Software Process Re-engineering and Improvement

Mikio Aoyama
Department of Information and Electronics Engineering
Niigata Institute of Technology
1719 Fujihashi, Kashiwazaki 945-11, Japan
Internet: mikio@iee.niit.ac.jp

Abstract
We discuss the methodology and lessons learned in the re-engineering and improvement of software process.

1 Revolutionary Change

It's widely recognized that process is the key technology to analyze, design and evaluate software development activities at every aspect [5]. For example, CMM (Capability Maturity Model) is adopted at numbers of organizations around the world [6]. However, we witnessed revolutionary changes in both software technology and society [3]. It's frequently asked whether we have to re-engineer or improve conventional process models and techniques.

Changes of software development are multifold as follows:

(1) Change of Business Goals: Time Development cycle-time became the major goal in software development. The RAD (Rapid Application Development) and concurrent-development process comes into real life [1, 2].

(2) Change of Computing Models The wide spread of the Inter/intra-net is changing the computing models, hence demanding the change of software development in many ways. Like CALS (Continuous Acquisition and Life-cycle Support), CE (Concurrent Engineering) and EC (Electronic Commerce), to use Inter/intra-net in software development is norm.

(3) Globalization of Software Supply and Demand Built supply and demand of software are globalized [4]. For example, many Asian companies take parts of contracted development and also import various package software. In global distributed development, people have to collaborate beyond the differences of languages, business practice, and culture.

(4) Change of Development Methodologies Object-orientation and componentware are coming into practice. However, they require different software process and management methodologies which are yet under discussions.

2 Can and How Process Work for Changes

To change process may affect every aspects of software development activities. It's sometimes painful to re-engineer and improve software process. We need to share the best practices and lessons learned. It's also needed to discuss the limitations of current technology and demands of new ones to cope with the changing paradigms.

Based on the motivations abovementioned, we are intended to discuss the following topics.

1) New demands to and changing paradigms of software development
2) Methods of software process re-engineering and improvement
3) Information and software technology for software process re-engineering and improvement
4) Best practices and lessons learned
5) Future directions

References