To keep their system and software development competitive, many companies are implementing orchestrated improvement programs for their engineering processes. Process improvement in today’s demanding business is crucial to improve competitiveness, expand markets, extend market penetration, better address customers’ real needs, manage various software assets, and ultimately, to be more profitable. If your processes don’t demand and receive continuous improvement, you won’t stay in business long.

This is where CMMI (Capability Maturity Model Integration) comes into the picture. If you don’t know where you are and where you need to go, change will never lead to tangible improvement. The Software Engineering Institute’s CMMI provides a framework for process improvement and capability assessments. It defines five maturity levels, together with a process improvement framework for your own improvement programs and to evaluate suppliers.

Industries and companies of all sizes worldwide use CMMI, so it provides a huge body of knowledge on topics along the entire product life cycle. Based on broad feedback of this worldwide user community, the SEI recently updated this content to CMMI v1.2. Originally based on the Software-CMM that appeared 20 years ago, it’s today also governed by the ISO to assure that suppliers are assessed comparably on a global basis.

New and widely improved


The book benefits from improvements to its style and presentation as well as improvements to the underlying model. The previous edition didn’t guide readers but left them in the dark as to translating the CMMI contents to their business needs.

This has changed. The model is now tuned to explaining where you need to go before describing how to achieve your goal. The authors embed business focus and practical guidance in each process area, and these process areas structure the model and the book.

The model is also less complex than the first edition. Many of the confusing elements (such as integrated supplier management and integrated process development) have been fully reworked. The SEI processed thousands of user change requests to make the model more useful. A new case study addresses using the CMMI for software and services. It’s hands-on and appealing to the many organizations whose business models combine software and service. Finally, the layout is improved, with the authors highlighting what’s most important and including many tips that illustrate how to do things better.

Having used both the new model and book for several months now, I see only a couple of places for further improvement. The book gives...
insufficient practical guidance for hardware, firmware, and systems development despite promising to deliver in these areas. And—not unimportant for the routine user—the paper quality is too thin to withstand daily use.

Why you should read it

Why should you read the book? Why should your organization embark on CMMI? Certainly, it’s all about competition. Companies have started to realize that change is critical: If you stand still, you fall behind! The business climate and the software marketplace have changed in favor of end users and customers. Companies must fight for new business, and customers expect process excellence. An industry-proven framework offers the benefit of improvement without getting lost in theory. CMMI offers concrete guidance in the form of structured goals, plus a wealth of best practices to implement. It presents maturity and capability levels to set up a goal-driven roadmap and a well-defined appraisal technique to assess processes and benchmarks. As a whole, the industry is becoming more mature. It’s all about business: Your competitors are at the same place as you (if not ahead). The goal is to further improve planning and decision making, lower costs, adhere to benchmarks. As a whole, the industry is becoming more mature. It’s all about business: Your competitors are at the same place as you (if not ahead). The goal is to further improve planning and decision making, lower costs, adhere to schedules, and improve product quality.

How to use it

CMMI: Guidelines for Process Integration and Product Improvement, Second Edition, is well-written and highly motivational. If you’re new to the field, chapter 1 will help you get started with an overview and useful examples. However, don’t try to read the book from beginning to end—it’s not made for that. The CMMI practitioner should use the book as a reference and benefit from its numerous tips on improving daily work.

Using the CMMI effectively means selecting from the model what matters to your business needs. CMMI training courses and the experience of CMMI professionals—be it engineering process group leaders or consultants—will help you to stay on track. Always remember that successful change needs not only a technical framework for guidance (that is, this book) but also concrete improvement objectives and, above all, leadership to drive the change in the direction you need.

Christof Ebert is managing director at Vector Consulting. He also serves on the editorial board of IEEE Software. Contact him at christof.ebert@vector-consulting.de.

No Borders, No Worries?

Anthony Akins


Let’s start with the positives. Software without Borders is very readable, with a lot of information about the challenges, benefits, history, and future of outsourcing software development. It has great references and several tools to help organizations prepare for outsourcing.

Throughout the book, Steve Mezak presents information, tools, questions, guidance, and recommendations on whether to outsource your software development. The book’s organization takes you from start to finish, explaining how to decide to outsource, select your vendor, describe your software for the most efficient and effective outsourcing, manage your outsourcing effort, and protect your intellectual property.

Software without Borders is strongly biased toward outsourcing, viewing it as something you can’t avoid. With that bias in mind, it aims to help make outsourcing your next project successful. Mezak sets the stage by presenting six development strategies you can employ. For all but the first, outsourcing is key:

- Using in-house development engineers—the classic approach to staffing your software development project.
- Onshore contract outsourcing—hiring a company in your own country to handle your software development.
- Offshore contract outsourcing—hiring a company outside of your country to handle your software development.
- An in-house and offshore blend— in-house engineers and offshore teams work on the same project, aiming for 24-hour round-the-clock development.
- Creating an offshore subsidiary—a good choice if your project requires 50 or more software engineers.
- Build, operate, and transfer—hiring an offshore company to provide engineers on contract. The offshore company builds your engineering team and operates it for you. After a year or two, you can transfer the engineers to your offshore subsidiary.

To help you choose the best development strategy, Mezak identifies 17 specific criteria. He introduces a simple outsourcing decision matrix early in the book, adding to it in each chapter as he presents more information. He divides the criteria into political, business, and technical criteria:

- Political: What will be the impact on employee morale if you outsource? Will your customers care if you outsource projects? How comfortable is your organization working with different cultures?
Business: How important are export restrictions on your product? Are there other reasons to consider going offshore? Is cost a major factor in the decision to outsource? How do you protect your intellectual property? How quickly does the project under question need to ramp up? Can the project tolerate schedule slippages? Is outsourcing a long-term approach for your company or just to meet a deadline? Do you see outsourcing as adding value to your company?

Technical: Do you need specific technical expertise? How well have you captured the project’s software requirements? How well-defined is your software development process or methodology? How many engineers do you need to develop your software? What kind of software development project are you considering outsourcing?

You can access the decision matrix and other tools and references at www.softwarewithoutborders.com. Before you decide to purchase this book, visit the site to look at the sample chapters, references, and downloadable tools.

Now, let’s consider the negatives. Ultimately, Software without Borders is a marketing tool for the author’s outsourcing services. Yeah, I know, all books can be considered marketing tools, but this book’s marketing aspect comes off strong. On one hand, I admire the honesty of the approach; no one’s fooling anyone here. If you want more help in successfully running your outsourcing effort, you’ll know who to contact shortly after you begin reading. By the time you finish the book, you’ll also have two discount coupons for jump-starting your outsourcing efforts or for beginning-to-end support and management of your next outsourced project. On the other hand, I wish the marketing pitch didn’t come off so strong. I feel like I just spent two hours at a used-car lot. When it was all over and done with, I might have got a good deal, but a strong sell always leaves me with a headache.

If you’re considering outsourcing, Software without Borders provides a great deal of information. More importantly, it points to information that will benefit any software project, not just outsourced projects. However, I recommend that it not be your only source of information.

Anthony Akins is a senior consultant at Valtech Technologies. Contact him at anthony.akins@valtech.com.