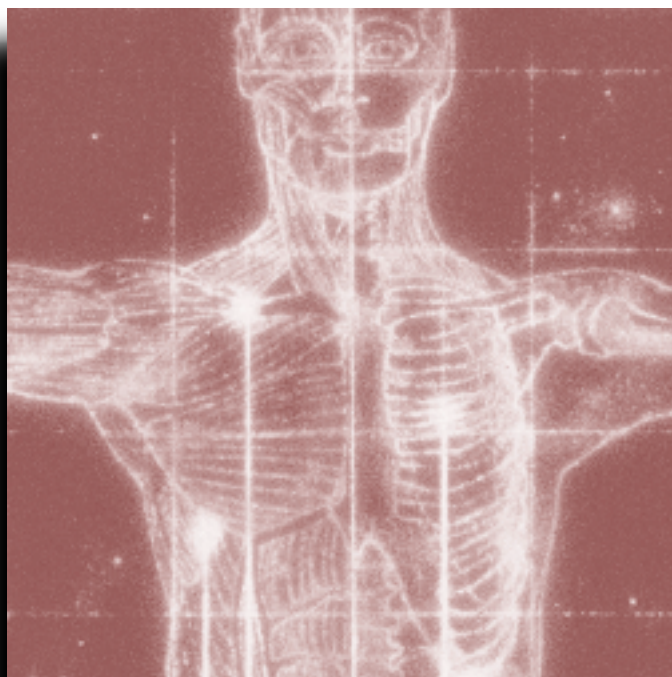


Creating Products *Customers Demand*



This article presents a re-engineering case study of the product requirements definition process at Digital Equipment Corporation. Based on cross-functional teams working in direct partnership with customers, the re-engineered process was designed to ensure customer satisfaction and business readiness before a line of code is written, an electronic circuit designed, or a service defined.

Several insights gained throughout the re-engineering experience had significant implications for effective cross-functional team performance. These insights addressed issues concerning the attainment of cultural change, as Digital sought to move from a technology-centric view of product development to a customer-centric view. Most

important among these insights were the concept of “whole products,” which had the effect of broadening the requirements definition scope to include requirements for marketing messages, pricing, and packaging, as well as deployment and service scenarios; and a shift in thinking about “process”, moving away from process as a series of sequenced procedures that, when executed, will deterministically yield desired results toward a view of process as a map of the product development terrain, serving as a guide through peaks and valleys to the final destination. As one of the software engineering managers remarked during an early pilot effort, “The *right* process is the one we invent.”

*Anthony F. Hutchings
Steve T. Knox*

Re-engineering Digital's Product Requirements Definition

In the summer of 1993, the central engineering organization in Digital began the implementation of a re-engineering effort

under the name of Achieving Engineering Excellence (AEE). This in turn was an integral element of Digital's overall re-engineering effort [3], inspired by a desire to streamline all our dealings with customers. A major goal of this effort was the reduction by 50% in new product development cycles. Internal data collected within Digital showed that the most significant contributor to excessive development cycles was a phenomenon known as "requirements churn" [1].

The AEE data, gleaned from a survey of hundreds of Digital's staff and an analysis of the corporate planning database, found that on average, 40% of the requirements specified in the Feasibility and Requirements Phase of the Lifecycle were redefined in the subsequent four Lifecycle Phases. The cost of requirements churn, using an industry-wide regression model, found that on average Digital spent 50% more than budgeted.

Three significant contributors to requirements churn were identified:

- Requirements for products were loosely defined. In the prescribed lifecycle management process, product requirements for central engineering were obtained from business groups in direct contact with targeted customers representing strategic industries. The business groups usually communicated their customers' needs to the engineering groups in the form of written requirements/specifications. It was common for development engineers, who had no direct experience with the context out of which these requirements were developed, to misinterpret the specifications. It was also common for the business group partners to modify their requests over the period of development as new customer needs arose or established needs were understood differently.
- A technology-driven central engineering organization mistrusted the technical feasibility of requirements handed to them by business groups staffed with marketing and sales representatives, especially when the requirements were ambiguously framed and/or changed during the development cycle. Also, in a number of cases, engineers preferred their own specific technology interest as a development path over that suggested by the business groups' specifications. In these cases, central engineering made a technology investment in those areas of their interest.
- In the Digital climate of a central engineering organization isolated from direct contact with the customers who would use their products, technology development took place without a contextualized sense of the problem to be solved. Without a common grounding in the customer's experience, conflicts over priorities often arose between engineering groups and business groups. The

source of these conflicts was from the observation that the criteria for prioritization were grounded not in customer experience but rather in separate organizational goals.

While the problems described were not found with every development effort studied, they were representative of the state of poorly managed requirements definition practices within the product development environment at Digital. However, the internal AEE study not only uncovered the factors contributing to requirements churn, it also came across a number of "best practices" by enlightened development groups that sought to address some of the problems mentioned:

- Cross-functional teams reduce misunderstandings and the time it takes to correct them. While most development groups consisted of the engineering staff and a product manager, some included marketing and field personnel as well as actual customer representatives. These teams had clearer requirements definition and specification, which mitigated the tendency to engage in interorganizational conflicts leading to changes in existing requirements or adding new ones late in the development cycle.
- Direct customer involvement in requirements definition was the single largest contributor to early agreement on product content by the cross-functional teams and subsequent reduction of requirements churn. Some of the development teams spent considerable time at customer sites to gain an in-context understanding of the problems they needed to solve. These teams would interview customers to understand their work-related goals, their organizational structure, their information technology environment, and the details of their work. Additionally, providing customers with low-fidelity prototypes of proposed solutions allowed early and rapid validation of appropriate technologies.
- Some teams designated a dedicated "design room," which they filled with nontraditional artifacts intended to capture and make public their understandings and agreements of requirements and specifications. These "artifacts" included wall-sized diagrams of the customer environment, hierarchies of problems encountered, matrices showing key customers and markets and their respective attributes, and priority lists of the benefits and capabilities desired, as well as hand-drawn storyboards of the product's use and nonfunctional GUIs that could be easily tested and revised.
- Requirements change control was a central part of the process for a few of the projects studied. These teams would agree up front on methods for evaluating requests to change or add new

requirements and for clearly delineating the cost-benefit criteria on which change request decisions would be based.

AEE Requirements Management Process

In response to the data collected, a Requirements Management re-engineering team was formed. The team comprised representatives from key organizations (hardware, software) as well as key functions (engineering, process, business, and marketing). Seeking a formal definition of requirements management, the team turned to the Software Engineering Institute's Capability Maturity Model (SEI/CMM)

to the issue of whether the requirements to be managed are the "right" requirements. This issue of "right" requirements was found to be a root cause of requirements churn and schedule overrun as experienced in many of the product development efforts at Digital.

Consequently, a re-engineering solution directed only toward the management of the stated requirements would not address the root problem. What was needed was a process to address the mistrust between the engineering and marketing organizations over the right requirements and specifications; and it was clear from the experience of the projects engaged in the best practices that the voice that determines the rightness of requirements was neither engineering nor marketing but rather the customers who would purchase and use the product.

Getting the requirements right became the grounding focus for the AEE Requirements Management team. This team developed a nine-step process, which is outlined in the four-field diagram of Figure 1. The column headings in Figure 2 refer to the functional perspectives represented on the cross-functional team. Each row represents a step in the process, which spans perspectives believed necessary for the satisfactory completion of that step. While the diagram suggests the steps are executed in sequence, in practice there

is a great deal of iteration and concurrency. (The reader will note the inclusion of four feedback loops in the diagram.)

This article will not detail every method and technique by which these nine steps are executed; in fact, one of the findings of the pilot efforts was that there were multiple alternative paths to accomplishing the intentions of a given process step. However, we will describe the high-level intentions of each of step, including their objectives and desired outcomes.

1) Understand system business needs. This step is the initial formation of the cross-functional team, comprising participants from, at a minimum, the engineering, marketing, and business functions. (Some products also require participation from sales and service personnel, as well as third-party partners.) The expected outcomes of this step are twofold: a) to build an awareness of the cross-functional dependencies by having each participant share their perspective on what the focus of the solution should be, listen to each other in a safe atmosphere and derive the points of common focus; and b) identify the candidate target markets and customer representatives to participate in the requirements definition.

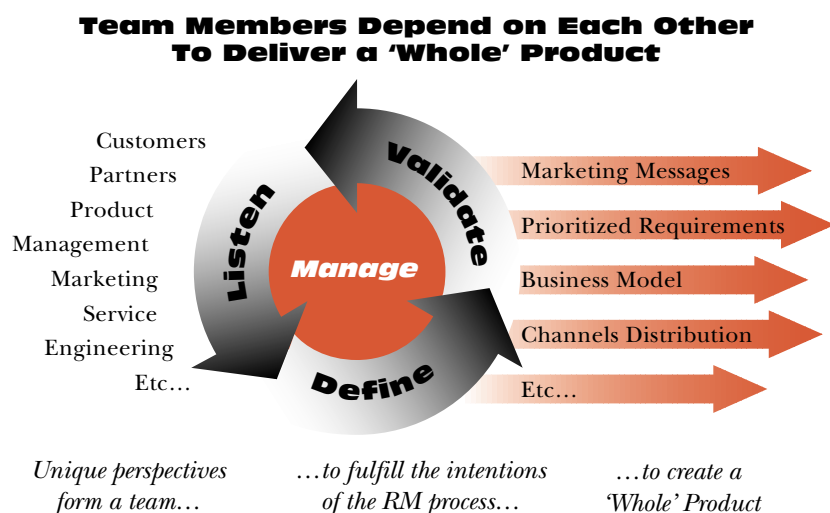


Figure 1. The "Whole Product" perspective of Requirements Management

[6]. The CMM defines requirements management as a level 2 key process that "involves establishing and maintaining an understanding and agreement with the customer on the requirements for the software throughout the lifecycle" [6]. The agreements referred to in this statement concern both the technical specifications and the delivery dates, which are meant to form the basis for estimation of the development effort and planning of the development activities. The SEI/CMM then details a set of goals, commitments, and activities directed toward ensuring that the system product requirements are unambiguous and testable and that policies and processes are in place to ensure these are properly analyzed and allocated into the plans, specifications, design, and test plans of the product.

The AEE team found that the SEI/CMM focus on requirements management was geared to ensuring (as it should) that the agreed-on requirements are managed and actually built into the product. However, the team also felt the CMM definition gave scant attention

2) Gather customer information. The intention of this step is to create a profound understanding of the problem to be solved from the perspective of those who would buy and use, and those who would partner with us in either developing or selling, the solution. (The “customer” then can be either the next element in the chain, as with a Value-Added Reseller or a direct consumer [user] of the solution. While there are a number of methods for collecting customer data, it was observed that the most useful data for requirements definition was derived from actual customer context. The findings of the internal AEE study acknowledged that customer surveys, focus groups, and advisory boards were particularly helpful in obtaining feedback about proposed product directions. However, in-context examination of actual customer work [4] provided the information most relevant to designing solutions. The expected outcome of this step is data that addresses the following questions: “What do the target customer/users do? What are the goals and intentions of their work? How do they go about realizing those goals? What gets in their way, and/or how could technology support, extend, or even transform their work to enable them to reach their goals with greater productivity and personal satisfaction?”

3) Translate needs into an internally consistent list of requirements. The intention of this step is to transform the qualitative data collected from customers into a concrete set of actionable requirements. To do this, qualitative data must be as rigorously analyzed as

quantitative data. The expected outcome of this step is the creation of artifacts representing the cross-functional team’s analysis of the data collected, centered on crystallizing the spoken and unspoken customer needs, prioritized according to their market potential and engineering feasibility.

4) Develop and prove key enabling concepts. Even with the top priority customer needs established and agreed on, there are still potential disagreements over the various technologies that can be brought to bear in meeting those needs. The intention of this step is to evaluate different technology aggregates, with the expected outcome being the discovery of the aggregate of features and functions that best “hangs together” according to the criteria set by the team. This step is expected to yield the “leadership” concept, which then becomes the chosen solution and the basis for design.

5) Compare chosen solutions against the competition. Almost never, in the old process, is the chosen solution compared feature by feature against the competition. Such information is vital to help understand product positioning in the customer’s mind and allocation of engineering resources toward technology development. The outcome of this step is identification of the points of market leverage for the chosen solution.

6) Define (diagram) top-level subsystem dependencies and commitments. The internal study found that most projects were very good at identifying their subsystem dependencies. What was almost always

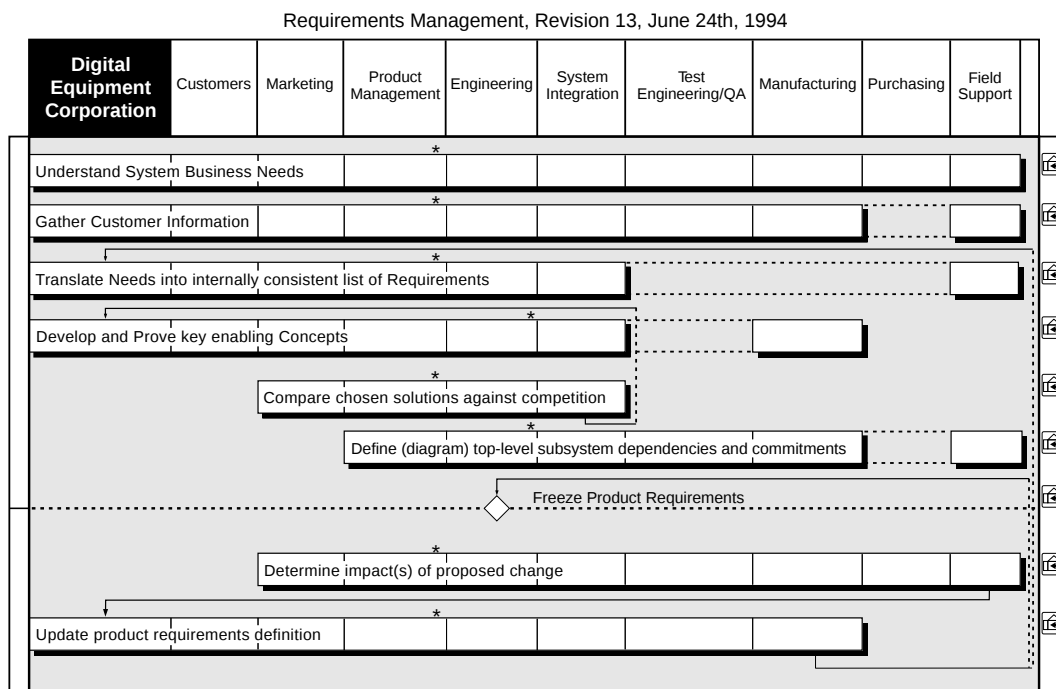


Figure 2. The Requirements Management process in detail

lacking was the translation of a known dependency into a dialogue for commitment. (There were a number of cases in which the status of critical dependencies was not fully known until the field test!) The dialogue or conversation is so important because it allows and encourages teams to truly share why they should support one another and what they can get out of the dependency in a win-win context. The result of this step is the formalization of a contractual commitment for each known dependency, which includes regular updates of development status.

7) Freeze product requirements. By this point, the product specification has been derived from customer data compared against the competition, and low-fidelity prototypes may even have been iteratively validated with the customers to ensure the functional match between the intended solution and the customers' work. At this point a stake is put into the ground (figuratively speaking), since low-level design and code are now being generated and changes to the product become extremely costly. Requirements are considered frozen at this step, with the delivery of the detailed Project Plan.

8) Determine impacts of proposed change. There will always be an impetus to change requirements. Some of it comes from customers, while other forces are brought to bear by partners, technology evolution, market shifts, and other factors. However, recognition of the severe impact changes have on the scheduled commitments results in this step in which any major modification or request for new requirements is accompanied by a cost/benefit analysis detailing the case for and the risks incurred by the proposed change. This data is analyzed by the full cross-functional team to reach a decision on whether the change should be implemented now, in a subsequent release, or in some cases, not at all.

9) Update product requirements definition. This step is meant to ensure that all representations of the product configuration (requirements, specifications, project plans, and designs) are consistent with each other and updated in the case of change. The results of this step are formal revisions of the product requirements when change occurs.

Experiences With The Pilot Implementations

The Requirements Management process was not designed separately from product development. In the re-engineering effort, the nine steps were prototyped and modified in the context of eleven pilot projects that were implementing and debugging the process. Each implementation was guided by a process facilitator, who maintained the integrity of the process by working with the team to modify and adapt methods and techniques in executing each of the process steps.

The pilots encountered their share of problems in working out the bugs in the process. (Before going any further, a word about the pilots. These projects

were a mixture of eight software projects, two service projects, and a hardware project. All were chosen because a senior member of the project, either a Product Manager or an Engineering Manager, acknowledged his or her concern for the kinds of problems the Requirements Management process claimed to address or even was a fan of one or more of the constituent processes). The re-engineering team paid considerable attention to studying these problems in the aggregate, trying to understand whether they shared any root causes.

Two classes of root-cause problems emerged from this study.

Problem 1: Many product development personnel have expectations that knowledge-driven, creative processes such as Requirements Management are similar in execution to deterministic, manufacturing-style processes. The symptoms of this problem showed up as dissatisfaction by some of the teams with the output of Steps 2, 3, and/or 4—the gathering and analysis of customer data and the creation and validation of enabling concepts. When the re-engineering team studied the difference between teams that felt very successful with these steps and compared their experience with teams that did not, it found that the underlying reasons were not due to the particular methods by which the steps were executed. Rather, the more successful teams had developed an enlightened view of the “knowledge” work implicit in Steps 2, 3, and 4

Common among the teams that were dissatisfied with the outcome of these steps was a view of process as a deterministic procedure. In this view, process execution meant inputting the outcome of a previous step into the sequenced procedures of the current step, thus producing the desired result. This expectation was grounded in a manufacturing-like view of process: execute the prescribed steps and predictably attain the desired result. For example, some teams executed step 3 by simply inputting the customers' problems discovered in step 2 into a decision matrix¹ format, such as Quality Function Deployment (QFD)¹ [2], and outputting a prioritized list of features and functions. In these cases, teams with the deterministic point of view were often dissatisfied with the quality of their results. This was because they had expected the process to do the creative work of contextualized data collection, interpretive data analysis, and responsive design. In their view, any process step should be mechanistically executed: put in the mate-

¹Quality Function Deployment, invented in Japan over 15 years ago, focuses on determining the key quality needs of customers and systematically rank-ordering them according to importance, how much improvement over the competition is needed, and how much marketing edge can be gained from meeting the need. The process concludes by correlating (and thus also rank-ordering) a set of metrics for the quality needs or even features of the proposed product concept with customer needs.

rial, turn the crank, and out come the results. And, to be sure, that is what happened for these teams. They got the predicted results, but for a variety of reasons they were less than completely satisfied with them.

However, the teams that were very satisfied with their results had in common a view of process as a guide, not as a deterministic sequence of procedures. These teams took a highly adaptive approach to the procedures they had employed, sometimes inventing their own. It is important to recognize the more successful teams were engaged not only in the design of a solution for the customer problems they were trying to solve but also in the creative design of the means by which they chose to define and resolve the problems.

For example, in step 2 (Gather customer information), some teams used focus groups or product definition forums, while others applied Holtzblatt and Beyer's contextual methods [3] to inquire into and develop models of customer work. In steps 3 and 4, some teams would adaptively apply Kano's methods [7], variations on QFD [2] and/or forms of rapid prototyping to develop and validate enabling concepts and reach agreement on the best solution.

This variety in methods and techniques applied by the successful teams brought home a major lesson: The quality of the results of a given process step was due less to the specific technique utilized than to the commitment and creative involvement of the team to its execution. That is not to say that all the techniques were equally valid. To be sure, some situations called for a clear preference of one method over another. However, the most critical factor to reaching a conclusion that would bring about satisfactory agreement among the members of the cross-functional team was that all the team members felt ownership of the method used to execute the process step. For any given step, no single method accomplished this better than others in all cases.

This clearly indicated that an engaged, involved cross-functional team felt ownership of all aspects of their work—not just the deliverables—but also the methods it adapted to attain results. No one method or technique would be right for all situations, and the more successful teams would invent their own, using the process step as a map of the terrain they needed to cross, which allowed them to embark on the journey of development by way of a path they forged together.

Problem 2: The work of the cross-functional team focused primarily on engineering deliverables. The symptom of this problem was a most common one: most of the pilots found it difficult to hold the cross-functional teams together over the entire development period. That is not to say that the team would fall apart, but there was an almost predictable pattern of early cross-functional involvement that would dissi-

pate over time, leaving the majority of members during the latter stages coming from engineering and product management ranks. The function that most often departed early was the marketing function (which was also usually the most difficult to enroll).

When members of the re-engineering team spoke with marketing personnel who dropped out, it was found that they were responding to very real pressures in their jobs: They were often called away by the central marketing organization to plan sales updates, create brochures, or stage customer engagement events. While these were seen as legitimate reasons to be called away, they also revealed the root cause for the difficulty of enrolling them and maintaining their committed participation: Marketing personnel did not see their work showing up in the Requirements Management process.

This insight revealed a significant flaw in the original design of the Requirements Management process. While the re-engineering team had valued the perspective marketing would bring to a cross-functional product development team, it was implicitly envisioned that their contribution would be supporting the engineering objectives (i.e., deliver a requirements document, functional specification, designs/prototypes, and code). Thus, the notion of a cross-functional team and the definition of its work came out of an "engineering-centric" mindset. The re-engineering team had not incorporated the *work* of marketing into the Requirements Management process, and consequently participants from the marketing function could not readily see the value of their participation in terms of the deliverables of their function.

Mindful of this lapse in thinking, the re-engineering team felt a critical need to develop an understanding of the goals and work of the marketing function. To meet this need, they decided to employ steps 2, 3, and 4 of the process itself as a guide to its revision. The re-engineering team's plan was to view the marketing function as the target customer, collect in-context data on their work, then analyze this data to create prototype revisions of the process and validate these prototypes with marketing personnel. The goal of the revisions was to enroll marketing to participate fully in pilots of the revised Requirements Management process.

A series of 14 contextual inquiries into the work of marketing were conducted. Analysis of the data found that nearly all of the important marketing deliverables were missing from the Requirements Management process. These included:

- Marketing Messages: Positioning the solution in the mind of the customer.

- Channels: Strategizing how the product will be distributed and deployed.
- Communications: Planning the media and events to induce buying.
- Pricing Model: The chosen solution must include the dimension of how much customers are willing to pay.

Process Revisions—An Opportunity for Concurrent Development

Interpretive analysis of the marketing function's contextual inquiry data presented the Requirements Management design team with an opportunity to redesign the process to achieve greater concurrency in the work of all the cross-functional team members. While the hallowed phrase "concurrent engineering" felt uncomfortable and outmoded, the team did feel it was moving toward a concept of "Concurrent development." The goal of this concept was to lay to rest once and for all the bogey of an engineering-only worldview. The collective experiences of the cross-functional pilots had clearly shown the design team there is much more to delivering a solution to customers than just its engineering content. This is not to belittle the quintessential importance of the engineering element but rather to acknowledge the other vital ingredients of a successful solution, including:

- The Marketing and Advertising messages
- The user information (documentation)
- The pricing and competitive positioning of the solution
- The sales and support service

A redesign of the Requirements Management process offered a unique opportunity to develop all these aspects of the "whole product" or solution concurrently.

The Requirements Management design team expected engineering resistance to incorporating the work of the other functions into the Requirements Management process. Yet, in conversations with technical and project leaders, it found a compelling message in the promise that the marketing messages, the support programs, the competitive positioning, the advertising campaigns, the user documentation and information could all be ready for delivery at the same time and in complete concordance with one another. The design team decided to challenge engineering with the feasibility of this approach by noting that there was nothing but their own intransigence to stop them developing all these facets of the solution in parallel. All that was required for this to become a reality would be for the cross-functional team to agree to do this, to work in parallel on their own unique deliverables, to keep in regular communication throughout, and to base all their work faithfully

on the frozen customer requirements.

The paradigm of the elements comprising the "product" had to change for the cross-functional team to envision the possibility of concurrent development. Instead of the product being conceived of as a set of capabilities delivered to the customer, the design team adopted the term "whole product" to refer to the additional elements the customer experiences in the full business relationship: marketing messages; the sales engagement; the distribution channel; the pricing, packaging, and licensing policy; and warranty and service agreements.

The Requirements Management process was revised to focus on concurrent development of whole products (Figure 2; see also [5]). While the original nine steps remained intact, they were hidden in the background of the process descriptions. As depicted in Figure 1, the process has been simplified into three broad phases, each of which involves customer feedback and validation of outcome. The intentions and expected results of each phase correspond to the original nine steps, with the revisions focused on ensuring that the deliverables of the full cross-functional team would be attainable out of the same work that delivered the solution definition.

Concept Development

In the marketing function contextual inquiries, one of the people interviewed was in the process of putting together the sales update for an upcoming announcement. A remark made by this individual at the time radically altered the thinking of the re-engineering team: "I've never understood why I am doing the sales update now. I should be doing it when the feasibility and requirements are being developed." This revolutionary thought provided the first opportunity to incorporate the work of marketing and its deliverables into the Requirements Management process. The new possibilities opened up by this thought were: What if the same in-context data used to understand customers' work and their problems could be used to derive the marketing messages for the sales update? and What if the initial prototype of the product included a prototype of the sales update so the cross-functional development team could seek feedback from the customer to validate the strategic positioning and vocabulary of the marketing messages?

Accordingly, the initial steps (1, 2, 3, and 4) of the Requirements Management process (Figure 2) were revised to be represented as Concept Development (Figure 1). The focus here was to develop the whole-product concept through a deep understanding of the problems customers experience in working to achieve their goals. As before, this understanding provided the focus for prioritizing the solution capabilities customers needed. But now it also served as the basis for formulating marketing messages that

expressed the value expected by customer's, at a given time and at a price they were willing to pay. Thus, the concept arrived at in step 4 was not just an expression of engineering specifications but also included the communications that would induce customers to buy. This whole-product concept was validated and refined by the customers from whom the in-context data was collected.

Product Scenario

The newly adopted notion of whole products led to the revision of steps 5, 6, and 7 of the Requirements Management process as the development of the whole-product scenario. The function of the scenario was to serve as the basis for the business readiness decision. Already the outputs of steps 3–7 contained many of the components of the engineering scenario: Concept Description, Prioritized Requirements, Low-Fidelity Prototype, Functional Specification, Design, and Dependencies. However, only step 6 delivered a component of what might be considered the Marketing Scenario, that being a Competitive Analysis. Accordingly, the work of steps 3–7 was expanded to include delivery of Marketing Messages, Pricing Model, Distribution Channels, and Communications Programs. As other functions are represented on the cross-functional Requirements Management team, the whole-product scenario would be further expanded to include their specific components. The idea here was early prototyping of these components and their validation and refinement with the customer-partners, thereby making a science out of the business readiness decision.

Rapid Prototyping

Both Concept Development and Product Scenarios include low-fidelity prototyping in the actual work of validating and refining concepts and scenarios with customers. However, the re-engineering team was acutely aware of the possibility that new requirements would be demanded once users had a working model of the product in their hands. Thus, in the very early stages of implementation, GUI “shells” and early baselevels are given to customers, with the feedback intended to refine the operational usability of the product. Recognizing that this level of prototyping will inevitably lead customers to ask for new capabilities from essentially frozen (step 7) requirements, the process was revised to handle customer feedback with the cost/benefit analysis of step 8 and, if a change is made, with the configuration management focus of step 9.

In addition, we saw rapid prototyping as a proactive way to gather rich, in-context data for subsequent releases. In this manner, we could envision our interaction with customer-partners as the basis for planning a family of releases. The goal is to adhere to the time constraints determined during the Concept Development and Product Scenario phases— giving

customers what they need as soon as possible and looking to expand functionality in a planned, orderly set of releases, while preparing the design now for the intended growth path.

The Human Dimension of Change

In addition to the organizational and functional issues encountered during this re-engineering process, we observed two key elements of the human side of introducing a process such as Requirements Management into an organization:

- Turning the cross-functional group into a real team
- Overcoming people's natural resistance to change

The Cross-functional Group as a Real Team

It was important that we carefully selected representatives from the key functions that have a stake in the project's success: product management, marketing, testing, manufacturing, usability, documentation, engineering, and support. In a company whose history has been dominated by engineering leadership, facilitation of the meetings had to be conducted especially carefully to accord balanced weight to these different functions. The early meetings were a carefully orchestrated attempt to form a functioning team. The facilitator's job was often getting the various team members to learn to listen to one another's views and appreciate their perspectives.

Resistance to Change

People have a natural resistance to change. This Requirements Management process in particular was very different from what Digital teams had been used to. The facilitator, now a change agent, had to continually show teams the benefits of the process and reassure them that the uncertainty they were experiencing (due to putting the customer in the driver's seat) and the time they were expending would indeed yield a valuable and rewarding experience. One of the ways to achieve this is to continually make visible the partial results as they emerge, in a public setting: for instance, to display the organized results of customer visits, before their prioritization into requirements, in the form of a large wall-hanging in the project room.

Conclusion

At this time, two additional pilots based on the revised process are in progress. While it is still too early to gauge their full experience, it has been clear that marketing messages can be derived from the in-context study of customer work and can be validated at the same time and in the same manner as low-fidelity (e.g., paper) prototypes.

It is also clear that the perspective of product

development as a social enterprise has been a missing point of view in a technology-driven company such as Digital. Exhortations to be more customer-driven all too often result in the designation of responsible functions to become accountable for customer contact. Ironically, the outcome is increased distance from actual customer experience and further fragmentation among the cross-functional perspectives needed to deliver a "whole" product.

The Requirements Management process being employed by many development projects at Digital today seeks to develop relationships among cross-functional perspectives and a direct relationship between the entire team and their customers. Out of these relationships commitments are made to *listen to* the customer, with integrity. With direct customer partners, the team prototypes, validates, and refines all the major components of the business relationship. Accountability is operationalized by continued direct engagement with customer partners throughout the development period.

The experience of those who have facilitated the Requirements Management process has been observed enthusiasm and an increasing respect for the value of multiple perspectives in making both the design choices and the business decisions. As with any development effort, there are certainly times of struggle and tension, but it has been

observed that the business-as-usual mistrust among the functions is being replaced by a positive sense of team ownership and a perception that no one function holds all the answers, that no one function can be successful alone.

Acknowledgments.

Several times in this article, we have made reference to the "Requirements Management re-engineering team." Just like the very process we describe, the work of developing this Requirements Management process was a cross-functional team effort, and we would like to acknowledge the great insights and help of the following people: Faye Allen, Michael Bendit (of Booz-Allen-Hamilton), Norm Celotto, Subhash Dandage, Len Fehskens, Steve Holmes, Fred Howell, Paul Huntwork, Michelle Lynn, Steve MacDonald, Ed Paterline, Reza Sisakhti, Tom Szczepanski, Anne Warner, and Paul Wood. ■

References

1. Achieving Excellence in Engineering Project Profile Survey, conducted by Booz-Allen-Hamilton, Inc. for Digital Equipment Corporation, May 1993.
2. Cohen, L. Quality Function Deployment: An application perspective from Digital Equipment Corporation. *Nat. Prod. Rev.* (Summer 1988).
3. Hammer M., and Champy, J. Re-engineering the Corporation. *Harper Bus. Rev.* (1993).
4. Holtzblatt, K., and Beyer, H. Making customer-centered design work for teams. *Commun. ACM* 36, 10 (Oct. 1993), 92-103.
5. Moore, G.A. Crossing the chasm. *Harper Bus. Rev.* (1991).
6. Paulk, M.C., et al. Capability Maturity Model for Software. Tech. Rep. CMU/SEI-91-TR-24, Software Engineering Institute, Carnegie Mellon Univ., 1991.
7. Walden, D., Ed. A Special Issue on Kano's Methods for Understanding Customer-Defined Quality. *The Center For Quality Management J.* 2, 4 (Fall 1993).

About the Authors:

ANTHONY F. HUTCHINGS is a vice president in J.P. Morgan's Corporate Technology group, in the area of software methods and quality. After leading Digital Equipment Corporation's VLSI CAD group, he formed a software process and quality consulting group, the Software Engineering Technology Center, within the company. Current research interests include software complexity management, software metrics, and requirements engineering. **Author's Present Address:** J.P. Morgan, 60 Wall Street, M/S: 5/75W, New York, NY 10260; email: hutchings_tony@jpmorgan.com

STEVE T. KNOX is a consultant engineer with Quantum Corporation Disk Manufacturing. Current research interests include statistical optimization of disk-drive manufacturing processes. **Author's Present Address:** Quantum Corporation, 333 South Street, SHR1-3/M10, Shrewsbury, MA 01545; email: knox@ezdisk.enet.qntm.com

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and /or specific permission.