

Achieving and Maintaining CMMI Maturity Level 5 in a Small Organization

Davide Falesi and Michele Shaw, Fraunhofer Center for Experimental Software Engineering

Kathleen Mullen, Keymind

// Economic achievements, success factors, and lessons learned from real-life examples offer a unique reference for practitioners intending to pursue high-maturity CMMI levels, particularly in small organizational settings. //



RESEARCHERS AND PRACTITIONERS alike have studied process improvement for many years.¹ CMMI (Capability Maturity Model Integration) models, provided by the CMMI Institute, “consist of best practices

that address development activities applied to products and services.”² CMMI doesn’t tell you how to implement your project and organizational processes, but it specifies specific and generic practice requirements.

CMMI-rated maturity levels range from 2 to 5, with a 5 denoting an organization said to be “optimizing” because it has highly mature practices that leverage the sophistication of quantitative techniques to measure projects and effectively implements organizational process improvements. High-maturity organizations foster an organization-wide environment of continuous, measured, process improvement and high performance.

Achieving a high-maturity CMMI rating in a small organization (fewer than 50 people) is usually more difficult than at medium or large ones,^{3,4} primarily because there are fewer experience reports for small organizations.^{5,6} That said, most small organizations don’t apply CMMI because it’s perceived as being too difficult.⁷ However, as businesses recognize the benefits of maturity models, there’s growing demand for mature organizations, even if they have fewer than 50 people (<http://fremle.talkingvillage.com/resource.x/1585>).

According to current CMMI Institute database information (<http://cmmiinstitute.com/assets/presentations/2013MarCMMI.pdf>), of the approximately 5,500 organizations rated worldwide, only 344 organizations have been rated level 5 (78 of which are in the US—of these, only 3 percent are small ones, and less than half are non-military). This article describes our eight-year journey in achieving, and one-year experience in maintaining, maturity level (ML) 5 in an organization called Keymind, a division of Luminpoint. Keymind provides software development, strategic marketing, and user-focused services to both commercial and government customers.

Our Journey

When we started our journey, we were convinced of its value, but we

had serious concerns, as many organizations do, about CMMI appropriateness for small organizations. However, despite potential pitfalls, we recognized that no standard is perfect. Moreover, CMMI MLs are clearly required by certain types of customers (federal) for certain projects, and regardless of customers' constraints, we believed that applying the CMMI to our processes would be beneficial, if properly interpreted. Achieving a maturity rating, no matter the level, requires an investment of time and resources that needs to show a return; achieving a high ML requires further investment. Therefore, we defined specific and quantifiable business goals such as increasing profits and numbers of clients by providing desired product quality and capabilities, on time (or faster than expected) and at a competitive cost, while still improving the organization's overall stability.

As Figure 1 shows, our CMMI investment has provided outstanding returns. The blue line, which represents profit per employee, has more than doubled within 10 years; the best performance came during the high maturity years (post ML 3). The figure also shows how the size and quality of our projects increased over time. The purple line—our releases over five years, as measured by requirements per release—shows how we've been able to effectively deploy more functionality in each release. Moreover, the number of bugs per requirement (the green line) decreased 10 times in five years. Because we can deliver more functionality per release to our customers with fewer bugs, the result is higher revenue per employee.

We show size and complexity in terms of number of requirements

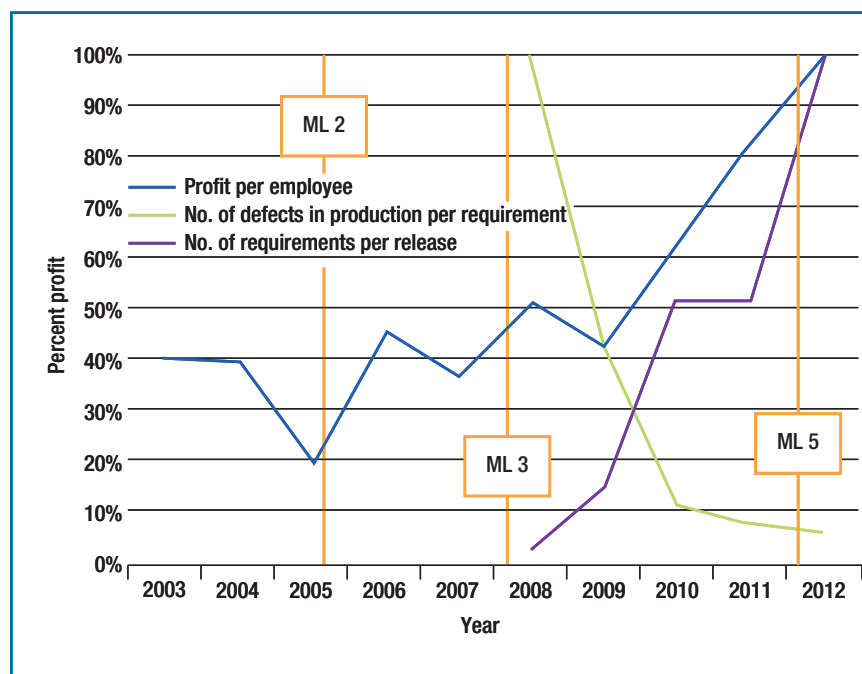


FIGURE 1. Outstanding returns. The trend over the years of profit per employee, the number of requirements per release, and the number of defects in production per requirement shows big gains to our company. We intentionally normalized the y-axis to hide sensitive data.

rather than LOC because the latter is a less reliable metric in our context. Both the “number of bugs per requirement” and “number of requirements per release” have a limited temporal range—that is, we weren’t able to show reliable trends before 2008. This data is unreliable before we were ML 3, a point that introduces an important yet overlooked paradox: it’s difficult to show quantitative improvement over pre-ML 3 years because data collected can be less reliable or insufficient for application of statistical analysis across the entire organization.

How We Started

To understand what to do to comply with CMMI, we started by searching for experience reports. Unfortunately, not much was available at that time (around 2004), so we be-

gan talking to and working with local CMMI and process improvement experts. Our primary goal was to establish practices that both fit the specific organization and were CMMI compliant. Initial practices included project estimation, planning, and configuration management defined in a way that could be applied to every project in the organization. After achieving ML 2 and then ML 3, we started to see productivity benefits, and, at the same time, people in the organization became galvanized by our process improvement initiative. When we learned that we had to re-appraise every three years to maintain our ML 3 rating, it only made sense to further improve using CMMI and to target ML 5.

To successfully move from ML 3, we started by analyzing the literature, looking for reports on how

other organizations achieved ML 5. The analysis of the few available studies revealed that the majority of reporting organizations focused on the peer review process.⁸ Peer reviews are useful because they provide a lot of data, which makes applying quantitative high-maturity techniques a bit easier. However,

of CMMI best practices into an organization competes with time-to-market constraints and project deadlines. Moreover, the same resource in small organizations usually wears many hats, so the need to compromise billable hours versus process improvement tasks becomes even more difficult in small

We expect that it would be almost impossible to achieve more than ML 2 without an internal dedicated resource.

they focus on the number and types of defects, which is a very relevant issue to safety-critical domains but not as much for our context. Our initial focus on technical processes such as the compliance of our technical architectures to standards and the number of defects in production was the best starting point because we had data available across multiple projects. Thus, we realized that applying ML 5 practices to the peer review alone wouldn't fit our context because it wouldn't provide useful decision-making information to our project managers. Conversely, a lot of worthwhile data in tools for configuration management and issue tracking could be valuable for analysis and addressing key practices required at MLs 4 and 5. Thus, our approach was to make the best use of these tools.⁹

What Helped

Several factors went into our successful achievement of ML 5.

Internal Dedicated Resource

The effort and time required to establish and monitor the application

of CMMI best practices into an organization competes with time-to-market constraints and project deadlines. Moreover, the same resource in small organizations usually wears many hats, so the need to compromise billable hours versus process improvement tasks becomes even more difficult in small organizations due to limited resources and short iterations. The presence of an internal resource and project dedicated to meeting CMMI objectives helped us communicate effectively with the rest of the organization what to do, how, and when. Given our experience in applying CMMI-based process improvement activities in small organizations, we expect that it would be almost impossible to achieve more than ML 2 without an internal dedicated resource.

Stability of Key Leaders

Key leaders in the organization were convinced from the very beginning about the importance of using CMMI as a framework for our organizational practices. After the ML3 rating, the "Drive for 5" became our leadership motto and created a shared vision for everyone in the organization. Fortunately, key leaders didn't change during our journey, but we envision that there would have been many more difficulties if they had and weren't fully aligned with our process improvement goals and our path.

Culture That Fuels Change

The organization demonstrated several key characteristics that promote the identification, definition, and rollout of process improvements, so change occurred relatively easily. These characteristics include shared vision, openness, flexibility, and creativeness. The inherent openness and ease of communication among personnel permeated all levels of the business and allowed for quick identification of best practices. Leadership worked well to establish goals, build excitement, and celebrate milestones at each CMMI victory along the journey.

Rigorously Selecting the Lead Appraiser

CMMI requires one certified lead appraiser to check the maturity level of an organization using a defined appraisal method (SCAMPI Class A). In general, CMMI requires human interpretation and subjective opinions about how the CMMI practices are interpreted and customized given the organizational context. Thus, from an organization perspective, it's hard to estimate if the lead appraiser will be aligned with the organization's interpretation of the CMMI. The CMMI Institute allows the organization to choose its lead appraiser. During our journey, we developed and applied a rigorous approach to interviewing and selecting lead appraisers. Moreover, we leveraged a CMMI ML 3 process area, called Decision Analysis and Resolution (DAR), in which specific practices are applied to making and documenting important decisions more rigorously. Along the same lines, we applied DAR and documented our results when selecting our CMMI lead appraisers. Interestingly, this information was even useful during

the appraisal as evidence of application of the CMMI DAR practice.

External Expertise

In small organizations, the dedicated process improvement manager is often one person and sometimes even part-time. Obviously, it's unusual for one person to have all the expertise needed to complete a major change initiative such as the one required to achieve ML 5, which includes process improvement, change management, quantitative measurement techniques, statistical analysis, appraisals, and tool support. In our experience, we found it essential to partner with reliable external expertise that provides a mix of resources, complementing the ones available in our organization, and includes people who are well-aligned with our process improvement strategy and share the same ideals and values.

Small Iterations

CMMI relies on data analysis about process and product. To make sense and provide reliable indications, data analysis requires a certain number of observations or data points. Having iterations lasting around three months on average helped us build a knowledge base with sufficient data points in a reasonably short period of time. We envision that having projects with long iterations would have implied waiting much longer (to get enough data points) for the quantitative analysis.

What We Learned

We learned a lot of lessons during our journey.

You Can't Fake Your Way through the Statistical Analysis

A successful application of CMMI practices, especially the ones charac-

terizing ML 5, requires deep knowledge of both the CMMI model and statistical concepts such as p-value, confidence intervals, and so on. This knowledge needs to be available sooner rather than later in the journey in order to appropriately transition from external partners to participants in the organization.

You Need Workshops for Training, Evidence Collection, and Process Implementation

Learning about new or changed practices is essential. However, what really matters is demonstrated application of those practices in the project or organization, which, from a CMMI appraisal standpoint, is demonstrated through the creation of evidence. We've always believed in the value and importance of training all employees, but training sometimes conflicted with project deadlines. Using a workshop approach let us take our training to the next level so that our employees learned experientially, and at the same time, we could collect evidence that our CMMI practices were applied. Moreover, workshops are a useful mechanism for ensuring that new processes are implemented.

Process improvement and quality personnel supported these meetings and followed up with project managers and appropriate personnel to ensure that practices were effectively institutionalized.

Make Continuous Improvement a Priority

Improvement is a continuous and never-ending process. More importantly, no size fits all. Despite the existence of best practices, people change and contexts, projects, and attitudes differ. We clearly understood that our improvement approach had to be iterative: you can't know in advance what to measure to get a useful estimation. It's important to get feedback from all the stakeholders and check that collected data and applied practices make sense. People need to be convinced about what they're doing; they need to perceive a particular practice's value and be able to spot and report red flags so that a practice can always be improved upon.

Automation Is Essential

Whether collecting data or reporting results, it's imperative that automatic mechanisms exist to support

Using a workshop approach let us take our training to the next level so that our employees learned experientially.

Our workshops were usually scheduled during lunch time (to eliminate any tension with project deadlines) and consisted of a short review of practices and expectations and a discussion about examples of how to apply them on individual projects.

these efforts. We initially thought that our process improvement team could use worksheets to address our high-maturity measurement-related practice efforts. It quickly became obvious that it was faster, cheaper, easier, and much more reliable for

us to pull data from existing tools (issue-tracking and configuration management tools, in particular) and to visualize data analyses in an automated way. In fact, quickly creating different report views facilitated discussions at project and organizational measurement meetings and allowed improvement opportunities to effortlessly become visible for action.

Perform Quantitative Management in One Meeting

Initially, we had measurement events occurring haphazardly throughout the project life cycle. Over time, we realized it was important to collapse important measurement activities into single meetings throughout the cycle. These meetings include essential project personnel working with the process improvement measurement experts to set project goals or review progress against goals, tailor project processes given decisions based on our prediction model, and apply project lessons learned using organizational baseline performance and experiences.

It's a Long Journey

Customizing the CMMI to your business, applying customized practices, putting data collection in place,

improvement from adopting CMMI. During this period, years may pass, and motivation can plummet.

Now that we're ML 5, we're focusing our process improvement initiatives on enlarging the proportion of projects where data is beneficial. Everybody now sees the value in collecting data and how using it drives informed decisions and effectively allows us to monitor projects. Thus, we're trying to apply our CMMI-compliant best practices to all projects and all developers. We're also finding in this process that the data requires further stratification, given the types of projects that we're working on. For example, we've found that our projects built using COTS software solutions have different performance baselines than projects where we build the software from the ground up.

We're still striving to collect higher-quality metrics. We've already collected bug information and classified each bug according to its type. However, we feel that the classification schema is a little subjective, definitions are vague or overlapping, and different people are classifying the same bug in different categories (which cancels out all the

more than 90 percent of developers classify 10 defects each, for the sake of validation. We then use feedback from the validation to further refine the new classification schema. This activity is fairly expensive in terms of time and effort, but we strongly feel that it's been well spent since the bug information is crucial to understanding our processes and improving our predictive models. Having achieved a continuous process improvement mindset, we expect to demonstrate quantitative return on our defect re-classification investment.

Finally, although data analysis is very useful (see point above), it can be dangerous as it can drive you to the wrong conclusions if not analyzed with enough rigor and scientific basis.¹⁰ Removing or at least reducing possible errors in data interpretation is important to ensure that wrong decisions aren't made, or at least made with a low confidence. For instance, we've now significantly refactored our measurement tool to better predict the defects in production. In the past, the system predicted the ranges of defects (low: 0 to 2, medium: 3 to 5, high: 6 or more). Initially grouping the defects was valuable, especially as a starting point to achieve confidence and see the value of a defect prediction activity. However, we later realized that the concept of low, medium, and high is subjective and can vary among projects. Therefore, we modified the measurement scale of the predicted variable, from the ordinal scale (low, medium, or high) to the real scale (that is, the number of defects; see the sidebar for more details).

Our "Drive for 5" initiative was successful. The lessons we learned on this journey remain at the forefront and have contributed to allowing us to measurably examine (and

Everybody now sees the value in collecting data and how using it drives informed decisions.

building tools for analyzing (collected) data to support decisions, and making better decisions—all are steps in this journey. And they need to be completed to demonstrate any real

benefits of the action). Thus, we've started to invest time and effort in discussing possible improvements, defining a new bug classification schema, and validating it by having



THE KEYMIND MEASUREMENT REPORTING TOOL

To quantitatively manage projects, Keymind has developed and institutionalized the Measurement Reporting Tool (MRT), the primary purpose of which is to improve the quality of our measurement reporting and to increase the speed at which we can look at process performance both within projects and across the organization. MRT provides several capabilities: project monitor, JIRA quality, defect analysis, code peer review, collaborative unit testing, architecture compliance, code smells, defect prediction, and process control charts. MRT is a Web-based tool, and the user can choose the project to analyze and the specific analysis to apply by using the tab at the top of the screen (see Figure A).

PROJECT MONITOR

Figure A shows a screenshot of the tool to monitor some areas of quality concern for a project. Specifically, four different aspects are monitored: peer review, collaborative unit test development, architecture compliance, and object-oriented principles. The user can choose which of these analyses to see by expanding or collapsing a specific part of the screen. In Figure A, the user analyzed the “collaborative unit test development” rule, which consists of checking that after a specific number of source files changes, unit test case files change as well. The specific developer ID (zilu, ptran) is reported by the tool, and the trend of rule compliance is analyzed over a six-month time frame using palettes (green = satisfied, white = NA, red = unsatisfied).

DEFECT PREDICTION

Figure B shows a screenshot of the tool to estimate the number of production defects. The aim is to let the user find the product and project characteristics that, within a specific confidence level (for example, a minimum 95 percent), will likely result in an acceptable product in terms of number and type of production defects (such as a maximum number of defects with high

severity must be less than two). We continuously improved this feature, even after we’ve been positively rated, to make better (informed) decisions. In the current version, the user enters product and project characteristics (right side with sliders), such as percent of God Classes, number of implementation tasks and change requests, and number of development days. The tool’s output is the chart on the left side, where the y-axis reports the maximum number of expected production defects (colors identify different severity level), and the x-axis reports the confidence level.

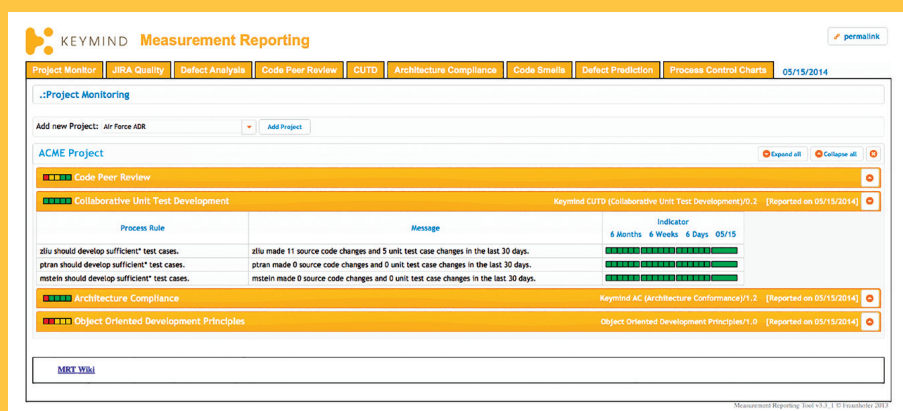


FIGURE A. A screenshot of the project monitor feature developed to support Keymind analyses of process performance data to determine its ability to meet identified business objectives.

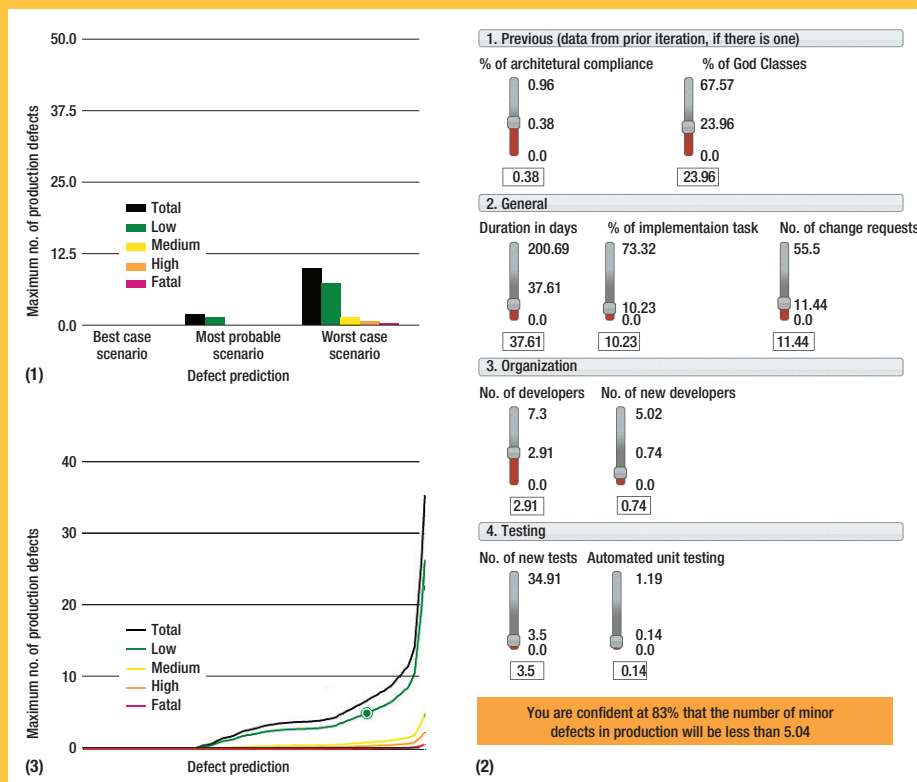


FIGURE B. A screenshot of the MRT Defect Prediction tab developed to estimate defects and comply with the ML4 Organizational Process Performance (OPP) process area.



DAVIDE FALESSI is the multimedia editor of *IEEE Software* and a research scientist in the Measurement and Knowledge Management Division at Fraunhofer Center for Experimental Software Engineering in Maryland. His main research interest is in devising and empirically assessing scalable solutions for the development of software-intensive systems with a particular emphasis on architecture, requirements, and quality. Falessi received a PhD in computer engineering from the University of Rome "Tor Vergata." Contact him at dfalessi@fc-md.umd.edu.



MICHELE A. SHAW is an applied technology lead for process improvement at the Fraunhofer Center for Experimental Software Engineering in Maryland. Her interests include transferring technological advances in project management, quality assurance, and other CMMI practices in small and medium-sized organizations. Shaw received an MSc in applied behavioral science from the Johns Hopkins University. Contact her at mshaw@fc-md.umd.edu.



KATHLEEN MULLEN is the process improvement manager at Keymind, a division of Luminpoint. She received an MSc in library and information science from the College of Information Studies, University of Maryland at College Park. Contact her at kmullen@axiom-rm.com.

improve) our performance at every level of our business. Even though the journey was long, and a significant investment was required, we have achieved huge returns in applying the CMMI model in a way that fits our business and supports decision-making by every stakeholder, including our employers and customers. Now we're committed to continue on this quantitative improvement path in an open-minded way and with an eye on CMMI. 🍷

Acknowledgments

We thank the several individuals who contributed to this ML 5 achievement including the executive steering committee (Kevin Riley, Doug Peardon, and Shane Oleson), Shannon Taylor, Mark Stein,

Nico Zazworka, Sally Ginsburg, Jessica Tischbierck, Marcel Schwarzmarm, Marcel Sinn, Alex Voegelé, and Johannes Vietze.

References

1. V.R. Basili and H.D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments," *IEEE Trans. Software Eng.*, vol. 14, no. 6, 1988, pp. 758–773.
2. M. Chrissis, M. Konrad, and S. Shrum, "CMMI for Development: Guidelines for Process Integration and Product Improvement," *SEI Series in Software Engineering*, 2011, p. 688.
3. K.C. Dangle et al., "Software Process Improvement in Small Organizations: A Case Study," *IEEE Software*, vol. 22, no. 6, 2005, pp. 68–75.
4. F.J. Pino, F. García, and M. Piattini, "Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review," *Software Quality J.*, vol. 16, no. 2, 2007, pp. 237–261.
5. S. Garcia-Miller, "Lessons Learned from Adopting CMMI in Small Organizations," *Software Eng. Inst.*, 2005; www.sei.cmu.edu/library/abstracts/presentations/Garcia-SEPG2005.cfm.
6. M. Staples and M. Niazi, "Systematic Review of Organizational Motivations for Adopting CMM-based SPI," *Information and Software Technology*, vol. 50, no. 7–8, 2008, pp. 605–620.
7. M. Staples et al., "An Exploratory Study of Why Organizations Do Not Adopt CMMI," *J. Systems and Software*, vol. 80, no. 6, 2007, pp. 883–895.
8. L. Harjumaa, I. Tervonen, and P. Vuorio, "Using Software Inspection as a Catalyst for SPI in a Small Company," *Product Focused Software Process Improvement*, LNCS, vol. 3009, 2004, pp. 62–75.
9. J. Schumacher et al., "Building Empirical Support for Automated Code Smell Detection," *Proc. 2010 ACM-IEEE Int'l Symp. Empirical Software Engineering and Measurement*, 2010, p. 1.
10. D. Falessi et al., "On Failure Classification: The Impact of 'Getting It Wrong,'" *Proc. 36th Int'l Conf. Software Eng. (ICSE)*, 2014, pp. 512–515.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

IEEE Software

FIND US ON
**FACEBOOK
& TWITTER!**

[facebook.com/
ieeesoftware](https://facebook.com/ieeesoftware)

[twitter.com/
ieeesoftware](https://twitter.com/ieeesoftware)