Effectively utilizing project, product and process knowledge

Christof Ebert a,*, Jozef De Man b

a Vector Consulting Services, Stuttgart, Germany
b Alcatel-Lucent and Ghent University, Antwerp, Belgium

Received 26 July 2006; received in revised form 16 June 2007; accepted 17 June 2007
Available online 4 September 2007

Abstract

Improving project management, product development and engineering processes is for many companies crucial to survive in a fast changing environment. However, these activities are rarely integrated well due to the diversity of stakeholders with individual knowledge about projects, products and processes. This case study shows how Alcatel-Lucent over time achieved effective interaction of engineering processes, tools and people on the basis of a knowledge-centric product life-cycle management (PLM). Starting from identifying project, product and process knowledge, we show how they can be effectively integrated for best possible usage across the enterprise. The case study provides insight into how to best embark on PLM and how to effectively integrate product development with supportive tools. It describes how the concepts can be transferred to software engineering teams and IT departments in other companies. Concrete results from several product lines, such as efficiency improvement and better global development underline the business value.

© 2008 Elsevier B.V. All rights reserved.

Keywords: CMMI; KM, knowledge management; PLM, product life-cycle management; Process improvement; Project management

1. Introduction

Systems and software development requires profound technology and product knowledge combined with effective usage of teamwork, processes, methods and tools. To reduce complexity and boost performance, it looks just rational to put a small group of software engineers at one place, share the project objectives, agree processes and technologies to apply and let the project run. Unfortunately, reality is different, especially in times of global development of solutions involving many different competences, components, interfaces, processes and tools, all contributing to increased complexity.

Today’s global software engineering with short release cycles, interacting product lines and product and solution development from many sources has advantages but also drawbacks. While the positive side accounts for faster cycle time, time-zone effectiveness and reduced cost, we should not close our eyes for severe disadvantages. Working in a globally distributed project for instance involves overhead for planning and managing people and suffers from language, psychological and cultural barriers. It often trans-lates into knowledge needs which are not satisfied in due time due to these obstacles.

Of growing importance in such ever-changing environments is effective management of software engineering knowledge. We will talk here less about specific technical knowledge on a dedicated software technology, which is typically provided from various sourcing channels. We look rather into the knowledge linked to the key assets of a company, namely project, product and process knowledge.

This article describes how Alcatel-Lucent over the past years has mastered challenges in product development by a clear focus on product life-cycle management (PLM) and engineering knowledge management. Specifically we will address what we did to integrate these two aspects, how our results can be transferred to other settings and what the benefits are. We will share results from several years of gradually building environments and a culture of
sharing software engineering knowledge. Integrating and utilizing project, product and process knowledge is an ongoing journey, involving continuous process improvement, on which it builds. Specifically we were interested in supporting product development and project management with efficient processes in order to substantially improve performance along the product life-cycle. The challenge we faced was centered on more effectively utilizing software engineering knowledge as a mandatory driver for business success in software-intensive product development.

The article is organized as follows. The next section provides a brief introduction to PLM and KM. Section 3 summarizes related work. Section 4 describes the environment of the case study. Section 5 briefly introduces the different elements of software engineering knowledge. Section 6 summarizes lessons learned from integrating project, product and process knowledge to a useful and used environment. Section 7 provides concrete results and the value to the enterprise, which we achieved over the past years. Finally, Section 8 concludes this industrial case study and suggests future work in knowledge management and PLM.

2. Product life-cycle management and knowledge management

IT and software companies are challenged with cost pressure, decreasing cycle time and very rapid technology and feature evolution. At the same time, several of these companies have over loaded their R&D capacity and capability with many technologies, complex roadmaps of multiple variants and unclear value contributions of products and new features. Portfolio Management is still insufficiently practiced for managing IT assets in a corporation [1]. The primary (and often sole) measurement is the financial performance. Often, too many projects run in parallel, without concrete and quantitative objectives and without tracking where they are with respect to expectations. Project proposals are evaluated in isolation from other ongoing activities. Projects are started or stopped based on local criteria, not considering the global trade-offs across all projects and opportunities [2]. Only one-third of all software engineering companies have techniques to effectively share knowledge on their products and development projects [1,2,20].

Fig. 1 shows the many challenges a software enterprise or IT department are confronted with, such as competition, cost reduction, quality and cycle time improvement but also technology hypes to be assessed and solutions to be created. The challenges are different for different companies but must be mastered in order to remain competitive.

Mastering these challenges demands continuous improvement along three axes, namely:

- **Consolidating**: Focusing on few relevant core products which distinguish the enterprise in the market; maximizing the business value from these products; value creation from solutions instead of mere shrink-wrap products; establishing eco-systems of customers, suppliers and contractors.
- **Industrializing**: Mastering projects, processes and knowledge by intelligent collaboration. This is the focus of this article and case study.
- **Globalizing**: This is optional and depends on the market to be addressed and size of the corporation. Small and medium-sized enterprises might use global suppliers, while larger enterprises establish a global software engineering capability. The third axis is pointless without the first two, as many examples of failed global development show [2].

We mastered these various challenges by embarking on product life-cycle management (PLM) several years ago – long before it became popular in IT. We were successful with PLM because we enriched it with a knowledge-centric approach instead of following a tools-centric approach as it happens often today [3]. A clear and objective-driven improvement strategy driven by business needs, such as cycle time reduction or improved predictability, avoided that we would build a theoretic knowledge management or process improvement framework that would be of little practical use.

We will briefly define product life-cycle management (PLM) and knowledge management (KM) to clarify principles that we applied. PLM is the overall business process that governs a product or service from its inception to the end of its life in order to achieve the best possible value for the business of the enterprise and its customers and partners. It combines processes, people and tools for the effective engineering of products – from their inception until end of service. KM is the process that deals with systematically eliciting, structuring and facilitating the efficient retrieval and effective use of knowledge. It involves tacit knowledge of experts and explicit knowledge, codified in procedures, process and tools. KM stretches from know-how to know-what and know-why.

In our view, PLM and knowledge management must mutually support each other. Knowledge-centric PLM
must bring together process, product and project knowledge from a learning perspective and thus improves engineering productivity. Fig. 2 illustrates this view of PLM and the need for good knowledge management. To work effectively, people need to handle a variety of different forms of knowledge from processes, technologies, projects and products. PLM helps to integrate those along the entire life-cycle of a release or product or beyond to an entire portfolio.

Our ambition with knowledge-centric PLM is to enable R&D organizations to more effectively handle knowledge within the daily operational activities. Often information is reused, but with high redundancies or manual overhead. At times, the redundancies create rework – as things are not done right first – or even errors that remain in the product. An example is handling of product requirements and business case information. If this information is not shared between stakeholders along the project, the development will end in exceeding time and budget with gold-plating and rework due to having the wrong focus.

Being able to not only reuse information but also embed knowledge into integrated workflows for specific tasks, generates immediate returns by making engineers more flexible. Consider the time and effort necessary to move engineers from one project to another. Having standard knowledge management around a standard product life-cycle reduces the learning curve to allow focusing on real technical challenges instead of organization overhead. We should however be aware that knowledge management is not reduced to workflow management, which we consider a facilitator for effective knowledge management.

Knowledge management systems offer different perspectives to allow for instance navigation based on work products, roles or processes. Technological innovation and successful new products are the results of well-oiled relationships and tightly choreographed teamwork, not only amongst the different business units or divisions of a corporation, but also between autonomous and geographically dispersed enterprises.

To ensure effective execution on project and product level, knowledge management should be closely linked and integrated in the company’s product life-cycle. Typical life-cycles might follow ISO 15288 (systems life-cycle [25]) or ISO 12207 (software life-cycle [26]). They have in common a gating process between major phases based on defined criteria. These gates enforce evaluating the overall status (both commercial and technical) and deciding on whether and how to proceed with the project. For illustration, we will use a simplified product life-cycle as it applies to software, systems and IT projects. Fig. 3 shows for each of these standard life-cycle phases a sample of decisions and actions that are taken to ensure progress against commitments and objectives. It highlights the various knowledge needs to guide those decisions. Our ambition to use PLM as a vehicle for synchronizing knowledge needs becomes obvious from this introductory picture.

In the early years, we nicknamed this initiative “e-R&D” [3], but later on called it PLM as this term got widely used in industry over the past years. The vision of PLM is to provide outstanding R&D performance. This is achieved through the three drivers of PLM, namely management efficiency, process excellence and technology effectiveness. Management efficiency targets lean management practices and ensures that people take realistic commitments and deliver accordingly. Process improvement addresses the R&D and product management processes and aims to improve our process capability. Technology effectiveness leverages on management efficiency and process excellence and looks into providing the right innovative technology to address our customers’ needs. The glue to bring these three drivers together is knowledge management.

---

**Fig. 2.** Knowledge-centric product life-cycle management integrates processes, people and tools for effective engineering.
PLM as a concept and solution applies to software engineering as well as systems or hardware products. It applies to different types and sizes of companies, because it is not prescribing a solution suitable only for big companies but rather a clear focus on processes along the life-cycle. We use it for complex solutions with multiple hardware and software components as well as simple software services. For better understanding, this article narrows down the scope toward systems built primarily from software components.

3. Related work

We will look here at the impact of related work in the domains of product life-cycle management, knowledge management and process improvement. Compared to other work in this domain, this article shows two major benefits, namely:

- Knowledge management embedded into PLM offers a solution for both efficiency improvement and complexity control in technical product development.
- Our longitudinal study in an industrial context shows change management over 5 years of incremental growth indicating that such solution needs sufficient time to evolve and become engrained in an organization and its genes.

KM has reached the software engineering community in the 1990s. Software engineering knowledge was rarely adequately managed before. It was often randomly collected and stored in corporate document vaults with little or no means for efficient and effective reuse. Increasingly, KM has gained ground as a discipline that needs dedicated attention, not only from a functional but specifically from a cross-functional perspective [4,5]. Initiatives such as building a coherent software process terminology [6] or creating an initial body of knowledge [7] helped building a foundation to bring together very different disciplines. A summary of the discipline of KM and its specific challenges in the software world is provided by Burke and Howard [8] and Dingsøyr and Conradi [22].

There are some publications that deal with this subject and show how to bring knowledge and experiences into software organizations [4,5,8–13,21,22,27]. Though they provide concrete examples, they mostly deal with handling only a single dimension of software engineering knowledge. For instance, how to reuse experiences from dealing with nonfunctional requirements is very pragmatically described in [9]. Most references, however, are still more on the theoretical side than answering practical questions from day-to-day project business [4,8,10,11,14,21]. Applicability and actual application in industry-scale application development and solutions integration are not always evident in many of the above-mentioned research papers. Based on a quality improvement paradigm, Basili et al. defined the experience factory [27] which tries to condense best practices and redistribute them in new software engineering projects. But time was not yet ready for putting such ideas into industrial software development, due to lack of formalized processes and the necessary support tools. Baskerville and Pries-Heje [13] mapped the capability maturity model on knowledge management and found that such approach could be enhanced with specific needs related to KM. Specifically, they indicated how to utilize KM on top of “classic” process areas, such as training, in order to mitigate the dependencies on specific skills and employees.

An initial framework for KM needs from a software engineering viewpoint had been proposed by Martinez...
et al. [14]. This study helps in determining an enterprise’s position versus a generic framework of potential requirements. It starts with evaluating the KM-specific needs and then maps them on the specification of requirements. The difficulty however is that it remains fairly abstract at this point and offers little help to companies in building their own environment based on existing processes and environmental conditions. Aurum et al. [5] and Dingsøyr and Conradi [22] provide some useful case studies which help using KM techniques in software engineering. They both underline with many industry and scientific cases the major problems in applying KM to software engineering. There is a lack of agreed modeling languages that allow effective tagging and retrieving of information. Software is also developed typically in a rather ad-hoc way, meaning that engineers prefer to invent something new but dislike digesting somebody else’s previous work. The latter has changed with the growing interest in open source software [23] which stimulates both knowledge sharing as well as building open processes for evolving software components or tools.

**PLM** is about mastering knowledge to the benefit of management efficiency, technology effectiveness and process excellence. PLM though it heavily talks about integrating processes has focused so far primarily on tools. It emerged from hardware tools, such as product data management, underlining the need for unique referentials to achieve reuse and configuration control. Often tool vendors market product life-cycle management as hardware-centric. This should not be a surprise since the support evolved from the tradition of product data management tools with focus on component descriptions, interface descriptions and structured representation of the “bill-of-material”. Strict configuration management is an integral part of this support due to the serious impact of any change and the need to synchronize with many stakeholders. In such hardware-driven solutions, referential data, such as product data, and respective engineering change management were integrated with collaboration platforms and the resulting engineering tool was labeled “PLM”.

While PLM tools interwork with many HW design and manufacturing tools, they only recently started to look into specific software engineering environments. Examples include Dassault/MatrixOne, Agile or PTC with their hardware-centric design and data management tools, which increasingly interwork with software engineering tools, such as IBM/Rational’s Clearquest for defect tracking or Telelogic DOORS for requirements management.

We consider such tools only as components supporting part of a broader process, covering all engineering processes and the interfaces to product management, supply chain, portfolio management and service management. In fact, portfolio management and product management functions in our organization benefited from such integrated view on engineering because we could provide seamless interfaces that ease decision-making along the entire product (and service) life-cycle. Processes without tools have not much value. This was confirmed in a recent study at the London School of Economics which we fully support with our own data (see Fig. 4) [2,24].

Other approaches merely extend enterprise data management to cover some interfaces to engineering processes and label it PLM. For instance, major ERM vendors evolved their environments to support engineering document management. CRM environments since recently integrate with defect tracking tools. However, such IT-centric approaches do not go beyond interface management and do not integrate with systems and software engineering processes. Their scope is limited to interfaces and front-end processes.

We see PLM as a management principle guided by the need for effective knowledge sharing and founded on management efficiency, process excellence and technology effectiveness. We do envisage and recently see it with some vendor statements, that our knowledge-centric view on PLM will be adopted.

PLM needs both process and tools support as indicated by Fig. 4. Tuning processes, improving project management and establishing visibility on new product introduction – techniques well described by common improvement frameworks, such as CMMI [7]—will not cure this if not adequately supported by tools. The prospect of new, high-margin products, combined with the delayed impacts of resource allocation decisions, seduce product managers into starting more projects than their development resources can handle. Like in manufacturing during the 1980s, today’s software factories must focus simultaneously on the above-mentioned three dimensions, namely management, process and technology.

The tight integration of project, product and process perspectives together with knowledge management accelerates process improvement in technical product development. This was so far not described in research and shows the value of our longitudinal study. While the SEI reports in its annual summaries a duration of over 3 years as mean time from CMMI maturity level 1 to maturity level 3 (which requests defined processes and shared engineering knowledge) [6], we could achieve at Alcatel-Lucent

![Efficiency gains with improved processes and tools. Tools without processes are nothing, processes without tools are not good enough.](image_url)
an acceleration by up to 50%. We could also accelerate the ramping up of new product development, e.g. in offshore situations with different cultures involved – depending on the underlying maturity level – up to few months for a maturity level 2 or 3 organization [19].

4. The environment of this case study

Alcatel-Lucent provides end-to-end communication solutions, enabling carriers, service providers and enterprises to deploy and manage a user-centric communication service to any type of user, anywhere in the world. Leveraging a long-term expertise in telecommunications network equipment, applications and network services, Alcatel-Lucent faces today an increasing need to design, integrate and deploy solutions that are built from equipment and services of a wide range of internal and external suppliers. Effective software engineering plays a key role in this picture, as a vast majority of R&D investment today is on software technology and applications.

To stay competitive with its systems and software development, Alcatel-Lucent has over the past years put in place an orchestrated improvement program of its processes and the supporting engineering tools environment [3]. But this was not considered to be enough. We needed to effectively integrate and utilize knowledge from three dimensions, namely projects, products and processes. We have approached PLM in three implementation tracks, namely strengthened process capability, visibility and workflow integration.

Strengthened process capability is key to our implementation of PLM. Since more than a decade Alcatel-Lucent has been successfully embracing the Capability Maturity Model (CMM) and, more recently, its successor, the CMMI, both owned and maintained by the Software Engineering Institute [6,7]. We combined this model with a strong focus on business objectives and metrics to follow-up of implementation of change. In fact, a close link to Alcatel-Lucent’s business needs was a major push to focus on what is essential for customers and shareholders. For instance, several years ago, our voice networks primarily focused on field quality improvement. Rapidly changing markets are pushing the carriers to offer new services at a rapid pace. So we had to cut cycle time to less than half over few years. Such seamless shift of priorities in a consistent and sustainable way is achieved with a strong organizational process focus.

How do engineering tools come into the picture? Processes without adequate tool support remain theoretic and difficult to enforce. Our objective is to improve visibility in engineering and mastering a variety of workflows and external interfaces related to R&D. PLM must bridge the needs of process improvement with tool support. Global workflows together describe how software engineering work products are designed and maintained. They also provide the link to corporate business processes and specific tools environments. Some are entirely internal to engineering, while others are at the boundary to other functions. An example for such a workflow and related interfaces is service request management. Service requests result from field operations and some are treated within R&D to implement corrections that are again deployed to the field.

In Alcatel-Lucent, these elements are centered on a standardized product life-cycle (PLC) [15]. For well-orchestrated product launch, development, post-launch and discontinuance, all functions of the enterprise must play their role in developing and executing an integrated plan. The potential for growth as well as replacement must be assessed based on a common framework. Fig. 5 shows the product life-cycle in a simplified format and how the different product-related processes are anchored. Note that we portray the PLC as recursive and applicable to products and their individual releases. Therefore, processes such as strategy or concept repeat each other for each single instance of a product release. Our implementation of PLM covers most of these processes to some degree, with focus on visibility (e.g., progress, documents) and interface management. We will describe here the framework used and how it was integrated to PLM.

To show concrete results from using PLM in day-to-day software engineering projects, we took snapshots and samples from the PLM product release portal. This portal and the underlying PLM processes evolved since 2001 and comprise today hundreds of products with their releases covering all business divisions of Alcatel-Lucent. Project size in effort ranges between some person-weeks and several 100 person-years. All projects that had recorded valid data in the history database were used. For empirical studies, such as presented later to show the value of PLM, no “outliers” have been thrown out, thus avoiding to overlook influences from other parameters than the four process elements that we analyze here. The projects, though primarily in the domains of embedded systems or software applications, cover a wide range of the entire world of software-intensive projects. Results therefore apply to other industries than communication. This is also supported by benchmarks.

![Fig. 5. PLM provides the process architecture and the handles for collaborative knowledge management and optimized tools solutions.](image-url)
we have been doing in the field of automotive, defense and information systems.

5. Elements of software engineering knowledge

5.1. Project knowledge

Project knowledge is the knowledge about resources, functional and attribute requirements, work products, budget, timing, milestones, deliverables, increments, quality targets and performance parameters. Project knowledge is closely linked with product and process knowledge. It deals with the instantiation of processes to deliver a product.

KM for projects should address concrete use cases of the initial target communities and then gradually grow to capture more use cases. For instance, a key use case that we initially defined and applied was: “Support the project manager to retrieve information from past projects that apply to her current own project.” A project management team is typically interested to learn from previous projects in order to better manage quality and reliability. A project line manager is interested to improve the portfolio he is responsible for, and to ensure the right baselines and evolution paths are agreed and implemented to serve an ever-changing market.

PLM for projects requires a clear definition of context, scope and objectives. These parameters are closely linked to business objectives of the product line. A software project team might be interested in identifying which checklist to apply to improve coding or peer reviews. Department leaders in software organizations are interested in skill evolution that is aligned with future technology needs. They might also insist on deploying peer reviews and similar techniques to improve maintainability and become less expert-dependent.

Being able to not only reuse information but also embed the respective processes into integrated workflows for specific tasks generates immediate returns by making engineers more flexible. Consider the time and effort necessary to move engineers from one project to another. Having standard knowledge management and standard PLM reduces the learning curve to real technical challenges, instead of organization overheads. We should however be aware that knowledge management is not reduced to workflow management, which we treat a facilitator for effective knowledge management.

5.2. Product knowledge

Product knowledge is the knowledge about product features, and how they relate to other products, standards, protocols and the like. Especially in telecommunications, which is characterized by a particularly rapid technological change and uncertainty in an environment that often integrates technologies of several decades, product knowledge is the key success factor for a solution supplier.

Developers need product knowledge to reuse successful implementation strategies and actual implementation already realized before. Product management needs it to manage release roadmaps and product portfolios. It is crucial for defining technology and marketing strategy and to evaluate whether products and solutions fit in these strategies. It is also key knowledge for senior management to establish the optimal organizational structure for developing, marketing and selling products and solutions. In a technology-driven company, product knowledge is structured hierarchically and maintained by multiple stakeholders.

5.3. Process knowledge

Process knowledge is the knowledge about business processes, workflows, responsibilities, supporting technologies and interfaces between processes. In software engineering – unlike hardware engineering – this aspect of knowledge management is often neglected and as a result, activities do not scale up and performance decreases.

A common product life-cycle across the entire company is the global framework for detailed processes, some of which are defined at the corporate level but many of which are tailored in the business divisions to adapt to their specific needs. We can definitely recommend to start with a generic product life-cycle (PLC) and then continue on a more detailed level, rather than working in the opposite direction. This top-down approach applies for the entire discussion around business process reengineering and e-commerce introduction. Start with a generic (business) process, apply it in pilots and product lines, in order to improve on business processes, and then allow specific tailoring.

Sharing and reused process knowledge is the most difficult amongst the three dimensions. A small example shows the intrinsic difficulties. To successfully deliver a product with heterogeneous architecture and a mixture of legacy components built in various languages, certain processes must be aligned on the project level. This holds for project management, configuration management and requirements management. It allows tracing customer requirements that might affect several components through the project life-cycle. It allows reusing requirements in a product line. On the other hand, design processes and validation strategies are so close to the individual components’ architecture and development paradigms that process standardization is very delicate. And, for efficiency reasons, the manager of such heterogeneous project or product line surely would not like that within each small team the work product templates or tool-based workflows were redefined.

6. Introducing and using PLM

As we have learned from the article so far, industrial systems and software engineering must integrate project, product and process knowledge with a perspective on organizational learning. The answer we investigate here is
primarily centered on enabling R&D organizations to more effectively handle knowledge within the daily operational activities. Often information is reused, but with high redundancies or manual overhead. At times, the redundancies create rework or even errors that remain in the product. An example of such redundancies leading to defects is handling of product requirements and business case information.

To benefit from improved business processes, the different functions of the enterprise plus potential external partners (e.g. component suppliers, outsource manufacturing) need to agree on processes and practices. They need to have common access to knowledge, performance metrics and decision-making protocols. They need to share information, communication and underlying resources.

The barriers to such harmonization and cooperation are not to be underestimated. They range over language barriers, time zone barriers, incompatible technological infrastructures, clashing product line cultures and not-invented-here syndromes. An obvious barrier is the profit and loss responsibility that results, especially in tough times, in a focus on current quarter results and not in investments in future infrastructure. Incumbents perceive providing visibility a risk, because they become accountable and more subject to internal competition.

Processes must be easily accessible for the practitioners and managers. They must integrate seamlessly. By focusing on the essence of processes, integrating processes elements with each other and providing complete tools solutions, organizations can tailor processes to meet specific needs and allow localized and problem- or skill-specific software practices, while still ensuring that basic objectives of the organization are achieved. This is what we call managed process diversity.

Practitioners do not look for heavy process documentation, but rather for process support, that exactly describes what they have to do at the moment they have to do it. Modular process elements must be combined according to a specific role or work product to be delivered. Still the need for an organizational process, as described by CMMI maturity level 3 is strongly emphasized and reinforced [7]. To bridge this gap, different approaches have been described recently for managing process diversity [16–18]. Fig. 6 shows the solution to above-mentioned use case of supporting a project manager to retrieve information. The information collected at the product and project level is aggregated at the enterprise level with filtering by business division, business unit and product line. This allows the available information to be used for portfolio management. Managers can also drill down efficiently to detailed information about ongoing and proposed projects.

Let us now look into how we built our knowledge-centric PLM solution. In a first step, we agreed on the sets of processes and process elements that should be subject to tailoring and those that should be invariant. This was captured in a simple process meta model (Fig. 7) including milestones (when?), roles (who?), work products (what?) and activities or workflows (how?).

In the next step, we investigated which criteria would determine selection of a specific process. We identified sev-

![Fig. 6. PLM with knowledge management to zoom from a product line summary into single engineering projects and their respective project status information.](image)

![Fig. 7. The underlying process meta model and how instances are built.](image)
eral criteria that determine the layout of processes: the project size in terms of effort; the product type (for instance whether it is a generic R&D project, or a customization or maintenance project, or a prototype); specific component criteria (e.g. design paradigm, programming language, development platform, industrialization parameters related to market introduction and customer interfaces) and the process edition (the process is subject to configuration management, especially for big projects which overlap with ongoing improvement activities).

Having an understanding of the organizational processes, the next step is to understand how processes are tailored in projects or product lines, and how they link to legacy process descriptions with a finer level of granularity. The process meta model helps with integrating and facilitating the sharing of process elements across the borders of product lines. Fig. 8 illustrates how such an instantiation looks like. For example, common project roles are defined at the corporate level independent of the development model, e.g. the Product Manager, Technical Project Manager, Product Quality Manager. The model can be tailored by adding specific roles such as the System Architect or Integration Test Leader for a specific development model.

Knowing about how process elements link with each other and how they are tailored, the next step is to integrate R&D workflows, such as software development or software maintenance with their (e-) business counterparts, such as supply chain and service request management. Our vision was centered on visibility in engineering and mastering a variety of workflows and external interfaces describing how software engineering artifacts are gradually generated. Some are internal to engineering, while others are at the boundary to other functions. They all have their own tool environments, often overlapping with each other. Many of these tools are proprietary, at times legacy and surely not intended originally to work with each other or to be managed externally.

We have not tried to immediately replace this collection of different engineering tools by a single comprehensive solution for various reasons, such as cost, risk and timing. Instead, we have developed a lightweight framework for federating the existing tools. This approach has many advantages. It can start small and still immediately achieve a corporate scope. Based on the meta model illustrated in Fig. 7 and as an integral part of our PLM approach, we developed a life-cycle portal and workflow management anchoring the dates and status of the major milestones, the names of actors of key roles and references to key work products as defined in the standard product life-cycle. The portal is shown in Fig. 9. The portal was linked to the organizational hierarchy maintained in one of the corporate referential databases. This modest initial system could already provide some powerful support: structured navigation through the organizational hierarchy, presentation of roadmaps at each of the levels of the organization, visibility on key actors. From the very beginning, we also established simple hyperlink interfaces to corporate solutions: the organizational referential database already mentioned, the global directory services for authentication and email support, the product catalog. Many embedded hyperlinks allow navigating with a few clicks to the final element the reader is interested in. In fact, we managed to get with four clicks from the Alcatel-Lucent corporate Intranet-homepage to any running project with its data, processes and knowledge.

Initially, this portal provided an integrated view on project and product knowledge. The process perspective was managed in a separate process asset library, partly at the corporate level (for the corporate standards) but mainly at the business unit level for the detailed engineering pro-

![Fig. 8. Instantiation of a tailored process model by reusing available process artifacts and building upon the common underlying meta model.](image-url)
cesses. The meta model on which the product life-cycle is based is also the framework for the process description. Process assets include checklists for decision reviews, templates and process descriptions for the creation of work products, role descriptions, recommended tools and associated user guides. It was a straightforward step to add links from the product release portal to the corresponding process assets.

The integration can easily be achieved at the corporate level and with corporate solutions but this is obviously not sufficient. How did we adapt this approach to support seamless integration with the heterogeneous set of processes and tools at the product line level? We addressed this first at the level of the meta model and defined an inheritance mechanism to allow tailoring of the model. The tailoring is typically used to add more detail but is not allowed to alter the corporate framework and, even more importantly, must be based on the same meta model. Although process descriptions in the product lines had been developed without such a model in mind, they could nevertheless be mapped easily to it due to its very generic nature. As explained above, the basic entities of the model address the basic questions that are addressed in any process description: what must be done, how, by whom and how. This allowed us to provide a first level of integration into the operational systems and asset libraries of the product lines. For more detailed support and navigation, these systems can take over at a lower level.

It should be clear that the support environment is not an ad hoc set of hyperlinked web pages and tools. The product life-cycle provides a standard and mandatory framework for all processes and instantiations of these processes in projects. The object-oriented inheritance mechanism of the product release portal ensures a seamless presentation of the corporate processes and the detailed product line processes in a single view from the operational context of a selected project.

Integration is much more than just hyperlinking of documents. But it is clear that key elements of the business case for our integrated PLM approach are to avoid multiple entries of the same data, thus reducing inconsistencies, inaccuracies and excessive rework. Projects are typically required to submit the same data in various local support tools, presentations for project reviews and corporate databases for aggregated reporting. To position the product release portal as a single source of information, we offer a simple export facility to enable local tools to extract data and merge it with more detailed information collected in those local tools. Also an import mechanism is available to support automatic uploading from local databases.

Another element of our strategy is to rely on existing corporate systems and standards as much as possible to avoid duplication of data. An example is the interface to document management systems. To avoid configuration management problems, documents are not stored in the product release portal but hyperlinks are provided into several document management systems. While documents can be stored in different systems, they are uniquely identified by means of a document number. When a document is entered in a document management system, some metadata including the identity of the system is copied into a global document number register. The product release portal is integrated with the register to generate hyperlinks into the document management system automatically.

The entry level for the portal is the organization structure which makes it equally accessible for all functions and not just for R&D. Examples are marketing (e.g. how far is a product from its delivery?), security (where are certain protocols or components embedded that might cause security threats?) or procurement (how much royalties do we have to pay in a certain region?).

Usability of any workflow support system is determined by the degree to which it can be adapted or tailored to the projects’ needs. There are organizational and project-specific environmental constraints, which make it virtually impossible to apply the workflow system out of the box. Most commercially available workflow systems therefore offer some adaptation of a standard workflow to a project-oriented instance, which ensures that each single activity supports the project targets. Adaptation is achieved by...
offering a set of standard workflows, which are selected (e.g. incremental delivery vs. grand design; parallel vs. sequential development; development vs. maintenance). On a lower level, work products are defined or selected out of a pre-defined catalogue. Some models distinguish among mandatory and optional components. Most of them are implemented based on object-oriented paradigms that allow building of classes of process elements and (limited) inheritance in case that hierarchical refinement is offered. Many vendors are offering platforms for enterprise portals with flexible mechanisms to display information from various sources including legacy systems, but also those systems fail to offer the support for integration and organizational tailoring we want to achieve [10].

Low-level process change management is exactly the point where current workflow systems for unified processes fail. Though these workflow systems offer lots of functionality from various application use cases, they do not well integrate process needs on the above-mentioned levels into a hierarchy with guided selection. We therefore decided to build the integration layer ourselves; based on simple and generally supported web standards, creating the necessary flexibility in a situation where not all requirements of the software engineering workflow management would already be known.

The tool itself was built entirely open to both external business processes and legacy R&D processes, strictly following the architecture described above. To support organizational tailoring, the organizational structure is included automatically from the corporate reference database where this information is maintained. All person-related information is also kept up-to-date through an automatic link into the corporate directory services, which are also used as the basis for a common authentication mechanism.

With the availability of shared knowledge, the next challenge was how to push its utilization. Only active knowledge management guarantees the information quality requested by its users. This is in many companies an unresolved chicken-egg problem, because KM is approached only with tools in mind – and not with people and processes. To break that circle of not using available knowledge, we want to emphasize concrete guidelines:

*Cope with distance and diversity.* Use different communication channels to address audiences that are less familiar with each other. Apply remote team building by having non-technical discussions or events by telephone or video. Distributed management demands more effort, which must be budgeted both in terms of effort as well as skills. As a rule, you should plan some 5–10% of overhead for managing these teams. In the worst-case scenario with highly fragmented tasks and loss of escalation to resolve conflicts, the overhead can grow to 20–40% as we experienced in some cases.

*Plan for sufficient training.* A common failure in global software engineering is the lack of necessary technical or process skills and thus delays. Assure that skills and competences ramp up in due time before they are needed in the project. Adapt training mechanisms to the variety of cultures and preferred communication means. Mix different formats, such as classroom (can be remote and virtual), life webinars or e-learning of predigested contents. Force departments, team leads and project managers to periodically assess skills and skill needs of their teams. Demand training plans for each single engineer. Always remember that sufficient training and right skills is one of the best motivational instruments. Different roles and competences must be individually coached on the specific knowledge needs. Consider sending managers and staff to remote sides. Rotate middle management across sites so they will not get into the “us versus them” mode. Assure that managers feel obliged to live for some time in various cultures and countries.

*Foster knowledge sharing.* Install performance objectives and indicators around the sharing of knowledge and the successful collaboration as a team. A very simple step could be to advocate shared team targets with a distinct percentage across a product or development activity. Dedicated awards might also help as we found out at Alcatel-Lucent. Agree explicit communication protocols with the teams. This might include the recommended communication channels and when and how to use them most effectively. For instance, it seems a normal pattern for many engineers to send e-mails if they do not know other people in person. It is a good practice to demand that your engineers also call unknown persons by phone in order to directly share technical questions. Have a common product release portal for all project-related information. We found for instance that as long as document attachments are sent, people effectively learn less and trust they could always call a peer to retrieve the information. Instead, we advocated sending pointers to repositories and link the various repositories by means of common context (e.g. by tagging) and through search engines. Today we can even use dynamically updated links depending on what project is chosen (Fig. 10).

![Fig. 10. Linking the dimensions of project, product and processes.](image-url)
7. Results

Knowledge is a crucial resource that drives future success and must therefore be adequately managed. We described a PLM concept and implementation that effectively integrates systems and software engineering process, tools and KM. Like any other improvement a knowledge management program is only effective if broadly used. We followed up usage and usefulness during the implementation. Quantitative measures give a feedback on number of unique visitors per site, number of page visits for critical information, such as project homepages, number of documents or number of visits directed by search engine results. More relevant even are qualitative measures like how much value has been created by the process of sharing project, product and process knowledge, reusing available knowledge, and what new competencies and expertise have been built-up in the organization and what is the value of it.

We introduced the PLM concepts in several steps. Managing process knowledge was already on the move in the various organizations when they started CMM(I)-based process improvements. Project and product information started to be managed via the product release portals when we had a critical mass of product lines embarking all at the same time on CMM(I) maturity levels 2 and 3. During the introduction years of this product release portal across the company, we reached a growth of access rate of 10% and more per month. Typically, all product managers and engineering management (e.g., line managers, team leaders, quality managers, support) utilize it. Today with quantitatively managed processes and continuous improvement, the platform facilitates access to company-wide performance data, creates roadmap visibility and offers standardized access to processes, workflows, roles and competence management. The system has a registered user community of over 10,000 people and around 450 different visitors each working day.

Although it is tempting to try and demonstrate the impact of knowledge management on the performance of the organization quantitatively, we need to recognize some important factors that influence the feasibility and confidence level of such measurements, especially in this case study.

- The deployment of knowledge management went hand-in-hand with process improvement and growing maturity of the participating organizations. Without a "control group" to assess the impact without knowledge management and all other factors kept equal, it is impossible to attribute measured improvements to knowledge management only.
- Often global data on the performance of business divisions only becomes visible with the introduction of the environment to support knowledge management. This causes a lack of comparable data for the period before knowledge management was deployed.
- The impact of knowledge management is not likely to be incremental and may not show in the traditional engineering and project measurements. The latter focuses on whether the product was built right, while PLM and KM have an effect on whether you built the right product or implement the right features at the right time.

Obtained benefits can be extracted with activities that would not have been possible without this framework and have a direct influence or the efficiency or effectiveness of the company.

- The members of the Project Core Teams of all product releases are registered in the product release portal linked to their role. Distribution lists for each role can be generated automatically to support selective targeting of notifications of dedicated training or other important message to all affected persons.
- The system supports the corporate Product Life-cycle but also registers any tailoring that is used in business divisions. This information is evaluated and used to guide the evolution of the corporate PLC, by identifying tailoring that is beneficial to the entire company and incorporating this at the corporate level for application across all divisions.
- The operational support environment is integrated with the process environment. A product release portal initially contains a template with work products that must be developed and pointers to the templates for these work products and links to document management systems where the documents can be uploaded. Once the document has been uploaded, the link is automatically replaced by a pointer to the document itself to support further updates. Process knowledge must not be searched in a separate database but is available just-in-time in the operational support environment.
- A survey mainly among Marketing, Pre-Sales and Sales functions revealed that 89% considered the information of the described solution as very important or important for their work. Sixty percent of the respondents frequently accessed the portal to obtain information. This was still not replacing person contact because 80% (also) frequently contacted the product manager and 70% the product marketing manager to obtain information. Eighty percent of the respondents would actually prefer to have the information available in the support environment. The main reasons for relying on personal contact were the desire to get the latest information and also a lack of knowledge about all features of the support environment. It should also be noted that the personal communication is well supported by the portal because all key actors of each product release are registered with a link to contact information (telephone, E-mail).

On top of these rather qualitative results, we also tried to identify controlled experiments where our PLM solution was introduced in settings that allowed measuring
the impact before and after the changes (for details on measuring processes and ROI see [2]). Fig. 11 shows the effects on engineering efficiency during the introduction of PLM to several software development teams. We took a snapshot of defect detection along the entire product development from comparable projects. Three scenarios are provided that lie roughly 5 years apart. They are represented as three scenarios in the graph. The horizontal axis provides relative project time between start and delivery to field usage. The axis is normalized to allow comparing projects of different duration. The vertical axis provides relative defect detection over project time, also on a relative scale. Combining these two views in one graph allows to compare when defects were effectively detected and removed along the product life-cycle. To flatter such defect detection curve in the beginning, the more defects are delayed in their removal to activities after the activity which introduced the defects, thus increasing cost of engineering [2].

The projects were customization projects (i.e. variant development) of a communication system at Alcatel-Lucent. The size of the projects is roughly 10 person years, while the size of the software organization is about 200 persons. The duration is about 1 year from start of specification until release. Expected release quality was the same due to existing frame contracts. This allowed a benchmarking of defect detection because most influencing parameters – except PLM introduction – were stable during the field study. Technology used is software embedded to hardware for the control and management of these systems. This type of software is comparable to what many other companies are doing and therefore shows an interesting benchmark on the effects of PLM.

Scenario A was our starting point before we introduced any change. It was the traditional process with design, implementation and integration exhibiting late defect detection and unpredictable release time. The S-curve in Fig. 11 indicates that defects are detected mostly during test thus creating extra rework and high cost of non-quality. We first embarked on process improvement based on the capability maturity model but without tools and knowledge management. Effects were visible with a more linear curve of defect detection. But still we were not satisfied as engineering work had become somewhat cumbersome due to heavy processes. We introduced PLM as described in this article with a strong focus on structuring engineering knowledge along the domains of process, project, product and technology knowledge. With these changes engineers were able to directly access specifications when doing a review, or testers could verify their test cases with original customer requirements, and so on. As a result, defect detection is now almost linear with around 40% of defects found more than 30% earlier than before contributing to a more than 30% reduction of cost of non-quality.

When looking into the different dimensions of knowledge within R&D, we distinguish product knowledge, process knowledge and project knowledge. PLM drives productivity improvement and thus frees resources for innovation. The business case combines several aspects (see also Fig. 1 which sets up the challenge of PLM), such as improved quality, reduced cycle time or better productivity. Specifically, the combination of KM (mostly implicit knowledge) and explicit project and process information in coaching of engineers and managers has tangible benefits. Such coaching will cost but it does pay off. Looking only at cost of non-quality, i.e. time to detect and correct defects...
defects, we found that projects with intensive coaching (ca.
1–2 percent of accumulated phase effort) could reduce the
cost of non-quality in the respective engineering phase by
over 20% (more details in [2,19]).

We can directly address the customer needs by linking
dedicated improvement objectives, such as return rate,
via the product release portal (with all type of performance
figures) to process changes in R&D. Over past years, Alcatel-
Lucent proved substantial field quality improvements and
defect reduction. Two-digit field quality improvements
have been shown in Alcatel-Lucent by applying this
approach. The efficiency and effectiveness of engineering
processes directly influence engineering cycle time [2]. For
instance, earlier defect detection means faster and more
comprehensive defect correction. A defect found during
development costs according to our own studies some 5–
20% of the effort to correct compared to its detection dur-
ing test. Utilizing a consistent product life-cycle and pro-
cess repository is a necessary condition for reducing cycle
time, as they reduce frictions of unclear interfaces and
responsibilities as well as cutting rework because of inconsis-
tent assumptions and cutting retrieval time for specific
documents and work products. Several product lines cut
cycle time to almost half with strong focus on product
life-cycle and continuous process improvement. Again the
visibility of performance together with knowing which
changes contributed to the effects helps in setting the stage
and effectively sharing best practices.

With decreasing size and duration of projects, engineers
need to be flexible to quickly start working in a new envi-
ronment. While technical challenges cannot be reduced,
the organizational and administrative overheads must be
managed and limited. Alcatel-Lucent relies on a consistent
product life-cycle across the company to ensure we can
deliver solutions independent of where the components
come from. In a company the size of Alcatel-Lucent, it is
key to align working environments and the development
process in order to reduce the learning curves when starting
a project. With process asset libraries linked to tools, we
are able to filter out and evaluate scenarios of how process
change affects tools, or where tool changes would influence
processes. So-called “best-practices” can be communicated
with related tools and procedures to move up engineering
effectiveness and learn from the best in class. Interfaces
to tools and their user guides can now be embedded in
the process support environment. PLM supports integrated
workflows and knowing how a process relates to a product
line or the current project development phase. For exam-
ple, clicking on a work product name can activate an inter-
face to a document management system and administrative
data, such as the document number, are automatically
derived from the project context. Information is presented
in a consistent way for all projects, avoiding replication of
data and reducing search time. The product life-cycle view
of the workflow system provides a dashboard with immedi-
ate visibility on key data and responsibilities contributing
to management efficiency.

Increasingly, project roles and also specific work prod-
uct templates or process related roles are standardized
and can be reused thus facilitating more consistent skill
and human resource management. Transferring products
to another location, as is today often the case, is facilitated
by standardized role descriptions and workflows. Ramp-up
time is shorter with new engineers responsible for such
transferred products. We found during the introduction
of our development centers in Asia that organizations with
defined and continuously improving processes using the
PLM framework were faster and more effective in master-
ing global software engineering.

8. Conclusion and outlook

We have shown the benefits of an approach to combine
knowledge management (KM) and product life-cycle man-
agement (PLM) at Alcatel-Lucent by bringing together
knowledge about products, projects and processes. PLM
has been successfully growing from one product line to
gradually cover the entire company today, independent
of the type of markets, products or solutions. Today
PLM concepts are being introduced in all product lines
in order to stimulate organizational learning, and improve-
ments. With this initiative, R&D is fully included into our
e-business evolution.

We do not rely on general-purpose mechanisms for
knowledge management but strongly rely on a commonly
applied corporate product life-cycle, which provides the
framework for knowledge management. Process assets
are linked to the product life-cycle (PLC) definition, and
product and project data are following the same structure.
The corporate PLC can be tailored to adapt to the specific
needs of the product lines. This tailored life-cycle provides
the interface to detailed processes and tools support of the
product lines. A product release portal supports the instan-
tiation of the PLC for each product release and provides a
federation framework for process, product and project
knowledge.

What did we achieve with PLM? Since we consider
knowledge management a regular management activity,
we followed through like in other improvement projects
by looking into performance results from real projects, as
well as some process-related aspects, such as knowledge
utilization. We found improved quality, reduced cycle time,
improved engineering flexibility, reduced overheads
improved communication, increasing alignment of pro-
cesses and tools and faster ramp-up time and skill manage-
ment. Initially, improved visibility and aligned terminologies and roles already brought huge gains, as they
facilitate a borderless solution building inside the company.

The results of the survey mentioned above gives us
important clues about future work. We should not expect
that all knowledge can be codified and made available via
the intranet. Personal contact will always be necessary to
provide context and analysis. The support system should
therefore be extended to facilitate interpersonal communi-
cation and evolve toward a global who-is-who not only at the operational product/project management level but also at the tactical and strategic level.

Correctness and completeness of the information is another aspect that needs to be worked on. We are convinced this cannot be achieved by imposing a reporting discipline only. It can only truly be achieved by ensuring that the provider of information is directly benefiting from making the information available. This can be achieved to continuing to evolve the system to ensure information can be captured as early as possible when it is required at the lowest operational level, e.g. by supporting period project reporting in presentation format to avoid information is presented first a set of slides for the project review before it is entered in the system.

To conclude, we recall a few of the key success factors for our approach:

- the systematic use of a standardized PLC as the framework for processes and tools support,
- the definition of a tailoring mechanism to enable lean adaptation of this PLC to the specific needs of business units,
- the development of a lightweight product release portal as primary access point for all project related information,
- a clear focus on knowledge and competences rather than IT infrastructure to master PLM needs across the company.

PLM as a concept and vision is currently on its way into the mainstream software and systems industry. Recent evolutions in the PLM domain signal that vendors realize the value of the more integrated PLM concept which we described in this article. Hardware CAD vendors like Mentor are moving into the software domain [28] software IDE vendors, such as Borland with their “application life-cycle management” [29] or Telelogic with their “enterprise life-cycle management” [30], are moving toward process improvement and are evolving dedicated PLM solutions; systems tools vendors such as Dassault [31] are growing into knowledge and systems data management; and dedicated systems engineering PLM environments built around knowledge sharing and collaboration such as eASEE [32] are starting to be used in systems and software industries rather than stand-alone engineering tools suites. Compared to the more traditional approaches, such as extending an existing PDM-tool, dedicated PLM-solutions like eASEE are built around concrete business needs such as reusing software components or integrating documents and product or variant data.

What is next in PLM? Knowledge management must be better linked to business. Aligned business objectives and metrics must guide and monitor the development processes, the product lines and the project teams. Take as an example a mobile phone or game design (with lots of embedded software). Being a commodity, business-oriented targets cover return rates or brand loyalty. Defects increase return rate and reduce brand loyalty with devastating business impacts. Looking to projects, products and processes will improve the design away from overly narrow focus on manufacturing aspects toward usability engineering. Knowledge and experience from past projects will be embedded into the underlying design processes. We stress the need for adequate knowledge management (KM) as a basis for success in product and solution development – an aspect going well beyond most PLM approaches of today.

References


