Chapter 10
Comprehensive Software Industry Analysis Model (CSIAM)

T.R. Gopalakrishnan Nair
Dayananda Sagar Institutions, Bangalore, India

R. Selvarani
Dayananda Sagar Institutions, Bangalore, India

Muthu Ramachandran
Leeds Metropolitan University, UK

ABSTRACT

This chapter discusses a comprehensive analysis model of the concurrent software industry which is a collection of different sub-models interacting and communicating objectively with a cause-effect relationship. This model is not merely process centric but it also deals with the resources, skills and methods with an entity model which forms an important factor of the comprehensive approach. Process is considered as an activity performed over the resources with a particular level of skill set transforming the input. In this comprehensive model, the factors associated with various sub models like process, technology, business, risk and multi domain are considered and integrated through communication links, threading all the models in transforming the system specification into the product within the cycle time with an active cost. The final model will be able to accept a spectrum of input from the industry, ranging from finances, human resources, business practices, technology and software production process and come out with metrics correlated to the observables of that particular company or group.

I. INTRODUCTION

The software industry has witnessed massive growth since its transformation from a pristine sector catering only to scientific research. This exponential growth has posed a lot of challenges related to standards and process models. The available models today, though partially useful, have been deemed inappropriate for the present growth scenario. The need of the hour is a comprehensive model that takes into account the aspects that implicitly or explicitly facilitate the industry. The present standards or models are process centric (http://www.sei.cmu.edu/cmmi, Carod, Martin & Aranda, 2004). The idea of
software development revolves only around the process neglecting other important aspects such as business aspects, risk aspects, technological aspects, and aspects of multi domain analysis. This paper intends to develop individual models for each of the aspects mentioned above and finally integrate them to arrive on a comprehensive model which can facilitate and identify the key factors of success in the software industry.

This comprehensive analysis model is making use of the rationality of the software engineering as a structured collection of practices carried out through suitable resources. It will be treated as an entity to interact with the various other limbs of the software industry contributing to the success of a software business operation. The existing models and ideas supporting this aspect will form an integral part of the process unit of this model. This model will develop a metric system which indicates a set of performance indices of the companies in response to the variation of best practices prevailing in those companies. These metrics on the software companies will distinguish one company from the other though all of them have achieved a stamping of high maturity in the process view in conventional methods and exhibiting behaviors contrasting one to another leading to puzzles and ambiguities for a commercial deal. Comprehensive Software Industry Analysis Model (CSIAM) presumes that a comprehensive metric system of industry like this can address the failure rate of software projects existing in the field (Menkhaus & Andrich 2005).

II. PROCESS TRANSFORMATION TO A SUPER RESOURCE SCENARIO

In fact software technology has moved quite a long way for the last 20 years from where process centric thoughts blossomed (http://www.sei.cmu.edu/cmml, & Morgan) and implemented improvements in process that could bring geometric response in the generation of software products. Nevertheless, the technology has changed and the resources have grown quantitatively and qualitatively in significant size. It continuously demands process changes to suit the growth and sometimes even consuming a part of the process into the intelligence of the active resource. This transformation is inevitable and will be on a growing path along with the crowding of the new processors and intelligence power, taking place in computing systems. So the productivity of the software generation system is not purely dependent on old process centric factors (Menkhaus, Frei & Wuthrich, 2006) which are evident from the data that equally process certified production units performing with different productivity and quality. This calls for an open and liberal view to the software engineering of today without historical binding of thoughts prevailing in software generation by process refinement.

Modern software generation is not entirely depended up on process. Even if unbelievable improvements are brought into process, there will be only marginal differences in the form, fit and function (FFF) of product and the quality of product because today’s process based production can not account the resources and skills aptly. In reality, these days the process itself is getting integrated with software generation intelligence which is concurrently getting transferred to machines (Menkhaus, Frei & Wuthrich 2006) which is remaining unnoticed. Every successful product requires a minimum standard of logically correct process. The active resources play a significant role currently and will play an entire role in the production of software, when increasingly autonomic systems are introduced in software industry. The present scenario needs a shift from process centric software generation to process and resource centric software generation models. It resolves many issues in the current software industry (Zhong, Madhavji & El Emam, 2000) including the differential performance of the software companies having the same process standards (Rauterberg & Aeppli 1996).
Related Content

An Approach to Discover the Stable Routes in BGP Confederations

Business Intelligence and Analytics Cost Accounting: An Empirical Study of the Perceptions of Stakeholders and Their Implications

A Roadmap for Software Engineering for the Cloud: Results of a Systematic Review
[www.irma-international.org/chapter/roadmap-software-engineering-cloud/77696/](www.irma-international.org/chapter/roadmap-software-engineering-cloud/77696/)

Software Security Engineering: Towards Unifying Software Engineering and Security Engineering
[www.irma-international.org/chapter/software-security-engineering/29532/](www.irma-international.org/chapter/software-security-engineering/29532/)

Modeling Transparency in Software Systems for Distributed Work Groups
[www.irma-international.org/chapter/modeling-transparency-software-systems-distributed/75759/](www.irma-international.org/chapter/modeling-transparency-software-systems-distributed/75759/)