Frameworks for Validation of Mobile Software Project Performance

Haeng Kon Kim

Abstract— Prediction of various attributes like cycle time, cost, effort, resource requirements, safety and reliability are of paramount importance to project managers. Advance knowledge of such attributes at intermediate stages during the project development not only provides the manager with the knowledge about the status of the project but also negative indicators warn of possible risks so that preventive measures could be initiated to minimize their impacts. Effective estimation of attributes such as the above not only requires a solid technical basis, but also knowledge of various parameters specific to the organization. However, in a realistic situation, much of the information about past projects which should help in the estimation of project parameters is unknown or uncertain.

Mobile software organizations are in need of methods to understand, structure, and improve the data they are collection. In this paper, we present model-based performance prediction at mobile software development time in order to optimize a project of organization and strengthen control of it.

Index Terms—Mobile Service Validation, ROI, Ubiquitous computing, Mobile Software Project and Organizations, Agent Based Architectures, Goal-Question-Metrics

I. Introduction

A software measurement and validation methodology is a systematic method of measuring, assessing, and adjusting the software development process using mobile object components. Within such a systematic approach, software data is collected based on known or anticipated development issues, concerns, questions, or needs. The mobile object components are analyzed with respect to the characteristics of the software development process and products, and used to assess progress, quality, and performance throughout the mobile components development. There are seven key components to an effective measurement methodology:

- **Define** clearly the software development goals and the software measures (data elements) that support insight to the goals.
- **Use** the Goals-Questions-Metrics (GQM) paradigm framework
- **Define** and develop a set of metrics
- **Collect** and validate the data
- **Processing** the software data into graphs and tabular reports (indicators) that support the analysis of issues.
- **Analyzing** the indicators to provide insight into the goals.
- **Using** the results to implement improvements and identify new issues and questions.

Haeng Kon Kim is with a School of Information Technology, Catholic University of Deagu, Korea.(e-mail:hangkon@cu.ac.kr)

Mobile software companies have tried to have an edge on other competitors in securing more extensive market and maximizing financial profits. To this end, they should develop strategies suitable to their vision and implement projects to measure performance attributes. Lynch and Cross[1,2] suggested a performance pyramid for developing strategies needed to accomplish an organization's vision and measuring whether the organization accomplished its vision.

In this paper, we measures quality and delivery attributes for an organization's external effectiveness and, cycle time and waste attributes for its internal efficiency. It also measures process capability and project capability through PCM by completing and analyzing a questionnaire based on GQM (Goal Question Metrics) to find a way to improve the process. Based on the analysis of results, ECM (Earned Value Calculation Model) can be designed to analyze financial performance (earned value) through which effective process improvement plan and project plans suitable to the organization's vision can be developed. In order to predict the project suitable to the organization's vision and optimize the process with analysis results gained through the ECM, this paper also suggests PPM which can predict, based on the organization's project-performing capability, how much manpower, time and capital should be invested to the project and what degree of quality the developed product will have.

II. Related works

2.1 GQM (Goal-Question-Metrics) Process

GQM is a top-down approach to establish a goal-driven measurement system for software development, in that the team starts with organizational goals, defines measurement goals, poses questions to address the goals, and identifies metrics that provide answers to the questions. GQM defines a measurement model on three levels as illustrated as figure 1.

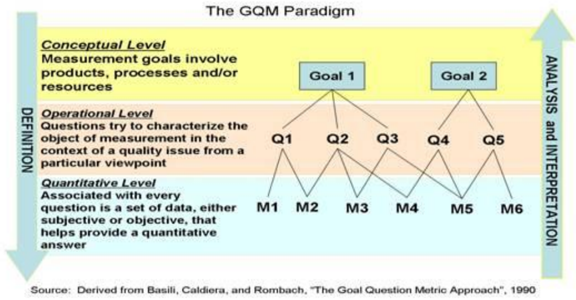

Fig. 1. The GQM paradigm

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The Goal-Question-Metric (GQM) practice focuses on following the GQM paradigm for establishing a metrics program to support software development and maintenance. Organizations typically implement GQM as part of an overall software process improvement initiative, but it is not limited to that role. The basic concepts of GQM can be used anywhere that effective metrics are needed to assess satisfaction of goals. It can even be used by individual members of a project team to focus their work and assess their progress toward achieving their specific goals as figure 2.

G1: To predict the schedule in order to manage it.
Q1: What is the actual vs. expected effort level? (G2) *effort*
Q2: How stable are the requirements? (G4) *req. cnt & modification*

G2: The system must release on time with at least 90% of the errors located and removed.
Q3: What methods are used for testing? (G2) *% tests/method*
Q4: When will 90% of the errors be found and all priority 1 errors closed? (G1, G4) *errors, effort, size*
Q5: What is the discrepancy rate of closure? (G1) *errors, closure status*

G3: Examine the maintainability risk from the point of view of the customer.
Q6: What percentages of modules exceed the guidelines for complexity and size? (G2, G4) *complexity, size*

G4: To evaluate the product in order to improve it.
Q7: What modules are "high risk"? (G2, G3) *complexity, size, errors*

Fig. 2. GQM Basic Process

It is difficult to distinguish between a business goal and a measurement goal; they may not always be mutually exclusive. What is important is that the driving goals originate from the group or organization which is responsible for the broader scope of a software initiative, the business environment in which the initiative occurs, rather than from within a particular project. It is not important whether the business goals are developed under the umbrella of GQM, or as a function of strategic planning. Business goals must exist; they must be identified and be the focus for establishing the measurement goals. Without them, the measurement program has no focus. Without this alignment, it is unlikely that implementing the rest of GQM will have a significant impact. When business goals exist, then multiple projects or sub-groups within an organization have a basis for identifying the measurement goals relating to their role or scope of influence within the organization.

The goals at the top of the GQM tree are the measurement goals that are the outcome of step 1 of the GQM process. They are conceptual, not quantitative. They are quantified by their linkage to questions and metrics as noted in the mapping. Some examples are provided later in this document to illustrate this point. Each GQM goal statement explicitly contains these facets as followings:

· **Object:** The product or process under study; e.g., testing phase or a subsystem of the end product
· **Purpose**: Motivation behind the goal (why); e.g., better understanding, better guidance, control, prediction, improvement
· **Focus**: The quality attribute of the object under study (what); e.g., reliability, effort, error slippage
· **Viewpoint**: Perspective of the goal (who's viewpoint); e.g., project manager, developer, customer, project team
· **Environment**: Context or scope of the measurement program; e.g., project X or division B

2.2 Metrics based Software Quality Management

Software quality management refers to the discipline of ensuring highest quality of software by rendering it free of defects and also by making it meet customer requirements. The overriding goal of software engineering is to produce a high quality system, application, or product. To achieve this goal, software engineers apply effective methods coupled with modern tools within the context of a mature software process. In addition measurement is used to assess the quality of the analysis and design models, source code and the test cases that have been created. Most commonly used metrics to assess software quality is computation of number of defects as a ratio to size of the software. The lesser the defect ratio the quality of software is finer. In addition many organization also focus on indirect metrics defined on quality factors viz. functionality, usability, reliability, performance and supportability (popularly known as FURPS quality factor). Guided decisions on reviews, testing, retesting and release of software are taken on basis of analyzing the above metrics.
Auxiliary measures as defects per review time, number of review defects to testing defects, number of defects reported by customer provide insight into the efficacy of each of the activities implied by the metrics.
Metrics to assess software quality need to be carefully chosen on basis of the nature of software, risk of poor quality as perceived by the customer, customer acceptance criteria etc. In each case it is quintessential to measure and compare the software (documents, program and data) to some datum and arrive at an indication of quality.

2.2 Earned Value Management

Promoting resources-managing ability to effectively invest IT and mobile resources and maximize their effect is becoming an essential field in the IT industry [2] [3]. IT emerged as a key area to reengineer and improve business process, along with using computers. Large corporations including IBM, Ford, and GE are enjoying 80% more effect from business process reengineering using IT than from the improvement just using computers.

The best way to calculate earned value is to accumulate data on the project to be implemented and conform to the following procedures[2].

Calculation Procedures for Earned Value

Step 1: Set objectives suitable to your organization's vision.
Step 2: Complete the questionnaire to find out the attributes helpful to improve the process
Step 3: Develop models and methods to measure attributes for evaluating accomplishment objectives.
Step 4: Identify the alternatives and measures through analysis

III. Design of Mobile PCM (Project Capability Model) Frameworks

In this section, we suggest PCM which can measure an organization's capability through completion and analysis of questionnaire. PCM calculates project-performing capability of an organization with GQM approach regarding each of 4 performance attributes in the performance pyramid (Lynch and Cross). GQM process is a series of procedures as followings:

- **Set** an organization's goals through GQM approach
- **Set** goals of project in each area on mobile applications

- **Make** questions and develop metrics measure accomplishment of the goals using the metrics.

Based on the performance pyramid, GQM quantitative questionnaire is made which enables calculation of an organization’s capability and earned value by using GQM approach. GQM quantitative questionnaire is composed of items with which external effectiveness and internal efficiency of an organization can be measured.

For evaluation of external effectiveness, performance attributes like quality and delivery are analyzed and for evaluation of internal efficiency, cycle time and waste are analyzed. In each area above, project goals are set again, strategies for process improvement are developed through GQM approach and measurement is carried out.

A. *Mobile Object Components Collecting Method*

We use Mobile Object Components from questionnaires on 20 tasks of corporations which are collected from SPICE (SW Process Improvement & Capability determination) assessment for mobile PCM frameworks. Mobile Object Components gained from the answers to the questionnaires are revised according to some defined rules to secure reliability of data on the assumption of T-distribution. Considering possible miscommunication between respondents and questionnaires and problem of representing quantitative data, data out of confidence intervals are revised according to revision rules.

B. *GQM Quantitative Questionnaire from Project Meta Data*

This section proposes GQM quantitative questionnaire made from general meta data. Procedures for making the questionnaire involving three steps as followings;

- ***setting goals:*** It is conceptual step. It consists of elements such as object, purpose viewpoint and focus. In this step, major goal are set.
- ***giving questions :*** It is operational step. In this step, questions are derived from the goals.
- ***gaining metrics:*** It is quantitative step in which proper answers are given to the questions.

Through the three steps, metrics system is made. 20 measurable metrics were made for 8 questions. GQM results gained though three steps are shown as <Table 1>

<Table 1> Metric Results based on GQM.

Goal	Question	Metric
Quality (To improve quality of product up to level of satisfying customers)	Defect density (In the project, how densely defect are found and properly dealt with.)	Defect rate of products
		Defect rate of technical documents.
		Defect rate of codes
		Defect management rate.
	Impact requirement (How much impact customer’s requirement of change has on project?)	Requirement change rate
Delivery (Shorten time needed to deliver product to customers)	Delivery time (Are products delivered to customers on schedule?)	General on-schedule-rate
		On-schedule-rate at planning/analysis stage
		On-schedule-rate at design stage
		On-schedule-rate at implementation stage
		On-schedule-rate at test stage
Cycle time (Shorten total processing time)	Man-Month: Effort distribution (To shorten cycle time, optimal MM is needed. Is MM optimized at each stage?)	Man-Month rate at planning/analysis stage
		Man-Month rate at design stage
		Man-Month rate at implementation stage
		Man-Month rate at test stage
Waste (Reduce waste of available resources when proceeding project)	Productivity (What is current productivity of project?)	Code productivity per person
		Documentation scale per code
		Documentation scale per person
	Reuse (How many codes are reused?)	Code reuse rate
	Rework (How much time is spent for rework?)	Rework hours

Meta data is derived from questions and metrics gained through GQM approach. Each factor of measure method of PCM model needed to calculate metrics is meta data. And measure method of each metric is a capability measure model factor of PCM. Meta data gained like this compose answers to GQM quantitative questionnaire.

C. *Project Capability Measurement Model*

This section proposes PCM to calculate project capability in terms of external effectiveness and internal efficiency of an organization.

Input data of this model is data collected from GQM questionnaire as suggested in 3.2. Factors of PCM to measure a project capability of an organization for 4 goals are shown in <Table2>

<Table 2> Factors of PCM to measure a project capability in terms of external effectiveness of an organization.

Question	Metric	Capability Measure Factor
Defect density (Quality)	Defect rate of products	Total number of defects
	Defect rate of technical documents.	(Requirement specification + design specification) number of defects and total number of pages of outcome
	Defect rate of codes	Number of code defects, total SLOC
	Defect management rate.	Number of complete correcting defects.
Impact requirement (Quality)	Requirement change rate	Number of requirement change, total number of requirement
Delivery time (Delivery)	General on-schedule-rate	Total delivery days, planned delivery days
	On-schedule-rate at planning/analysis stage	Actual delivery days, planned delivery days at planning/analysis stage
	On-schedule-rate at design stage	Actual delivery days, planned delivery days at design stage
	On-schedule-rate at implementation stage	Actual delivery days, planned delivery days at implementation stage
	On-schedule-rate at test stage	Actual delivery days, planned delivery days at test stage
Man-Month : Effort distribution (Cycle time)	Man-Month rate at planning/analysis stage	MM at planning/analysis stage
	Man-Month rate at design stage	MM at design stage
	Man-Month rate at implementation stage	MM rate at implementation stage
	Man-Month rate at test stage	MM at test stage
Man-Month : Effort correspondence (Cycle time)	General effort correspondence	Actual MM and planed MM, general MM
	Effort correspondence rate at planning/analysis stage	Actual MM and planed MM at planning/analysis stage
	Effort correspondence rate at design stage	Actual MM and planed MM at design stage
	Effort correspondence rate at implementation stage	Actual MM and planed MM at implementation stage
	Effort correspondence rate at test stage	Actual MM and planed MM at test stage
Productivity (Waste)	Code productivity per person	Effort SLOC
	Documentation scale per code	Number of document’s page
	Documentation scale per person	Total distributed effort
Reuse (Waste)	Code reuse rate	Number of reused SLOC
Rework (Waste)	Actual output to planned output ratio	Actual SLOC, Planned SLOC
	Rework rate cause by defects	Rework hours, total spent hours

Calculation forms to measure a project capability of an organization in terms of external effectiveness and internal efficiency for 4goals are shown in <Table3>

<Table 3> Calculation forms to measure a project capability of an organization in terms of external effectiveness and internal efficiency.

External effectiveness PCM(qd) = (PCM(q)+PCM(d)/2	PCM(q): quality effectiveness score	$\sum((100 - each\,defect\,rate) + defect\,management\,rate)/4$
	PCM(d): schedule effectiveness score	$\sum(100 - on\,schedule\,rate\,at\,each\,stage)/5$
Internal efficiency PCM(c,w)=(PCM(c))+PCM(w) /2	PCM(c): effort efficiency score	$\sum(effort\,correspndence\,rate\,at\,each\,stage)/4$
	PCM(w): resource efficiency score	$\sum(all\;\;factors) - (2 \times (100 - [rework\;per\,code]) / 6$

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

By calculating PCM for external effectiveness (q, d, quality and delivery) and PCM for internal efficiency(c, w, cycle and waste), benchmarking other competitors becomes possible. In addition, it also shows the degree of external effectiveness improvement. But effective process strategies cannot be developed with organization's capability alone. For example, in the case that PCM(q):85>PCM(d) : 75, no matter what you select out of two strategies(to heighten quality capability from 85 to 90 or to heighten time capability from 75 to 80) in order to increase external effectiveness, PCM result is same because both strategies is to increase 5.

But if you can get 1000 won from increased quality of 5 and get 500 won from increased delivery time of 5, it is not effective to increase external effectiveness by simply improving factors with lower figure. That is because it excludes the cost/profit the organization can get.

Therefore, in order to decide which capability should be strengthened by comparing quality capability and delivery capability of PCM, earned values should be calculated in fields of quality and delivery.

D. Design of ECM (Earned Value Calculation Model) for Mobile

This section suggests ECM with which project's cost for external effectiveness can be calculated, using the project capability results gained from PCM. Calculation procedure of ECM is as follows and the composition is shown in <Figure 3>

Calculation procedures of ECM

1. Calculate project capability with PCM.
2. Measure cost factors of quality and delivery belonging to external effectiveness of organization.
3. Design and calculate ECM for external effectiveness by using the measured values above.
4. Calculate expected cost for project improvement through analysis of the calculated results above.

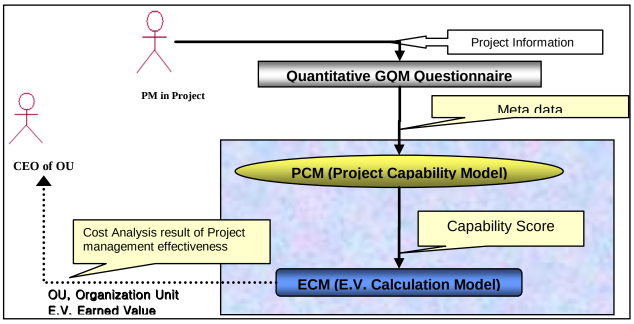

Fig. 3. Composition of ECM

<Table 4> provides ECM to analyze project cost, using PCM value of organization, with project input from GQM quantitative questionnaire.

<Table 4> ECM for project cost analysis

Goal	Earned Value	ECM Calculation model
Quality	Total cost for defect management	Rework hours/ 184 x average monthly salary
	Cost per defect	Total cost for defect management/ total number of defects
	Sigma level of current process	NORMSNIV(1-code defect rate/100)+1
	Number of defects which should be found to heighten sigma level of current process by one sigma.	Total number of SLLOC x code defect rate - (total number of SLOC(code defect rate-(1-NORMSDIST(sigma level of current process-1.5+1)))
	Cost for managing the defects found for 1 sigma level-up	Number of defects which should be found to heighten sigma level of current process by one sigma. Cost per defect
Gains from improvement	Cost/profit gained when a person to manage 100% increased number of defects	Rework hours/368 average monthly salary
Delivery	Total project cost	Project cost per day x total actual project days + total defect management cost
	Project cost per day	Average salary/23((total MM 23) /total actual project day)
	Total loss caused by difference between plan and actual result	Sum of loss at each stage + total schedule difference project cost per day
	Loss caused by difference between plan and actual result at planning/analysis stage	Schedule difference at planning/analysis stage * Project cost per day
	Loss caused by difference between plan and actual result at design stage	Schedule difference at design stage * Project cost per day
	Loss caused by difference between plan and actual result at implementation stage	Schedule difference at implementation stage * Project cost per day
	Loss caused by difference between plan and actual result at test stage	Schedule difference at test stage * Project cost per day
Gains from improvement	Gains caused when project is implemented on schedule.	Total project cost – total loss caused schedule difference
Cycle Time	Project cost at planning/analysis stage	(MM and average salary at planning/analysis stage)+ cost for defect management at planning/analysis stage
	Project cost at design stage	(MM and average salary at design stage)+ cost for defect management at design stage
	Project cost at implementation stage	(MM and average salary at implementation stage)+ cost for defect management at implementation stage
	Project cost at test stage	(MM and average salary at test stage)+ cost for defect management at test stage
	Gain/loss cause by difference between plan and performance at planning/analysis stage	(Planned MM – actual MM at planning/analysis stage) average salary
	Gain/loss cause by difference between plan and performance at design stage	(Planned MM – actual MM at design stage) * average salary
	Gain/loss cause by difference between plan and performance at implementation stage	(Planned MM – actual MM at implementation stage) * average salary
	Gain/loss cause by difference between plan and performance at test stage	(Planned MM – actual MM at test stage) * average salary
	Total gain/loss cause by difference between plan and performance	(Total planned MM – actual MM) * average salary
Gains from improvement	Gains caused by improved productivity	Total MM: total cost = 1MM: x (x= cost when reducing number of people by one) 100:272 = 1:Y (1% is for how may people?)
Waste	Earned value from reuse	Total SLOC number-Effort SLOC number)/400* (Total project days/184) * * average salary
	Cost caused by not doing reuse	Earned value from reuse + total project cost
	Earned value from reuse of 1SLOC	Total earned value from reuse
Gains from improvement	Gains caused by distributing manpower as planned	Gains when project is implemented 100% as planned/capability to be increased for hitting the target of 100%

By using PCM results and expected gains from improvement, effective strategies for external effectiveness of organization can be developed.

All that PCM and ECM can calculate is only current project capability and expected gains from improvement.

For example, when a strategy to increase delivery time by 5 is selected because the gains from shortened delivery time by 5 is larger than gains form improved quality, you cannot expect how much cost will be spent or how much days will be needed for the project.

Therefore, this paper also suggests PPM(Project Predict Model) which can calculate schedule and manpower for future project by analyzing PCM and ECM results.

E. PPM (Project Predict Model) for Mobile Applications

This section suggests PPM which can predict schedule, cost, manpower, and quality of future project by using PCM results.

This model is designed to predict schedule, cost, and manpower when an organization plans a new project by using PPM and PCM results and earned value from ECM. Calculation Procedures are as followings;

1. Calculate project capability through designed PCM model.
2. Calculate expected gains from improved project by using ECM for external effectiveness.
3. Design a project predict model for schedule, cost, manpower, based on PCM and ECM results
4. Calculate project predict value for effective strategies to improve quality and delivery, based on experience of individual organization

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

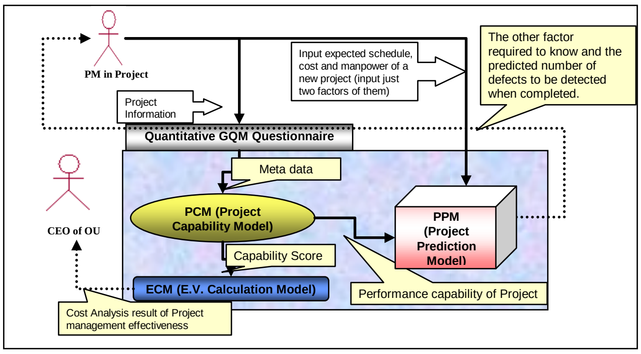

Fig. 3. Composition of PPM

Figure 3 shows the composition of PPM as we describe it. We assume for the PPM as followings;

Assumptions of PPM

Assumption 1: a new project is planned through PPM
Assumption 2: The organization to implement the new project already answered GQM quantitative questionnaire and thus has PCM and ECM results.
Assumption 3: The new project belongs to the same team within the organization as the project for which GQM questionnaire was answered.
Assumption 4: Scale of the project is predicted. (expected SLOC)
Assumption 5. 2 out of 3 factors(schedule, cost, manpower) has been determined.

(ref)) implement the project whose scale is 17, 721 SLOC with 50M/M within 6 months.(manpower and schedule has been determined) How many days will be spent? (Cost has not been determined)

On the assumption like this, PPM (project predict model) can be used in following 3 cases. It is assumed that SLOC has been determined for all cases.

Cases in which PPM can be used

Case 1 : total expected cost and man-month have been known, but total expected schedule has not been known.
Case 2 : total expected schedule and man-month have been known, but total expected cost has now been known.
Case 3 : total expected schedule and cost have been known, but total expected man-month has now been known.

IV. Practical Use with PCM and ECM for Verification of Reliability for PPM

A. *Case study of PCM and ECM*

This section verifies reliability of PCM and ECM through case study using data on GQM quantitative questionnaires collected from SPICE assessments during 2010 to 2011 periods from small and medium mobile application development companies in Korea. This paper uses data of three organizations as project mobile components as in the <Table 5> provides specifics of each of the three organizations

<Table 5> Specifics of each organization

	A company	B company	C company
Nature of task	Mobile App. Development task	Mobile Commercialization task	Mobile Development task
Existence of mother task	Exist	Exist	Not exist
Project Cost	About 50 million won	About 500 million won	About 100 million won
Motive of project	For commission from other organization.	For commission from other organization.	For internal study of the organization
Project domain	Mobile	Software and Computer	Mobile Multimedia

B. *Verification of reliability using PCM and ECM cases*

For case study, this paper used data which were collected from GQM questionnaires of three companies by using Excel as an automation tool. Reliability verification results of 3 companies through case studies are as follows.
<Table 6> and <Table 7> are summarized results of case studies.

<Table 6> Analysis results of case studies of PCM

	PCM capability score	Capability score of each goal	
A company	74.36 PCM(QD)	82.09	66.64
		Quality > Delivery	
	45.50 PCM(CW)	54	37
		Cycle Time > Waste	
B company	80.16 PCM(QD)	88.97	71.34
		Quality > Delivery	
	58.18 PCM(CW)	69.12	47.23
		Cycle Time > Waste	
C company	86.66 PCM(QD)	87.42	85.9
		Quality > Delivery	
	73.03 PCM(CW)	59.49	86.57
		Cycle Time < Waste	

<Table 7> Analysis results of case studies of ECM

	Gains from Quality improvement	Gains from Delivery improvement	Gains from Cycle Time improvement	Gains from Waste improvement
A company	1,027,174 won	55,897,345 won	630,385 won	928,543 won
	Quality < Delivery		Cycle Time < Waste	
B company	4,500,152 won	398,897,460 won	1,638,600 won	5,522,763 won
	Quality > Delivery		Cycle Time > Waste	
C company	1,250,000 won	17,537,021 won	212,286 won	1,029,000 won
	Quality < Delivery		Cycle Time < Waste	

A company should develop a strategy to reduce delivery time. As a result of checking ECM (q,d)results to confirm whether the strategy is effective, it was found that gains from delivery improvement is larger than gains from improvement in other fields. Therefore, if there is 100 % improvement in the field of delivery, 55,897,345 won can be gained. In addition, PPM also showed reliable results. Through these case studies, two effects can be expected. First, project capability can be predicted based on performance attributes before starting project. Second, Earned value (E.V.)'s reliability can be verified by comparing the E.V. calculated from ECM and values for SPI (Software Process Improvement) effects obtained from answers to GQM and finding the cause of difference through difference analysis.

As shown in Table 10, the results measured through ECM model and SPI effects felt by the developers in the organization are the same. The value flow of efficiency and effectiveness is as shown in <Figure 4>.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

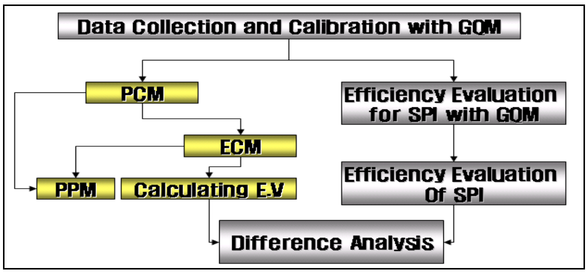

Fig. 4 Value Flow of efficiency and effectiveness

V. CONCLUSION AND FUTURE WORKS

Looking metrics as closing interacting and interlinked with other project management discipline is the key for successful and purposeful implementation of metrics in projects. It should be borne in mind that metrics enables us to assist in planning, tracking and control of software project and also in assessing quality of the software. Proper goal setting combined with proper identification of correct metrics helps projects to use metrics as a good decision support system to guide in project decision-making. Needless to mention that once this value of metrics is realized, it is never a separate activity in project manager's agenda and the same becomes a useful routine in effective project management.

In this paper, case studies were implemented based on 3 mobile components collections. Therefore, the reliability analysis was carried out on the assumption of T-distribution for mobile applications. If number of data collection exceeds 30, data reliability can be analyzed on assumption of F-distribution because data of all models that can be analyzed by F-distribution shows normal distribution. In this case, 4 performance attributes (quality, delivery, cycle time, and waste) are represented by using each typical performance variables. By using GQM based questionnaire we can analysis and define quality and delivery for the external effectiveness, and cycle time and waste for the internal efficiency to performance pyramid.

Acknowledgments. This work was supported by the Korea National Research Foundation (NRF) granted by the Korea Government (scientist of Regional University No.2012-0004489)

REFERENCES

[1] Geovorgyan, R., &Arzumanyan, K. (2008).Self-Assessment as a Tool for QA, *SEUA MMS Department Case Study, International Quality Assurance: Experience, Problems, Trends, Yerevan State University*, Armenia, 23-24 September, Proceedings, pp. 83-87

[2] Marukhyan V., Gevorgyan R., Mamyan S., The self-assessment results of the SEUA academic units. *Proceedings of International Conference (Tempus JEP-27178-2006) on Quality Enhancement: Experience, Challenges and Perspectives for Armenian Higher Education*, October 5th and 6th, Yerevan, pp.83-87, 2009

[3] Siakas K., Gevorgyan R. &Georgiadou E. (2010). The Education Quality Enhancement Strategy Implementation at the State Engineering University of Armenia, in J. Uhomoibhi, M. Ross and G. Staples (eds). *e-Learning and Social Responsibility*, Proceedings of the 15th INternational Conference on Software Process Improvement - Research into Education and Training, (INSPIRE 2010), 29 – 31 March at British Computer Society, London, UK, pp. 141-151

[4] Kyung-whan Lee, "Modeling for High Depending Computing", The fifth Korea Information Science Society's Software Engineering Association,Feb.20. 2003

[5] Boehm, C. Abts, A.W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Riefer, and B. Steece, "Software Cost Estimation with COCOMO II", Prentice Hall, 2000.

[6] Steece, B., Chulani, S., and Boehm, B., "Determining Software Quality Using COQUALMO," in Case Studies in Reliability and Maintenance, W. Blischke and D. Murthy, Eds.: Wiley, 2002

[7] ISO/IEC JTC1/SC7 15504: Information Technology-Software Process Assessment, ISO TR, ver.3.3, 1998

[8] KSPICE (Korea Association of Software process Assessors), SPICE Assessment Report http://kaspa.org, 2002～2003

[9] Frank Van Latum, Rini Van Soligen, "Adopting GQM-Based Measurement in an industrial Environment", 1998, IEEE software

[10] Young-jun Yoon, "Easy 6 sigma- Renovation of management quality", Future management technique consulting, 1998.

[11] Tim Kasse, "Action Focused Assessment for software process improvement", Artech House, 2002.

[12] Williams A. Florac, Anita D. Carleton, "Measuring the software process", 1999, SEI Series, Addison Wesley.

[13] Bohem, "Software Cost Estimation-COCOMOII , PH, 2000, pp34-40.

[14] Tom Gilb, "Software Inspection", Addison-Wesley, 2001.

[15] Ki-Won Song, "Research about confidence verification of KPA question item through SEI Maturity Questionnaire's calibration and SPICE Level metathesis modeling", SERA03, San Francisco, 2003.06