

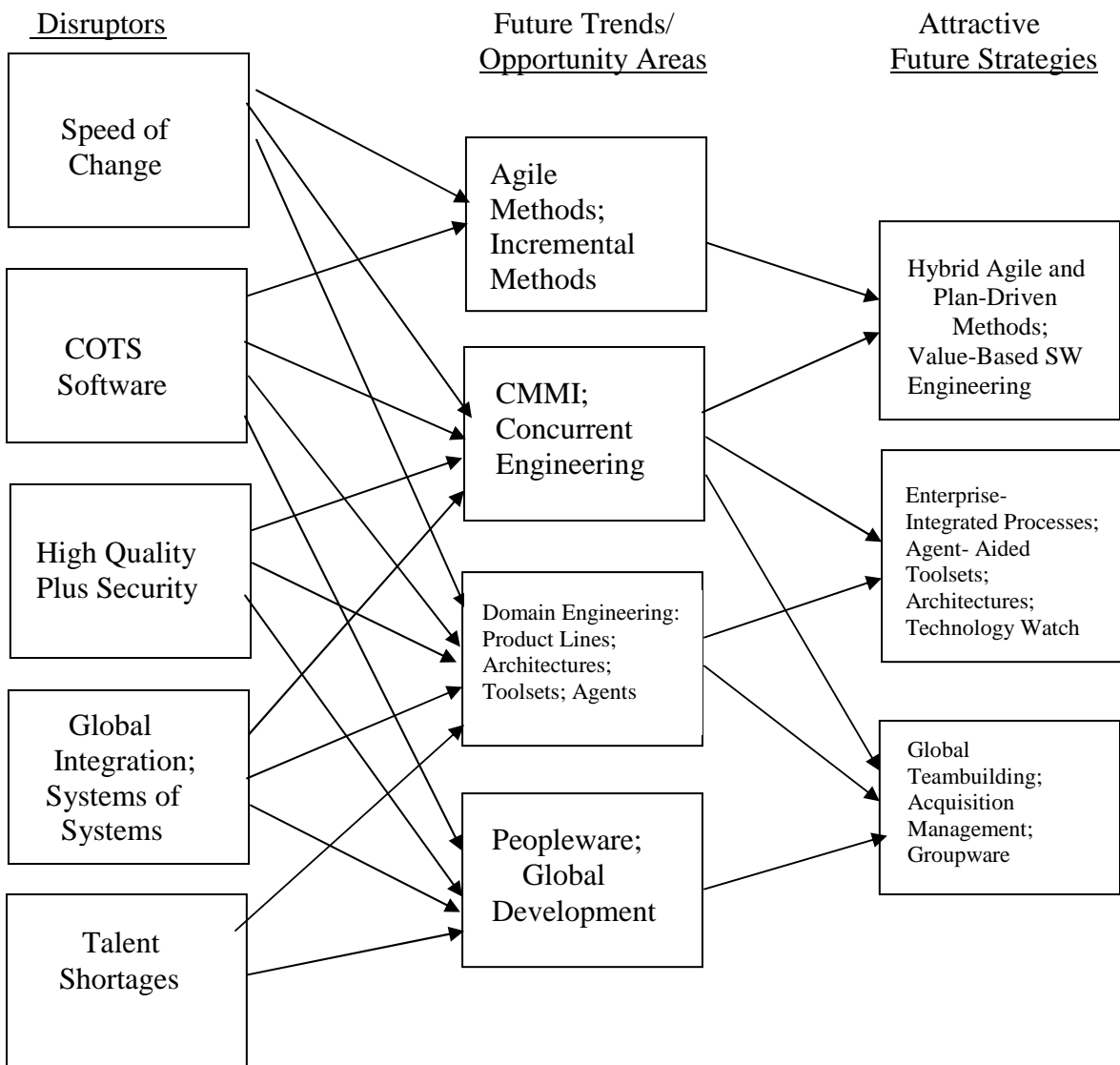
# Software Process Disruptors, Opportunity Areas, and Strategies

Barry Boehm  
USC-CSE-2005-500, January 2005  
Copyright USC-CSE 2005

## 1. Introduction

The near future (5-10 years) of software processes will be largely driven by disruptive forces that require organizations to change their traditional ways of doing business. This report begins with a discussion of the major current and near-future disruptors in the software process area and how they interact. It then discusses major trends in terms of opportunity areas for dealing with various combinations of disruptors. Based on the opportunity areas, it then identifies some attractive future strategies that appear to have high payoff. Figure 1 provides an overview of the relations among the future disruptors, opportunity areas, and strategies.

Figure 1. Software Development Process Trends and Strategies



## **2. Disruptors**

D1. Speed of Change. Increasingly rapid changes in technology, competition, and the business environment will continue to disrupt any long range plans an organization may have for the future. The pace of change will inflict heavy penalties on overly bureaucratic and document-intensive software processes.

D2. Commercial Off the Shelf (COTS) Software. The distribution of USC-CSE's e-services projects has gone from 28% COTS-intensive applications to 70% between 1996 and 2002. The 2000 Standish Group survey reported 54% COTS-intensive applications. Using COTS is economically essential in most application domains, but COTS opacity and lack of controllability make it a serious engineering challenge.

D3. High Quality Plus Security. Software quality attribute engineering and tradeoff analysis were already difficult enough before 9/11/01. The post-9/11/01 emphasis on security and the increasing use of hard-to-secure COTS products have made the job even harder.

D4. Global Integration; Systems of Systems. Everything and everybody is increasingly connected to everything and everybody else. Once-standalone systems become parts of large systems of systems they were not designed to interoperate within. Global integration adds localizability to the already long list of quality attributes to concurrently engineer.

D5. Talent Shortages. Demographics and cyclic trends in computing career attractiveness will reduce the supply of people talented enough to cope with the added complexities above, while the level of personnel capability needed to address the complexities is concurrently increasing. Systems engineering as well as programming skills will be increasingly needed.

## **3. Future Trends in Terms of Opportunity Areas**

T1. Agile and Incremental Methods. Although current agile methods do not scale up to address the full complexities of large systems of systems, they provide new perspectives and techniques for dealing with rapid change and COTS products. The increased speed of change puts a higher premium on incremental methods vs. start-over methods (e.g., for regression testing).

T2. Capability Maturity Model-Integrated (CMMI) and Concurrent Engineering. The CMMI's emphasis on concurrent software and systems engineering, integrated teams, and risk management provide a framework for incorporating more agile methods into a disciplined software/systems engineering process.

T3. Domain Engineering: Product Lines, Architectures, Toolsets, Agents. Corporate product line architectures, components, and software patterns are a good

investment for this opportunity area, although the increased speed of change and obsolescence must be factored into the investment strategy. Emerging technologies such as model-driven and aspect/feature-oriented development and agent technology will pay off earliest within relatively mature application domains.

T4. Peopleware and Global Development. Considerable talent is necessary to succeed with these new opportunity areas. Concurrent progress in support for distributed group collaboration, continuing education, career path development, and multinational teams provide opportunities for pro-active organizations to grow and retain their talent base. Agent technology offers opportunities for replacing labor-intensive tasks.

#### **4. Attractive Future Strategies**

S1. Hybrid Agile and Plan-Driven Methods; Value-Based Software Engineering. Risk/opportunity-driven process model generators such as Spiral Development; Lean Development, and “Lean RUP” provide ways to tailor agile and plan-driven methods to specific project situations. The WinWin Spiral Model is an example of a stakeholder-value-based software engineering (VBSE) process. VBSE processes are a significant step forward from current value-neutral processes such as object-oriented development and waterfall processes, in which every requirement, use case, object, test case, and defect is equally important.

S2. Enterprise-Integrated Processes, Architecture, and Toolsets. To cope with systems of systems of systems, COTS, and rapid change, there needs to be an enterprise-wide framework within which domain-specific processes, architectures, and toolsets can interoperate. Adding agent technology and incrementality to toolsets can significantly increase their power and ability to adapt to rapid change. A continuous technology-watch activity is necessary to keep the enterprise framework from becoming obsolete.

S3. Global Teambuilding, Acquisition Management, and Groupware. Global outsourcing and teambuilding requires advanced acquisition management capabilities (also being incorporated in versions of the CMMI). It also requires retraining software engineers to think like acquirers rather than like developers. User interfaces for both developers and end-users need to better reflect the transition from individual to group-oriented performance.