AGILE METHODOLOGY IN CMM FRAMEWORK: AN APPROACH TO SUCCESS FOR SOFTWARE COMPANIES IN CHINA

Yan Xu, Texas Tech University, Tel: (806) 747-3247, yxu@ttacs.ttu.edu
Zhangxi Lin, Texas Tech University, Tel: (806) 742-1926, zlin@ba.ttu.edu
William Foster, Arizona State University – West, Tel: (602) 543-6209, wafoster@asu.edu

Abstract: CMM has been well accepted by the world as the standard for software development process control, quality improvement, and software company capacity evaluation. Aiming at the software outsourcing service market, more and more Chinese software companies are involved in the CMM certifications rush. However, to most of them, CMM is a very challenging goal. As China's software industry is still at a primitive stage, it becomes critical to seek a suitable way to promote the competence of the small- and mid-sized Chinese software companies in worldwide competition. This research-in-progress paper suggests that Chinese software companies adopt agile development methodologies (ADM) while following the CMM standard. Using eXtreme Programming (XP) as the example, the paper shows that the ADM methodology can well fit the CMM framework. A further research plan is proposed for the case study in order to obtain the evidence to support the arguments in the paper.

INTRODUCTION

Inspired by a bright future, the mushrooming Chinese software companies have been trying to be more competent in a rapidly changing environment. When looking at the success of India's software industry, Chinese software companies realize that Capability Maturity Model (CMM) is a well-accepted standard for process improvement and product quality control (Wang and King, 2000). CMM describes the maturity levels of a software company in five levels: Initial, Repeatable, Defined, Managed and Optimizing. CMM is proposed to help software development companies and organizations to establish a mature software development process with high predictability and low risk (Persse, 2001). In the global software service market, whether a company has passed CMM certification and at which level the company has qualified will help the company to win the software contracts.

Now more and more Chinese software companies are pursuing CMM certifications. The reality is that over 90% of software companies in China are small-sized with fewer than 100 employees. Most of them have little experience in servicing clients outside of China. While the CMM certification is costly and its procedure is lengthy, most of companies cannot afford to pay for the certification nor will they benefit in a reasonable time frame. Meanwhile, the most important challenge to these companies is how to survive in the tough market.

To solve this problem, we suggest that Chinese software companies adopt the agile development methodology (ADM) for software development in conjunction with the CMM framework. Agile software development was proposed in the late 1990s. The idea of the agile software development is to adopt the light-but-sufficient rules of project behavior and to use human and communication-oriented rules (Cockburn, 2002). It includes a set of innovative software methodologies, for example, the widely accepted eXtreme Programming (XP). These methodologies are more code-oriented and people-oriented rather than those traditional document-oriented and process-oriented. They focus on human factors, communications, and fast development. With this alternative choice of software engineering approach, the questions that many Chinese software companies are facing include: at the very beginning stage what is the best strategy for the company's growth – focusing dedicatedly on pursuing CMM or starting from an agile methodology and then approaching to CMM? If the latter is the right decision, how to identify critical success factors in using the agile methodology in the software service market of China? This study is intended to answer these questions.

SOFTWARE INDUSTRY - CHINA VERSUS INDIA

As a developing country with the largest population in the world, China has a big domestic market of IT industry and has experienced fast growing IT sector, but it is far behind India in software industry. For example, China's software export is about 0.6 billion USD in 2002 (http://www.csia.org.cn), while India's software export to the US alone was 6.2 billion USD in 2001 (http://www.ciol.com/content/news/repts/101110203.asp). According to NASSCOM-McKinsey's prediction, by 2008, the projected revenues of India IT industry will reach US $87 billion, which includes US $19.5 billion from software products.

The myths of India's success in software industry may be owing to several factors: quality software development process and management complying to CMM standards, English facility and culture background, promotion of government, and systematic IT training and education. Among these, CMM adoption is the distinctive one from others. According to SEI's report, 42 out of the 57 organizations worldwide that have reached CMM Level 5 on the CMM scale are headquartered in India. Hundreds of India's companies have been certified CMM Level 4 or CMM Level 3. As many India software companies are competent with their CMM certifications in the world market, no wonder that software services have become one of India's biggest export successes in the past decade. Compared to the current status in India, only five of the software companies in China have passed CMM Level 3 and twenty-one of them have obtained CMM Level 2 certification, two of them has reached CMM Level 4 or 5. Apparently, as a whole, China is far behind of India in software quality certification and project management skills.

AGILE PROGRAMMING IN A CMM FRAMEWORK

There are two main adversities for Chinese software companies to compete in the world market: Lacking workshop operation experience, and Insufficient English language communication skills and international business cultural background. We argue that these two weaknesses inhered by Chinese software companies could be complemented by the adoption of ADM in CMM frame, and Chinese software companies can take the advantages of agile methods without losing benefits of CMM the final goal.

On February 11-13, 2001, Agile Alliance was formed by seventeen software professionals in eXtreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, and Pragmatic Programming (Table 1). They intend to promote the concepts of agile software development, and help organizations adopting those concepts listed in their Manifesto for Agile Software Developments (http://www.agilemanifesto.org):

?? Individuals and interactions over process and tools
?? Working software over comprehensive documentation
?? Customer collaboration over contract negotiation
?? Responding to change over following a plan

Table 1: Agile Development Methods **(**extracted from Abrahamsson et al. 2002**)**

Name	**Core Ideas**	**Scope of use**	**Current status**
XP (Extreme Programming)	12 key practices. (such as refactoring, test before coding) No process to fit every project.	Good for small and medium size team, 3—20 people	Growing. More practical experience than academic research.
Crystal	A set of methods. Suggest development cycle within 4 months. Emphasis on communications. Allow adoption of other agile methods.	Not good for life-critical system. Up to 40 persons local development.	Four proposed Crystal methods, two of them exist.
Scrum	Do not require specific practices, but need management practices and tools	Suitable for small team. < 10 people.	More going-on researches try to integrate Scrum and XP together.
DSDM (Dynamic System Development Method)	Application of controls to RAD. Emphasis time and resource.	Team size between 2 and 6, multiple teams exist. Can be used in large system, if the system can be split into components.	Widely used in U.K E-DSDM was released in 2001.
ASD (Adaptive Software Development)	Emphasis on incremental, iterative development	Focus on developing large system. No built-in limitation.	No significant research.
FDD (Feature-Driven Development)	Focus on design and building phase. Emphasis iterative development. Needs other supporting approaches	Claims to be suitable for the development of large software project.	Some consulting firms advocating it. Relatively new and still evolving.

There is a prevailing mis-concept in software industry that CMM and agile are not compatible to each other because of the emphasis in the documentation (Ken, 2002). We have found that agile methods are containable to CMM. "*SEI's CMM is not a methodology for software development. It is not a production template, nor is it a set of* *process law. … … CMM is an approach for process improvement, a set of guideline that will help you forge and refine a development environment based on consistent repetition, measurement, and refinement*." (See Persse 2001). When comparing CMM with the agile software development, one will find that CMM tells what need to do, but does not say how to do it. An agile software development method provides a set of best practices to teach you how to do. CMM does not specify any concrete software development methodology, but provides a set of criteria for choosing an appropriate methodology. So CMM can include agile software development since it is methodology oriented. In fact, many CMM practices are satisfied directly by some kinds of agile methods. Relatively, Agile methods are more "light". It is aimed at fast and efficient development practices, so that they can support the rapidly changed business environment without following those rigorous traditional methods.

In the CMM framework, Key Process Areas (KPAs) are used to describe each maturity level except level 1. A KPA contains several Key Practices (KPs), which are the smallest unit in CMM and need to be carried out to fulfill KPAs. Taking XP as a representative agile method, as shown in Table 2, its 12 critical practices are well addressed in CMM Level 2 and 3, except in CMM level 4 or 5 (Paulk, 2001). It means that XP is more suitable to technical work than to management and infrastructure issues and more suitable to premature software companies. In fact, XP is getting popular and software companies' experience in successfully using XP is accumulating. So far, the respondents of using XP take about 40 % among those using agile software development methods.

Table 2: The Mapping of XP Practices to CMM KPAs (Level 2-5)

CMM Level	**KPAs**	**XP practices applied**
Level 5: Optimizing	Defect prevention	Continuous integration
	Technology change management	Not applied
	Process change management	Not applied
Level 4: Managed	Quantitative process management	Not applied
	Software quality management	Not applied
Level 3: Defined	Organization process focus	Focus on team
	Organization process definition	Metaphor
	Training program	Not applied
	Integrated software management	Not applied
	Software product engineering	Metaphor, Simple design, Testing refactoring, Coding standards
	Intergroup coordination	On-site customer, Pair programming
	Peer reviews	Pair programming
Level 2: Repeatable	Requirements management	On-site customer Continuous integration
	Software project planning	Planning game, Small releases
	Software project tracking and oversight	Small releases
	Software subcontract management	Not applied
	Software quality assurance	Pair programming
	Software configuration management	Collective ownership

Based on the above analysis, we reach the following two hypotheses:

> Hypothesis 1: *CMM level of a software development process adopting ADM will be higher than the level of the process without adopting ADM for small or middle size Chinese companies.*
>
> Hypothesis 2: *A software development process adopting ADM will fit more KPAs of CMM than the process without adopting ADM if they are in the same CMM level for small or middle size Chinese companies.*

In addition to benefiting Chinese software companies in pursuing CMM, ADM has the potential to reduce the software development cost for Chinese software companies in providing outsourcing services in the international software market. Software cost estimation usually means the predications of the likely amount of effect, time and staffing levels required to build a software system (Fenton and Pfleeger, 1997). Currently there are four approaches to software cost estimation: expert opinion, analogy, decomposition and models. They involve a more thorough analysis of software project factors that affect cost, such as project size, experience of developers, implementation language, degree of reuse and so on. In reality, English language is the major hurdle for most Chinese software companies to conduct cost-effective services mainly due to the overhead in documentations, because of either lacking English staffs or paying too high salaries for the employees with qualified English proficiency. With its light documentation requirement, ADM could be an idea choice for these companies to build up their international businesses.

Hypothesis 3: *Adopting ADM in software development process will reduce the documentation cost for small or middle size Chinese software companies.*

Hypothesis 4: *Adopting ADM in software development process will increase competence of small or middle size Chinese software companies in the international software outsourcing market.*

APPROACHES TO ADOTING ADM

For most small or middle size companies or organizations, adopting ADM should not lose the way to CMM. ADM is more suitable for software projects, which are small, time-critical and in a rapidly changed business environment. In particular, for most Chinese small or middle size software companies, which mainly service in the market of small projects, there are three approaches to adopting ADM:

?? Follow ADM methodology fully and strictly.

?? Implement ADM map as the development process and comply with most of rules and practices (Wells, 2000).

?? Enhance current software development processes with some of ADM rules and practices to improve the productivity and reduce costs.

The first two approaches are suitable to start-up companies, which have no defined or documented software development process. The problem here is that they need to fully understand what is ADM and how to apply it. Most small- and middle-sized Chinese companies who struggle in the software market have already used or followed some software development processes and methodologies. They can tailor the processes in use and adopt the ADM methods to fit the needs for different projects. Applying some ADM rules and practices will help them to speed up time-critical projects in a more effective way.

RESEARCH PLAN

Recently, some software companies in both United States and China have started to adopt ADM for their software processes and productivity. By studying and comparing the software processes with and without ADM in these companies, we will analyze how ADM can help a company to improve productivity, reduce cost on documentation and pursue CMM. We will choose some companies that have already adopted APM for an empirical study. For each company, we will analyze their software development processes with ADM and without ADM. Since ADM teammates communicate intensively and keep the documentation to the minimum, instead of writing masses of technical documents, we will compare the ratio of documentation cost to total software development cost of their software development process with ADM and without ADM. We will also conduct a mock CMM assessment on the software process using ADM and the process before using ADM in the same company. The assessment procedure developed by Michael K. Daskalantonakis at Motorola Inc. will be used. The purpose of this assessment process is to help engineers and managers to evaluate an organization's current status with regard to CMM and identify the weak areas to be improved (Daskalantonkis, 1994). It can be used as an internal tool to help companies and organizations prepare for a formal CMM assessment. Based on the analysis in the research, the expected result is that adopting ADM in software development process will help companies or organizations to pursue CMM consequently. The analysis results will help small to middle sized Chinese software companies to clearly understand why ADM is a good strategy to help them to pursue CMM. Currently we are collecting and analyzing software development processes and documents from our research target companies.

References:

Please contact authors for complete references.