Quantitative Development Management Multi-vendor Management Process Reform

Technology Reports

Implementing Standardized Quantitative Management Processes for Development of Smartphone Applications

Communication Device Development Department

Takashi Tosaki Yoshihiro Yamada Hiroyuki Hattori

In the initial period of development for smartphone applications at NTT DOCOMO, the need to rapidly implement a wide variety of applications and the growing number of vendors led to numerous inconsistencies in development management at the various workplaces. Meanwhile, knowledge of development management processes tended to become limited to certain individuals. To deal with this situation, we created standard forms for vendors to report the development management status. We also created standard forms for application development groups at NTT DOCOMO to report the quality upon completion of development. We also established and implemented in-house standardized quantitative management processes for development. This article describes our initiatives to reform the above processes.

1. Introduction

The rapid shift from feature phones to smartphones in recent years has required NTT DOCOMO to quickly develop and deploy a wide range of applications to meet diverse demands for quality, cost, and delivery times. The vendors conventionally contracted for development (hereinafter referred to as "vendors") have been unable to meet these demands or else lacked resources, so NTT DOCOMO has sharply increased its adoption of vendors. However, the need to prioritize quick service launches and the dependence on vendors and in-house application development groups reduced consistency and left only a limited number of people able to handle each development. These inconsistencies in the degree of detail of development management information caused the content of the quality reports used in decision-making meetings for commercial release to differ widely, and the high number of question-and-answer sessions required to reach decisions on quality meant that release judgments took time. Furthermore, there was a high possibility that quality checks might be overlooked under these circumstances.

To address this problem, the process reform team created standardized forms for the vendor development status reports and quality reports required for commercial release decisions, which were then implemented in the development workplace in various ways. We also engaged in ongoing educational activities to embed the quantitative development management*1 processes used

^{*1} Quantitative development management: A development management method based on objective data and facts. Whether quantitative development management is implemented has a big influence on the success or failure of a development project.

Copies of articles may be reproduced only for personal, noncommercial use, provided that the name NTT DOCOMO Technical Journal, the name(s) of the author(s), the title and date of the article appear in the copies.

with these forms into the organizations. This article describes our process reform initiatives to date.

2. Development Processes and Issues

2.1 Roles in Development and Organizational Structure Related to Application Development

Figure 1 gives an overview of roles in development and organizational structures related to application development. The major in-house organizations comprise application development groups, a Project Management Office (PMO)*² overseeing developments, and maintenance management groups. Application development groups create definitions of requirements^{*3} based on the needs of the department in charge of the service and contract with the vendor to develop software (from basic design through to comprehensive testing). Once vendor development and in-house acceptance testing is complete, the organization director, PMO supervisor, and maintenance management supervisor decide whether the software is ready for commercial release.

2.2 Issues to be Resolved

At (1) in Fig. 1, development management between application development groups and vendors entailed regular meetings to share information regarding development status (hereinafter referred to as "information-sharing meetings"). Despite these meetings, inconsistency in development management methods remained, with only a few people capable of managing each development, which meant that the development status information from vendors to NTT DOCOMO was a mixture of quantitative/objective and qualitative/subjective reporting and that large variations occurred in terms of the level of detail. Depending on the development workplace, this could cause major issues, such as NTT DOCOMO being unable to properly grasp the state of development at the vendor or potential risks, leading to concerns over whether suitable countermeasures or risk preventions were being enacted.

Moreover, differences in the level of



- *2 **PMO:** A division dedicated to overseeing, administering and supporting management of individual projects in an organization.
- *3 Definitions of requirements: A document that contains an overview of the functions and specifications that the customer demands etc. A sourcebook for development – products delivered for subsequent processes must meet all of the requirements in the definitions of requirements.

detail presented by vendors led to variable content of quality reporting in release decision processes at (2) in Fig. 1. This meant those involved in making release decisions had to interpret reports appropriately to make objective quality judgments and supplement insufficient information through question-and-answer sessions with reporters. Both issues were serious.

3. Process Reform Activities

Following in-house interviews, the root causes of the variable reporting with development management processes and release decisions were deemed to stem from inconsistencies in vendor development status and quality reports when the time came to make release decisions and a lack of means to develop knowledge and expertise about quantitative management processes for development within an organization. To counter this, we have disseminated quantitative management methods for development throughout the organization using standardized forms that report development status from vendors to application development groups and quality at times of decision-making for commercial release.

3.1 Creating Standardized Forms and Initial Deployment

- Creating Development Status Reports
 In creating the development status
- *4 Metrics: Measurement methods and scales defined for quantitatively determining the quality of software and development processes. These include scale of a development, and the time and man-hours required for its processes etc.
- *5 Microsoft[®] Excel[®]: A trademark or registered trademark of Microsoft Corporation in the USA and other countries.

report form, considering the need for ten or more diverse vendors to use the form quickly and to keep management costs to a minimum, we determined the minimum number of metrics*⁴ (scale, progress, estimated quality, and actual results) required for the report and adopted the widely-used Microsoft[®] Excel^{®*5} file format.

Vendors present development status reports to NTT DOCOMO application development groups at weekly information-sharing meetings so that development status can be shared with transparency. Then, using the details in the report, both parties discuss measures against actual and potential risks. Making the most of Excel functionality, alerts are displayed automatically with red or vellow hatching in areas of concern so that both vendors and NTT DOCOMO can be sure that all risks are checked. Also, so that vendors do not get confused or overlook details when filling in the form, areas with input rules and conditional expressions or those requiring entry are shown with hatching.

The form is divided into four sheets – (1) summary, (2) development functions and scale management, (3) expected and actual progress management*⁶, and (4) expected and actual quality management - depending on objectives. **Figure 2** illustrates the summary and the expected and actual quality management sections of the development status report (for the others, refer to [1]).

- *6 Expected and actual progress management: Managing the differences between plans and actual performance.
- *7 Management unit: Units in software structure for measuring good and bad quality and taking actions as necessary to improve quality.
- *8 Review density: The amount of review per the scale of the product under review. A metric

- (1) To carry out a limited number of information-sharing meetings efficiently, we have designed the form so that development status and risk can be quickly grasped and shared with a brief look at the summary section, which includes a digest of data on the scale, progress, and quality of the development; current issues; and vendor actions (Fig. 2(a)).
- (2) The development functions and scale section is used for managing the function list, which is split into appropriate management units^{*7}, and the development scale for the management units during development planning and at end of the process. Development risk can be assessed by checking transitions in scale.
- (3) The expected and actual progress management section applies for each management unit. It enables sharing of delayed functions and development processes, their causes, and the period required for countermeasures. Progress is visualized so that risks can be assessed.
- (4) The expected and actual quality management section is used for managing quality index targets and performance for review density*⁸, test density*⁹ and review identification density*¹⁰, bug

that indicates the sufficiency of the amount of review. Here, a review refers reading intermediate developmental results (design documents and source code) by a number of people, including the authors. Extracting deficiencies and issues from a number of perspectives in this way raises the quality of products.





density^{*11}, and test status in processes from binding tests onward for each development process and management unit. Quality risks can be assessed and handled at an early stage by sharing analysis results and actions to rectify deviations from quality index targets. A visual representation of the zone analysis^{*12} results (Fig. 2 (b)) is provided as a mechanism for preventing both vendors and NTT DOCOMO from overlooking quality risk actions. Refer to [2] for the creation of the expected and actual quality management section.

Taking into account management costs and the degree of detail of development management, we have created an abridged version of the development status report form for small-scale development projects that incur fewer risks. A threshold of development scale and cost is exploited to select whether to use the normal or abridged version of the development status report. In general, the more detailed the development status report, the easier it is to quickly uncover and respond to risks as management costs increase. Hence, applying the abridged version of the development

- *9 **Test density:** The number of test items per the scale of the program development. A metric that indicates the depth of testing for each process. Here, a test means actually running created programs on a computer, and confirming that expected results can be obtained.
- *10 Review identification density: The number of review identifications for the scale of the

product under review. A metric used for the degree of extraction of issue identified from reviews, used to judge the quality of the product under review.

*11 Bug density: The number of bugs detected for the program development scale. A metric used for the degree of extraction of bugs in processes, used to strengthen tests and judge whether retesting is required.

*12 Zone analysis: A method of analysis in which given analysis themes are split up into zones focusing on certain characteristics. Creating zones enables more detailed responses compared to overall actions.





status report to projects with low development risks puts the focus on costcutting rather than detailed reporting, and by simplifying management processes and using simpler quality index values determined by NTT DOCOMO for these low risk projects, all development functions, scale, progress, and quality information can be handled with brevity on one sheet. For details of the structure, refer to [1].

2) Creating Quality Reports

In creating quality reports as forms to use for judging commercial release, we have adopted the Microsoft Excel file format for its ease of data collection and processing and its affinity with the development status report. We also considered the minimum amount of data needed to enable those making release decisions to judge quality while avoiding adding to the workload of the application development groups unnecessarily. The quality report source data is limited to the development status report used by the application development group during development plus acceptance test data regarding the number of test items and failures. Additional information gathering is not required. Furthermore, with performance data contained in the development status report, decision-makers can make comprehensive judgments using information including quality in upstream processes.

Figure 3 shows the quality report form in five sections: (1) an application overview including the provided application name and development scale; (2) a summary of quality data in development processes (omitted with abridged development status report) based on development status; (3) a reliability growth curve*¹³ created after comprehensive vendor testing and in-house acceptance testing detailing number of tests and failures; (4) the vendor-DOCOMO test/ failure ratio; and the qualitative perspective. We have maximized use of the Excel format to devise a form that is easy for decision-makers to read and reporters to complete. For example, the form includes input checks using input rules and conditional expression functions. It also automatically converts numeric data input to high-visibility graphs and visualizes quality risks by displaying high-risk quality data with yellow or red hatching to alert the reader. 3) Initial Initiatives to Deploy This Form

We have taken both a top-down and bottom-up approach to the implementation of this form (see reference [3] below). First, to deploy the form quickly in all development workplaces simultaneously we used a mandatory top-down approach. For the bottom-up approach, process reform team members participate in information-sharing meetings be-



*13 Reliability growth curve: A graph used for confirming project progress and quality status etc. The horizontal axis depicts dates, test time and number of test cases, while the vertical axis depicts the cumulative number of bugs discovered. These are often drawn as an S-shaped growth curve. tween application development groups and vendors that involve explanations of development backgrounds and form completion methods together with identification and checking of vendor completion methods and report details, including examples to be passed on to application developers. Enabling direct workplace support that includes both application development groups and vendors greatly reduces the feeling of being compelled to participate that stems from the top-down approach, and this significantly reduces resistance to incorporating the reforms. The combination of these approaches enabled us to deploy the forms in all development workplaces within approximately six months.

3.2 Embedding Quantitative Development Management

After implementation of the form was complete, we engaged in the three actions below by focusing on firmly establishing the quantitative development management methods and undertaking ongoing improvements.

1) Further Upgrading Forms and Other Documentation

We continued to improve the forms so that vendors can complete forms easily and progress and quality can be grasped regardless of development group management skills. To accomplish this, we interview application development groups and vendors, both officially and unofficially, and take proactive steps to hear opinions and suggestions from the development workplace so as to foster awareness of participation in process reform activities, taking into account ease of acceptance of any changes made. Furthermore, we provide comprehensive reference documentation to enable effective use of the forms. For vendors, we provide guidelines for completing development status reports, and for application development groups, we provide a know-how reference for the practice of quantitative development management including information on how to interpret development status and quality reports.

We have also enabled more objective quality judgments of projects by accumulating development performance data and using statistical data on past performance for quality alerts on the quality report. For example, in the reliability growth curve at (3) in Fig. 3, for each period, bug density in the 75 to 85 percentile*¹⁴ of past statistical data is shown as a yellow alert while bug density in the 85 to 100 percentile of past statistical data is shown as a red alert, indicating high-level quality risks.

 Implementation of Support Tools for Application Development Management After this form was launched and spread in the development workplace, the demands of workplaces that wanted to manage development with greater accuracy and efficiency and the load on

process reform team members to collect and aggregate main development data for all projects (including reliability, productivity, scale, and man-hours) increased. To counter this, we created support tools comprising two functions: (1) development management support for application development groups and (2) data aggregation support for process reform teams. Figure 4 shows an overview of these application development management support tools. The tools use Agile development*15 so that functions can be added quickly and flexibly while measuring the progress of process reforms.

- (1) Development management support functions are aimed at assisting application development groups and contain functions to solve development management issues that come to light through interviews and observations in the development workplace. Specifically, the functions analyze development plans based on the expertise of the process reform team and in-house experts while extracting factors of quality and slow-progress risks; clarify differences from recent development status reports for greater efficiency in information-sharing meetings; automatically create quality reports based on development status reports; analyze information on failures that oc-
- *14 Percentile: Units created by rearranging the distribution (variation) of measured values from small to large and displaying them as percentages so that measurement can be made from a value position on a percentage scale. For example, the 65th percentile indicates a value positioned at 65% counting up from the minimum value.
- *15 Agile development: A development methodology based on the Agile development declaration, a generic name for light development methods for rapid and adaptive software development.





cur with acceptance testing; and provide development feedback from project development data analysis and other areas. These functions have made development management more efficient, promoted a more voluntary approach, and embedded quantitative development management in the workplace. Tool usage logs are also automatically collected from application development groups and log analysis mechanisms are included for easy verification of tool effectiveness and workplace diffusion. This enables individual briefings in workplaces where tool diffusion is low and quick decision-making to improve underused functions.

(2) The data aggregation support functions are used to collect, accumulate, sort, and output data in development status reports. These functions enable quick, easy collection and accumulation of data for statistical purposes from a variety of development workplaces.

3) Full Human Support

Now that the support enabled through improved tools and documentation and an accumulation of know-how and experience in the workplace has accelerated the shift to proactive, autonomous quantitative development management, the process reform teams have gradually begun to reduce their participation in development workplaces. Nevertheless, whenever new versions of the forms are released or new functions added to the development management support tools, briefings are held for the entire department to educate people about the need for quantitative development management, introduce specific expertise for analyzing development risks using the development status and quality reports, and share both best and worst practices regarding development management.

In this way, while intentionally reducing the need for human support in the development workplace, we remain aware that process reform essentially relies on people. To this end, we still provide full support in the development workplace in high-risk situations, such as projects with high development risks from the perspective of degree of difficulty and scale or phases in which the personnel have changed.

4. Results of Process Reform Activities

Below, we describe the results of process reform activities.

4.1 Optimized Development Management and Objective Release Judgments

Using objective, quantitative indices, vendors and application development groups can now share development status, which enables mutual awareness of risks and early responses in terms of progress and quality. Also, implementing uniform quality reporting and setting conditions for quality alerts based on past statistical data for all development projects has made it possible to judge the quality of development projects objectively using side-by-side, chronological comparisons.

Figure 5 shows specific results of progress management reforms. The progress delay rate is defined as the percentage of information-sharing meetings involving delays greater than one day from among all such meetings for all projects in which development status reports were implemented. Although the progress delay rate exceeded 55% in the first six months after implementation, the improvements that followed have brought it down below 20% in recent times, indicating that the process reform actions have improved progress management.

Figure 6 shows specific results of quality reforms. We created two box plots^{*16} from bug density data aggregated from all tests after comprehensive vendor testing for all projects in the first six months and the most recent







six months, and found that the median values in the former fell to two-thirds of those in the latter. Popularizing the quantitative management of development helps incorporate quality in upstream processes and enables confirmation of software quality improvements.

4.2 Maintaining Development Data Statistics

Using data aggregation support functions with application development management support tools, we created a white paper on software development data by aggregating and sorting main

to the box

^{*16} Box plot: A type of graph used in statistics to display data with a lot of variation in a way that is easy to understand. In general, these graphs express the 1st quartile, the median, the 3rd quartile, and the maximum. The 1st, 2nd (median) and 3rd quartiles are represented as a "box," while the minimum and maximum values are represented by the "whiskers" attached

development data for all projects (including reliability, productivity, scale and man-hours) to publish in our department. For the content and structure of the white paper, we referenced software development data white papers [4] issued by the Information-technology Promotion Agency, Japan/Software Reliability Enhancement Center (IPA/SEC)*17. Along with the addition of the latest data, the software development data white paper (in-house version) has had three past revisions, and a restructured, derivative version is also available for vendors. The software white paper has been sent out to all application development groups and managers in our department, gets cited in feedback on completed developments and confirmation of adequacy for development plans, and is contributing to reinforcing the development PDCA cycle.

4.3 Structures of Mechanisms for Application Development Groups to Autonomously Optimize Development Management

Quality reporting has been implemented so that accurate quality reports can be output if development management is correct while alerts will be frequently displayed in quality reportage to indicate quality risks - a fact that is now well understood by application development groups - if the development management is faulty. Providing development management support functions with application development management tools has enabled application development groups to easily, instantly, and automatically output quality reports during the intermediate stages of application development. This provides them with quality report alerts during development and enables them to distinguish risk in vendor progress and quality reporting while promoting discussion of suitable additional responses to issues, thus leading to proper and autonomous development management in the development workplace.

4.4 Spreading Development Management Systematically with Application Development Management Support Tools

While combining usage log analysis of management support tools for application development with interviews in the development workplace, we have continued to upgrade functionality and expand the number of people using these tools by reinforcing application development groups' understanding of the tools' merits through presentations and other means. We have rolled out systematic development management to an assumed 20% of users after three months from initial deployment, expanding to 80% after six months. As a result of the ongoing popularization of tool usage, all application development members currently know about and are using the tools.

5. Conclusion

This article has described process reform initiatives for developing smartphone applications. We quickly embedded new form usage by creating standardized forms of development status report forms designed to be used by a wide range of vendors and standardized forms of quality report. We also established processes for quantitative development management in our department through document and tool upgrades and efficient human support. These process reform initiatives have brought about major improvements in the quality of development management and have therefore enabled better quality software development with fewer progress delays.

We also organized approximately two years of these process reform initiatives and presented the findings at the Japanese Software Quality Symposium*18 [1]. The forms we designed to be used by many different vendors became a central feature of the NTT DOCOMO presentation and attracted high levels of interest from symposium participants, who asked many questions about their details. It is unusual for a company that orders software development to give a presentation at this symposium, but it was effective in promoting knowledge sharing between ordering companies and vendors, which contributes to the advancement of the software development industry.

^{*17} IPA/SEC: An organization that studies and creates standards and visualization methods for development processes as well as quantitative quality management methods etc. with the aim of spreading quantitative project management in software development.

^{*18} Software Quality Symposium: The largest software quality related event in Japan, held to share practical technologies, experiences and research findings and exchange opinions related to software quality. As well as presentations and panel discussions by celebrities, the symposium also accepts presentations from general participants.

Going forward, we intend to take initiatives to improve in areas of concern by enabling closer linking between accumulated data, lifting the level of managerial skills and awareness, and handling Agile development processes.

REFERENCES

[1] T. Tosaki, Y. Yamada and H. Hattori:

"Development process improvement initiatives in ordering companies," Software Quality Symposium, 2015 (in Japanese).

[2] Information-technology Promotion Agency, Japan Software Engineering Center: "Recommendations for quantitative quality prediction," SEC BOOKS, 2008 (in Japanese).

[3] K. Hotta, H. Seki and Y. Miyazaki: "Soft-

ware quality assurance practice and system building," Soft Research Center, 2008 (in Japanese).

- [4] Information-technology Promotion Agency, Japan: "Software development data white paper," SEC BOOKS (in Japanese).
- [5] Software Quality Symposium Web site. http://www.juse.jp/sqip/symposium/