Capability Maturity Model for Reducing Serious Troubles of IT systems

Hiroshi Ohtaka, Motomu Koumura, Hiroshi Nishikawa, Masahiro Isokawa, Kouichirou Oshika

Abstract

Although capability maturity models have been developed and applied to software developers for many years, serious troubles of IT (Information Technology) systems such as magnificent overruns of software development cost and piling up of social and economic activities due to software defects are still increasing in the software industry. Since it has not been clarified what are insufficient in the models, we cannot expect any improvement of the situation of the software industry in the future. This paper surveys not only legacy technologies of capability maturity model but also facts of the actual software industry and clarifies what are insufficient. This paper, based on the surveys, also presents a shape of an alternative capability maturity model for breaking through to reduce the serious troubles sustainably in the software industry, as well as a future vision that indicates how actual problems of the software industry should be improved in the future, which legacy capability maturity models have not been able to clarify yet.

Keywords: Capability Maturity Model, Software Industry, Serious Troubles

1. Introduction

Serious troubles of IT systems including software have not been discontinued so far. For example, increasing IT defects, which influenced negative effects to our social and economic activities, are reported by Information-technology Promotion Agency, Japan (IPA) [1] (Figure 1.).

![Figure 1. IT Defects/year Broadcasted by Japanese News Media (IPA)](image)

Other than the quality (Q) issue, many cases of the serious troubles have been observed. Magnificent losses caused by software development projects have been very often reported in financial statements of major software development vendors. These are examples of the serious troubles, which caused the management loss due to overrun of cost (C) for software development. As far as delivery (D) is concerned, we also often observe news of cases where postponements of IT service launches due to delays of software development or cases of aborting software development projects.

Software development vendor, developer or company (hereafter, SD in short), which include IT system integrator but exclude component package software vendor, has been suffering from such serious trouble of QCD (which is defined as either caused the delay of the customer service starting plan, resulting in enormous problems broadcasted by the news media, or caused a magnificent financial
loss to the management) repeatedly. That is, the following history has been observed in the software industry as illustrated in Figure 2.

1) Serious troubles outbreak in a SD.
2) The management of the SD empowers an organization often called like QA (Quality Assurance) or PMO (Project Management Office) and requests the organization to carry out its orders reactively.
3) The SD successfully decrease the number of serious troubles. After several years, the management of the SD decide to reduce resources allocated to the organization.
4) The power of the organization for checking and inspecting software development projects degrades, because the resources decrease. Because the power of the organization are weaker than before, the check/inspection is likely to become a mere formality.

Figure 2. History of serious troubles in the software industry

Serious troubles are apt to outbreak again in this situation. After several years, the serious troubles outbreak again and the status return to step 2).

The management at step 2), probably who is newly assigned due to loss caused by the troubles under the former management, may establish a new inspection regime and reporting style for the project risk check, rejecting the system established by the former management.

However, for tens of years after that, we have been observing that the same history repeat itself, where serious troubles outbreak again and again even in major prime SDs, which represent Japanese software industry.

The cause of such history is thought to lack of capability maturity of SD organization (organizational skill) to prevent serious troubles. It is probable that one of the causes is that we have only insufficient methods to visualize such organizational skill to prevent serious troubles in the actual software industry.

Based on such background, this paper first surveys methods of capability maturity model to visualize organizational skill. We also survey facts of the actual software industry. Next, based on these surveys, we clarify what are insufficient in the traditional models. We also derive an alternative capability maturity model to visualize the organizational skill, which improves the insufficiencies, and present a future vision for the software industry to improve the situation of increasing serious troubles, which legacy capability maturity models have not been able to present yet.

2. Literature review (surveying capability maturity models)

A lot of capability maturity models, such as CMMI, OPM3, P3M3, SPICE, BPMM, (PM)2 Model, etc. [2-7] have been proposed for improving organizational performance. And also many surveys of the models are practiced to clarified the difference between the models [8][9]. However, none of them have surveyed on viewpoint of the serious troubles of the software industry mentioned so far. Also most of the models have not been widely applied to the software industries for more than a decade, except the CMMI and the OPM3. Thus this paper focuses on the two models to investigate details.

A. CMMI (Capability Maturity Model Integration)

The CMMI [2] proposes an organizational capability maturity model and has contributed to make progress to standardizing software engineering processes by aggregated sufficient best practices of
engineers. It specifies five levels of the capability maturity based on the sufficient processes for software engineer as illustrated in Figure 3.

However, we often observe that software products, which are developed in SD organization with certification of the CMMI level five, are released too late or with full of bugs. Although the CMMI has sufficient processes for software engineers, it does not necessarily cover all software related processes needed for other stakeholders in the same organization.

For example, by comparing with the PMBOK [10], the CMMI is said to lack specifying clear processes for project managers, such as processes for risk management (like risk planning, risk identification and analysis, response to risk), integrated change control (like authorizing changed baselines or managed scope/cost/time corrective action, etc.), and other systematized processes to plan/execute/control project with every management area (like scope, cost or stakeholder). Without such management processes, the organization suffers from high risk that the serious troubles outbreak.

Moreover, since the CMMI level is certified by investigating only processes of a few sample projects, it is not difficult for a SD organization to be certified higher level of the CMMI, even if business of the SD organization is occupied almost by dispatching business of engineers, not by project contract business in which project management capability is required. Thus the CMMI certification is not sufficient incentive for SD organization with less project management capability to obtain higher project management capability.

B. OPM3 (Organizational Project Management Maturity Model)

Project Management Institute (PMI) shows a standard to identify capability maturity of organizational project management in general organization [11] by classifying targets of organizational management to “Project” [10], “Program” [12] and “Portfolio” [13] by the OPM3. Four levels are defined (Standardization (S), Measurement (M), Control (C), and Continuous Improvement (I)) to identify organizational capability maturity regarding project management as illustrated in Figure 4.
Although the OPM3 may be a common standard for all industries or services including construction, medicine or any other firms, it is very hard for SDs in the software industry to map the idea to their actual field. Even if the idea of the maturity model is understood, it is difficult for software vendors to apply the idea to step up their level of organizational project management to the higher one in the industry.

3. Surveying facts of software industry

When developing a capability maturity model for industrial organizations, it must be developed by classifying industries and carefully investigating facts in organizations of each of the industries. Particularly when developing the model for reducing the serious troubles in the software industry, the following facts (stakeholder and business in organization) should not be ignored.

A. Relationship between Serious Troubles and Organizations in the Software Industry

There have been a research on relationship between software troubles (problem projects) and their organization by analyzing who are responsible for each of actual 193 cases problem projects (107 serious troubles and 86 projects with less problem, which were summarized by expertized members of software development projects in a working group named “Project Mieruka” organized by IPA). In this research, the criteria of serious problem project (SPP), which is accompanied with serious trouble, is defined as either caused the delay of the customer service starting plan, resulting in enormous problems broadcasted by the news media, or caused a financial loss more than one hundred million yen [14]. The research clarified that although responsibility of project managers is still important, the ratio of cases, where senior managers and salespersons should take responsibility, increase, while the ratio of cases where only project managers should be responsible, decrease in serious trouble cases (Figure 5). When we focus on 86 cases other than SPP in the figure, senior managers and salespersons were responsible for 21%. On the other hand, as far as the SPP cases are concerned, the senior managers and the salespersons were related to more than 76% of the SPP cases.

It is suggested that not only the project managers but also whole organization in which the senior managers and the salespersons are involved should practice necessary processes for avoiding serious trouble of software development as described in the paper [14].

Such software related processes of stakeholders cannot be found in the common standard (the PMBOK/OPM3, which does not necessarily specify all of the software specific knowledge), nor in the software specific standards (any traditional model including the CMMI, which payed attention to engineers and project managers but payed less attention to stakeholders in the organization, like senior managers or sales person). This is one of important facts to consider effective capability maturity of organization in the software industry.

<table>
<thead>
<tr>
<th>Break down of responsibility for 86 cases other than SPP</th>
<th>Break down of responsibility for 107 SPP cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM</td>
<td>SM/SP</td>
</tr>
<tr>
<td>68 cases</td>
<td>1 case</td>
</tr>
</tbody>
</table>

Responsibility related to SM/SP (①+②) = 21.0%  
Responsibility related to SM/SP (①+②) = 76.6%

(note) PM; Project manager, SM; senior manager, SP; sales person

Figure 5. Break down of responsibility for problem projects

B. Surveying Actual Business of Software Industry

There are small business companies whose scale is too small to organize a software development project, among software related enterprise. Business of such companies is consulting or dispatching business of engineers with special software technology.
Even if we exclude such small-scale companies and major SDs or package software vendors, most of the other SDs do not aim to be software development contractors, which take responsibility for QCD goals of the software development project.

One of the causes might be that the managements of such SDs fear for business losses caused by cost overruns of software development contract. In actual, Nikkei BP often says decision by the managements of SDs to withdraw from contracting business for software development after cost overruns. It is not visible for the managements how to set a goal and how to achieve the goal for individual SD, to improve organizational project management skill enough to be a project contractor. Therefore, the SDs sink in business without taking project contract risk (just supplying software engineers to projects in the major contractors and regulating amount of required resources to receive/feed).

The other cause may that the SDs can keep survived without introducing project management process to all of the organization. They also can be even certified higher level of the CMMI, since the CMMI do not require project management process like the PMBOK nor organizational process in the OPM3 as mentioned before.

We categorize such SDs as CAT1, whose SDs are defined as companies which sink in dispatching business by supplying engineers, although their scale and engineering resources are sufficient enough to organize a software development projects.

The number of SDs in Japan is said to be approximately 20 thousands. Never the less, more than 90% of the whole SDs, including companies, which had been small SDs in initial stage but have grown enough to be listed in Tokyo Stock Exchange No.1, are said to give up to step up to upper level contractors (that is, staying in the CAT1 category).

Thus another important fact is that project management for software development has not been applied wider enough to most of SDs in the software industry. Such industrial structure of software, which has been not changed for tens of years, may be indirect cause that serious troubles have not been reduced in the industry.

The other SDs can be categorized as follows based on survey so far.

CAT2: SDs stepping up to be project contractors by decreasing dispatching business.

CAT3: SDs already have achieved the stepping up. Most of major prime software vendors are included in this category. However, they suffer from periodical serious troubles and repeating the same history as mentioned before.

CAT4: SDs trying to prevent repeating the same history of the serious troubles.

CAT5: SDs already achieved preventing serious troubles sustainably.

It is difficult to find any evidence that such details of software business are recognized not only in the common standard (OPM3) but also in the software specific standard (CMMI and any other models).

4. What are insufficient

Although the CMMI has aggregated sufficient best practices of engineers, it fails to be aware of importance of management practiced by stakeholders in organizations including senior managers or sales persons, to prevent serious troubles, without which serious troubles (SPPs) occurred in actual as shown in the previous section. The serious troubles mentioned so far are thought to indicate that management of the organization is poor. Since there is no plan to involve necessary management processes of sufficient stakeholders in the SD organization to the CMMI, we cannot expect any strategy or future vision for reducing serious troubles by the CMMI.

As for the OPM3, although it is a common standard of organizational management for all industry, we cannot observe any process of organizational management practiced by senior managers or sales persons for avoiding serious troubles in the software industry. Actually, there have been no report proving that higher level organization in the OPM3 could decrease serious trouble much more than the lower level organization in the software industry.

Moreover, as mentioned in the previous section, the CMMI has not achieved remarkable achievement for SDs to introduce project management wider to their business in the software industry. This might be another indirect cause of failure to prevent serious troubles.

Therefore, we can think that the traditional capability maturity models have the following insufficiencies for reducing the serious troubles sustainably.

1) Insufficient implication of management processes of stakeholders inside SD organization
2) Insufficient incentive for SDs to introduce project management wider in the software industry
3) Impossible to present any future plan for the software industry to reduce serious troubles

5. An alternative capability maturity model to eliminate the insufficiencies

An alternative model that improves the insufficiencies of the traditional models can be derived as follows.

Individual SD may be classified to either level in the following hierarchy of organizational capability maturity in software development, based on the categorization mentioned so far.

SD LEVEL 1: SDs corresponding to CAT1.

Engineer resources are required to the SDs from software development projects in major prime vendor contractors. However, the projects are managed by project managers of the prime vendors. This SDs are just practicing dispatch business for taking role of resource regulator. Even if the SDs practice any project contract business, the business scale is far smaller than the whole business of the companies. The management of the SDs give up to take project contract risk and make rightful project contract basically.

SD LEVEL 2: SDs corresponding to CAT2.

In the SDs, one can observe increasing project contract business, by replacing legacy dispatching business to step up to be project contractors. The SDs are enhancing skill of project managers in their own companies.

SD LEVEL 3: SDs corresponding to CAT3.

In the SDs, most of sales are achieved by project contract business. However, we can observe the followings.

- Project checklist:

   Methodology is “standardized” to check software trouble risk of projects as specified in the OPM3. However, the amount of check list is too large for field project managers to check all items in the list without extra load. It is also not easy for the management to be sure whether the checklist is effective or not.

- QA:

   The QA “standardize” to practice “measure” and “control” as specified in the OPM3. However, field projects not necessarily rely upon the QA members. Even when some QA members are highly skilled and fully relied upon by field projects, they are just temporally assigned to the QA team. When the business units lack resources of project managers due to increase of software development orders, they will be in a meanwhile moved to business units for managing field projects.

- Serious troubles:

   Periodically outbreak of serious troubles is observed. Soon after that, the management takes countermeasures for strengthening QA activities to reduce and prevent the serious troubles, by recalling the highly skilled project managers to the QA team as its members. However, for a mean while after decreasing the serious troubles and increasing the orders, the QA activities are to be weakened, serious troubles occur again (i.e. the same historical events are re-produced).

SD LEVEL 4: SDs corresponding to CAT4.

The SDs are trying to step up to be contractors with countermeasures to prevent outbreaks of serious troubles sustainably, by the followings and by practicing their “continuous improving”.

- Project checklist:

   The SDs have much more compact methodologies than LEVEL3, which both the managements and field project managers can use them without extra load.

- QA:

   Even when the business units lack resources of project managers due to increase of software development orders, the managements of the SDs continuously assign members to QA teams, who are highly skilled and fully relied upon by field projects.

- Serious troubles:

   Outbreak of serious troubles become to be observed less frequently than LEVEL3.

SD LEVEL 5: SDs corresponding to CAT5.

The SDs have achieved preventing serious troubles sustainably more than ten years, even if the management is changed to the new management. The SDs keep practicing “continuous improvements” without termination like  \textit{Kaisenz} in Toyota.
We can derive a new capability maturity model, by constructing a hierarchical structure from LEVEL1 to LEVEL5 mentioned above. The new model is illustrated at Figure 6.

Although the model is based on the OPM3, it makes software firm much easier to understand than the OPM3 and introduce it to software industry. That is, the “Portfolio” in the OPM3 is eliminated, since it is not necessarily proper maturity axis for SDs but necessary axis for software user or buyer.

Practical examples of suggestion for stepping up to higher LEVEL are shown in Appendix1, which involves management processes of stakeholders inside SD organization.

<table>
<thead>
<tr>
<th>LEVELs of New CMM</th>
<th>LEVEL1</th>
<th>LEVEL2</th>
<th>LEVEL3</th>
<th>LEVEL4</th>
<th>LEVEL5</th>
<th>Corresponding levels of OPM3</th>
<th>Corresponding levels of CMMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sinking in dispatch business</td>
<td></td>
<td>Increasing project contract business</td>
<td>Mostly project contract business</td>
<td>Trying to sustainably prevent outbreaks of SPPs</td>
<td>Achieved preventing SPPs sustainably</td>
<td>Level1~5</td>
<td>~Level5</td>
</tr>
</tbody>
</table>

Figure 6. Alternative capability maturity model

6. Comparing traditional models with alternative one

Usually, a valuable capability maturity model never neglects major problems which is happening actually in industries, to which the model is applied. However, the following A and B indicate that the traditional capability maturity models regretfully have less ability to cope with difficult serious problems in the software industry, than the alternative model.

A. The Traditional Capability Maturity Models

Most of existing SDs, whose number is said to be 20 thousands in Japan, are classified to only two groups in reality. One is major SD contractors with high project management skill and the other dispatch business SD companies with less project management skill. Such structure of the software industry have been continued without any change for tens of years. However, the serious troubles have not decreased as explained before, even after the CMMI were widely applied and other maturity models became to be applicable. The serious troubles have brought not only magnificent profit loss but also opportunity loss to the managements of SD companies, since much resources, who should have increased sales by other normal project contract, are consumed for shooting serious troubles.

Nevertheless, the traditional models have not even suggested any solution for the SD companies to cope with the serious problems for many years. They also have given less incentive for the SD companies which have less project management skill, to become prime contractors. Consequently, project management has not applied wider to the software industry for many years. Although it is hardly a wise option for the software industry to keep the existing structure of the software industry also in the future, the traditional models have not even suggested any vision for the software industry to improve the situation.

Thus, less person can undertake the traditional models as valuable in the viewpoint of resolving such serious problems in the software industry.
B. The Alternative Model

On the contrary, the alternative model copes with the problems of the software industry and tries to satisfy “what are insufficient” as follows.

First of all, after organizational skill of individual SD are visualized by applying the alternative model, most of major SD contractors will be certified as LEVEL3, and the other dispatch business SD companies may be LEVEL1. If the first goal is achieved, it may become natural for individual SD to make efforts for stepping up its organizational skill LEVEL to higher one, which offers incentive for SDs of LEVEL1 to upper LEVELs by introducing project management. And if most of SDs have the same idea, the movements for stepping up LEVELs in the software industry will be accelerated.

If such continuous endeavor are initiated, existing structure of the software industry will incrementally change from the existing two groups (LEVEL1 and LEVEL3) to 5 groups corresponding to 5 levels (LEVEL1 to LEVEL5) as illustrated in Figure 7.

In the last goal of this figure, the followings are achieved.

1) Software development is practiced under project contract business much wider in the software industry than before, which contributes to empowerment of project management wider in the industry.
2) Serious troubles are expected to decrease by higher skill of SD organizations, which is developed by stepping up to higher LEVEL and also by empowered project management obtained by 1).

The last goal in the figure illustrates future vision of how the software industry should be, in a view point of coping with the serious problems, which the legacy models have failed to visualize.

In this sense, it is possible to think that the alternative model is much valuable for the software industry, than the traditional models.

<table>
<thead>
<tr>
<th>Structure of the Software Industry</th>
<th>Software Industry As Is (Approximately 20 Thousand companies)</th>
<th>1st Goal; Visualizing Organizational Skill of Individual Company</th>
<th>Last Goal; Software Industry To Be</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Assumption based on the ideal maturity model</td>
<td>Step Up</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LEVEL 5</td>
<td>LEVEL 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LEVEL 4</td>
<td>LEVEL 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LEVEL 3</td>
<td>LEVEL 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LEVEL 2</td>
<td>LEVEL 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LEVEL 1</td>
<td>LEVEL 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Least of Software Vendors (most of which have survived by dispatch business for the Major Vendors.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Major Contractors (IT system integration contractors)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7. Future vision of the software industry

7. Conclusion

By surveying not only capability models but also facts of the software industry, we clarified “what are insufficient” in the legacy models. Under such “insufficient” situation, the software industry have been failing to reduce serious troubles for many years. The traditional models also have not been able even to suggest any possible future vision for the software industry to improve the situation.

The serious troubles may be obstacles also for developing new industries and services using new IT technologies such as AI (Artificial Intelligence) or IoT (Internet of Things) in a meanwhile. Therefore, to eliminate such obstacles as soon as possible, this paper also dared to suggest a shape of an alternative capability maturity model which shows a future vision for moving forward to reduction of the serious troubles as illustrated in Figure 7.

It is suggested for any organizations of information processing and management, which include developers of the traditional models, to quickly develop such capability maturity model, apply it widely to the software industry, and achieve reducing serious troubles sustainably in actual. We would like to continue research activities by collaborating with them, for visualizing organizational skill and aggregating best practices including IT managers and stakeholders in IT Mieruka Institute.
8. References


Appendix1 (Practical suggestions for SDs to step up to higher LEVEL)

The followings are recommended for those who have will to step up individual SD level in the alternative capability maturity model, as far as we have experienced.

[SD LEVEL1 to LEVEL2]

There must be project managers for managing software development project, when the SD try to become from LEVEL1 (a human bank of engineers) to project contract enterprise. Thus the SD needs to specify proper method to develop project managers among employees in the SD. However, as far as software development is concerned, the PMBOK has insufficient method to develop project managers, since it lacks software specific knowledge. In spite of recent software extension of the PMBOK, it is also difficult to say it is sufficient, since the software extension involves less tacit knowledge of experienced project managers, with which they actually have been preventing IT project failures in actual projects.

Thus it is recommended to use project management method specified in books named “Mieruka (Visualization) of IT project”, which have been published by IPA (four books via Nikkei BP, Minato-ku, Tokyo, Japan, 2006-2008). The method involves variety of practical tools such as bird-eye-view diagrams, compact check sheet, summary of problem projects, quantitative management tools and
integration management tool, which are developed by visualizing tacit knowledge of expertized project managers in major prime vendors.

[SD LEVEL2 to LEVEL3]

The SD increases project contracts comparing to dispatching business every year by developing skill of project managers (enlarging manageable project scale and complexity). The SD also increases amount of project managers. The SD should start trying to “Standardize” and “Measure” project management to meet the company style. The SD should also start trying to practice “Control” projects by organizing QA and by evaluating skill of project managers and engineers quantitatively.

[SD LEVEL3 to LEVEL4]

The SD may practice countermeasures for continuous improvement to sustainably prevent serious troubles. The following organizational actions are included.
- QA: By assigning QA members among expertized senior project managers, the SD organizes QA team under the top management. The members should have much higher skill than ordinary project managers and have been highly trusted by field projects. The top management keeps the members staying in the QA team, after problems of serious troubles are resolved, and even when business units lack project manager resources due to increase in orders.
- Project checklist: The SD simplifies the checklist so that both load levels of the management and field projects may be acceptable, by reducing large checklist to check items which are mandatory for prevention of serious trouble. For example, one way to practice the reduction is to link individual check item to each case of problem project like Mieruka method, and eliminate check items which have no link to cases of serious troubles.
- Tacit knowledge: The SD visualize tacit knowledge which project managers have obtained in their experience of serious troubles, before their retirement. It is also advisable to learn how to visualize tacit knowledge by participating in research activity to practice it by non-profit based organization.

[SD LEVEL4 to LEVEL5]

The SD establishes its own mechanism to prevent serious troubles sustainably.

For example, the SD has quantitative evaluation inventory of skill to prevent serious trouble in every stakeholders (i.e. not only project managers or engineers but also senior managers or sales person). A methodology to prevent serious troubles sustainably may be operated by quantitative management to check correlation between stakeholder skill and risk of serious trouble, using the inventory. Senior managers or sales person, who have been assigned by inventory, never fail to request customers to fix serious problem in the early stage of software development (e.g. when software owner is absent or not clear in the customer company, changeable software specification often cause serious troubles, which should be prevented by them).

Also QA team has strong leadership for avoiding serious trouble. QA may influence the management decisions.

After keeping such continuous improvement, the SD has been not suffered from any serious trouble sustainably.