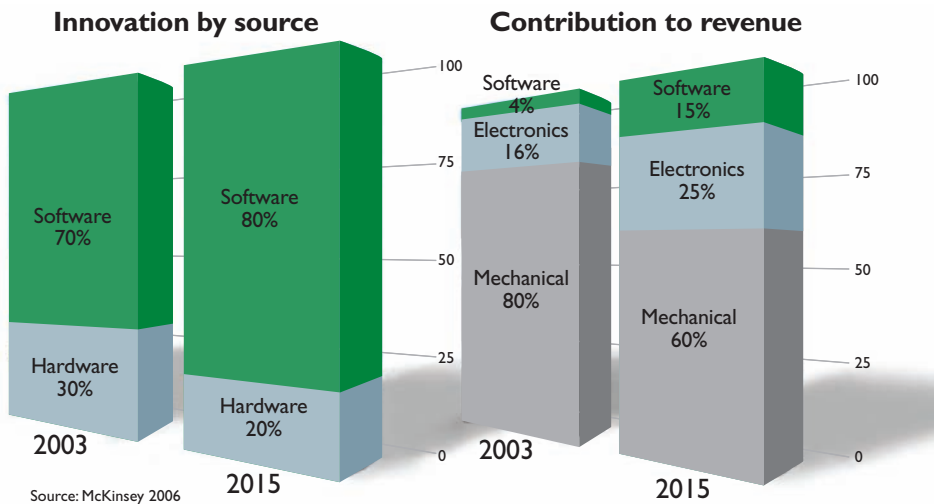
In virtually every industry segment, product complexity is increasing and software is now a critical element in industries that traditionally provided hardware-oriented products. Many of the advanced features offered in today's products are enabled through the use of software-driven electronics. For example, digital cameras use embedded software to stabilize images or optimize picture quality. Cell phones have digital multimedia and mobile gaming. Home appliances have software-powered "brains" designed to conserve energy usage by optimizing washing/drying cycles. Similarly, the use of advanced safety features, infotainment centers and service diagnostics have turned many automobiles into "processors on wheels".

Smaller, more powerful hardware, such as system-on-chip and digital signal processors (DSPs), low cost, denser memory, enhanced functionality and networking capability have increased the importance, complexity and size of today's embedded software applications.

Along these lines, software-powered digital photography has dramatically changed the way we capture, enhance, use and share visual images. High-end cameras leverage software-enabled functions to capture both high resolution images and action video and encode/decode this data in multiple formats. Even lower-end cameras use software to monitor and enhance image quality, display data from an infocenter and send data to multiple output devices.

Software as a competitive advantage

Recent McKinsey and Boston Consulting Group research studies reveal the dominant role that software components play in delivering innovative products. Companies in many industries see software-based features as a way to address market pressures, differentiate their products and gain competitive advantage. As the accompanying charts indicate, the majority of innovations and new revenue is expected to come from the use of embedded software and electronics.



These studies indicate that in virtually every market segment, software development constitutes a large and growing portion of industry's R&D expenditures. In fast moving markets, such as the cell phone segment, embedding high-tech electronics, time-to-market and new functionality are critical success factors.

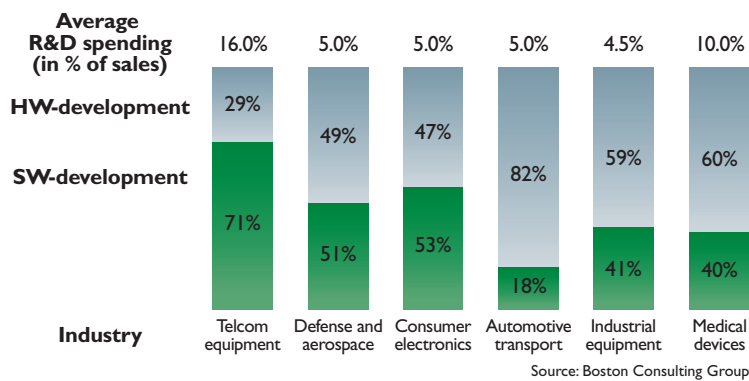
Product software: A source of competitive advantage

Averages from major products per company in benchmarking sample

Key product differentiators	SW driven	HW driven	Weight of differentiators
Reliability	71%	29%	8%
Ease of integration	74%	26%	12%
Usability/flexibility	71%	29%	12%
Features/functions	58%	42%	36%

Cost
Design
Weighted average differentiation

Large fraction of R&D expenses in high-tech industries devoted to product software



Isolated software development and today's product problems

As already indicated, the key value drivers for product manufacturers are innovative features, accelerated development time and improved product quality/reliability. However, because of the difficulty in managing globally dispersed design teams, software development's historic isolation from other domains in the product development process and the increased complexity of today's software applications, it is becoming more difficult to achieve those goals. According to one independent survey, half of all software projects are at least a month late or cancelled.

"15 percent of projects fail outright; 51 percent are challenged"

The Standish Group

Software-related product delays and recalls are also common, resulting in missed market opportunity or huge warranty costs. Frequently, delays or recalls are caused by software not working correctly when it is merged with the rest of the product – or when design changes are not properly communicated during the development process.

In fast moving industry segments, such as the high-tech cell phone and camera/imaging markets, bringing a system to market as early as possible is often more important than minimizing overall development costs. To accelerate product development and improve quality while reducing cost, design teams must be able to re-use proven design modules. Re-using software enables companies to speed up systems production since both development and validation time can be reduced by implementing effective re-use initiatives.

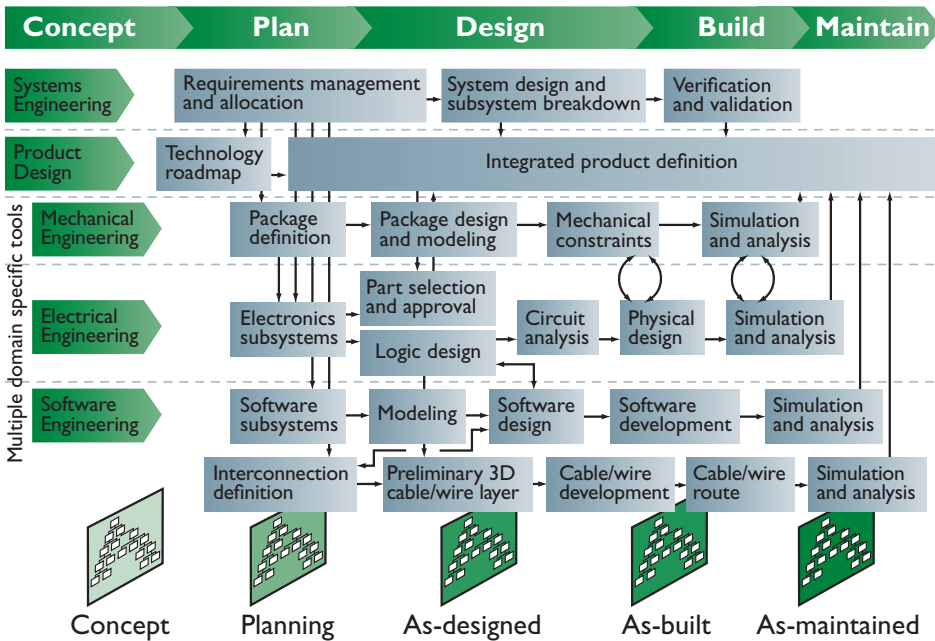
Mechatronics (electromechanical design)

Problems in software development and its integration with the rest of the product are serious concerns for product manufacturers who introduce software into their development process. “Mechatronics” and “electromechanical design” are terms frequently used to describe the products and systems developed from the synergies between software, electronics, mechanical and control system technologies. Regardless what nomenclature you use to describe this process, the increased electronics and software content being added to your next generation products requires you to integrate these domains and intelligently manage their interdependencies.

A recent IEEE magazine article pointed out that product makers are getting pretty good at component engineering – but still are deficient at systems engineering. As the article indicates, “90 percent of electronics components function as designed, (while) 50 percent...fail when integrated with their systems.”

To address software’s increasing complexity and reflect its interdependencies, today’s product makers must be able to manage the software lifecycle in the context of the entire product, including its software, electronic, control system and mechanical components. Two major requirements drive the integrated product development process:

- Componentization of software into management objects that have their own lifecycles
- Management of the engineering tools and processes used in the software lifecycle



Managing the entire electromechanical product development process

Software as an integrated part of the whole product

The ability to track, manage and configure software as an integrated part of the product lifecycle process requires a single source that builds relationships between software components (software “parts”) and the rest of the product. By treating software as a part, software functions are mapped with functional and non-functional product requirements, which will drive the systematic validation of the software’s interaction with other system components.

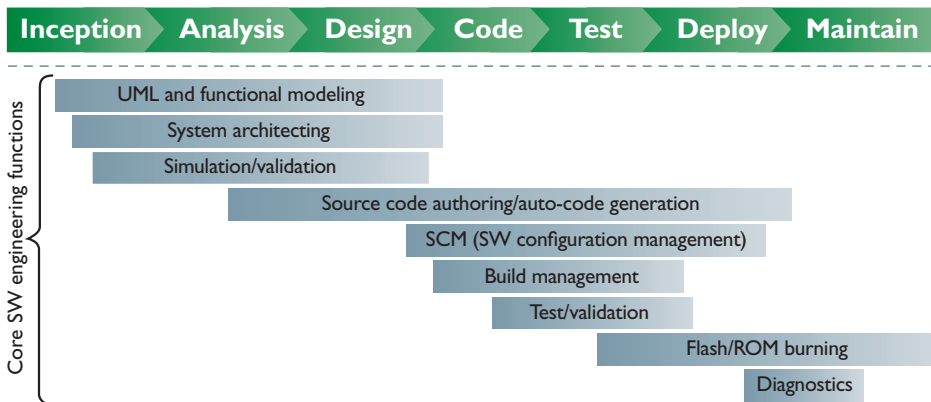
By treating software as a part in the product configuration and establishing the appropriate dependencies and compatibility during product development, product teams can ensure that:

- Software identification, auditing, accounting and configuration are enabled
- Various software modules are compatible with one another
- Software is compatible with the electronics hardware
- Changes to software and hardware modules are managed from inception to end-of-life
- Impacts of software changes are identified before changes are made
- Software development implements features based on a requirements-driven process

Software development

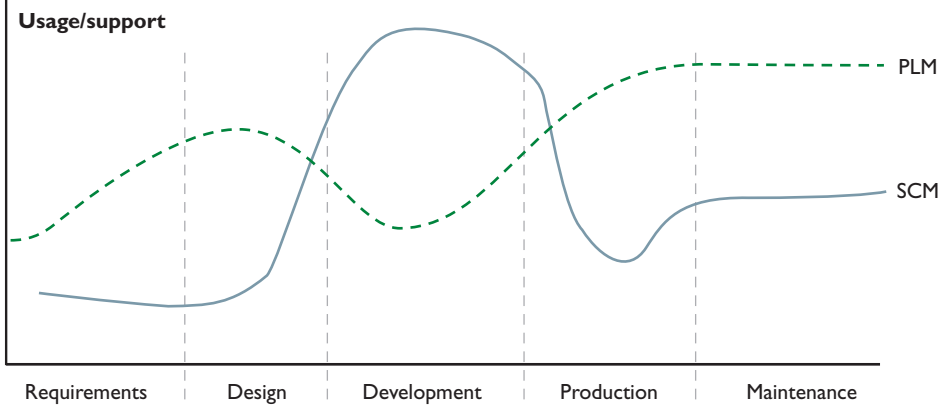
To analyze the existing software development environment in detail, you can segment the environment into core software development tools and processes, such as system architecting, UML and functional modeling, simulation and validation, source code generation, build management, binary management, and flashing or production.

However, software development is only one aspect of the software lifecycle and a single component of the overall product configuration. Dozens of other individuals around the globe, such as project managers, QA team participants, hardware engineers and others, are involved in the product development process. These stakeholders require access to a wide variety of information and documentation relating to software projects.



Unlike standalone software configuration management (SCM) or application lifecycle management (ALM) solutions (which only focus on the software development portion of the software lifecycle), product lifecycle management (PLM) solutions address and integrate the entire product development process.

PLM/SCM system support for software during product lifecycle

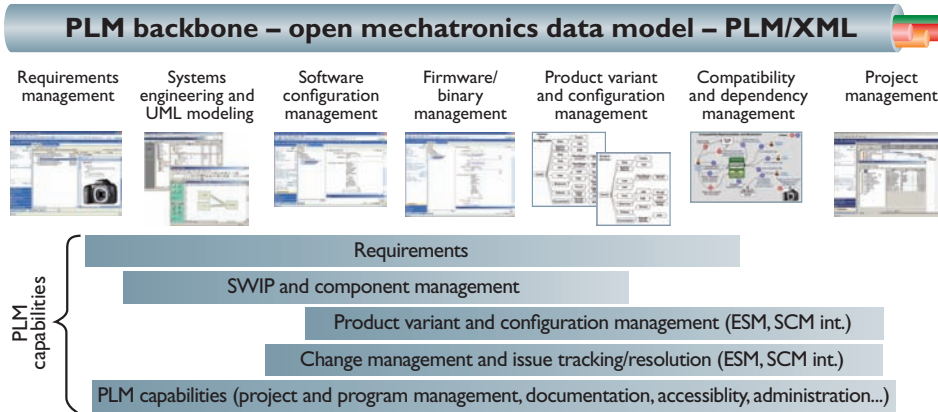


As the accompany graph indicates, the usage and support provided by PLM and SCM/ALM tools varies during the development of software and the lifecycle of a product. In the initial phases, PLM is needed for requirements management, system design and hardware development. In contrast, SCM/ALM systems are primarily used during the software development phase.

As development nears completion, PLM plays a more dominate role in the configuration, production, support and maintenance phases of the product. From a purely functional point of view, the PLM and SCM environments complement each other to address the entire software lifecycle. As a result, PLM solutions like Siemens PLM Software's Teamcenter play a crucial role in ensuring that your company is capturing, implementing, developing and maintaining what your customers want and what they will actually buy.

► Using Teamcenter to manage your software cycle

To ensure your company is designing, implementing and maintaining products that your customers want, Teamcenter-enabled PLM solutions establish a single-source of product and process knowledge for your product development environment. This knowledge and Teamcenter-driven processes tie your software, electronics and mechanical domains and their tools into an integrated whole. Teamcenter also provides the cross domain applications you need to perform requirements management, systems engineering, component management, configuration management, change management, build management, and product and project management on a global basis.



Teamcenter solutions for the software lifecycle

Teamcenter's software lifecycle management capabilities

Teamcenter capability	Function
Requirements management	Captures and traces software functionality, ensuring design teams are implementing what your customers want
Systems engineering and UML modeling	Enables your development teams to define the system and model's functional behavior; improves quality and streamlines the development process by providing all stakeholders with a system-level understanding of the product and how each of its components fits together
Software configuration management	Leverages integration with third-party software configuration management solutions, enabling development teams to facilitate software change management and elevate change management to the system level, thereby improving productivity across the software development lifecycle
Software dependency and compatibility management	Manages the interdependencies of multiple signals, binary-to-binary, binary-to-hardware and hardware-to-hardware, thereby ensuring product quality
Product configuration and variant management	Provides a holistic approach for controlling product changes at the system level; defines which software modules can or should be flashed onto processors for various product configurations, thereby improving product re-use and ensuring product reliability

Teamcenter capability	Function
Change management	Incorporates software into the overall change management process and verifies software changes in relation to the entire system, enabling your teams to analyze change impact on the basis of product dependencies and thereby improve product reliability
Product and process management	Allocates resources, defines schedules and hardware/software dependencies and monitors progress to enable your teams to meet product cost and delivery schedule objectives
Multi-site design collaboration	Manages both hardware and software product development projects on a 24/7 basis by sharing data across multiple sites

Whether you have a single development site or are operating on a global scale, Teamcenter provides the PLM backbone and mechatronics data model that addresses the needs of the software lifecycle. Teamcenter meets these objectives by facilitating system-level integration and enabling you to share and manage data for a wide variety of internally developed and integrated third-party applications. By addressing the software lifecycle as a piece of the whole product, Teamcenter accelerates product introduction, lowers cost and improves quality.

Teamcenter’s requirements management capabilities

According to a recent study by Standish Group International, “only 67 percent of originally defined features show-up in the finished product, of those 45 percent are never used.” Failure to capture, document and link customer requirements to specific implementation projects predisposes your project to constant redefinition, increases costs and reduces product quality and reliability. This failure also extends the software development process as unnecessary features are added to the product. Just as importantly, quality assurance teams find it harder to test for required functionality when customer and market requirements are either unknown or unrelated to the product’s design elements.

To deliver testable, high quality software, developers must fully understand a product’s requirements and share a common vision of the product with other members of the development team. There also must be a clear association of product requirements to the development project.

For example, high-tech and electronics companies may be required to deliver a camera that supports a variety of recording formats, pixel specifications, resolution rates, file sizes, aspect ratios, and image types.

Teamcenter enables you to capture and import product requirements from a variety of sources, including Microsoft Word, Excel and Visio, as well as from many other applications. The captured requirements can then be displayed in a familiar Microsoft format. During this process, requirements are extracted from these source documents and stored in Teamcenter while the original structure of these documents continues to be maintained.

Teamcenter provides complete traceability back to the source. It also allocates or links individual requirements to subsystem-level and component-level requirements. This level of traceability improves product quality and test coverage while eliminating feature creep and unnecessary rework that leads to schedule delays and cost overruns.

You can also leverage Teamcenter–managed requirements in enterprise-level processes, including workflows, change management processes, review/approval cycles and rules-driven procedures, as well as to support your options and variant management initiatives. This approach enables you to include software requirements in your enterprise knowledge management system.

Teamcenter's systems engineering and UML capabilities

Systems engineering. Teamcenter's systems engineering capabilities enable you to better understand what an entire product or system is suppose to do, how it will be constructed and how its hardware and software components are expected to interact. Product teams leverage these Teamcenter capabilities to create high-level views of the system. These views can then be linked to customer requirements and program constraints, thereby providing whole product visibility for cross-product or cross-view optimization.

Software architects use systems engineering to define the system and its subsystems with "building blocks" that define the product functions and software modules. These building blocks associate the system's functional and logistical requirements to the physical implementation of its software features.

Teamcenter enables software development teams to simulate the communications, controls, signal and video interaction required to ensure a system does exactly what it is designed to do in a consistent and reproducible manner. From this systems-level view, design teams can see how a change ripples across other parts of the system.


Just as importantly, the entire product development organization now has a better understanding of a product's true quality definition. This detailed level of system definition, modeling and simulation helps software development teams create the well-defined, high-quality reusable software modules needed to accelerate your new product introduction processes.

For example, a leading high tech and electronics company uses system engineering to decompose a camera into building blocks, such as the camera's body, electronics and software. Since the camera's development is driven by product requirements and performance targets, the product team segments the software blocks into specific capabilities for picture mode, shutter speeds, recording format and other factors. Where needed, the product team can further segment the subsystems (e.g., segmenting the picture mode into other sub-blocks, which will require software modules for jitter stabilization and format encode/decode). Systems engineering then enables the product team to simulate a change and see how picture resolution might affect the software module controlling jitter stabilization and the memory needed for the electronics.

UML and SysML modeling and simulation. To cope with the rapid growth in software complexity and lines of software code per device, many developers are using a model-driven development (MDD) strategy. The use of these models helps manufacturers deal with increasing software complexity by raising the level of abstraction to the behavioral level and communicating through a standard graphical language that simplifies these problems. Two common modeling languages are UML and SysML.

Teamcenter integrates UML and SysML modeling applications such as Sparx Systems Enterprise Architect and IBM/Telelogic's Rhapsody to help developers separate product behavior from product functionality. By using models to raise the level of abstraction, these models enable developers to generate and validate source code more efficiently. The Teamcenter integration allows developers to execute UML/SysML models with the same code that's run on the target system.

By providing traceable links to specific product requirements, Teamcenter's systems engineering and behavioral-level modeling tools enable software developers to design, implement and test high-quality code in a MDD environment. This approach shortens development times, improves code quality and increases the potential for re-use on other projects.



System simulation. Many products with embedded software include communications or control functions whose performance varies depending on a variety of factors, such as power, time or temperature. To understand the effect these factors will have in the product, development teams must be able to simulate them prior to implementation.

Teamcenter's integration with Matlab/Simulink enables software developers to verify their designs by creating algorithms and testing a variety of time-varying systems, including communications, gaming, controls, signal and video processing. In essence, the Teamcenter integration establishes an environment for the multi-domain simulation of dynamic systems. As with other Teamcenter integrations, traceability is provided from the requirement down to the model that implements it.

Teamcenter's software configuration management capabilities

To address the specific software configuration management needs of the software development team, Teamcenter integrates with ClearCase. Through this integration, information about the software design elements created under Rational ClearCase is brought into the single PLM environment that you manage with Teamcenter.

Software developers use ClearCase to perform all of the functions typically associated with the source code development process, including defining development workspaces, branching and establishing baselines. In this scenario, the software source code and configuration files remain in ClearCase so as not to disturb the work environment of the individual software developer.

Along these lines, consider the way high tech and electronics companies might use these capabilities. Working in their native ClearCase application software, developers establish the views that contain the image enhancement software files they are developing. As the development process progresses, ClearCase creates and maintains a list of all objects, files and versions required to build or compile the image enhancement module code. As the builds are constructed, the metadata associated with the image enhancement application and configuration files are checked into Teamcenter – where they will be linked to other parts of the product and made available for other product team members for viewing.

Once the software elements are ready to be shared on an enterprise basis, the integration links these definitions to the overall product configurations managed under Teamcenter through a standards-based exchange process. In essence, the integration provides a window into the software development environment, seamlessly presenting ClearCase information as if it was stored in Teamcenter. At this point, users who do not regularly work with ClearCase, such as project managers, QA team members and design managers, can open and access views and VOBs, or simply query the captured ClearCase data without tying-up a ClearCase license. Product managers can use product configuration to define and identify software parts involved in a particular product variant or software modules required to execute downstream processes, such as build management and media packaging.

Structured workflows and change processes are created within the Teamcenter environment. Similarly, requirements traceability can be established to software modules (source code and binaries). The Teamcenter integration allows software developers to participate in an integrated, enterprise-wide design environment, where the code they generate can be leveraged by multiple team members. This collaborative approach improves productivity while increasing product quality and reliability.

Teamcenter's software dependency management capabilities

Teamcenter's Embedded Software Manager enables the software binaries generated from compiled source code to be treated as a "part" in the product bill-of-material. Development teams use the Embedded Software Manager to manage the complex dependencies between signals exchanged from one system to another, one binary to another, binaries to electronic processors and processor to processor.

Signal management. Guiding the behavior of even the simplest of software features requires the generation of multiple signals or messages. To ensure all the signals are accounted for and messages being generated are correct for each feature, development teams need to document and track the generation, source and receiver of each signal. Teamcenter enables developers to create and associate messages and signals when building the product's electronic network. This approach helps development teams manage the communication matrix and identify the dependencies between binaries.


Dependency management (binary to binary). The execution of many embedded software features may require the generation of multiple binary files, such as application software, configuration files, calibration files and bootloaders. To ensure that correct and compatible versions of these binary files are used in conjunction with one another, Teamcenter enables developers to view, manage and track the interdependencies within and among each of these software modules.

Dependency management (binary to processor). Incorporating, or flashing, the software part (binary) into its hardware counterpart (electronics processor) is another critical step in the software lifecycle. When software updates are required, technicians need to know all of the software in the assembly that is affected by changes to the updated module. This enables the technicians to identify and update the corresponding ECUs at the same time. By tracking and managing software as a part, the correct versions of software can be transferred to the flashing station where the new hardware/software assembly is made ready for installation – or when software updates need to be incorporated into existing assemblies.

Dependency management (processor to processor). Some products employ a more distributed approach to the software functions being performed. In these situations, one processor may require communication with, or input from, one or more other processors. To ensure that your development teams use processors with compatible software, Teamcenter tracks and manages the interdependencies within and among multiple processors. This capability is especially useful when service personnel update an ECU's software, or need to determine what replacement parts are compatible across the system when a problem is identified in one of the system's multiple ECUs.

This detailed level of dependency management enables software developers and product managers to exercise greater control in configuring, tracking and managing software binaries. During development, these users can participate in structured workflows and change management processes. When product problems are encountered, users can trace the defects from the software binaries and their associated processors back to the original source code, systems definition, and product requirements if necessary

Going back to this white paper's camera and a simple exposure function example, many signals are processed by the system to ensure the proper exposure for a photograph. The light source is measured, which triggers information for shutter speed and Fstop setting. If either function is unable to adjust enough – or if it has been overridden manually and a good quality picture is not possible, information is sent to the flash to help increase the amount of light on the source. In this situation, it is important to identify and track the sender and receivers in the signal matrix.



Suppose then that the product development team is working with multiple camera models that have different features. In situations like this, developers must ensure that the software modules which perform each function are correctly matched and included. For example, a lower-end camera may only take still photographs requiring image enhancement and an encode/code function that stores the data in jpeg format. In contrast, another model capable of capturing action video will require the encode function to support jpeg, RAW and mpeg4 formats. The correct configuration and configuration files must also be loaded into each camera to ensure that the proper exposure settings and calculations are available for the best picture quality. Should warranty issues arise, the local service center needs to know exactly what software modules to use and what other configurations are compatible.

Teamcenter's configuration and variant management capabilities

Within today's products, one platform can have multiple product options driven by software. These software combinations and variations grow even more complex as various software modules are re-used on other product platforms and across multiple projects. Too frequently product quality and reliability problems result when unproven software modules are integrated or when the configuration mistakes occur.


Teamcenter's configuration management capabilities make it possible to relate that software file to the product usage, model and platform it is used in. Teamcenter enables product teams to understand important attributes about the software file, such as its programming protocols, network protocols, memory sizes, disk file sizes and the memory address space of the hardware.

Validation algorithms can report discrepancies between engineering metadata, software files and their in-service use. These reports help product teams detect bad software files at the source and get them corrected before they are distributed further. Issues such as header information, memory size consumed, format of the binary file, binary file part number, certification document, test cases and configuration file are checked against the product's requirements. Teamcenter's structured lifecycles ensure that changes (to address field concerns) are audited from early in detection process to final engineering resolution.

By incorporating the software development process into the overall product lifecycle and facilitating a complete configuration management process, Teamcenter helps product teams improve overall product quality. Teamcenter enables the product team to define each product option and variant in the same product structure and establish all of the appropriate connections and dependencies between each part in the product configuration, including its software, hardware and documentation links.

Teamcenter helps team members quickly identify relevant data for any product variant and avoid data duplication. These connections and dependencies also help product managers identify what other parts of the product will be affected when the software changes, as well as how specific electronic or mechanical part changes impact the software.

Software configuration and calibration data management helps team members identify and resolve issues by merging parameters. It also enables teams to use the same data model across multiple models and types and logically group parameters for better performance and access control. This approach enables software development teams to minimize maintenance and schema changes.



Software configuration management also promotes the use of self-contained software entities, which can be leveraged as building blocks while designing larger systems. This practice exposes well defined ports and interfaces between software components that can be used by other systems. It also enables software content to be used an arbitrary number of times within the same systems, as well as in different systems.

This level of product configuration component management makes it possible to identify and re-use proven software modules and other intellectual property. Since the initial use of any software module reveals implementation faults (which are detected throughout the software lifecycle and then fixed as the software matures), the re-use of more mature software should be more dependable than newly developed software. Teamcenter's ability to facilitate whole product configuration management promotes efficiency across the development process, improves product quality and lowers warranty cost.

Once again turning to the white paper's camera example, the product in question consists of a camera body (used for multiple models), an assortment of lenses, manuals, straps, and electronics hardware and software that perform multiple functions, such as recording formats, shutter speeds, Fstops, action video and jitter stabilization. By managing all the various components, Teamcenter can track and manage multiple product configurations with a variety of variants and options. By viewing a single whole-product definition that includes the camera body, lens, manual, strap, electronics hardware and software for jpeg recording format, shutter speeds, Fstops, and image enhancement, marketing and sales teams can configure an entry-level camera for the novice user. Using the same basic set of parts with electronics hardware and software for jpeg and mpeg-4 recording format, action video and jitter stabilization, these teams also can configure a camera for the professional photographer.


Teamcenter's change management and issue management capabilities

When Teamcenter's comprehensive issue and product change management capabilities are integrated with an SCM system, product managers, project managers, hardware developers and software developers can identify and document open issues and resolve them in the context of the entire product. These integrations provide end-to-end software change traceability by tracking what code changes have been implemented to address specific issues (defects and enhancements).

A Teamcenter-SCM integration provides a key benefit by ensuring that the assessment of the quality of the product sources is consistent with the actual product sources themselves. This approach helps product managers plan the tasks associated with delivering the product. Second, it enables SCM users to quickly and easily navigate between related information that appears in both the defect tracking system and SCM system.

A Teamcenter-SCM integration can provide crucial strategic planning information, including information that enables project managers to determine software status and establish quality-related metrics that apply as the product evolves. In this type of environment, project managers can leverage an integrated solution to:

- Determine the rate of issue finding and fixing
- Identify which project/product areas are most costly to maintain
- Assess the quality of a release candidate
- Ascertain which issues have been fixed and which issues are still present in a release candidate



When issues are discovered in any product, the problem in question is documented in Teamcenter and, if necessary, a change request is distributed to an appropriate “owner” for investigation. The owner diagnoses the cause of the problem as a software or hardware issue and determines the nature of the fix. The owner can use Teamcenter’s configuration, compatibility, dependency and signal management capabilities to identify what other parts of the product will be affected.

Next, a responsible party in product development identifies specific work tasks, defines a project plan and assigns resources to the task at hand using Teamcenter’s project management capabilities. For software related issues, the developer will use the links created from the binary to identify the specific source code module within which to implement the fix. Simultaneously, hardware engineers can begin implementing the changes needed for any electronics or packaging parts.


Once the changes are incorporated and builds are generated and checked-in, documentation is updated and attached to the change. Then, Teamcenter notifies the quality assurance (QA) group. A QA analyst can test the software fixes in conjunction with any other parts of the system that were affected. Finally, as the change review board validates the change, the change request is closed and appropriate notifications are routed.

Because Teamcenter is able to facilitate an end-to-end issue tracking and change management process that includes a complete and accurate product change document and change package, your development organization eliminates errors, reduces costs and ensures total product quality.

Continuing the white paper’s camera example, suppose a customer encounters a problem where the photographic images being captured are consistently under exposed. This may indicate a defect in the image enhancement software. As part of a systematic problem reporting process, Teamcenter is leveraged to capture this issue and route it to a design engineer. Using Teamcenter’s dependency and product configuration management capabilities, the engineer determines that the light metering software and the electronic flash hardware also need to be modified. Following the configuration links, the software developer quickly identifies the specific source code modules that have to be modified. Once the hardware and software engineers implement the changes and check-in their data, the changes can be tested and approved for release. This Teamcenter-driven process includes a complete set of change documentation that reflects all aspects and parts included in the change.

Teamcenter’s program and project management capabilities

To improve productivity, an integrated approach to program and project management allows product, project and software managers to more effectively integrate, schedule and allocate software resources across the entire product lifecycle. Teamcenter’s resource and schedule management capabilities enable product teams to quickly link both software and hardware related project tasks to the requirements that drive them. Teamcenter can assign and track resource workloads (human and tools), identify dependencies and resource constraints across the product, and determine the progress and current status of each item defined in the project. Dashboards provide critical information that accurately displays the interrelationship of this data. Each of these capabilities helps mitigate project risks and eliminate last-minute surprises by balancing and optimizing the project’s time and resource constraints,



Companies that implement best practice approaches, such as CMMI or stage-/phase-gate, into their product development processes can use Teamcenter's program and project management capabilities to tie project tasks and resources into intelligent schedules and managed workflows. To consolidate multiple projects into a virtual workspace, Microsoft Project Desktop has been integrated with Teamcenter. Through this integration, users can view the interdependencies among multiple projects and perform critical-path analysis, as well as examine costs and resources at the program level.

Going back to the white paper's camera example, the product manager for the camera product line is able to identify the hardware, software, documentation, testing and manufacturing costs and resources for a given project. As the project proceeds, new requirements arise, or unexpected issues appear (such as slippages in the image enhancement software), the product manager can quickly identify what impact these activities may have on the overall schedule or whether team members from other software projects are available to assist with the development effort.

Teamcenter's multi-site design capabilities

Today's development teams and product suppliers operate on a global scale. The effective management of project schedules and resources may require you to provide continuous support for your software and hardware development projects by moving them from one development site to another.

To address this requirement, Teamcenter supports multi-site design collaboration through two key technologies:

- A shared object repository providing a single database instance shared across multiple sites
- A network-based multi-site data replication environment with import and export capabilities

These two technologies enable any site that wants to share an object to "publish" the object in the shared repository by indicating what network-based sites are authorized to import that particular object. The destination site then imports the objects into its local database, transferring ownership of the data.

Widely dispersed design teams are able to use these technologies to more readily share data, collaborate on design issues and coordinate development activities. By sharing objects at multiple sites, Teamcenter lets you maximize the network's data access performance and your global productivity.

Turning to the white paper's camera example, as schedule slippages in the image enhancement software are encountered, the product manager determines that other development resources must be added to the project. These resources are distributed across multiple geographic zones. Teamcenter's multi-site collaboration capabilities enable the project to be worked on a 24/7 basis. By leveraging Teamcenter's multi-site support capabilities, the entire software development environment and associated files for the image enhancement software, or any other software project, can be shared across and among each development site.

▶ PLM – a key component of the software lifecycle

Software content in today's products is expanding rapidly. To address this reality, companies that value product and process innovation are looking to incorporate the software lifecycle into their product development process by:

- Facilitating global design collaboration
- Integrating their domain-specific innovation tools
- Leveraging powerful cross-domain applications
- Integrating and managing the whole product lifecycle

By using Teamcenter to manage software as part of the entire product lifecycle, best-in-class companies accelerate the development process and ensure product quality and reliability. Just as importantly, this state-of-the-market approach ultimately enables product makers to meet, and even exceed, their delivery schedule and product cost targets.



About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 5.9 million licensed seats and 56,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

Siemens PLM Software

Headquarters

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
972 987 3000
Fax 972 987 3398

Americas

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
800 498 5351
Fax 972 987 3398

Europe

3 Knoll Road
Camberley
Surrey GU15 3SY
United Kingdom
44 (0) 1276 702000
Fax 44 (0) 1276 705150

Asia-Pacific

Suites 6804-8, 68/F
Central Plaza
18 Harbour Road
WanChai
Hong Kong
852 2230 3333
Fax 852 2230 3210

www.siemens.com/plm

© 2009 Siemens Product Lifecycle Management Software Inc. All rights reserved. Siemens and the Siemens logo are registered trademarks of Siemens AG. Teamcenter, NX, Solid Edge, Tecnomatix, Parasolid, Femap, I-deas, Velocity Series and Geolus are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks used herein are the property of their respective holders.