

# **CMM in changing environment with uncertainty**

## **-Injecting ancient Chinese philosophy into modern science and technology**

**Zhou Zhiying**

Department of Computer Science and Technology,  
*Tsinghua University, Beijing, P.R.China 100084*

**Software engineering attempts to turn the software development from a workshop fashion into industrialization** for the larger scale, more complex, higher qualities and controlled schedule. The workshop fashion emphasizes the capability of individual, and characters the software development processes as prone mistakes, low speedy, unstable and uncertain in no order, but embedded the possibility of innovations based on individual subjective initiative. The industrial activities are based on repeatable, objective and/or scientific laws or rules with characters of accuracy, stable and speed. The standards or disciplines in industry provide the facilities of learning and automation.

Software engineering is based on projects and products, but pursues over projects, products, developing teams, organizations and individuals. Therefore, **the fundamental of software engineering is objectivity, which is the generic within software development**. The objectivity of software engineering determines that all activities in software engineering depending on facts, as characters of science: observable, public and repeatable. Documentation is important to software engineering, which recorded developing history of facts: the objective representations of understanding and progress, objective foundation of analysis and improvement.

**CMM reflects the best practice in the stable environments**, refer to figure 1. It provides the identified processes and activities in software development and declares orders of executing repeatable activities to efficiently reach missions of organizations. Organizations with higher CMM level behave better under stronger repeatable, sophisticated discipline and/or automatic tools as the matured experiences based on many past successes. CMM represents the main streams in software engineering.

The processes of economic globalism promote the diversity, the variation, the complexity, and uncertainty of technologies and environments. Successful projects continually emerge with the different ever conflict approaches. The popular confident **CMM [1] faces the challenge from lightweight methodology [2]**. In some times CMM suffers those complains as a root of failures. The debates on software process improvement are fulfilling forums of academic researchers and industry practitioners.

Here, I try to **study and understand these confusions in current reality**.

At first, we should understand that the **theorems, methods, practices, and the variety of standards on software engineering are different from the correspondents of physical world in classical sense**. Software development strongly depends on the specific environments and executive individuals. The nature of different individual in different environment would hardly be repeatable as the machine. In other side, it is hard to completely indicate the features of specific environment, conditions, the capability or behaviors of individual as the attempt of counting the water drops in the sea [3]. Therefore, all theorems so far on software engineering are just the approximations or models of the partial. It could never be repeated completely or never be executable without the substantial complement in popular developing projects. In case of the software engineering applied in the real environment, it is necessary for developers to identify the characters of projects and contexts, and adjust the difference initiatively. **The essential difficulty to model technology (or software engineering) is the evolution of the human being, the society and the world**. There are still the pretty rooms for software engineering development.

Then, I find **it is interesting to take the advantage of sight of TAIJI picture**, refer figure 2. TAIJI picture uncovers the relationship of two opposite poles with ancient Chinese philosophy of understanding the world: inter-transmutation and inter-involvement in gradual way or sudden way from two directions (forward and backward) of recursive evolvement. Suppose the two opposite poles are represented by colors: white and black. In any case, one pole, as black, always involves the other one, a

small white spot inside the black. The radius within a pole could be turned at bi-direction; this is the transmutation of poles. In one direction the pole could be gradually weaker until the strong opposite occupied, and in another direction a pole could be gradually stronger with involved opposite spot until the progress is broken and the weaker opposite pole occupies suddenly. Then the transmutation continues in different level or at different interpretation as in YIJING, the book of changes [4]. The world evolves in different interpretations forever. **TAIJI picture really is a generic model of evolution world.**

Back to software engineering, in early years of software develop, the manned work occupied as workshop fashion. The top mission of traditional software engineering is toward automation. But **software process improvement of concepts on only one fixed direction with some hard nuts still is today's reality.** Facing the rapid shift world, it is better to accept the concepts of evolving world. We could enrich software engineering with TAIJI spirit. Two poles in software development are manned work and automation of discipline.

Let us **interpret the TAIJI picture to two directions of software process improvement.** If the black half represents the human actions and the white half represents the characters of automation (the machine). At first, we worked in the workshop developing fashion. It is full of unstable or prone mistakes but complete flexibility in no order. There were always some repeatable activities in the workshop developing fashion, the white spot inside the black. After understanding those repeatable in projects, the rules were constructed to avoid certain mistakes and promote productivity. After composing the discipline/rules as standard processes, the discipline/automation gradually was substantial as process maturity being higher. This is **one direction of software process improvement, as regular CMM described.** But, whatever the satisfaction of the discipline/automation is, some thing should be more flexible than discipline/automation could be provided, as the black spot inside the white. The changes are increased and the white have to be turned off suddenly, the manned work with flexibility and innovation would resolve the inside spot over the capability of recent discipline/automation (XP seems as black one). This is one direction of the improvement cycle of black and white. **Fails of CMM in some recent cases are the victims of "Einstellung"** [5]. This German word for set tells the general factors that hinder problem solving by fixation. When people were confronted by the problem of the approaching tree, they attempted to solve it by performing a maneuver that had performed many times in the past and which had worked successfully then. But when the situation is different now, the work should not be planed as before. Experienced organizations with higher maturity could be blinded or masked some easier, more directive evidences by Einstellung. In this case of different context of projects, reducing the inference from past successes and increasing the flexibility for alternative would be better trade-off. Giving up the gains from past success, the work with less discipline/automation would be prone more mistakes, but the flexibility provide the alternative, which could reach new success easier, and mistakes may uncover vital alternative in uncertainty. This concludes that we need the other direction of improvement. **The backward direction also provides the possibility for improvement: reducing strength of discipline/automatic based on past, finding/adding new rules based on current, and re-composing the discipline to adapt the rapid change.** Specially, we could see facts of sudden way of this direction in software world. The new products of software innovations by manned work suddenly occupy the markets, breaking the gradual developing/ improving steps and directing to some different automatic. **Both directions could be the way of true improvement.**

However, one direction improvement is not enough to reality today. It is better to slightly **update CMM at sight of TAIJI into figure 3 with loop structure of bi-directive improvement,** comparing the original in figure 1.

CMM in figure 3 remains all elements of original CMM, except improving direction. In evolution world it is necessary **to refine the understanding of maturity, capability and satisfaction of organization's mission.** CMM declares to provide the framework/model of capability maturity with what should be done, but without how to do. The past success experiences institutionalize "how to do" of CMM's what should be done. Highest maturity or capability of organization should be understood as equipped the quantified and automatic tools of past experiences under the style of CMM framework. The past experiences of different organizations are different. Actually the maturity metrics only reflects the goals of activities, not the contents of activities with different complexity. Therefore the meaning or contents of same maturity level in different organizations are different. It should not be understood as a measurement of the absolute capability of performing tasks in all organizations. It ever

should not be understood as a measurement of the absolute capability of performing tasks in same organization in different periods of changing. Furthermore, **CMM's level is a relative maturity metrics of organization themselves in a relative stable period.** The maturity metrics and capability/improvement metrics should be different in the evolution world; especially **the metrics of capability/improvement should be changeable according to the role of values in evolving organization's mission.** They should direct the satisfaction of organization's mission. Organization with higher maturity benefits from merits of discipline/ automation, and suffers from Eintellung. Organization with Low maturity benefits from flexibility and human initiative, and suffers from weakness of human being. It is obvious in daily life that the steps to maturity are not as same as the steps to better capability to perform specific tasks. It is true only that the measurement for maturity and improvement could be same in one organization in the stable world. CMM in figure 1 provides the steps to maturity, not the exact steps to improvement for next mission.

The concepts of process improvement must be modified in the evolution world. In sight of TAIJI, every thing is temporal and changeable; there is no best forever, except the evolution forever. There are no definite and absolute best for the organization improvement practice in the evolution world, as **there is no best level in the cycle of improvement of figure 3, except suggesting two directions (increasing maturity or reducing discipline) with the uncertainty.** The theorems of uncertainty on expected value, decision under risk, payoff of matrix, nature of states and possible preferences, could help to predict and select which direction to adopt. We should not pursue so called best level or high maturity without assessing the risks of changes. Understanding and selecting suitable one from opposite directions is the best practice of making continue improvement in own case of evolution.

However, CMM could help software organizations to assess the current position in the improvement cycle, and give suggestion to possible improvement directions (increasing maturity or reducing discipline/automation) with uncertainty. Organizations themselves have to understand the risks of selecting improvement direction based on prediction. **Software processes improvement in the changing environment with uncertainty is activities based on insight, understanding and risk spirits.** Both CMM and theorem of uncertainty should be applied to each level of the process improvement in rapid shift world.

CMM in figure 3 gives clear insight of software processes improvement by spirit of TAIJI picture in evolution world. CMM in figure 1 became the reduction of CMM in figure 3 for stable environments. CMM in figure 3 also helps to avoid misunderstanding and misuse of CMM even in stable environments. It seems better to explain the recent phenomena and to help organizations taking actions in reality.

In additions, figure 3 surprises me what interest it is. There are many models and theories with stage styles of one developing direction in modern science and technology, such as economic, management, cognition, etc. They played important role in past time and have to face the challenge now. However it is not difficult to recognize two poles in these theorems: bottom and top. Then, the similar mapping as from figure 1 to figure 3 is quite easy. It is interest to notify that fixation in the mapped (models and theories) disappear. It may open some separation of concepts as capability from maturity in CMM interpretation. Then **mapping models and theories by two directions and two ways of selection in an evolution sense with open interpretation shine flexible and fitness to the reality in the changing environment with uncertainty.** Theorem of uncertainty would enrich traditional solutions in spirit of TAIJI picture. Specially, the connection and interchange of the opposite two suddenly could create surprise sense, and they could initiate innovations at sight of TAIJI.

Try to explore the generic treasury of an evolution world and accept the invaluable gifts from ancient Chinese philosophy.

## Reference:

- [1] CMMI Product Development Team. CMMI<sup>SM</sup> for Systems Engineering/ Software Engineering, Version 1.02, CMU/SEI 2000-TR-029, November 2
- [2] http: // [www.extremeprogramming.org](http://www.extremeprogramming.org)
- [3] Jaime Gil-Aluja. Elements for Theory of Decision in Uncertainty. Applied Optimization Volume 32. Kluwer Academic Publishers. 1999.

[4] The complete translation of YIJING ( In Chinese). Edited by Cunliang. Inner Mongolia people press. 1998

[5] Moshe F. Rubinstein, Kenneth Pfeiffer. Concepts in problem solving. Prentice-Hall, Inc., Englewood Cliffs, NJ. 1980

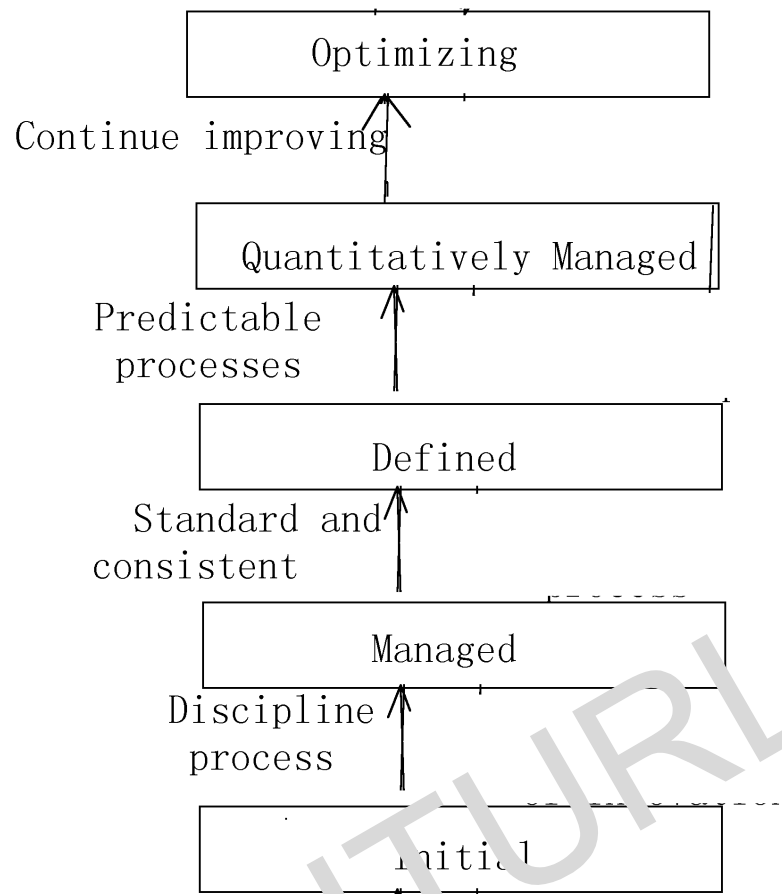


Figure 1 Original CMM presentation

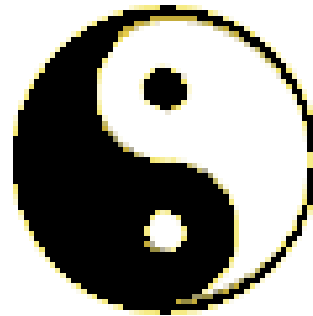
**Individual  
(Subjective)**



**Machine (tools) , Automation  
(Objective)**

Technique, methods, architecture, platform,  
Software/hardware separation, principle of computing

**Unify**



**Diversity**

Figure 2 Interpretation in sight of TAIJI

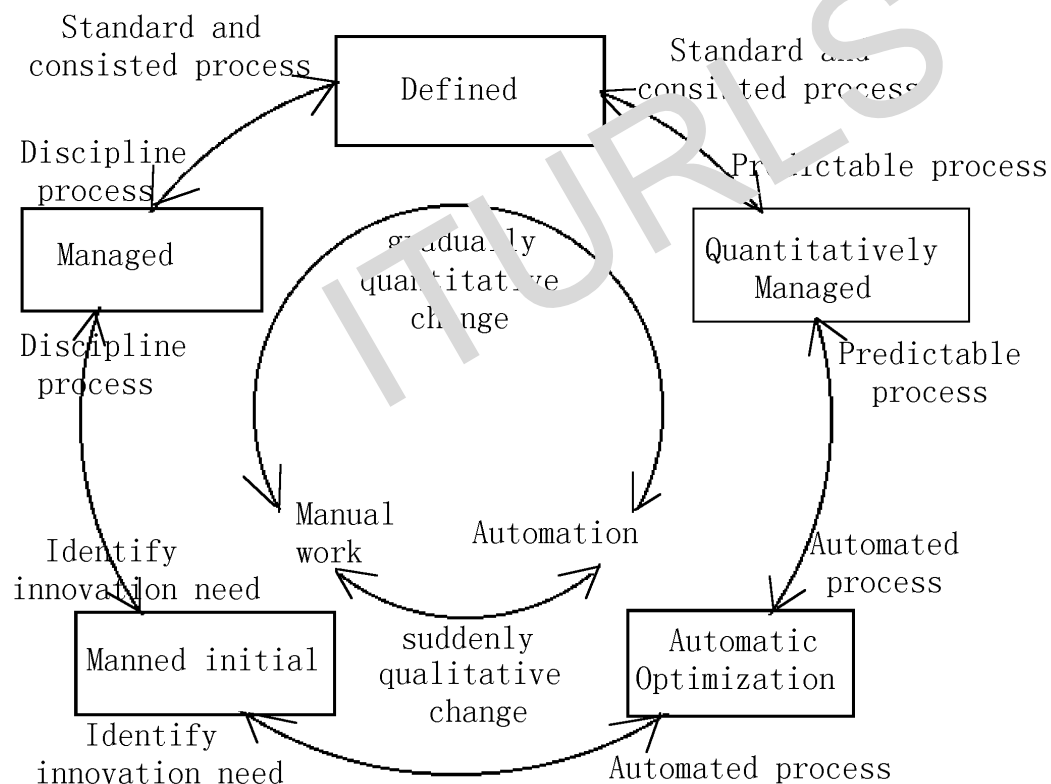


Figure 3. Mapping CMM in sight of TAIJI