acmqueue

Major-league SEMAT— Why Should an Executive Care?

Becoming better, faster, cheaper, and happier

Ivar Jacobson, Pan-Wei Ng, Ian Spence, and Paul E. McMahon

In today's ever more competitive world, boards of directors and executives demand that CIOs and their teams deliver "more with less." Studies show, without any real surprise, that there is no one-size-fits-all method to suit all software initiatives, and that a practice-based approach with some light but effective degree of order and governance is the goal of most software-development departments.

SEMAT (Software Engineering Method and Theory) is a collaborative initiative whose mission is to provide software engineering with a foundation, based on a kernel of practice-independent elements, supported by a language that allows best practices to be described in as light or as detailed a way as needed. These can then be selected by teams for the context of their endeavor, thus ensuring the organization neither falls into a practice free-for-all nor constrains its business into a process straitjacket.

Executives have reason to care about SEMAT. The initiative supports the goals of "more with less" by delivering value at the team and organizational levels. Initiatives will always remain at a theoretical level unless and until proven through experimentation, and the case for SEMAT is strongly supported through real-life case studies.

CIO PRIORITIES

According to Gartner's "Top 10 CIO Business and Technology Priorities in 2013," CIO imperatives are split between business enablement and efficiency, reflecting the importance of the CIO's contribution in both these areas of the organization.[1] This is obviously a tough challenge—and the phrase "getting more with less" is often heard in conversations with IT executives and their leadership teams. What can be done to gain a real effect, and how can the software functions of major-league IT organizations react?

In support of this challenge, respondents to a survey conducted at the 2013 Gartner AADI (Application Architecture, Development, and Integration) Summit pointed out that trying to do more with less by attempting to standardize around a single development process was not the answer.[7] The results reflect that no "silver bullet" universal method has been found (no surprise there!) and that a software endeavor needs to use the most effective practices for its particular context. The respondents went on to suggest that at a foundational level, ensuring an ability to measure the effectiveness of endeavors end to end was a key requirement across all applications development.

The survey acknowledged that while the agile movement has helped accelerate development, respondents and their CIO leadership are still looking for some degree of rigor, structure, and governance in order to align better with business, behave more predictably, measure and address risk, and finally, deliver business value better, faster, and cheaper with happier stakeholders—the BFCH metrics.

Of course, "being agile" can be said to be a goal for any software endeavor, or any software organization (after all, who would not want to be agile?). It is not necessarily the case, however, that the same set of agile practices can be applied universally across all programs and teams. Indeed, in some endeavors, particular agile techniques or practices may not be possible. Take, for example, the case of a bank's portfolio of programs. At any time there is huge diversity across many dimensions: pure development projects require a different approach from maintenance projects; highly regulated software initiatives such as financial clearing may require more rigorous requirements documentation than new consumer-driven, graphical front-end development; and simple standalone apps may require a radically faster cadence than architectural changes such as the replacement of the enterprise application server. This, therefore, gives rise to the need for multiple practice alternatives within a single organization.

The conclusions of the AADI Summit survey—that a standard one-size-fits-all process should not be a 2013 priority and that some order and governance are necessary to prevent what could become a practice free-for-all if left unchecked—certainly make sense when considering the reality of a large organization's application-development landscape, as in the bank example.

The SEMAT initiative supports the findings of the AADI Summit survey, as well as the more general CIO goal of "more with less" through delivery of value at both the organizational and team levels.

INTRODUCING SEMAT

Since 2010, experts from industry and academia have collaborated under the SEMAT banner in an explicit attempt to "refound software engineering based on a solid theory, proven principles, and best practices." As a first step on this journey they have created Essence, a kernel and language for the improvement of software-development methods, which is soon to be ratified as a standard by OMG (Object Management Group).[5] It will be clear that the standard is an underpinning of the various practices available in the industry—agile or not.

The SEMAT initiative and the Essence kernel enable organizations to establish an environment where they can make use of the right practices for the right endeavors and right contexts. The practices are built on an independent, solid foundation (the kernel) that incorporates a lightweight yet appropriate level of order and measurement for the business. This approach represents a first of its kind in software engineering.

The Essence kernel includes "things we always have" and "things we always do" when developing software. It is universal to all software-development endeavors—agile or pre-agile. SEMAT has adopted several fundamentally new ideas. One of them is called the *alpha* concept, which allows teams to focus on the *doing* rather than on the *describing*. Teams can measure the progress and health of their endeavors in a practice- or method-independent way. At any moment the team can identify where it is now and where it is going next. It makes the team results-focused instead of document-driven or activity-centric. The alpha concept supports several other ideas—for example, the idea of a method being a composition of practices built on top of the universal kernel. Thanks to the alpha concept, SEMAT has been able to create a robust, extensible, intuitive kernel.

More complete descriptions of SEMAT are available in previous writings.[2,3]

THE VALUE OF SEMAT FOR LARGE ORGANIZATIONS

It is a fact that the software world has a huge number of methods—perhaps more than 100,000 have been developed, although many have not been used more than once. Some are known by brands or labels such as RUP (Rational Unified Process), Scrum, and XP (Extreme Programming), but most are homegrown with ideas picked up from textbooks or other resources, with zero opportunity for reuse outside of the single context for which they were built. The development of SEMAT has demonstrated that underneath all of these methods is a common ground, or a *kernel*, of "things we always have" and "things we always do" when developing software. These things form the essence of software engineering and are now included in Essence.[5]

As discussed earlier, large organizations have a diverse range of projects carried out by a diverse range of people. Without an accepted kernel to share, coordinating development endeavors is unnecessarily complicated, because teams must spend scarce time and attention on deciding how to work at the start of every project. This results in significant wasted effort, project delays, frustrated developers, and dissatisfied customers. All of these factors have the potential to compromise that all-important CIO priority of achieving "more with less." The inefficiencies occur both at the project/team (endeavor) level and the organizational level. To quantify this, consider the value it could bring to an organization if the development team did not need to come up with a specific new way of working, but instead could start from a common ground and then add existing well-proven (possibly agile or lean) ideas published in a library of practices.

The Essence kernel has benefits at both the team and organizational levels. To make this discussion something readers can identify with, let's introduce two characters: Smith, a project leader, and Dave, an executive in a large company that has been trying to improve the way it develops software. The company had invested in and achieved a CMMI (Capability, Maturity, Model Integration) rating, then adopted the Unified Process, and recently experimented with and, in some teams, successfully applied agile methods.

GRASS-ROOTS LEVEL—ENSURING THE TEAM'S SUCCESS

At the team level, the objective is to ensure the success of the endeavor, which Smith is responsible for. Success means delivering what his customers really want in the given time frame and budget, and with the right quality. Success also means that his team is motivated and collaborating effectively—fundamentally, a happy team. This requires four skills: using the right method for the right job; measuring project progress and health; effective collaboration; and decision-making based on a middle ground.

Using the right method for the job at hand. The battle-seasoned Smith understands (as did the Gartner AADI survey respondents) that there is no one-size-fits-all, future-proofed software-development method. As mentioned, Smith's organization had adopted CMMI, the Unified Process, and agile methods. Smith is not against any of these approaches and, in fact, thinks there is something good in each one of them. The problem is that, in the past, his organization at any point in time mandated just one way of running development.

Although Smith's colleague who runs small enhancements found Scrum to be useful, Smith found Scrum by itself to be inadequate for the large-scale enterprise development he is running. Smith was also mindful that his organization's previous CMMI- and Unified Process-based approaches were also inadequate. In each project, Smith still had to make significant adjustments to the mandated way of working to achieve a suitable approach. In effect, he was adding to the 100,000-methods problem by creating his own variant. Making and establishing such adjustments was no small feat, as Smith had to communicate them to his teammates, each of whom had different backgrounds and opinions.

Smith wanted a better approach for his teammates to come quickly to agreement on how they should work on each project.

The lightweight kernel and its support for systematically combining practices to create a variety of "just enough" methods, all based on the same common kernel, allows organizations to provide development teams with the right methods to undertake each job at hand, supporting each organization's specific culture, standards, and mandated practices.

Ability to measure project progress and health accurately. If Smith has learned anything, it is that software development is a complex endeavor, and the progress and health of a project have to be considered from multiple perspectives—for example, stakeholders' satisfaction; requirements (and requirements grow as the software system evolves); software system; and so on. The problem, however, is that Smith's company does not have governance procedures that holistically help each team evaluate progress and health. In fact, no commonly accepted framework has previously been able to do this.

Each alpha (i.e., opportunity, stakeholders, requirements, software system, team, work, and way of working) covers a dimension, and the states defined for each alpha provide an indication of progress and health. For example, Essence identifies states of the software system as being architecture selected, demonstrable, usable, ready, operational, and retired. Each of these states provides an indication of how far the development of the software system has come, and what a team should do next to move the progress and health of the software system forward. Modern approaches do exist that recommend a similar way of thinking but only in a single dimension (e.g., Kanban focuses on the work alpha). Essence, on the other hand, is more comprehensive and holistic, since it is multidimensional (i.e., it focuses on all the alphas), and thus gives a more accurate picture of progress and health. What's more, Essence does this in a lightweight manner, unlike previous attempts such as within CMMI and the Unified Process.

Ability to get team members to collaborate effectively. Each time Smith begins a project, he has to assemble a team with members from different parts of the organization, and possibly even from different contractor organizations. The team members need to agree on how they should collaborate. For example, they need to agree on which practices to use. Because they have different backgrounds and experiences, this often turns out to be nontrivial. Preparing team members can be as complex as a reengineering effort. As evidence, in the literature published over the past 20 years on the causes of failed projects, poor communication among team members frequently ranks very high.[6]

Individuals from different organizations have a common base from which to start. This helps the team communicate more effectively and therefore become productive more rapidly, even when their native or favorite ways of working are significantly different.

Decision-making based on a middle ground. The basis for decision-making should be quick health checks rather than inflexible processes—or nothing at all. An integral part of Smith's job is deciding with his teammates where to place their efforts to achieve progress and deal with project risks. Some requirements might be vague and need further exploration with customer representatives; some technical issues might need investigation; some defects might need attention. Previously, however, Smith had to work within prescribed governance procedures that required the production of specific documents at specific milestones within the project and specific activities to be conducted in a specific order. This was a heavyweight, ineffective approach that did not help Smith and his teammates make relevant decisions. With Essence, however, the focus is on the results generated and dealing with risks within the project's context.

Essence provides a framework that enables faster, more effective, more autonomous decision-making and can help organizations evolve their current governance practices to meet the needs of present-day challenges. One example of how Essence accomplishes this is through the alphas, alpha states, and checklists that keep the team focused on what is most important at any point in the project, based on the current project context. That provides an independent check of how well the team is doing at achieving the intended result. This check can be applied regardless of the method and practices the team is using. Today we understand far better the importance of each project's context when it comes to managing cost and performance effectively. This is where Essence can be a great differentiator from previous approaches, because it helps ask the right question at the right time based on the current state and the target goal state.

To summarize, the use of the SEMAT kernel provides many benefits at the team level. Visualize a library of interchangeable practices, written using a language that can be as light or as formal as the situation requires, and built around the immutable fact that all software endeavors are successful based on the progression of certain key elements. The team can select from that library based on the needs of the endeavor and *compose* their way of working rather than *create* it. They can build the optimum way for their situation. They can measure the progress and health of their project using standard metrics that are independent of those practices and methods—again invention and distraction that the team can avoid. New members of the team need not learn everything from scratch. All of these factors contribute positively to improving time-to-effectiveness for the team. Reducing such wasted time and energy, in addition to helping the team achieve its goals, clearly has cumulative benefits for the organization at large, thus helping attain the CIO's goal of "more with less."

ORGANIZATIONAL LEVEL—GROWTH AND COMPETITIVENESS MEAN "MORE WITH LESS"

As discussed at the beginning of this article, at the organizational level the objective is business growth. In our scenario this is what Dave is responsible for. He knows that his business will grow only if his customers are happy and keep coming back, providing more business opportunities for the organization. He knows that speed to market and relevance are improved through having a highly motivated and driven team. He also knows he needs to run a tight ship. Therefore, Dave wants to introduce mechanisms to nurture a continuously learning and inspiring culture. Not only does Dave want his teams to learn from one another, he also wants them to learn from others in the industry. This requires some changes.

A common vocabulary and body of knowledge. One would have thought that Dave's team would share a vocabulary because they work for the same company, but this is not the case. Many teams within his organization use different terminology, and often the same word can carry different meanings. In order for one team to learn successful practices from another team, it has to cross this unnecessary language barrier.

Essence tears down this language barrier by enabling practices to be described using a single vocabulary—the kernel. In addition, Essence provides a structure to organize practices such that they can be discovered, assessed, and applied quickly.

Teams can also contribute what they have learned back to these practices, thus building a structured library of knowledge—namely, practices. This also provides a learning roadmap for individuals in the organization. The more practices from the library they can master, the more competent they become. The benefits of a common practice library reach beyond single projects. For example, it can substantially help with the communications problems documented on many failed IT projects.[6]

Empowering teams to change their way of working safely. This must be done not just in minor steps, but also in major ways. Dave wants to encourage his teams to be innovative, but at the same time he worries that they may venture into chaos. As such, his approach has tended to err toward the conservative, and the flexibility given to team leaders like Smith was rather limited. With Essence and the layering of practices on top of the kernel, however, organizations have a proven framework from which they can safely change their way of working. This allows Dave to focus on mandating a minimum number of specific practices—those that he needs the teams to practicein a consistent manner. At the same time he can also empower the teams to innovate in the other areas of the method, adding and replacing practices as and when it is appropriate.

Teams get a controlled and intuitive way to replace an existing practice or add a new one. This empowers teams to right-size their methods safely and in major ways, if appropriate.

Continual performance improvement/true learning organization. As mentioned previously, Dave's organization had adopted CMMI, the Unified Process, and, lately, agile practices, each providing a different emphasis. When embracing a new approach, however, his organization unfortunately threw away many of the good practices they had learned in the past, along with the parts they wanted to free themselves from. This was a great loss for the organization. In fact, it led to significant and costly reinvention. For example, when they introduced agile practices they had to reinvent the way architectural work was accomplished. By adopting the Essence kernel and working with practices rather than monolithic methods, Dave's organization became a true learning organization—continually growing and improving the set of practices they use and improving their development capability.

Teams can easily understand what is working and where they need to tune their practices—dropping or refining the unsuccessful ones and adding appropriate new ones to address the changing needs of their development organization.

The result is a software-development department that is not only allowed but encouraged to explore the possibilities presented by new techniques and practices in the industry. Successful practices (the good ideas) can be slotted into the lightweight Essence framework and incorporated quickly into the organization's way of working. Such freedom is highly motivating in a knowledge-worker environment—and motivation is key to attracting and retaining the best employees in the industry, and therefore to heightened effectiveness in the software organization.

Organizations benefit from SEMAT, too, and not simply because of the roll-up benefits from teams. The practice-based approach to ways of working, or methods, means that the organization invests at this level and has a greater chance of reuse, rather than creating bespoke descriptions of project-specific methods. Resource groups master practices rather than methods, arming them with the ability to move more easily among teams and giving the CIO greater flexibility in the workforce. This more efficient management of practices, combined with higher adaptability and effectiveness of people, supports the "more with less" requirement imposed by the business.

SEMAT SUCCESS STORIES

Without doubt, Essence is a powerful tool, but it has to be used appropriately. The SEMAT approach has helped a number of large-scale organizations and development endeavors for many years, both in general and in specific scenarios.[4]

OFFSHORE DEVELOPMENT

The ideas behind Essence have been applied in a major initiative involving a large, well-known Japanese consumer electronics company that had outsourced development to a vendor in China. This client had never outsourced, nor did it understand the impending risks and see the value of iterative development in a use case and test-driven manner. The Chinese outsourcer was accustomed to following methods that were dictated by its customers. These methods tended to be traditional waterfall approaches. Thus, getting started with good communication and a clear vision was a huge challenge. The solution was to start off using the Essence kernel alphas to agree on how teams should be organized and composed. The lightweight, intuitive nature of the alpha-state cards helped both the client and the vendor team leaders visualize the way they could work together.

The endeavor started with eight client staff and gradually grew to 10 client staff/20 vendor staff, then to 30 client staff/50 vendor staff. At the end of the endeavor, there were 80 client staff and 200 vendor staff involved.

When members joined the development team, they were given a quick briefing on the kernel and the new practices (iterative development, use-case-driven development, and test-driven development). Project managers monitored the state of development and the size of teams. When a team became too big, it was split, and the new team leader was trained in the kernel to understand the forces acting on the team. Clearly, the involvement of each team and individual was not static, even within this one major endeavor. Not only was Essence able to help the teams get started, it also helped them grow.

COLLABORATIVE GLOBAL SOFTWARE DEVELOPMENT

Building on the foundation provided by the Essence kernel, a major global reinsurance company has established a family of collaboration models that cover the whole spectrum of software- and application-development work. Four collaboration models have been built on the same kernel from the same set of 12 practices. The models are (1) exploratory, (2) standard, (3) maintenance, and (4) support. Each defines a lightweight governance process suitable for the kind of work being undertaken and provides enough practices for the teams to get started.

This has many organizational, as well as local, benefits, including:

- **The unification of the people working in the service organization.** The application-development group is organized into a number of independent services. Teams are formed by drawing on the members of the various services as they are needed. The use of the kernel provides the common ground that allows these teams to come together quickly and ensure that they all speak the same language.
- **The enabling of flexible sourcing policies.** Adopting a kernel-based approach also allows the various services to flex their capacity safely by drawing on external resources and suppliers. The use of the kernel allows everyone to get up to speed quickly with the way-of-working and easily understand his or her roles and responsibilities.
- **Increased agility and improved project performance.** By focusing on the practices that join up the services and the teamwork needed to develop good software, the organization has been able to

empower the teams and the services that they draw on to continuously inspect and adapt, and safely embrace agile and lean thinking in their day-to-day work.

- **Global collaboration.** The company's application-development teams are located all over the world. By providing a standard, lightweight way of working, greater global collaboration and in some cases global-sourcing models have been achieved.

STANDARD APPLICATION LIFE-CYCLE MANAGEMENT TOOLING

A number of companies have used the Essence approach to help establish practice-independent tooling and application life-cycle management frameworks, including:

- A major systems integrator assembled an open source, low-cost tooling environment to support distributed teams working around the Essence kernel, as well as its support for practice composition and execution. The environment runs as an organizational service and has been successfully used on a variety of projects. The flexibility inherent in Essence has allowed the same tooling and practices to be used in support of both agile and waterfall ways of working.
- A major UK government department introduced a kernel-based agile tool set to enable disciplined agility and the tracking of project progress and health in a practice-independent fashion. This allowed teams to adopt the agile practices of their choice within a consistent, effective, lightweight governance framework. It also helped the teams stay on track and avoid some of the pitfalls that can occur when transitioning to an agile way of working.

In both cases the common ground provided by the kernel and the results focus of the alphas enabled the widespread adoption of the tooling without compromising the teams' autonomy and empowerment.

BECOMING BETTER, FASTER, CHEAPER, AND HAPPIER

The effectiveness of a method should be measured in how much better, faster, cheaper software you develop and how much happier your customers and employees are. Better methods results in improvements in all these factors. Better software means increased competitiveness for your products. Faster is critical to getting products on the market. Cheaper often comes as a side effect of faster, but also from automation. And happier customers come from many sources, one being the creation of better user experiences.

Better, faster, cheaper, and happier are all elements of the CIO's priorities. SEMAT provides the foundation that allows a large organization's application-development department to improve all these elements, resulting in a more effective team of motivated professionals truly contributing to the competitiveness of the organization.

Organizations do not need to wait to benefit from the Essence kernel. Teams and departments can start benefiting today. The primary value that Essence brings is preventing the costly reinvention and unnecessary relearning of what is already known.

Essence can help your organization get to where you know you need to go, and it will help you get there faster and cheaper, so you are ready for whatever the future brings.

LOVE IT, HATE IT? LET US KNOW

feedback@queue.acm.org

REFERENCES

1. Gartner Inc. 2013. Top 10 CIO business and technology priorities in 2013; www.gartnerinfo.com/sym23/evtm_219_CIOtop10%5B3%5D.pdf.
2. Jacobson, I., Ng, P.-W., McMahon, P., Spence, I., Lidman, S. 2012. The essence of software engineering: the SEMAT kernel. *ACM Queue* 10(10); http://queue.acm.org/detail.cfm?id=2389616.
3. Jacobson, I., Ng, P.-W., McMahon, P. E., Spence, I., Lidman, S. 2013. *The Essence of Software Engineering: Applying the SEMAT Kernel*. Addison-Wesley.
4. Jacobson, I., Ng, P.-W., Spence, I. 2006. The essential unified process. *Dr. Dobb's Journal* (August).
5. Object Management Group. 2013. Essence: the kernel and language for software engineering methods; http://semat.org/wp-content/uploads/2013/02/Essence_final_submission_18Feb13.pdf.
6. Rosencrance, L. 2007. Survey: poor communication causes most IT project failures; *Computerworld*; http://www.computerworld.com/s/article/9012758/Survey_Poor_communication_causes_most_IT_project_failures.
7. Serena Software Inc. 2013. 2013 IT priorities survey: app dev gets down to business. Conducted at the Gartner Application Architecture, Development and Integration (AADI) Summit; http://www.serena.com/index.php/en/solutions/app-dev-delivery/infographic-application-development-priorities-2013/.

DR. IVAR JACOBSON, the chairman of Ivar Jacobson International, is a father of components and component architecture, use cases, the Unified Modeling Language, and the Rational Unified Process. He has contributed to modern business modeling and aspect-oriented software development. He is an international honorary advisor at Peking University, Beijing, and holds an honorary doctorate from San Martin de Porres University, Peru.

DR. PAN-WEI NG coaches large-scale systems development involving many millions of lines of code and hundreds of people per release, helping them transition to a lean and agile way of working, not forgetting to improve their code and architecture and to test through use cases and aspects. He is the coauthor, with Ivar Jacobson, of *Aspect-oriented Software Development with Use Cases*. He believes in making things tangible and practical and has been an active contributor to ideas behind the kernel, including the use of state cards.

IAN SPENCE is CTO at Ivar Jacobson International and the team leader for the development of the SEMAT kernel. An experienced coach, he has introduced hundreds of projects to iterative and agile practices. He has also led numerous successful large-scale transformation projects working with development organizations of up to 5,000 people. His current interests are agile for large projects, agile outsourcing, and driving sustainable change with agile measurements.

PAUL MCMAHON (pemcmahon@acm.org) is an independent consultant focusing on coaching project managers, team leaders, and software professionals in the practical use of lean and agile techniques in constrained environments. He is a Certified ScrumMaster and a Certified Lean Six Sigma Black Belt. He has been a leader in the SEMAT initiative since its inception.