

Understanding the Product Life Cycle:

Four Key Requirements Engineering Techniques

Christof Ebert, *Alcatel*

A field study involving many industry projects revealed that only those that took a requirements engineering perspective in four key product life-cycle management activities were successful.

Many enterprises view time to market and schedule performance as the key differentiators between market leaders and followers. By maintaining schedule commitments and shortening cycle times, companies become more reliable suppliers and can optimize profitability. Pressed to accelerate project handover and new-product commercialization, companies have improved R&D execution over the years with instruments such as CMMI (Capability Maturity Model Integration).¹

They continue, however, to face challenges in cross-functional coordination, which result in project delays and long cycle times.

Specifically, organizations frequently commit to requirements and contracts to boost short-term revenues without properly aligning sales, product management, project management, and marketing. Such misalignment results in insufficient capacity planning or product-development resource allocation, thus delaying projects.

For this field study, Alcatel's R&D Effectiveness team investigated products and solutions within Alcatel's different business groups. We extracted in the first part of the study four seemingly successful product life-cycle management techniques. Then we quantitatively evaluated the impacts of these techniques on those projects. Results verified that when used in parallel, the four techniques can indeed significantly reduce delays.

Requirements and the product life cycle

Creating a successful product requires identifying market needs and translating them into a product vision and scope, which are executed following sound project management principles. Product management is the role of governing a product from its inception to delivery to generate the greatest possible value to the business. Requirements are the basic building blocks gluing together different phases of the *product life cycle*—the sum of all activities needed to define, develop, implement, build, operate, service, and phase out a product and its related variants. However, the 2003 CHAOS report showed that only 52 percent of originally allocated requirements appear in the final released version.²

The traditional rule of thumb for allocated requirements is to expect them to change by 1

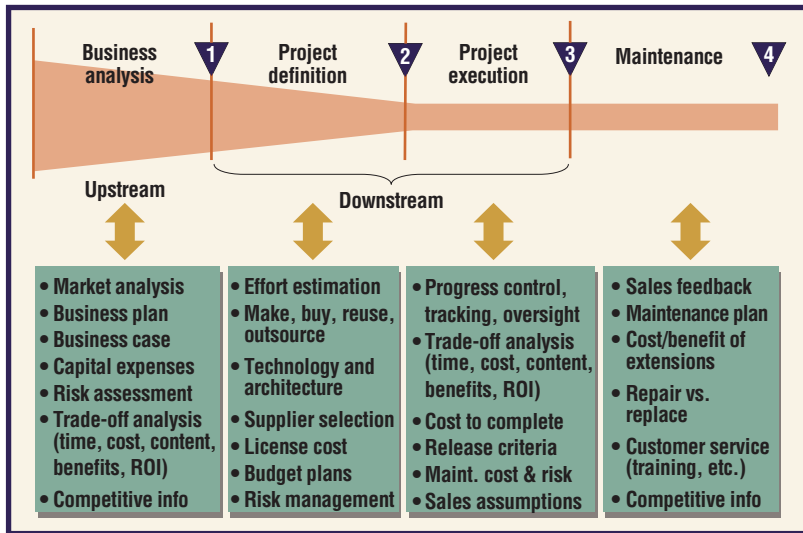


Figure 1. A simplified product life cycle with four decision gates and upstream (left) and downstream (right) processes. The product manager can stop product development at each gate.

to 3 percent per month.^{3,4} For a two-year project, this typically translates into changing (adding, deleting, or modifying) more than 30 percent of your requirements. Because such changes almost always lead to delays, many contractors and clients prefer projects that last no longer than one year. Research has shown that such projects perform better,² but shortening project duration and working with iterations doesn't always solve the problem.

Reducing requirements changes requires effective *product life-cycle management*, which assures understanding and collaboration across the entire product life cycle to deliver the greatest business value to an enterprise, its stakeholders, and its customers. Figure 1 shows a simplified product life cycle based on the cycle that Alcatel and many other companies use. The upstream process covers everything prior to the first decision gate (marked "1"), while downstream processes relate to the different activities between gates 1 and 3. (Typical software life cycles follow IEEE 12207 or IEEE 1074 standards, both of which implement a gating (or review) process between major phases.⁵)

Although research and practice of project management and requirements engineering (RE) have paid a lot of attention to downstream processes around project execution,¹⁻³ they haven't sufficiently researched upstream processes, even though they're also a part of RE. This is often because of these processes' complexity (that is, their heterogeneous and overlapping ownerships, vague processes, and unclear impacts of stakeholders). At most, the upstream activities are mentioned in the re-

quirements elicitation process. A major barrier to implementing strong upstream processes is the short-term profit-and-loss responsibility, which provides incentives to focus on current-quarter results (that is, ongoing projects and contracts) rather than future products or platforms. During our study, we aimed to view product life-cycle processes, such as implementing gate reviews or creating empowered project teams, as part of the RE discipline.

We viewed this as important because poor upstream processes can result in insufficient project planning, continuous changes in the requirements and project scope, delays, configuration problems, defects, and overall customer dissatisfaction (owing to broken commitments or unmet expectations). In 2004, Robert Cooper studied new-product development in 105 business units from various industries in the US.⁶ He found that the top 20 percent delivered 79 percent of their new products on time, while the average companies delivered only 51 percent of projects on time. Furthermore, from that same set of top-ranked companies, 66 percent broke down their resources according to strategic needs. Only 31 percent of the average companies aligned project resource utilization with strategic needs.

Another disturbing trend we wanted to address is the increased duration of the project's analysis (or elaboration) phase. Product managers, system analysts, and engineers have become so paranoid about the expected number of requirements changes and delays that they often spend too much time on requirements analysis before the project even starts. This syndrome, often called "paralysis by analysis," contributes to project duration and cost and results in additional delays. We thus face two vicious circles, both caused by insufficient or nonexistent upstream processes and both contributing to delays (see figure 2).

Alcatel's field study

Alcatel, which has 56,000 employees in 130 countries, develops technologies and provides communication solutions for telecom operators, Internet access providers, and enterprises. Our goal was to study how to reduce project delays. (We didn't investigate impacts on sales, revenues, or customer satisfaction because the market situation can greatly influence these variables, so they're difficult to compare.)

We used all projects that had recorded valid

data in our history database and had been completed between January 2002 and October 2003. We didn't throw out any "outliers," thus avoiding overlooking influences from parameters other than the four techniques we analyzed. Project size in terms of effort was between a few person-weeks and several 100 person-years.

We designed the field study as a two-step approach. First, we collected evidence and performed a thorough root-cause analysis in a rather small sample (one representative product line). We carefully evaluated all projects in this representative product line to better understand why requirements change during a project. The product line used in this sample check had 15 projects, all developed in 2002. On average, these projects changed 73 percent of their requirements after the project started. One third of these changes were technical (such as an infeasible specification) while two thirds were commercial (covering the majority of requirements uncertainties). More often than not, the changes involved deleting or replacing requirements, but because requirements have different impact and effort needs, replacing one requirement with another doesn't always preserve the project effort.

We gained several key insights from this analysis:

- RE must start early and connect portfolio considerations with project management.
- All relevant stakeholders must be available and empowered.
- The product life cycle must be mandatory for all projects, independent of type, size, or scope.
- All elements of the portfolio must be equally and easily visible.

To design the empirical study, we translated these initial observations into four techniques we felt would be the key to better project life-cycle management. We treated them as independent variables and checked them against each of the projects:

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

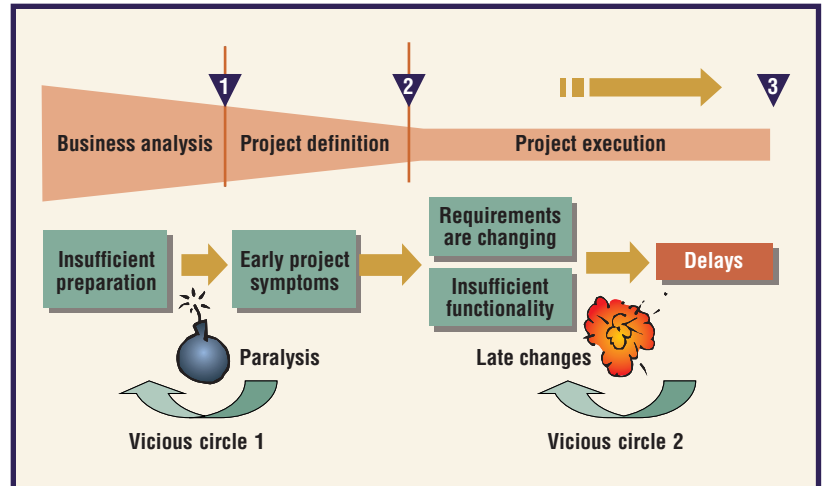


Figure 2. Insufficient upstream processes cause paralysis and late changes and ultimately delays.

We deduced these four independent variables from Alcatel's history database by evaluating the availability of project core teams, noting whether early gating reviews had been conducted and reported, evaluating the accessibility and use of requirements, and checking the availability of product release information. We used completeness and sanity checks to verify each technique's existence and metrics. If the metrics looked odd (typically the result of typing errors), we double-checked them with the owners.

Having these four independent variables closely combined with each single project in our study, we could extract each variable's impact on performance. This avoided the shotgun approach with uncontrolled independent variables.⁴

We counted requirements on the basis of product requirements as opposed to customer or component requirements. These project requirements reflect an external view of the project and explain what the project should do, not how it will be designed. We baselined the requirements by counting them at the project's start. Changes to the requirements included all modifications, additions, or deletions made after the project's start. We normalized the number of changes with the number of requirements in the specific project so that we could compare the number of changes across projects.

We measured the project duration as the time between the gating review that kicked off the project (figure 1, gate 2) and the gating review that decides to release the project to the customer or market (figure 1, gate 3). Latest, at project start, the core team had to commit to a firm end

Table 1**Summary of projects from the field study**

No. of techniques used	No. of projects	Minimum size (person-years)	Maximum size (person-years)	Average schedule (%)
0	134	0	47	136*
1	30	2	148	147
2	30	0	780	121
3	17	2	177	113
4	35	1	225	103

* 100% would indicate an on-schedule project.

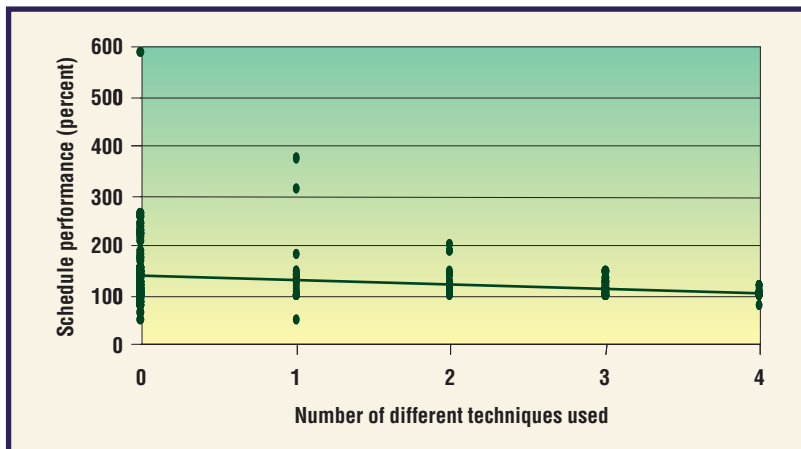


Figure 3. Schedule performance and volatility improve with combined use of three or four of the techniques.

date and record it with the other planning data. So, to measure project delays (and thus schedule performance), we compared the project-end milestone to the reported end date. We normalized this absolute difference in calendar days using the originally committed project duration in calendar days. If a project was initially scheduled to run for 100 calendar days and finished after 120 calendar days, it had a (project) delay of 20 percent and a project (schedule) performance of 120 percent. This metric is the dependent variable we analyzed in the study.

Table 1 provides an overview of the 246 projects we evaluated. The first column shows our control variable—the number of techniques used. If a project used one, two, or three techniques, we don't detail which combination of techniques were used, because we found that it didn't matter (this is discussed later). There is no order or dependency in using these four techniques. The last column shows our study's dependent variable as an average of the schedule performance is good enough (in the neighborhood of 10 percent overruns) only when using three or four of the techniques.

Having implemented these techniques in various orders and at various speeds let us correlate the results (project delays) with the number of techniques used. We found that schedule performance was independent of project size—within the range of project size that we had in our set.⁷ So, smaller projects didn't necessarily perform better than the larger ones. In fact, delayed projects were almost evenly scattered across project size. A Pearson rank correlation showed $r = 31$ percent, indicating no relationship between size and delays. A double-sided chi-square test evaluated whether using the four techniques related to project size. We could accept the homogeneity hypothesis (a statistical test) on a significance level of $\alpha = 1$ percent. Use of any particular technique didn't depend on project size.

We also determined that none of the four techniques were used in correlation with each other, which we verified using a double-sided chi-square test. We could accept the homogeneity hypothesis on a significance level of $\alpha = 5$ percent (or lower) for each single pair of the four techniques, indicating that they weren't correlated in usage.

The four techniques

Figure 3 summarizes the results of the impact analysis of the four techniques. Performance significantly improves with three or four of the techniques. Using none, one, or two techniques doesn't have much impact on project delays. A double-sided chi-square test accepts the homogeneity hypothesis on a significance level of $\alpha = 1$ percent. There is a significant impact on reducing delays when using three or four techniques. The number of projects delivered on schedule improved by 20 percent when using three or four techniques. The number of projects that were late 5 percent of the time or less increased from 45 percent to 63 percent with three or four techniques used, and projects that were late 10 percent of the time or less increased from 56 percent to 77 percent.

The four techniques are typically reinforced in organizations committed to process improvement. Organizational entities (business units or regional entities with product responsibility) that view life-cycle processes and performance improvement as management fads or approach a technique as a single-shot change won't see sustainable results. An entire

organization must buy into product life-cycle management and capture all functions. Training and clear role descriptions are key to making any of the techniques a success.

Install an effective core team for each product release

Three roles dominate decision-making during the product life cycle: the product manager, marketing manager, and (often technical) project manager. They must build a multifunctional core team fully accountable for the product's success. They represent not only the major internal stakeholders during product or solution development but also the sales and customer perspective. This core team must have a clear mandate to own the project. If such a team is available but underlying commitments aren't baselined, the team 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 the projects reach their objectives. This is owing to the empowered team and the clear line of command for all relevant stakeholders involved with the entire project.

The product manager must consider the long-term business goals beyond the single product. She determines what to make and how to make it and is accountable for business success within an entire portfolio. She approves the roadmap and content and determines what and how to innovate. She's 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's 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. He 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's 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

minibusiness without continuous external interferences. They evaluate the portfolio's cost and benefits and make it an overall success.

Focus the product life cycle on upstream gate reviews

Gate reviews are essentially a communication and risk-management tool. Each gating review evaluates the project status on the basis of predefined criteria and decides whether and how to proceed. The top driver is the portfolio, which depicts all products within the company and their markets and respective investments. Each product should have a feature catalog across the next several releases covering the vision, market, architecture, and technology. The team can then drive a technology roadmap from this product roadmap, letting it select suppliers or build partnerships. It also drives the individual roadmaps of releases and projects, which typically have a horizon of a few months up to one year.

The product life cycle must be mandatory for all projects. This implies that it's 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 shouldn't result in lengthy meetings; rather, all attendees should receive online checklists before each meeting so they can quickly decide whether to move to the next phase. In fact, most project information should be available online.

A useful product life cycle must acknowledge that requirements might never be complete and might indeed be in a state of continuum. Sometimes requirements are purposefully incomplete. The product life cycle should include defining stopping criteria so that team members can determine what is good or stable enough.

Evaluate requirements from various perspectives

The entire core team evaluates requirements to ensure different perspectives are considered. The team must justify each requirement to support the business case and to help manage changes and priorities. Often, market

A useful product life cycle must acknowledge that requirements might never be complete and might indeed be in a state of continuum.

**Assuring
dependability
means
maintaining
commitments to
agreed-upon
milestones,
contents, or
quality targets.**

demands are mistranslated into ad hoc project contents, resulting in ever-changing requirements. Impact analysis is thus based on requirements⁸ as well as on priority setting and *portfolio management*—a dynamic decision process aimed at having the right product mix and selecting the right projects to implement a given strategy.^{4,9}

Proper impact analysis from product management, marketing, and technical perspectives helps assure the team remains focused on what it can do without overcommitting itself. The evaluation considers 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 the product or requirements and for what reason? Are the requirements necessary for a solution or just inherited from an incumbent approach, perhaps having become obsolete in the meantime? To answer these questions, the core team must document requirements in a structured and disciplined way. They must express the requirements using both technical and business judgment. Also, they should review any incoming requirements, with the product catalog and global product evolution in mind, to evaluate marginal value versus marginal costs. Doing this avoids mistranslating market perceptions into unfocused requirements.

Assure dependable portfolio visibility and release implementation


Assuring dependability means maintaining commitments to agreed-upon milestones, contents, or quality targets. For instance, within a product-line architecture, the underlying generic product, platform, or components upon which many customization products will be built must be on time and must provide the agreed-upon contents. Otherwise, numerous ripple effects will follow. Naturally, project management techniques differ between a generic and an application product—the platform must build in resource buffers while the application product can easily work with feature prioritization and *time boxing* (that is, delivering flexible content within a fixed timeframe).

Having a project plan that's directly linked to requirements is mandatory. If a plan or contents change, the entire team must approve the change and synchronize it with the rest of the project. Requirements as well as other relevant product and project information must be ac-

cessible online. The team can use different tools, starting with simple spreadsheets.¹⁰ A Web-based company-wide dashboard (or project release portal) with standardized metrics (which Alcatel uses) facilitates easy and standardized visibility of the project's progress and increases the stakeholders' understanding of project dependencies.

Full portfolio visibility in software projects means that relevant stakeholders can access and assess all projects continuously and in their totality. Such a portfolio report should build an online database using project information directly from operational databases so that such information is accessible for further impact analysis and project planning. To facilitate smooth and continuous data collection and aggregation without generating huge overheads, extraction should be highly automated and accessible from intranet portals. If requirements, constraints, or assumptions change, the core team must adapt the evaluation. The quality of the data and requirements lists are key to trusted decision making.

The projects we evaluated, though primarily in the domains of embedded systems and software applications, cover a wide range software-dominated projects, so the results apply to industries other than just communication. (Benchmarks we've done for automotive, defense, and information systems also support this.^{3,4}) The techniques discussed here assure visibility, agreement, and commitment to reduce requirements changes. Together with strong project management from a team of empowered stakeholder representatives, the techniques improve predictions and schedule performance. Integrating upstream processes with product-life-cycle-related RE processes can help reduce delays by over 20 percent per year.

Furthermore, the techniques are tangible and can be formally introduced to projects during the launch period, thus reducing the change impact (there's no big bang). Practitioners in engineering, product management, and marketing accept them because they yield results and stimulate empowered project teams. At times, they require customer education, but this is feasible given the benefits the customers will experience with more predictability and faster reaction time. 

Acknowledgments

We thank the Alcatel product managers and engineering teams that shared over past years their best practices and helped in setting up product-life-cycle management.

References

1. M.B. Chrissis, M.Konrad, and S. Shrum, *CMMI: Guidelines for Process Integration and Product Improvement*. Addison-Wesley, 2003.
2. *CHAOS Chronicles v. 3.0*, The Standish Group Int'l, 2003; www.standishgroup.com/chaos/toc.php.
3. W. Royce, *Software Project Management*, Addison-Wesley, 1999.
4. C. Ebert et al., *Best Practices in Software Measurement*, Springer, 2004.
5. M.E. McGrath, *Next Generation Product Development: How to Increase Productivity, Cut Costs, and Reduce Cycle Times*, McGraw-Hill, 2004.
6. R.G. Cooper et al., "Benchmarking Best NPD Practices: Research-Technology Management," Part I, 2004, p. 31, Part II, 2004, p. 43, Part III, 2004, p. 43; www.apqc.org.
7. C. Ebert, "Requirements BEFORE the Requirements: Understanding the Upstream Impacts," *Proc. Int'l Conf. Requirements Eng.* (RE 05), IEEE CS Press, 2005, pp. 117-124.

About the Author



Christof Ebert is the director of R&D at Alcatel, Paris. His research and consulting covers process improvement and requirements engineering. He received his PhD with honors in electrical engineering from the University of Stuttgart, where he lectures on software engineering. He's a member of the Alcatel Technical Academy and an SEI-authorized CMMI instructor. He's also a senior member of IEEE Computer Society and the editor of *IEEE Software's* Open Source column. Contact him at Alcatel, 54, rue la Boetie, F-75008 Paris; christofebert@ieee.org.

8. J. Giesen and A.Voelker, "Requirements Interdependencies and Stakeholder Preferences," *Proc. Int'l Conf. Requirements Eng.* (RE 02), IEEE CS Press, 2002.
9. L. Gorchels, *The Product Manager's Handbook: The Complete Product Management Resource*, McGraw-Hill, 2000.
10. J. Doerr, B. Paech, and M. Koehler, "Requirements Engineering Process Improvement Based on an Information Model," *Proc. Int'l Conf. on Requirements Eng.* (RE 04), IEEE CS Press, 2004.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.

PURPOSE The IEEE Computer Society is the world's largest association of computing professionals, and is the leading provider of technical information in the field.

MEMBERSHIP Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEB SITE

The IEEE Computer Society's Web site, at www.computer.org, offers information and samples from the society's publications and conferences, as well as a broad range of information about technical committees, standards, student activities, and more.

BOARD OF GOVERNORS

Term Expiring 2006: Mark Christensen, Alan Clements, Robert Colwell, Annie Combelles, Ann Q. Gates, Robit Kapur, Bill N. Schilit

Term Expiring 2007: Jean M. Bacon, George V. Cybenko, Antonio Doria, Richard A. Kemmerer, Itaru Mimura, Brian M. O'Connell, Christina M. Schober

Term Expiring 2008: Richard H. Eckbouse, James D. Isaak, James W. Moore, Gary McGraw, Robert H. Sloan, Makoto Takizawa, Stephanie M. White

Next Board Meeting: 16 June 06, San Juan, PR

IEEE OFFICERS

President : MICHAEL R. LIGHTNER

President-Elect: LEAH H. JAMIESON

Past President: W. CLEON ANDERSON

Executive Director: JEFFRY W. RAYNES

Secretary: J. ROBERTO DE MARCA

Treasurer: JOSEPH V. LILLIE

VP, Educational Activities: MOSHE KAM

VP, Pub. Services & Products: SAIFUR RAHMAN

VP, Regional Activities: PEDRO RAY

President, Standards Assoc: DONALD N. HEIRMAN

VP, Technical Activities: CELIA DESMOND

IEEE Division V Director: OSCAR N. GARCIA

IEEE Division VIII Director: STEPHEN L. DIAMOND

President, IEEE-USA: RALPH W. WYNDRUM, JR.



COMPUTER SOCIETY OFFICES

Washington Office

1730 Massachusetts Ave. NW

Washington, DC 20036-1992

Phone: +1 202 371 0101

Fax: +1 202 728 9614

E-mail: bq.ofc@computer.org

Los Alamitos Office

10662 Los Vaqueros Cir., PO Box 3014

Los Alamitos, CA 90720-1314

Phone: +1 714 821 8380

E-mail: help@computer.org

Membership and Publication Orders:

Phone: +1 800 272 6657

Fax: +1 714 821 4641

E-mail: help@computer.org

Asia/Pacific Office

Watanabe Building

1-4-2 Minami-Aoyama, Minato-ku

Tokyo 107-0062, Japan

Phone: +81 3 3408 3118

Fax: +81 3 3408 3553

E-mail: tokyo.ofc@computer.org



EXECUTIVE COMMITTEE

President:

DEBORAH M. COOPER*

PO Box 8822

Reston, VA 20195

Phone: +1 703 716 1164

Fax: +1 703 716 1159

d.cooper@computer.org

President-Elect:

MICHAEL R. WILLIAMS*

Past President:

GERALD L. ENGEL*

VP, Conferences and Tutorials:

RANGACHAR KASTURI (1ST VP)*

VP, Standards Activities: SUSAN K. (KATHY) LAND

(2ND VP)*

VP, Chapters Activities:

CHRISTINA M. SCHOBERT*

VP, Educational Activities: MURALI VARANASIT

VP, Electronic Products and Services:

SOREL REISMAN†

VP, Publications: JON G. ROKNE†

VP, Technical Activities: STEPHANIE M. WHITE*

Secretary: ANN Q. GATES*

Treasurer: STEPHEN B. SEIDMAN†

2006-2007 IEEE Division V Director:

OSCAR N. GARCIA†

2005-2006 IEEE Division VIII Director:

STEPHEN L. DIAMOND†

2006 IEEE Division VIII Director-Elect:

THOMAS W. WILLIAMS†

Computer Editor in Chief: DORIS L. CARVER†

Executive Director: DAVID W. HENNAGE†

* voting member of the Board of Governors

† nonvoting member of the Board of Governors

EXECUTIVE STAFF

Executive Director: DAVID W. HENNAGE

Assoc. Executive Director: ANNE MARIE KELLY

Publisher: ANGELA BURGESS

Associate Publisher: DICK PRICE

Director, Administration: VIOLET S. DOAN

Director, Information Technology & Services:

ROBERT CARE

Director, Business & Product Development:

PETER TURNER

rev. 6 March 06