Four Key Requirements Engineering Techniques

Abstract

Requirements are the initial and basic building blocks combining processes in a product’s life-cycle. We take the position that only by taking an requirements engineering perspective in four key product life-cycle management activities, the underlying projects will be successful. The article describes a field study with data from 246 industry projects in the domains of software platforms, embedded systems and software applications. We found that these four life-cycle techniques must be used simultaneously to achieve tangible performance improvement measured by schedule adherence. Each technique is elaborated by concrete, practical experiences.

Keywords: PLM, portfolio management, product management, project management, product life cycle, process improvement, requirements engineering.

Copyright

Copyright by Vector Consulting Services GmbH and IEEE Computer Society Press. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

1 REQUIREMENTS AND THE PRODUCT LIFE CYCLE

Time to market and schedule performance is considered by many enterprises to be the key differentiator between market leaders and followers. Matching schedule commitments and shortening cycle time until release assures the perception as a reliable supplier as well as overall profit optimization. Pressed to accelerate project handover and new product commercialization, companies have improved the execution of R&D during the past years with instruments like CMMI. They continue however in facing challenges in cross-functional coordination which result in project delays and long cycle-time. Specifically requirements and contracts are frequently committed without proper alignment of sales, product management, project management and marketing in order to boost short-term revenues. Such misalignment results in insufficient capacity planning or product development resource allocation, thus making projects late.

Having a winning product requires being successful in identifying market needs and translating them into a product vision and scope which is executed following sound project management principles. Requirements are the initial and basic building blocks gluing together different phases of the product life cycle. A recent Chaos report showed that only half of the originally allocated requirements appear in the final released version [1]. This is primarily the consequence of this process not having a clear corporate owner with assigned accountability for its success. The Bermuda triangle in many IT and software companies lies in between sales / marketing, strategy and product management and technical and R&D management. Not that the people don’t talk to each other, but often they are unable to integrate their different agendas into winning products following a pipeline approach as we know it from pharmaceuticals.

Different studies look to the effect on requirements engineering on product success [2,3]. In a study looking at new product development from a broader scope, Cooper found in 105 business units from various industries in USA that the top 20% of enterprises deliver 79% of their new products in time, while the average delivers only 51% of projects in time. From that same set of top 20% ranking companies 66% relate their resource breakdown to strategic needs. Only 31% of the average companies are aligning their strategies with resource utilization in projects. Typical results from poor upstream processes are insufficient project planning, continuous changes in the requirements and project scope, delays, configuration problems, defects, and overall customer dissatisfaction due to not keeping commitments or not getting the product they expect.

The traditional rule of thumb indicates a change rate of 1-3% per month in terms of effort related to the allocated requirements [2,3]. This translates into more than 30% changes (i.e., new, deleted or changed requirements) to overall requirements for a project duration of two years. As a consequence, many contractors and also clients strongly urged to reduce project duration towards one year maximum. With this reduction of project duration, projects indeed performed better. We faced in the past years an increase of change rates to even double that amount, i.e., 30% in a one-year project. This cannot anymore be fixed with only shortening project duration and working with iterations. A common denominator of requirements changes is that they practically always correlate with project delays.

Let us briefly explain some terminology which we use in the article. The product life cycle (PLC) is the sum of all activities needed to define, develop, implement, build, operate, service, and phase out a product or solution and its related variants. Product management is the discipline and role which governs a product (or solution or service) from its inception to the market/customer delivery in order to generate biggest possible value to the business. Product life cycle management (PLM) is the process for guiding products and solutions from in-
ception through retirement to deliver the greatest business value to an enterprise, its stakeholders and its customers. Portfolio management is a dynamic decision process aimed at having the right product mix and selecting the right projects to implement a given strategy.

Fig. 1 shows a simplified product life cycle. In this example the upstream process covers the funnel until the first gating review (triangle marked with “1”), while downstream processes relate to the different activities between gate 1 and gate 3. Effective PLM assures understanding and collaboration across the entire product life cycle – from inception (i.e., long before requirements are elicited and defined) to end of support and phase-out. To benefit from improved business processes, the different functions of the enterprise plus potential external partners (e.g., outsource manufacturing) need to agree on the related processes, cycle, tools and practices. They need to apply common access to knowledge, performance metrics and decision-making protocols. They need to share information, communication, and underlying resources.

Figure 1: Simplified product life-cycle with four decision gates and upstream (left) and downstream (right) processes. Product development can be stopped at each of the gates.

Downstream processes around the project execution have received much attention both from project management as well as from requirements engineering perspectives [1,2,4]. Unfortunately the upstream processes were not getting much attention in research, although they are also part of RE. This is often because of their complexity (e.g., heterogeneous and overlapping ownerships, vague processes, unclear impacts of stakeholders). At most the upstream activities are mentioned in the requirements elicitation process. A major barrier is the short-term profit and loss responsibility that provides incentives to focus on current quarter results (i.e., ongoing projects and contracts) and not to invest in future products or platforms. We want to emphasize with this study to consider product life-cycle processes, such as gate reviews or empowered project teams, part of the RE discipline.

The article is structured as follows. Chapter 2 describes briefly the setting of this field study and investigates root causes of insufficient project performance. Chapter 3 provides lessons learned and concrete data from using the described practices and how they can be generalized for transfer into other settings. Finally, chapter 4 summarizes results.
2 FIELD STUDY LAYOUT

For this field study we investigated products and solutions within different business groups. 246 projects were taken from our history database. Project size in effort was between some person weeks and several 100 person years. All projects that had recorded valid data in the history database were used. No “outliers” have been thrown out, thus avoiding overlooking influences from other parameters than the four process elements that we analyze here. The projects evaluated in this study, though primarily in the domains of embedded systems or software applications cover a wide range of the entire world of software dominated projects. Results therefore apply to other industries than communication. This is also supported by benchmarks we have been doing in the field of automotive, defense and information systems (for published records see also: [2,3]).

The field study was designed in a two-step approach. First we collected evidences and performed a thorough root cause analysis in a rather small sample (one representative product line). From this root cause analysis we distilled four winning techniques which we studied in a second step across all projects in a given time-frame. Such funneled approach reduces time to perform such study. It theoretically allows being corrected because considering further root causes and new techniques is not stopped during the second step. We did not receive further such evidences during step 2. We could however evaluate in detail any combination of using these techniques and judge from the achieved results on what has biggest impact on schedule performance.

We take in this study a clear focus on reducing project delays. This will be our dependent variable. Late projects are a primary cause for shrinking margins and insufficient business case validity. Being late makes customers angry because they miss themselves their windows of opportunity. Delays impact follow-on projects and make the overall portfolio and resource planning almost impossible. Evidently schedule adherence at the cost of inflexibility is not the target. Some requirements changes may actually help improve both schedule performance and fit-for-purpose/market success of the solution. This however must be balanced with a focus on keeping commitments.

A disturbing consequence from the risk of being late is that product managers, system analysts and engineers have become so paranoid on everlasting requirements changes and delays, that we observed an increasing duration of the analysis (or elaboration) phase of a project, that is, before the actual project starts. This syndrome, often called “paralysis by analysis” contributed to project duration and cost. This results in additional delays. We thus face two vicious circles caused by weak upstream processes that both contribute to delays (fig. 2).

We carefully evaluated all projects in one representative product line to better understand why requirements change in the course of a project. This product line had 15 projects. 73% of a project’s requirements were changing in average over the 15 projects (median: 50%) after project start. By “change” we mean modifications to existing requirements or – more often the case – deletion or replacement of requirements. Requirements in projects have different sizes, thus replacing one requirement with another one does not preserve the project effort. For instance if one project reduced the amount of delivered requirements, it would mean that some specific key functionality was delivered to the customer, while others had been postponed or re-evaluated. We found that one third of the changes are of technical nature (e.g., a specification was infeasible for design), while two thirds are of commercial nature (covering the majority of requirements uncertainties).

Our key insights from this first in-depth root cause analysis are as follows. To have winning products, RE must start early and has to connect portfolio considerations with project man-
agreement. Effective RE must ensure that all relevant stakeholders are available and empowered. The product life cycle must be mandatory for all projects, independent of type, size and scope. All elements of the portfolio must be equally easy visible.

![Vicious circle 1](image1)

![Vicious circle 2](image2)

Fig. 2: Insufficient upstream processes cause paralysis and late changes and ultimately delays

To design the empirical study we translated these initial observations into four techniques we felt would be the key to better PLM and thus improve schedule commitments:

1. Install an effective core team for each product release
2. Focus the product life-cycle on upstream gate reviews
3. Evaluate requirements from various perspectives
4. Assure a dependable portfolio and release implementation

The “existence” or practicing of each of these four independent variables could be deducted from our company-wide history database by means of evaluating availability of project core teams, analysis whether early gating reviews have been conducted and reported, access to and usage of requirements, and availability of product release information. We use completeness and sanity checks to verify existence and validity of the metrics. If metrics look odd (typically the result of typing errors), the owners are notified to double-check.

Having these 4 independent variables closely linked to the projects in our study, we could extract impacts of each single variable. This avoids conclusions of the type described in experimental software engineering as shotgun approach with uncontrolled independent variables [5]. The subsets of projects used here to explain results are not overlapping. This means that effects attributing to one result would not attribute simultaneously to another and thus hide the real explanation (see last section for chi square test on this statement).

Requirements were counted on the basis of project requirements. They reflect an external view on the project and explain what the project should do, not how it will be designed. Requirements are baselined as a counted number at the time of project start. Changes to requirements are all changed, added or deleted requirements after project start. Number of changes is normalized with the number of requirements in the specific project thus allowing comparing across projects. Project duration is measured between the gating review which kicks off the project (fig. 1, milestone 2) and the gating review which decides to release the
project to the customer or market (fig. 1, milestone 3). Latest at the time of project start, a firm end-date of the project has to be committed. This end-date is then recorded together with other planning data.

Delays of projects (and thus schedule performance as we use it in this study) are measured by comparing the project end milestone as committed at project start with the actually achieved project end milestone. This absolute difference in calendar days is normalized by the originally committed project duration in calendar days. If a project has been initially agreed to run for 100 calendar days and is actually finished after 120 calendar days, it has a (project) delay of 20% and therefore a project (schedule) performance of 120%. This metric is the dependent variable which we analyze in the study.

Table 1 provides an overview of the 246 projects evaluated in this study. The first column is what we call the control variable of the study. We look into 4 specific techniques and the degree to which they are applied. So we have projects where none of the techniques are applied (labeled “0” in first column) up to projects where all 4 techniques are applied (labeled “4”). If only 1, 2 or 3 techniques were used, we do not identify which subset of techniques were used, because what we found (and explain later) is that it does not matter. There is no order or dependency in using these four techniques.

The further columns in table 1 tell how many projects fall into each category. A majority had used none of the techniques. The others are roughly 20-30 projects for each category which is sufficient for our calculations, since we have only one control variable. Minimum and maximum size (in person years) is shown in the columns 3 and 4. Size is uniformly represented from very small projects (few person weeks) up to several hundred person years. The last column shows the dependent variable of our study as an average of the schedule performance. It already highlights that only with using 3 or 4 of the techniques, the schedule performance improves.

<table>
<thead>
<tr>
<th># techniques used</th>
<th>#projects</th>
<th>min size</th>
<th>max size</th>
<th>average schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>134</td>
<td>0</td>
<td>47</td>
<td>136%</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>2</td>
<td>148</td>
<td>147%</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>0</td>
<td>780</td>
<td>121%</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>2</td>
<td>177</td>
<td>113%</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>1</td>
<td>225</td>
<td>103%</td>
</tr>
</tbody>
</table>

Table 1: Summary of projects from the field study

Having implemented these techniques with different order and speed allowed us to correlate results (i.e., project delays) with the amount of techniques used. We found that schedule performance is independent of project size – within the range of project size that we had in our set [10]. This means that smaller projects do not necessarily perform better than larger projects. In fact delayed projects are almost evenly scattered across project size. A Pearson rank correlation showed \( r = 31\% \) which indicates no relationship between size and delays. A double-sided Chi-square test evaluated whether usage of the four techniques would relate to the project size. We could accept the homogeneity hypothesis on a significance level of \( \alpha = 1\% \). Usage of any particular technique is not depending on project size.

We also evaluated whether some of the four techniques would be used in correlation with each other and found that they were used independently at the time when the projects were executed. A double-sided Chi-square test looked into relationship of the four techniques. We could accept the homogeneity hypothesis on a significance level of \( \alpha = 5\% \) (or below) for each single pair of the 4 techniques, indicating that they are not correlated in usage.
Fig. 3 summarizes the results of this impact analysis of the 4 techniques. Performance significantly improves with 3 or 4 of the techniques being used. Using none, one or two techniques (from the set of four techniques above) have no impact on project delays. A double-sided Chi-square test accepts the homogeneity hypothesis on a significance level of \( \alpha = 1\% \). There is no impact on delays when using only one or two of the four techniques. A double-sided Chi-square test rejects the homogeneity hypothesis on a significance level of \( \alpha = 1\% \). There is a significant impact on – reducing – delays when using three or four of the techniques! The number of projects delivered on schedule is improved by twenty percentage points when using three or four of the mentioned techniques. Projects being 5% late improve from 45% to 63% with 3 or 4 techniques used and projects being 10% late improve from 56% to 77%.

The four mentioned techniques are typically reinforced in organizations which are committed to process improvement. Organizational entities (i.e., business units or regional entities with product responsibility) which view life-cycle processes and performance improvement as a management fad or approach a technique as a single-shot change won’t see sustainable results. Product life-cycle management must be bought by an entire organization, capturing all functions. Training and clear role descriptions are key to making any of the techniques a success.

Fig. 3: Schedule performance improves only with combined use of 3 or 4 of the techniques (right side)

### 3 FOUR TECHNIQUES WHICH MEAN SUCCESS

#### 3.1 Install an effective core team for each product release

Three roles dominate the decision-making during the product life cycle, namely the product manager, the marketing manager and the (often technical) project manager. They must build a multifunctional core team fully accountable for the success of a product. They represent not only the major internal stakeholders in product or solution development, but also sufficiently represent the sales and customer perspective. This core team must have a clear mandate to “own” the project. We found that if such a core team is available but underlying commitments are not baselined (see ch. 3.2 and 3.4), it is of no value. The more power the core team has (as opposed to ineffective line organizations) to make and implement decisions, the faster problems are solved and the better projects reach their objectives.

The product manager has the more long-term business responsibility beyond the single product. She determines what to make and how to make it and is accountable for the business success within an entire portfolio. She approves roadmap and content and determines
what and how to innovate. The product manager is responsible for the entire value chain of a product following the life-cycle and asks: What do we keep, what do we evolve, what do we kill? The (product) marketing manager determines how to sell a product or service. He is accountable for market and customer success. He has a profound understanding of customer needs, market trends, sales perspectives and competitors. He communicates the value proposition to sales and customers. The marketing manager drives the sales plan and execution and asks: What markets will we address? The project manager determines how to best execute a project or contract. He ensures the project is executed as defined. He is accountable for business and customer success within a contract project. He manages the project plan and its execution and asks: How do we get all this done? Together these core team members run the project and product line portfolio as a mini-business without continuous external interferences. They evaluate cost and benefits of the portfolio.

3.2 Focus the product life-cycle on upstream gate reviews

Typical software life cycles follow IEEE 12207 or IEEE 1074 standards. Both have in common a gating process between major phases [6]. Such gate review is essentially a communication and risk management tool. Each such gating review evaluates on the basis of predefined criteria the project status and decides on whether and how to proceed with the project. The top driver is the portfolio, which depicts all products within the company and their markets and respective investments. For each product there should be a feature catalogue across the next several releases covering the vision, market, architecture and technology. From such product roadmap a technology roadmap is derived, which allows for instance to select suppliers or build up partnerships. It also drives the individual roadmaps of releases and projects, which typically have a horizon of few months up to maximum one year.

The product life cycle must be mandatory for all projects. This implies that it is sufficiently agile to handle different types of projects. Standardized tailoring of the life cycle to different project types with predefined templates or intranet web pages simplifies usage and reduces overheads. Its mandatory elements must be explicit and auditable. Some online workflow support facilitates ease of implementation and correctness of information. Gate reviews (decision reviews) must be well prepared. They must not result in lengthy meeting, but are rather prepared with online checklists so all attendees are prepared and can decide in short time the go/no go for next phase. Project information should generally be available online.

A useful product life-cycle has to acknowledge that requirements may never be complete and may indeed be in a ‘continuum’ state. Sometimes requirements are purposefully incomplete and RE must deal with such situation. The product life-cycle should guide with defining stopping criteria, i.e., determining what is good enough or stable enough.

3.3 Evaluate requirements from various perspectives

Requirements are evaluated by the entire core team (see ch. 3.1) to ensure that different perspectives are considered. Each single requirement must be justified to support the business case and to allow managing changes and priorities. Often market demands are mistranslated into ad-hoc project contents, resulting in ever-changing requirements, almost assuring the project will fail. Impact analysis thus is based on requirements, as well as priority setting and portfolio management.

Proper impact analysis from product management, marketing and technical perspectives assure focus on what can be done without over-commitment. Evaluation looks into several dimensions: What are the requirements? How do they relate between markets and correlate
with each other? What is their impact? What markets have asked for it and for what reason? Are they necessary for a solution or just inherited from an incumbent approach perhaps becoming obsolete in meantime? To address these questions requirements must be documented in a structured and disciplined way. They must be expressed allowing both technical as well as business judgment. Any incoming requirement should be reviewed with the product catalogue and global product evolution in mind to also evaluate marginal value versus marginal costs. Doing this by the entire core team avoids the mistranslation of market perceptions into unfocussed requirements.

3.4 Assure a dependable portfolio and release implementation

A fourth element within the PLM is to assure a dependable portfolio and release implementation. Dependable means that agreed milestones, contents or quality targets are maintained as committed. For instance within a product-line architecture the underlying generic product, platform or components, on which many customization products build upon, must be in time and provide the agreed contents. Otherwise there will be numerous ripple effects. Naturally project management techniques differ between a generic and an application product. While the platform has to build in resource buffers, the application product can easily work with feature prioritization and time boxing.

Having a project plan that is directly linked with requirements is mandatory for all projects. If there is a change to plan or contents, both must be synchronized and approved by the entire core team. By definition requirements must be accessible online together with other relevant product and project information. Different tools can be used, starting with simple spreadsheets. A web-based dashboard with standardized metrics enables the communication of project progress and increases understanding of project dependencies. Full portfolio visibility in software projects means to access and to assess all projects continuously and in totality. Project information builds an online accessible history database to base further impact analysis and project planning upon. To facilitate smooth and continuous data collection and aggregation without generating huge overheads, extraction should be highly automated and accessible from intranet portals. If constraints or assumptions are changing, the evaluation must be adapted. Data quality of project information and requirements lists is key to trusted decision-making.

4 CONCLUSIONS

Late projects are a primary cause for shrinking margins and insufficient business case validity. From a business perspective we thus evaluate in this study how to reduce project delays by reinforcing product life-cycle management. We did not investigate effects such as on sales, revenues or customer satisfaction because they have much more influences from the market situation and are thus difficult to compare. Analysis of 246 industry projects from domains such as small software applications and services, embedded systems (e.g., mobile phones) to large switching, routing or mobile communication systems showed that four techniques have to be used in parallel to significantly reduce delays:

1. Install an effective core team for each product release
2. Focus the product life-cycle on upstream gate reviews
3. Evaluate requirements from various perspectives
4. Assure a dependable portfolio and release implementation
The four mentioned techniques assure visibility, agreement and commitment. By doing so, they reduce requirements changes and together with strong project management by a team of empowered stakeholder representatives improve predictions and schedule performance. By strong integration of upstream processes with the product life-cycle related RE processes, we could achieve reduction of delays of over 20% per year. The techniques are tangible and can be formally introduced to projects during the launch period, thus reducing the change impact (i.e., no big bang). Practitioners in engineering, product management and marketing accept them because they yield results and stimulate empowered project teams. At times they mean some customer education, which is feasible given the benefits our customers will face with more predictability and faster reaction time.

Acknowledgements

We would like to thank the product managers and engineering teams that shared over past years their best practices.

References


Company profile

Vector Consulting Services is a globally active consulting firm with focus on optimizing technical product development. Renowned companies from automotive, information technology, healthcare, transport and aerospace rely on the professional solutions for improving product development, product strategy, and organizational change management. A subsidiary of the Vector Group with 1200 employees, Vector Consulting Services supports its clients worldwide with sustainable consulting solutions covering the entire product life cycle and the related processes and tools. The firm is managed by partners. This assures independent and customer-oriented consulting.

Contact: www.vector.com/consulting .

Biography

Dr. Christof Ebert is managing director at Vector Consulting Services. He supports clients around the world to improve product strategy and product development and to manage organizational changes. Prior to that, he held global management positions for ten years at Alcatel-Lucent. A trusted advisor for companies around the world, member of industry boards, he lectures at the Universities of Stuttgart and at the Sorbonne in Paris. He authored several books including his most recent book "Global Software and IT" published by Wiley. He received the IEEE distinguished visitor award and is a member of the Alcatel Technical Academy.

Contact him at christof.ebert@vector.com