Measuring and Improving Software Process in China

Qing Wang, Mingshu Li
Institute of Software, Chinese Academy of Sciences
wq@itechs.iscas.ac.cn, mingshu@admin.iscas.ac.cn

Abstract

Measurement is an important facility to support effective and reasonable management. Software development has its inherent property of high-dependency on the personal capability of software engineers that causes the variance and instability of software processes. Therefore measurement for software process has long been a challenge. This paper discusses the major problems in software process measurement, presents an active measurement model (AMM) to support software process improvement (SPI). Based on the AMM, software organizations can establish adaptive measurement process and execute the measure just close-related the process goals, which focuses on their particular business environment. Subsequently, some methods of establishing the adaptive measurement process and the appropriate software process performance baseline to support higher level quantitative management are suggested. The application and implementation of AMM and the related methods is introduced in the end. These works were integrated in a toolkit called SoftPM, which was used widely in China.

1. Introduction

The goal of software engineering is to ensure the software development and production under budget, on schedule and with high quality [1].

Software process engineering focuses on achieving these objectives by providing effective and efficient process management methodologies and techniques. Measurement is a primary facility to enable the software process performed with predictable performance and capability, and to ensure process artifact meeting their requirements [2].

Enlightened from manufacturing industries, software process management involves similar methodologies and technologies [3, 4, 5] which are widely applied in these traditional industries. Quantitative process management is one of the key characteristics of higher maturity level process capability. But Software development and produce is very different from manufacture. The methodology and technology, which applied for software process management should be improved appropriately and compliantly. Measurement is same too, It's very important that achieve the expected effects of software process improvement based on effective measurement.

Without measurement, it's unlikely to well understand software process and produce high quality products. When you can measure and describe the thing with data, you can understand what is actually going on. Effective measurement can be used to help identifying, analyzing and solving the problems arising during the development process, to evaluate and improve the capability and maturity of processes, and to predict the quality of process products [6, 7].

With appropriate measurement, software organizations can predict the performance and capability of their software processes, ensure the quality of software products and improve their processes continually and effectively. Section 2 of this paper discusses the difficulty of software process improvement. Some investigation information from software organizations in China was applied. Section 3 presents AMM - an Active Measurement Model that help establish an adaptive measurement process driven by the process goal depending on organization's business environment and mission to support continual process improvement. Section 4 then discusses the principle and method to define and establish an adaptive measurement process and establish and refine the appropriate process performance baseline evolutional. Section 5 introduces the application of AMM. SoftPM, a toolkit that implements and integrates AMM and related methods was brief introduced. Finally, Section 6 presents conclusions.

2. The Difficulty in SPI

We have investigated 38 software organizations in China. We designed survey forms consisting of multiple choice and open-end type questions to search the key problem for software process improvement.

Table 1 and Table 2 show the results according to all of the survey data for all of the investigated organization and just small-medium enterprise (SME) of them respectively.

Table 1. Major Problems for all

Rank	Problem	Frequency
1	Over-complex and dogmatic process	65％
2	High cost	52％
3	Other	＜5％

Table 2. Problems for the SMEs

Rank	Problem	Frequency
1	High cost	92％
2	Over-complex and dogmatic process	85％
3	Other	＜5％

The feedback information shown cost is a universal problem, it is more severe for SMEs. To understand deeply why cost influences SMEs more than the larger ones, we compare three companies' SPI investment and revenue. Company A has more than 500 employees, company B has about 150 employees, and company C has 60. See Table 3. From the table, the large company investment-revenue ratio is 0.7%, which is much lower than 5-13% for the SMEs. For large companies with decent revenue, 0.7% of yearly revenue is not a heavy burden, but for the small and medium size company, 5-13% is unbearable.

Table 3. Cost comparison among large, medium and small companies

	A	B	C
Effort (man-year)	15	13	10
Capital /year（kRMB）	1100	800	500
Revenue（kRMB）	200000	22000	6000
Investment/revenue ratio	0.7％	5％	13％

That means the research and development work to cut down the cost of SPI is very important and significant. Measurement will be an effective solution to address the problem.

The second major problem is "Over-complex and dogmatic process". When interviewing SPI consultants of these organizations, we find that since some standards such as CMM/CMMI tells just "what" to do", SPI consultants often provide advices on "how to do" at strategic level while leaving intensive, time-consuming tactic SPI efforts to the clients, such as deeper understanding of CMM/CMMI, carefully defining processes and determining the priority of software process improvement which match both their own environment and CMM/CMMI requirements. Unfortunately many software organizations simply take the advices of how to define a set of CMM/CMMI-compliant processes and the process performance baseline. They forget process usability is hard to improve in a short period of time. Only the adaptive process improvement will bring benefit to organization.

1/3 of the software organizations complain about lack of automated supporting tools. This make the above two problems even worse.

From this investigation, we can say the major problem for software process improvement is the lack of understanding for software process. It results in the high cost and bad cost-benefit of SPI directly.

Measurement plays an important role in process management. If you cannot measure, you cannot manage. Actually, in CMMI Measurement and Analysis is a basic PA was categorized in support process and maturity level 2.

Unfortunately, most of the Chinese software organizations haven't use measurement to support their process management. One reason is that many software engineers hate measurement. They consider that the software process are too uncertain to measure.. Another reason is the lack of effective method and technique to guide their work. Manual data processing will brings extra management overhead and not get the better cost-benefit. Measurement is being the bottleneck of software process improvement, that blocks software is developed with industrialization.

3. An Active measurement Model for SPI

The objective of measurement is to understand the objects with appropriate data and method. To make measurement effectively and appropriately, the primary problem should be addressed is that describe the features of the object which is measure, construct the metrics, collect data and analyze the results. Software process measurement includes all above elements as well.

3.1. Problems Issued by Software Process Measurement

- Describing the features of software process

Describing the features of an object is the first step in measurement. Software process has many features, but not all of them should be measured the same way. Different organizations focus on different process goals, sometimes even the same organization under different business scenarios and at different maturity levels. This necessitates that different process features should be handled carefully. The first problem that the software process measurement should address is how to extract and describe the process feature, reflecting the aspects of related process goal exactly and easy to be measured. Formalized description is an effective and useful method in this regard.

- Modeling the infrastructure of software process measurement

Software process has its inherent property, such as highly volatile, knowledge intensive, concurrent and coupling among different categories of processes. This adds tremendous difficulties in software process measurement, and explains why the higher level of process capability is harder to be achieved. As we know, quality is not free. It is very important to build an adaptive measurement to support software organizations to deploy appropriate measurement in a cost-effective way. Measurement model, which constructs the framework and infrastructure of measurement, must be established. Many researches have contributed to it, such as GQM [8], GDM [9], PSM [10], and so on. This paper presents an integrated measurement model called Active Measurement Model - AMM. With AMM, organization can generate a specific-adaptive measurement process to measure and analyze the features related to process goals with respect to a particular organization and its business scenario.

The algorithm of measurement is another primary difficulty for software process measurement. Many research and practices applied statistical technology, such as Statistical Process Control (SPC) [11, 12, 13] and Six-Sigma [14,15]. But the precondition of SPC is hypothesizing the process is stable and for small piece and large batch repeat production. Software projects are often long term and haven't absolutely repeatable process. Every repeat of any process will produce different process products. This brings the challenge for software process measurement. How to determine the measurement entities, construct reasonable data samples, and apply appropriate algorithm is the key.

- Establishing the performance baseline for quantitative management

The baseline of software process performance is more difficult to establish than manufacture. In manufacture, the process capability is depended on devices mainly. The process performance is assured after the product-line is built. But for software process, the process capability is highly dependent on available resources. Different people maybe deliver same scale process artifact with different quality and different productivity, even though the development environment and devices are the same. The process performance baseline cannot only rely on experiment, it must be established and refined evolutionally depending on process data and appropriate measurement and analysis.

- Constructing the strategy for data collection

Data is the basis of measurement but is also the bottleneck of measurement. Some organizations collect lots of data but never really use them, many others lack effective method to gather data. Huge amount of data means high performance measurement facility needed and expensive cost. Poor data means it is insufficient to reflect the real world. Otherwise, the quality is formed in the process of production. The data to describe the product and process must be collected along with activities and progress of software development.

- Providing the information for process improvement

The goal of process measurement is to support process management, especially provide the objectivity and reliable information for process improvement. Unfortunately, many software organizations just measure seemingly. They applied some measure to control process, but these measures fail to provide sufficient information to express why the process is not good enough and how to improve. Figure 1 illustrates the process improvement based on measurement.

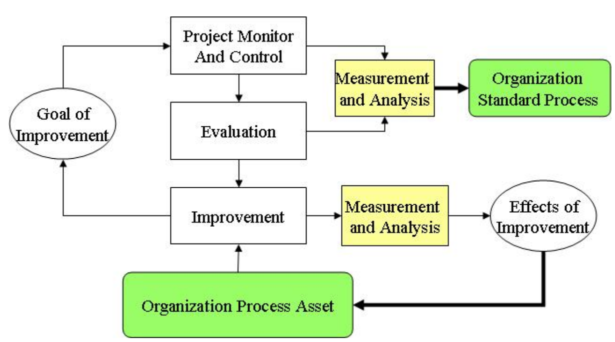

Figure 1. Measurement and improvement

3.2. Active Measurement Model - AMM

Quantitatively process management ensures the organizations archive their business goal with technique and management procedure. On one hand, it makes the production activities in the order and under control. On the other hand, it provides the reliable evidence for customer satisfaction. Measurement is the means of management but not the goal of management. The goal of process measurement is to improve the efficiency of management and evaluate whether the process and business goals are achieved with the qualitative and quantitative analysis, as shown as in figure 2.

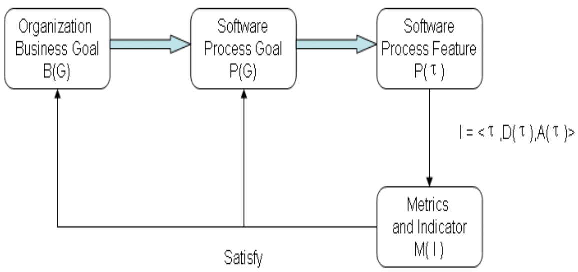

Figure 2. Business goal to measurement

Nowadays, there are many software organizations which can not understand and support their process management quantitatively used data. One of the reasons is that many of them don't know what needs to be measured, and another reason is that they don't know how to measure effectively.

There are many features, which could be measured. But the measurement has cost too. Too much measurements result in cost overrun and unnecessary waste. Poor measurement cannot give the higher level management a clearly vision to understand the significance of measurement. It also causes avoidable failure in decision making, because of the insufficient supported data. Organization must apply appropriated measure based on the process capability and specific project and business environment. The essential principle of measurement is as follow.

- Justify **why** we apply measurement. Any measurement with aimless, excessive and ineffective is insignificant. Measurement should serve to process control, product quality assessment, process improvement and so on.
- Determine **what** needs to be measured. Under the goal of measurement, we need to analyze and extract the entities, which reflect the behavior and status of cared process and process goal. The entities are the objects of measurement.
- Decide **how** to measure these entities. Design and apply the effective measuring algorithm to provide the appropriate information for understanding the object and support process improvement as soon as possible.

To follow these principles, some basic framework should be built to support organization to establish adaptive measurement, reduce the cost of measurement and improve the efficiency of measurement.

To solve this problem, we present the AMM model. It provides an infrastructure to support software organizations establish an adaptive measurement automatically driven by its process goal.

AMM model is based on GQM method[8]. GQM emphasizes the importance of clarifying implicit business objectives and needs for information and translating them into measurable objectives. AMM model fills this gap. In AMM, firstly, we provide an infrastructure to describe and construct the features of process. These features answer the question related the goal and are enclosed by a set of indicators. This infrastructure is opening. The goals, feature, indicators and relative metrics can be added and modified as improvement growth. After organization determine the goal use the any methods, such as GQM, they can use AMM to drive an adequate measurement process and bind to related processes. Then when project is going on, which data and information should be collected, when and how they can be collected will be clear. Based on this data and information, the measurement will be made and the more analysis and evaluation will be done subsequently.

Definitely, feature is the core of AMM. The feature expresses the behavior, status and goals of process. As the result of the changes of business environment, business goals as well as the process improvement, the process goals are also changing. Consequently, the features of software process are changing as well.

At first, we abstract and encapsulate feature by mathematical model, it includes related data and the method that how to get this data. We call it indicator, I, as defined in definition 1.

Definition 1: We define indicator I as:

$$I = \langle \tau, D(\tau), A(\tau) \rangle \quad (1)$$

In definition 1, τ is the feature of process, $D(\tau)$ is the data-set to express the feature τ and $A(\tau)$ is the method of collect data, such as testing, formal verification and review.

Indicator is the basis of measurement, analysis and evaluation. There are close relations among process goal, process feature, and indicator, as shown as in figure 3.

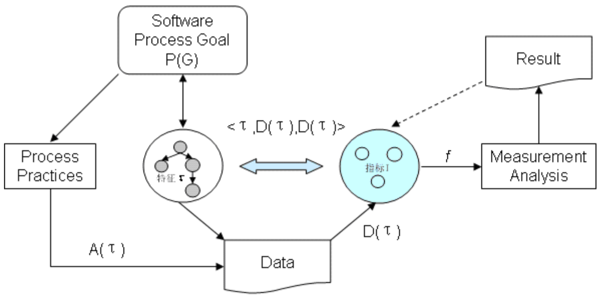

Figure 3. The relation among measurement elements

AMM is a model that organizes these measurement elements properly. The AMM is defined in definition 2.

Definition 2: We define AMM as follow:

$$\text{AMM} = < \text{Tset}, \ \text{Iset}, \text{Metric-set}> \quad (2)$$

In definition 2, Tset = $\{\tau_1, \tau_2, \ldots, \tau_n\}$. It is the global set of process features. Iset = $\{I_1, I_2, \ldots, I_n\}$. It is the global set of process feature indicators.

I = <τ, D(τ), A(τ)> , as definition 1.

The elements in Tset and Iset have M-N multiple map relationship.

Metric-set = $\{\text{Metric}_1, \text{Metric}_2, \ldots\ldots, \text{Metric}_L\}$, It is the global set of metrics. Metric is defined as definition 3.

Definition 3: We define Metric as follow:

$$\text{Metric} = <I, \ f(I),> \quad (3)$$

In definition 3, f(I) is the measure algorithm applied on indicator I.

With AMM, we can derive and export the measurement process automatically based on the process goals which organization focus as input, as shown as in figure 4.

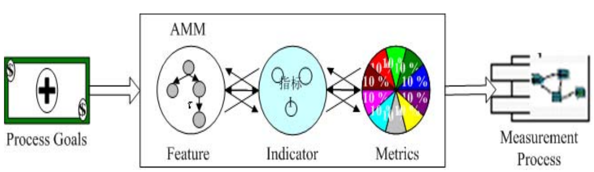

Figure 4. AMM and the measurement process

The benefit of AMM as follow:

- The measurement process exported by AMM is focused on the process goals. Make the objective of measurement legitimate, the cost of measurement affordable, and the effect of measurement significant and appreciated by higher level management.
- With the process goal as input, AMM can derive the measurement entities as illustrated in figure 3, and then decompose to the related feature subset. As definition 1 and 3, we can get the indicators and the metrics applied on these indicators. As yet, AMM address the problem what is measured and how to measure.
- The result of AMM measurement can be used in three ways: 1) control the object process; 2) provide the finding for process improvement; 3) self-optimize and refine the model, include enriching the feature set, refining the indicator which abstractly encapsulate the feature, and optimizing the algorithm which applied on indicator.

4. A Solution Based on AMM

Base on CMM, some methods such as how to define and build an adaptive measurement process and how to establish the appropriate process performance baseline were developed. It provides a solution to measure and improve software process.

4.1. Exporting Measurement Process Based on AMM

The measurement process, which is exported based on AMM, is a series of measure activities. Figure 5 illustrates the relation between activities in measurement process and the other software processes.

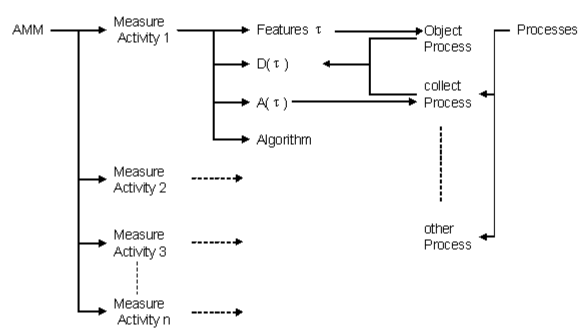

Figure 5. Measure activity with software process

The object process in figure 5 is the other software process, such as engineering process, management process and support process (e.g. configuration, review and so on). As illustrated in figure 4, each measure activity relates to some processes. An measure activities is compose by <τ, D(τ), A(τ), f(I)>. A(τ) is the technique to collect the required data, and is directly depending on some software processes , such as test, review, audit, which is called collect process. This process maybe is different from the process which is measured, which is called object process. For example, we can use review to collect the data of design process. Review and design are two different processes. Both are connected with measure activity. Review is designated by A(τ), Design is derived from the input process goals and identified by process features τ. Sometime both of the collect process and object process are same, such as measure the efficiency of review. Both of the collect process and object process are review. D(τ) is the data of object process and collected by collect process. *f*(I) is the algorithm applied on D(τ). According to the relation, the measurement process exported from AMM can be merged with other software process easily. Consequently, AMM provides a well defined infrastructure to establishing the adaptive measurement and a feasible strategy and facility to collect data. Figure 6 illustrates the method of exporting the measurement process based on AMM.

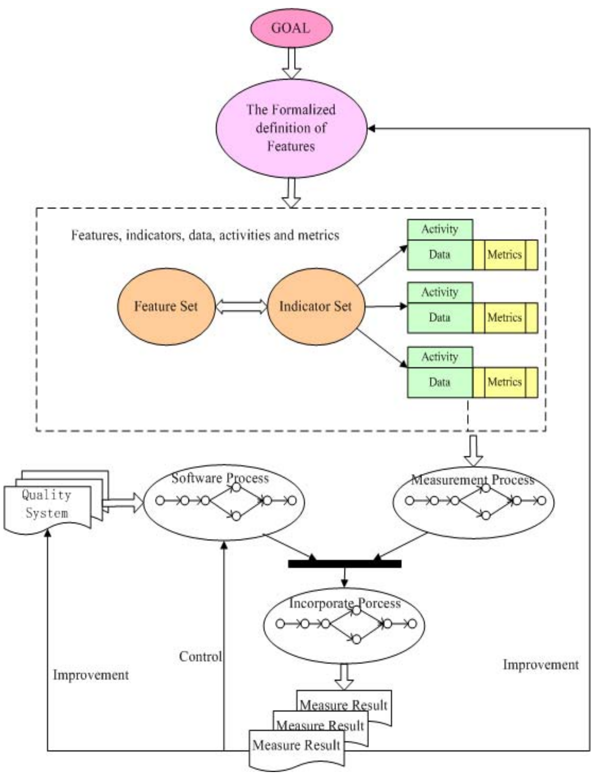

Figure 6. The method of software process measurement based on AMM

As illustrated in figure 6, according to the input of process goals, we can derive the process feature subset which characters the process goals. Then AMM can export the adaptive measurement process. The measurement process merges with other software process to generate an integrated software process. This process is performed through the software life cycle to produce the software products, control and assure the quality of processes and products and support the decision making for process continual improvement.

4.2. Establishing the Appropriate Software Process Performance Baseline

For the higher level process quantitative management, there are two key attributes to evaluate the process. They are performance and capability of process. Performance indicates the expected objective of the process. Capability is the scope of the deviation around the objective. For example, some organization has its defect ratio of design process as 10%, and has the normal deviation is [8%, 12%]. It means that the performance of design process quality measured by defect ratio is 10%, and the corresponding capability is in [8% 12%]. There are many performance baselines for one process depended on different measurement indicators. Also for the example of design process, we can use defect ratio to evaluate the quality status of process, unit effort-cost to evaluate the management status of process, gap between plan and real to evaluate the compliant of estimate, and so on. Each measurement should have a performance baseline for control when the process achieves higher level and quantitative management.

But the performance baseline must be established and refined evolutionally. In fact, organization will have different interest with different process capability and maturity. When they are in lower level, the process goal they focus on is to repeat the successful process and force them evolve to stable. Absolutely, the measurement in this period should consider the distribution of deviation of process, analyze the cause of deviation, take the corrective action to improve it and establish the performance baseline eventually.

Process performance baseline is the foundation of higher level quantitative management. It relies on the history data and the measurement and analysis on these data. Many organizations establish their process performance baseline just for similar organizations or consulter's experiment. It is not recommendatory. Because the improvement must be gradual, evolutional and continual, the next step must build on the prior one. The higher level quantitative management must be based on the lower level process management. So that evolving the baseline by measure and analyzing the related data is a correct and effective method. Some recommended algorithm will be continually discussed in section 5.

5. Practices in China

Institute of Software, Chinese Academy of Sciences (ISCAS) developed some tools to help software organizations manage their software process. SoftPM is a toolkit which integrated these tools to provide a collaborative working environment for senior managers, project managers, developers, SQAs, customers, suppliers and so on.

Platform for Quality Management (PQM) is the most important tool in SoftPM. PQM is based on TQM and compliant with ISO9000/CMM/CMMI. PQM includes four tools to support primary project and process management activities. The four tools can be combined according to increasing process improvement requirement and compliant with the CMM/CMMI. PQM covers most process activities and organizes these activities based on the PDCA cycle as shown in figure 7.

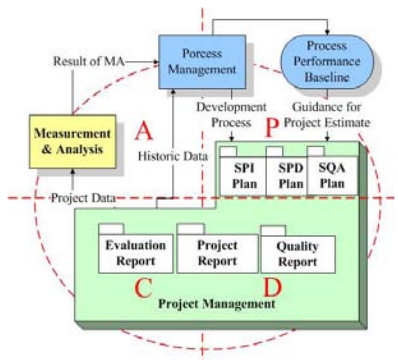

Figure 7. PQM organize activities based on PDCA

PQM provides an opening framework for measurement and analysis based on AMM to support organization to bind the measurement activities with the expect process quantitative management objectives. In PQM a well–defined structure will be integrated into PQM to enable data collection from project management activities easily. When an organization achieves high level maturity or capability, statistical process control (SPC) can be used to measure and control the performance of selected processes stability.

As we discussed above, AMM provide a model for software organization to establish an adaptive measurement process to support reasonable quantitative process management. PQM applied AMM to support organizations to plan and perform appropriate measurement process and merge with other software process to complete the effective management.

Actually, organizations focus on different process goal when they in different capability and maturity level. as shown as in figure 8.

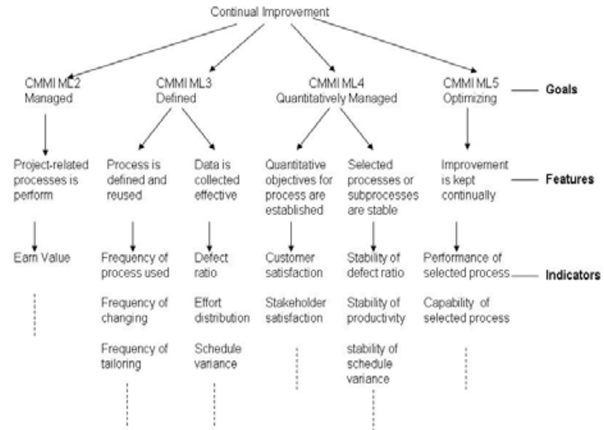

Figure 8. An example of Process goals, features and indicators

As illustrated in figure 8, we construct a relation tree of the goals, features and the related measurement indicators based on CMMI [16] as a sample. The indicators will be different according to the different process goals.

- **Improving in CMMI maturity Level 2**

When software process is at lower capability and maturity level, the process goal is to perform the processes repeatedly. The processes in this period are concentrated on project management. The related measure should support the project management, such as earn value measurement. PQM provides a graphic earn value measure. The figure 9 shows the earn value measure for project cost.

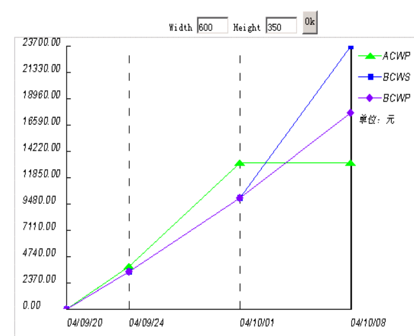

Figure 9. Earn value of project cost

- **Improving to CMMI maturity level 3**

For each process in level 3, the goal is to institutionalize a defined process. There are two features to character the goal. They are 1) process is defined and reused; 2) the data was collected effective and reflect some information for improvement. As illustrated in figure 8, the indicator related to the first feature are the frequency of process used, the frequency of changing, and the frequency of tailoring; those related to the second feature are defect ratio, effort distribution, and schedule variance.

By PQM, when organizations focus on this goal, it can provide the related measures. Figure 10 shows a graphic measure for effort distribution.

Figure 10. The effort distribution

The objective of measurement in this period is to collect the quality data of process and artifact, statistical calculate the quality defect, understand the cost distribution, predict the variance of schedule and identify the possible risk. Along with this, measuring and analyzing the cause result in defects, continually improving them to evolve the process performance baseline are more important.

As we discussed in section 4, the performance baseline must be established evolutionally. In PQM, we provide some algorithm to do it. For example, let we observe the project schedule variance. We use $\mu \pm \sigma$ to measure the distribution of schedule variance.

Measure algorithm is shown in Algorithm 1.

Algorithm 1. Distribution of schedule variance

(1) We construct schedule variance as sample X,
$X = (\text{Real} - \text{planed})/\text{planed} * 100\%$

(2) Calculating μ as average of sample X

(3) Calculating σ as standard deviation of X

We collected 10 samples[1] for the two processes respectively, and then applied algorithm 1. Figure 11 and figure 12 show the measure result respectively.

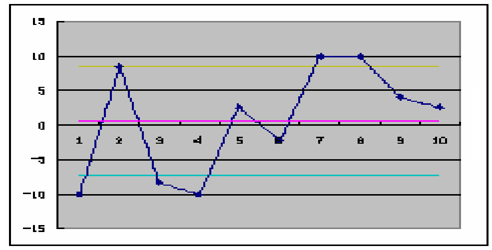

Figure 11. The distribution for schedule variance 0.71 ± 7.97

For figure 11, the average of schedule variable is 0.71, which means the schedule executing is conform in the mass. But the standard deviation is 7.97, it means the capability of the team varies largely. The corrective action should be taken to improve the personal process capability and team process capability. We train the engineers of this process and get the obvious improvement effect.

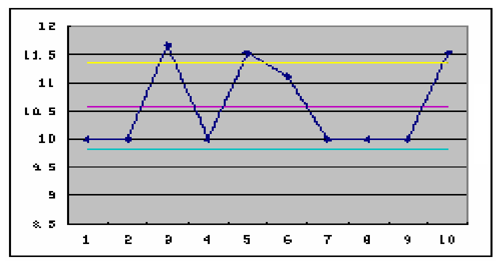

Figure 12. The distribution for schedule variance 10.58 ± 0.76

For figure 12, the average of schedule variable is 10.58, and it means the schedule is always delayed about 10.58%. But the standard deviation is 0.76, it means the capability of the team is stable, and the corrective action should be taken to improve the estimate and planning. After measuring and analyzing the cause of schedule variance and improve continually, we make the schedule variance trend stability, as shown as in figure 13.

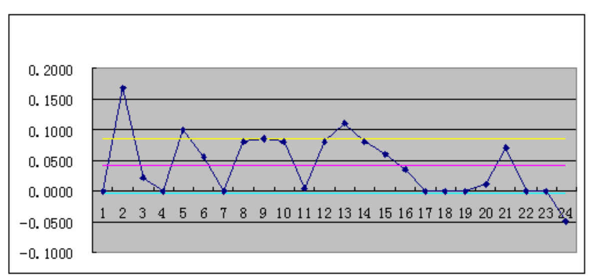

Figure 13. The distribution for schedule variance 0.04 ± 0.05

[1] All sample from ISCAS

As illustrated in figure 13, the performance is also improved along with the establishment of the baseline.

- **Improving to Level 4**

For each process in level 4, the goal is to institutionalize a quantitative process. There are two features to reflect the goal: 1) the quantitative objectives for process are established, 2) the selected processes or sub-processes are stable. As illustrated in figure 8, the indicators related to the first feature are customer satisfaction and stakeholder satisfaction; those related to the second feature are stability of defect ratio, stability of productivity, and stability of schedule variable.

In this period, organizations have established the related process performance baseline. PQM supports organization to select quantitative objective and control them by measures, such as SPC. Figure 14 shows the interface to identify the processes and projects to be control based on the organization's standard process to establish the quantitative objectives.

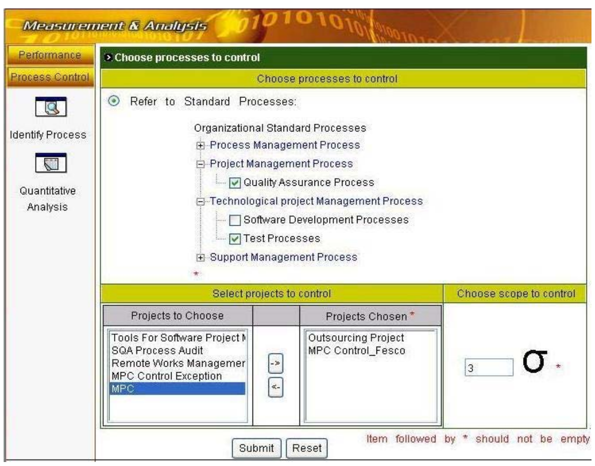

Figure 14. Identify the quantitative objective by PQM

After the quantitative objectives were identified, the measures will be focused on the stable control of these objectives. SPC should be applied as an effective technique. Figure 15 shows the data and performance baseline, which is established as control limit.

Process Control[All Processes, Control Limit: 3 σ]

Project Name	Schedule Variance (%)	Size Variance (%)	Effort(%)								
			Engineering Activity		SPI Activity		Management Activity		QA Activity		Review A...
			Distribution	Variance	Distribution	Variance	Distribution	Variance	Distribution	Variance	Distribution
Project based on web											
MPC Control Exception	-20	212.5	0	-100	--	--	15.15	566.67	--	--	--
MPC Control_Fesco	-33	16.67	65.31	86.67	0	--	1.75	0	0	--	0
MPC	8	10	34.06	20.16	1.94	6.25	0.8	16.67	3.89	183.33	6.86
CL	48.5	5	58.56	-0.37	1.83	6.25	1.03	12.5	5.91	5.77	13.91
UCL	306.52	84.8	88.64	30.91	2.04	39.5	6.54	79	7.59	56.92	26.01
LCL	-209.52	-74.8	28.48	-31.65	1.62	-27	-4.48	-54	4.23	-45.38	1.81
ScheduleControl Chart	SizeControl Chart	EffortControl Chart	DefectControl Chart	ProductivityControl Chart	Delivery Quality...						
Project for quality assurance											
SQA Process Audit	747	--	--	--	100	-92.9	--	--	--	--	--

Figure 15. Data and control limit for identified processes and projects in PQM

Figure 16 illustrates the measure for productivity of identified processes and projects by $\overline{x}-S$ chart.

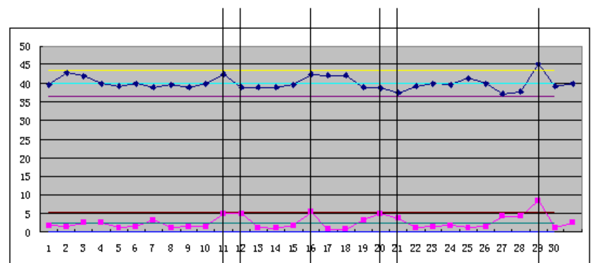

Figure 16. The measure for productivity stability by $\overline{x}-S$

As figure 16 illustrates, we use $\overline{x}-S$ chart to measure and control the productivity of the process. Meanwhile, the important thing is that we must observe the average-chart and deviation-chart together, especially for the special points. We should analyze and understand why some points close or exceed upper or lower limit and some points appear the non-random distribution. For example, the points 11 and 12 in figure 16, the deviation stays high, but the average drops down. It means the capability of engineers in this process is different and the diversity is bigger, that affects the capability of the process and results in dropping down of the productivity of this process. Obviously, measure like this can provide useful information to support process improvement.

- **Improving to Level 5**

The goal in CMMI capability or maturity level 5 is to analyze and identify the root causes of defects and other problem and take action to prevent them from occurring. Some measurement such Six-sigma, Pareto will be applied more in this period. The key consideration of SPI is to improve the performance and capability of process continually. We will not discuss in succession here.

6. Conclusion

Effective measurement can be used to help identifying, analyzing, and solving the problems arising during the development process, to evaluate and improve the capability and maturity of processes, and to predict the quality of process products.

In our solution, measurement can support any kind of software process improvement, not only for higher maturity level organizations, but also for organizations at lower level. AMM provide the model and framework to establish an adaptive measurement process and a well define infrastructure for data collection. Based on AMM, the measurement prcess is deployed driven by the goal of process management in the context of specific project and SPI requirement. It can prevent software organizations from the waste of overrun measurement and the insufficiency of poor measurement. So it will help to make the cost of SPI in a reasonable ratio of return of investment. AMM has been adopted and implemented in SoftPM, it can support software organizations manage their process effectively under the appropriate measurement.

SoftPM has been applied in many areas and organizations in China, such as the national software industry parks, 863[2] software incubators and more than 100 software companies. SoftPM helps them define the standard and project's processes, establish and maintain the process assets library, perform project management and quality assurance activities, collect the data for measurement, measure and evaluate the status of process performing and so on. With SoftPM, 35% effort of SPI can be saved according to the feedback information of customers.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under grant Nos. 60473060, 60273026 as well as the Hi-Tech Research and Development Program (863 Program) of China under grant No. 2004AA112080.

Both authors would like to acknowledgement the efforts of the participants and thank them for taking part in the study.

Reference

[1] Lan Sommerville, *Software Engineering* 7th Edition, Addison Wesley Publishing Company, 2004

[2] Evelyn Stiller, Cathie LeBlanc: *Project-Based Software Engineering*, Addison Wesley Publishing Company, 2002

[3] Sun Jing, *Quality Control for Process with Zero-Uncomfortable*, Tsinghua University Press, Beijing, 2001.3.

[4] http://www.isixsigma.com/st/control_charts

[5] http://www.hq.nasa.gov/office/hqlibrary/ppm/ppm31.htm

[6] Jim Lawler, Barbara Kitchenham, "Measurement Modeling Technology" *IEEE Software*, Vol.20 No.3, pp.68-75, May/June 2003

[7] Stephen H. Kan, *Metrics and Models in Software Quality Engineering* 2nd Edition, Addison Wesley Publishing Company, 2003

[8] Basili VR, Caldiera G, Rombach HD, "The goal question metric approach". *Encyclopedia of Software Engineering* - 2 Volume Set, pp 528-532, John Wiley & Sons, Inc 1994.

[9] Robert E. Park, Wolfhart B. Goethert, William A. Florac, Goal-Driven Software Measurement — A Guidebook, CMU/SEI-96-HB-002, 1996.

[10] John McGarry, David Card, et al, *Practical Software Measurement: Objective Information for Decision Makers*, Addison Wesley Publishing Company, 2001.

[11] William A.Florac, Anita D. Carleton, *Measuring software process-Statistical process control for software process improvement,* Addison-Wesley Publishing Company, 1999.

[12] Nancy Eickelmann, Animesh Anant, "Statistical Process Control: What You Don't Measure Can Hurt You!", *IEEE Software*, Vol.20, No. 2, Mar/Apr 2003 pp. 49-51.

[13] Mark C. Paulk, "Applying SPC to the Personal Software Process SM," 2000, Proceedings of the Tenth International Conference on Software Quality, New Orleans, LA,16-18 October 2000.

[14] Six Sigma Forum Magazine Volume 2 Issue 3,May 2003.

[15] Mala Murugappan, Gargi Keeni , "Blending CMM and Six Sigma to Meet Business Goals", *IEEE Software*, Vol. 20, No. 2, Mar/Apr 2003, pp. 42-48.

[16] Mary B. Chrissis, Mike Konrad, Sandy Shrum, *CMMI: Guide for Process Integration and Product Improvement,* Addison-Wesley Publishing Company, 2004.

[2] 863 Program, Hi-Tech research and development program of China