

Software Process Improvement Paradigms for IT Industry: Why the Bottom-Up Approach Fits Best

Dr Lin Zucconi
CSIRO Division of Information Technology
PO Box 664, Canberra ACT 2601
Australia

Improving software process with the objectives of improving software quality, productivity, and deliverability is a major goal of many organisations in the IT industry. Various paradigms for improvement and for process capability modeling are in use or in trial. These approaches fall into two basic categories: top-down, such as the Software Engineering Institute's Capability Maturity Model (CMM), and bottom-up, such as Prof. Victor Basili's Quality Improvement Paradigm (QIP) [1]. At the same time, a number of software quality-related national and international standards are being promulgated and some are seeing wide spread adoption.

These factors raise two issues that must be discussed within the software engineering community. These are the

- relative merits and deficiencies of the various software process improvement paradigms and their relative applicability to various organisational cultures, and
- relevance of the various software quality standards and process standards to actual improvements in quality or productivity.

Businesses are focused on producing innovative product and services. In order to retain market share and to be able to remain in business, they must produce their products or services minimising time-to-market and with a level of quality appropriate for their marketplace. The largest segment of the world IT economy is that populated by small and medium size companies. These do not have deep pockets that can support significant R&D or large process improvement projects. Changes in these organisations need to be evolutionary, not revolutionary. They need to precisely fit their business objectives. Meanwhile, engineers and managers must focus on getting product out-the-door.

Yet increasingly, these companies realise that to remain competitive and innovative, they need to improve the processes by which they build their products. The process changes must be driven by each company's goals, characteristics, product attributes, and experiences. The local environment will define the type and scope of and the need for each change, not an external model or universal set of standards. These can offer only guidance to assist the company engineers in understanding the nature of their problems with software development and maintenance practices.

The bottom-up improvement approach assumes that each company must first completely understand its processes, products, software characteristics, and goals before it can select changes appropriate to improving its process. It recognises that it is essential to first understand the software business of the organisation before any

change can be made. It ensures that any change is driven by the needs of the company and guided by the company's experience and not by an externally generated set of general concepts.

The fallacy of the top-down approach is that it assumes that a set of universal practices exists and that these must be applied to all software development. This approach promulgates a "one-size-fits-all" approach to software development no matter what the size of the organisation, the nature of its software or its market, or of its culture. As Frank McGarry states "the bottom-up approach assumes the most critical element of the change is the detailed understanding of the local processes, products, characteristics and goals. Only after developing this insight can a measurable, effective, change program be implemented." [2]

Similarly, in general, the relationship of software standards to actual improvements in quality of the *product* has not been demonstrated. Software engineering standards are heavy on process and light on product. In other engineering disciplines, the reverse is true. Also in other disciplines, engineering standards include methods to measure compliance. [3] So until we can produce standards that describe validated practices, with descriptions of means to measure compliance, our standards may only result in the production of tonnes of paper and in organisations receiving "certifications" of little real value.

There is a light at the end of this tunnel. Wide application of the bottom-up process improvement paradigm will yield industrially proven best practices that can be used to build validated software engineering standards where compliance can be measured. And most importantly, these practices will have been shown to directly affect the quality and productivity goals of the IT industry.

References

- [1] V. Basili, and S. Green, "Software Process Evolution at the SEL," IEEE Software, V. 11, N. 4, July 1994, pp. 58-66
- [2] M. Thomas, and F. McGarry, "Top-Down vs. Bottom-Up Process Improvement," IEEE Software, V. 11, N. 4, July 1994, p. 12-13
- [3] S. L. Pfleeger, N. Fenton, and S. Page, "Evaluating Software Engineering Standards," IEEE Computer, V. 27, N. 9, September 1994, pp. 71-79