The CeBASE Framework for Strategic Software Development and Evolution

Barry Boehm, USC and Victor Basili, UMD
Center for Empirically-Based Software Engineering (CeBASE)
EDSER-3 Position Paper

1. Introduction

One of the challenges highlighted in the EDSER-3 Call for Papers is for a symmetric approach to cost, risk, benefit, and opportunity modeling and management. This position paper offers the CeBASE Method as a response to this challenge. It is the result of an effort by the CeBASE principals at USC and U. of Maryland to reconcile their various models of software phenomenology into a common framework for pursuing empirical software engineering research and for organizing the results into a useful experience base.

The USC system of models is encompassed within the Model-Based (System) Architecting and Software Engineering (MBASE) method [5]. It includes compatible adaptations of the stakeholder win-win model [6], the DMR Benefits Realization Approach [Thorp, 1998], elements of the Rational Unified Process [Royce, 1998; Kruchten, 2000], the COCOMO suite of estimation models, and the Spiral Model with its associated anchor point milestones and risk management models. The Maryland system of models is encompassed within the Experience Factory method [2], and includes the Quality Improvement Paradigm and the Goal-Question-Metric method [3; van Solingen-Berghout, 1999].

The resulting unified CeBASE Method shown in Figure 1 has the following characteristics that respond to the EDSER-3 challenge:

- Complementary and symmetric closed-loop feedback processes at the organization/portfolio level and at the project level. These focus on establishing, monitoring, and adjusting stakeholder shared-vision and value propositions as these encounter progress/plan/goal mismatches, which may be shortfalls, opportunities, or risks.
- Common and mutually supportive elements at the organization/portfolio level and at the project level. The first element establishes the success-critical stakeholders, their shared vision, and value propositions. The second establishes the people, process, and product plans. The third monitors progress and the environment with respect to the shared vision elements and plans, and applies corrective action (including shared vision, plans, and experience base updates) where appropriate.

2. The CeBASE Method's Integrating Framework

As we explored the details of Maryland's Experience Factory and Goal-Question-Metric approaches and USC's MBASE approach, we found that they were expressing very similar principles and practices. The Spiral Model's initial focus on system objectives was essentially the same as GQM's initial focus on organizational goals, and MBASE's focus on tracing system goals back to organizational goals. The Experience Factory's focus on organizational learning to understand a system's operational stakeholders and their goals corresponds strongly with MBASE's stakeholder win-win approach to mutual stakeholder understanding and development of a shared system vision.

As seen in Figure 1, the CeBASE method is organized around the three strategic themes of the stakeholders' shared vision for the organization or project; thorough plans for process, product and people; and continuous monitoring and control. These themes express both the operation of EF-GMQM at the organizational level and the operation of MBASE-GMQM at the project level. Figure 1 also shows how progress/plan/goal mismatches feed continuous learning and improvement at both the organizational and project levels; and how learning at each level is used to condition learning at the level above or below.

3. The CeBASE Method and the Integrated Capability Maturity Model (CMMI)

We did a validity check of the integrated CEBASE method by comparing it with the contents of the 24 process areas in the CMMI [SEI, 2000; 1]. We found not only a strong correspondence, but also an almost complete coverage or the CMMI's practices by the

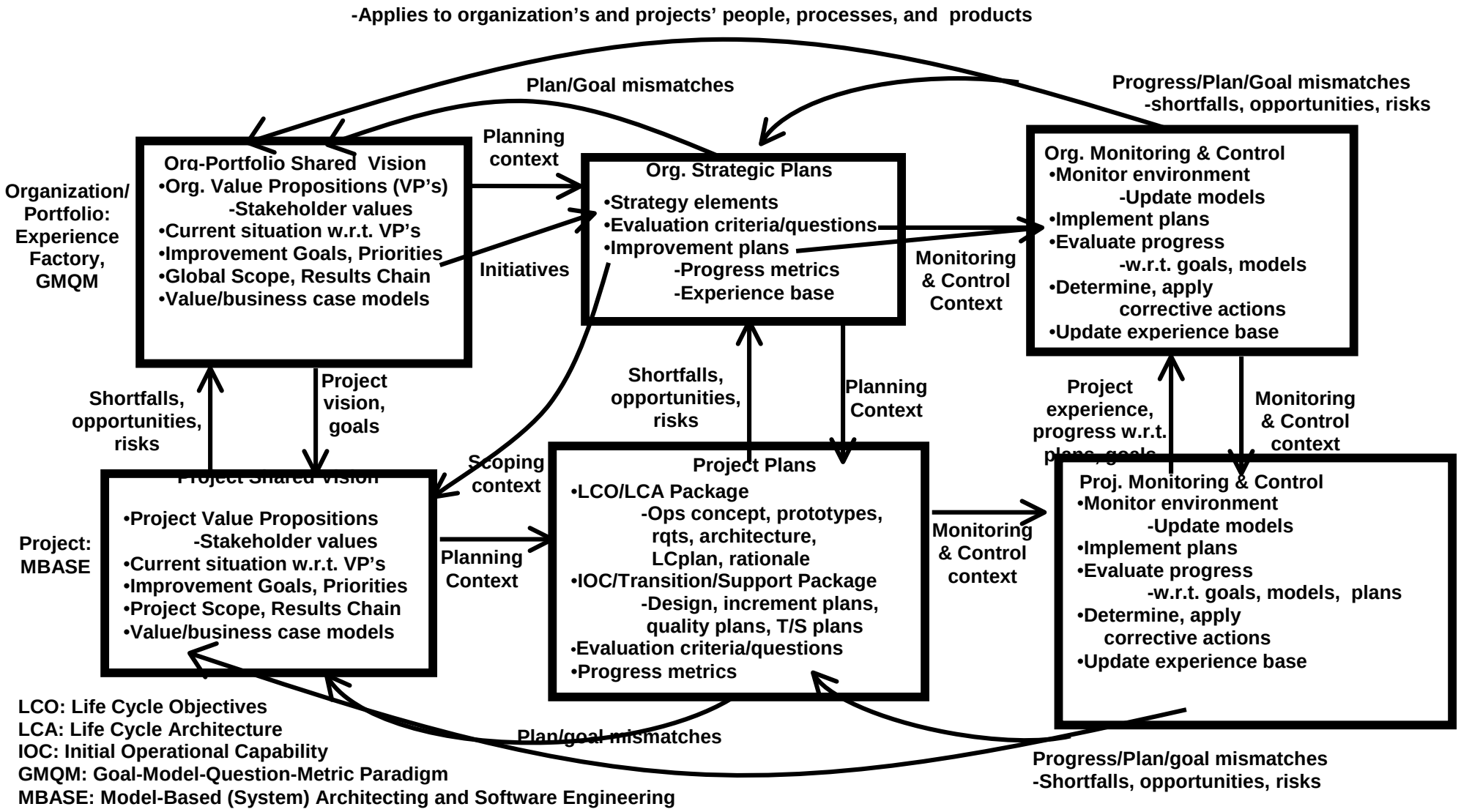

Figure 1. The CeBASE Method

organizational and project components of the CeBASE method. A summary of the percentage coverage of the CMMI process areas by the CeBASE method is shown in Table 1. The "+" annotations in Table 1 indicate that the CeBASE method's coverage goes considerably beyond that of the CMMI. For example, it covers not just an organizational process focus but also an organizational product and people focus. The "-" annotations in Table 1 indicate that some areas in the CeBASE method still remain to be fleshed out, such as detailed guidelines for organizational training plans, although they are covered in principle.

The CeBASE Method also provides a prescriptive approach for an organization to use in tailoring the CMMI's generic practices to its particular culture, environment, and value propositions. Thus, an e-commerce organization's value propositions (rapid time to market, rapid adaptation to change) will cause it to adopt more flexible processes, but such elements as the Anchor Point milestones will balance this flexibility with sufficient discipline to keep the overall process under control. And the value propositions of an organization developing safety-critical products or services will cause it to emphasize more rigorous specifications, processes, and practices, but in ways that enable it to cope with rapid change (e.g., by capturing evolution requirements and architecting systems to accommodate future change; by building in buffer periods to synchronize and stabilize processes [Cusumano-Selby, 1996]; or to adapt to potential schedule or budget slips by dropping lower-priority product features.) The counterpart [Port-Boehm, 2001] position paper on "Risk-Based Strategic Software Design" shows how risk considerations can be used to tailor the level of COTS evaluation involved in system architecting to the project or organization's levels of risk in evaluation throughness and delivery schedule.

Table 1. CeBASE Method Coverage of CMMI

- Process Management
  - Organizational Process Focus: 100+
  - Organizational Process Definition: 100+
  - Organizational Training: 100-
  - Organizational Process Performance: 100-
  - Organizational Innovation and Deployment: 100+
- Project Management
  - Project Planning: 100
  - Project Monitoring and Control: 100+
  - Supplier Agreement Management: 50-
  - Integrated Project Management: 100-
  - Risk Management: 100
  - Integrated Teaming: 100
  - Quantitative Project Management: 70-
- Engineering
  - Requirements Management: 100
  - Requirements Development: 100
  - Technical Solution: 60+
  - Product Integration: 70+
  - Verification: 70-
  - Validation: 80+
- Support
  - Configuration Management: 70-
  - Process and Product Quality Assurance: 70-
  - Measurement and Analysis: 100-
  - Decision Analysis and Resolution: 100-
  - Organizational Environment for Integration: 80-
  - Causal Analysis and Resolution: 100

4. Example: The Schedule-As-Independent Variable (SAIV) Process

To illustrate the points about delivery schedule, the CeBASE Method enables a project with a tight schedule constraint to use a Schedule-As-Independent-Variable (SAIV) process. This process uses the steps shown below to ensure that a project can deliver at least a stakeholder-satisfactory core capability within the fixed schedule, and as much additional next-priority capabilities as time permits [4].

1. Shared vision and expectations management
2. Feature prioritisation
3. Schedule range estimation
4. Architecture and core capabilities determination
5. Incremental development
6. Change and progress monitoring and control

The counterpart [Huang, 2001] case study illustrates the application of the SAIV process model to the development of a Web-based library information service with an inflexible delivery date.

5. Conclusions

The CeBASE Method appears to provide a supportive framework for value-based strategic software development and evolution at the organizational, portfolio, and project levels. We look forward to obtaining feedback on its appropriateness and coverage at EDSER-3.

References

[1] D. Ahern, A. Clouse, and R. Turner, CMMI Distilled, Addison Wesley, 2001.

[2] V. Basili, G. Caldeira, and H. Rombach, "The Experience Factory," in J. Marciniak (ed.), Encyclopedia of Software Engineering, Wiley, 1994.

[3] V. Basili, G. Caldeira, and H. Rombach, "The Goal Question Metric Approach," in J. Marciniak (ed.), Encyclopedia of Software Engineering, Wiley, 1994.

[4] B. Boehm and A.W. Brown, "Mastering Rapid Delivery and Change with the SAIV Process Model," Proceedings, ESCOM 2001, April 2001.

[5] B. Boehm, D. Port, "Escaping the Software Tar Pit: Model Clashes and How to Avoid Them," ACM Software Engineering Notes, January 1999, pp. 36-48.

[6] B. Boehm and R. Ross, "Theory W Software Project Management: Principles and Examples," IEEE-TSE, July 1989, pp. 902-916.

[7] M. Cusumano and R. Selby, Microsoft Secrets, The Free Press, 1995.

[Huang, 2001]. L. Huang, "Strategic Design Issues Encountered with the SAIV Process Model: A Case Study," EDSER-3 Workshop, May 2001.

[Kruchten, 2000]. P. Kruchten, The Rational Unified Process (2nd ed.), Addison-Wesley, 2000.

[Port-Boehm, 2001]. D. Port and B. Boehm, "Risk-Based Strategic Software Design: How Much COTS Evaluation is Enough?," EDSER-3 Workshop, May 2001.

[Royce, 1998]. W.E. Royce, Software Project Management: A Unified Framework, Addison-Wesley, 1998.

[Thorp, 1998]. J. Thorp and DMR, The Information Paradox, McGraw Hill, 1998.

[van Solingen-Berghout, 1999]. R. Van Solingen and E. Berghout, The Goal/Question/Metric Method, McGraw Hill, 1999.

[SEI, 2000]. SEI, CMMI web site: www.sei.cmu.edu