(Excerpt pages 37-41, 3rd Ed. *CMMI for Development: Guidelines for Process Integration and Product Improvement* by Mary Beth Chrissis, Mike Konrad and Sandy Shrum, ISBN 0321711505, Copyright 2011 Pearson Education, Inc.)

Applying Principles of Empiricism

by Victor R. Basili, Kathleen C. Dangle, and Michele A. Shaw

Thinking Empirically

Thinking empirically about software engineering changes the way you think about process improvement.

Software engineering is an engineering discipline. Like other disciplines, software engineering requires an empirical paradigm that involves observing, building models, analyzing, and experimenting so that we can learn. We need to model the products, the processes, and the cause/effect relationships between them in the context of the organization and the project set. This empirical mindset provides a basis for choosing the appropriate processes, analyzing the effects of those selections, and packaging the resulting knowledge for reuse and evolution; it drives an effective process improvement initiative.

Note: Systems and software engineering have a lot in common; both require human-intensive implementation approaches and fundamentally focus on the issue of design, unlike manufacturing. As such, this empirical thinking can be applied to systems engineering as an underlying approach to improving systems engineering outcomes.

Empirical Principles

There are several principles associated with software engineering. In what follows, we discuss a few of these principles as they relate to process improvement.

P1. Observe your business. Organizations have different characteristics, goals, and cultures; stakeholders have different and competing needs. Well-engineered systems and software depend on many variables and context plays a significant role in defining goals and objectives for what can be and what must be achieved.

Organizations must strive to build quantitative and qualitative models to understand the cause and effect relationships between processes and products in the context of the development/maintenance efforts.

How else can these organizations articulate the differences and similarities among projects in the organization so they have a basis for selecting the processes to use to achieve their goals?

P2. Measurement is fundamental. Measurement is a standard abstraction process that allows us to build models or representations of what we observe so we can reason about relationships in context. The use of models in conjunction with experience, judgment, and intuition can guide decision-making. Measurement through models provides a mechanism for an evidence-based investigation so that decisions are supported with facts versus a system of pure beliefs.

P3. Process is a variable. Processes need to be selected and tailored to solve the problem at hand. In order to find the right process for the right situation, organizations must understand the effects of the process under differing conditions. This means a process must be measurable so that its effects can be quantified. It also means organizations must compile evidence that shows what works under what circumstances.

P4. Stakeholders must make their goals explicit. There is a wide range of stakeholders for any project (e.g., customers, end users, contract managers, practitioners, and managers). The organization itself and different stakeholders have different goals and needs. Organizations must make these goals and needs explicit through models and measures so they can be communicated, analyzed, synthesized, evaluated, and used to select and tailor the right processes. Making them explicit allows them to be packaged so they can be remembered and used again.

P5. Learn from experience. Organizations have the opportunity to learn from their experiences and build their core competence in systems and software engineering. For process improvement, the focus should be on learning about processes and how they interact with the environment on each project. This learning is evolutionary and each project should make the organization smarter about how to do the next project. But this learning must be deliberate and explicit or it will not be available for the organization to leverage.

P6. Software Engineering is "big science." Improving software engineering processes must be done through observation and experimentation in the context of where the actual products are being

developed. There is a synergistic relationship between practice and research. Industrial, government, and academic organizations must partner to expand and evolve systems and software competencies. There are so many facets to software engineering that it requires multiple talents and differing expertise. We need real-world laboratories that allow us to see the interactions among teams, processes, and products.

The Role of Empiricism in CMMI

At CMMI level 5, process improvement is intended to be an empirically-based activity. Each project is planned and executed using practices that are selected based on the context of the environment, the project needs, and past experiences. A level 5 organization understands the relationship between process and product and is capable of manipulating process to achieve various product characteristics. It is this capability that provides the greatest value from process improvement to the organization.

Empirical thinking shifts the process-improvement mindset from "putting processes in place" to "understanding the effects of processes so that appropriate processes can be adopted." Different and better decisions are made regarding how we choose improvement initiatives (prioritize), how we implement practices, and how we manage efforts in projects and in organizations when our explicit approach is based on empirical principles. That mindset should be in place at the beginning of the process improvement initiative, thereby focusing the effort on the real objectives of the project or the organization, the specific product and process problems, relevant experience with methods, and so on.

Organizations that are effective at implementing process improvement understand and apply empirical principles. Additionally, practices that support these principles are evident within CMMI. CMMI prescribes that data be used to make decisions about process definition at the project level as well as process change at the organizational level. Measurement and learning are catalysts for all of the practices; that is, they provide the bases for why specific practices are selected and how processes are implemented. Systems and software engineering practices are refined and fine-tuned as their effects are better understood.

Some Empirical Techniques That Support CMMI

Many techniques and methods exist to assist organizations and projects realize their goals by implementing CMMI practices. Below

are some techniques that the Fraunhofer Center has effectively implemented that you may wish to consider:

Goal/Question/Metric (GQM) Approach—An essential technique for measurement in any context, GQM can support realization of P2 and P4. Originally, GQM was defined for NASA Goddard Space Flight Center to evaluate software defects on a set of projects.[2]

Quality Improvement Paradigm (QIP)—A phased process for organizational improvement that integrates the experience of individual projects with the corporate learning process, and can help realize Empirical Principles P1, P3, and P5. QIP includes characterizing the organization through models, setting goals, choosing appropriate processes for implementation on projects, analyzing results, and packaging the experience for future use in the organization.[3]

Experience Factory (EF)—A concept based on the continual accumulation of project experiences that are essential to organizational and project improvement, EF highlights the logical separation between the project organization and the factory that processes the project experiences in an organization experience base to make it reusable. EF can help realize Empirical Principles P1, P2, and P5.[4]

GQM+Strategies—An approach that supports strategic measurement by extending GQM to support goal definition and alignment, strategy development, measurement implementation, and assessment across an enterprise.[5] GQM+Strategies assists the organization in tying together its approach and motivations underlying the implementation of the Empirical Principles.

Adopting these empirical principles early, understanding their role in CMMI, and deliberately selecting techniques to support your effort can have a profound effect on the success of the organization as you embark the process improvement path.

2. V. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric Approach," *Encyclopedia of Software Engineering*, pp. 528–532, John Wiley & Sons, Inc., 1994.

3. V. Basili and G. Caldiera, "Improve Software Quality by Reusing Knowledge and Experience," *Sloan Management Review*, MIT Press, vol. 37(1): 55–64, Fall 1995.

4. V. Basili, G. Caldiera, and H. D. Rombach, "The Experience Factory," *Encyclopedia of Software Engineering*, pp. 469–476, John Wiley & Sons, Inc., 1994.

5. Victor R. Basili, Mikael Lindvall, Myrna Regardie, Carolyn Seaman, Jens Heidrich, Jurgen Munch, Dieter Rombach, and Adam Trendowicz, “Linking Software Development and Business Strategy Through Measurement,” *IEEE Computer*, pp. 57–65, April, 2010.