Management of Exhaustion in Continuous Product Development: The Case of a Mobile Game Company

Wei HUANG^{a)}

Abstract: In case of products under continuous development, such as mobile games, (a) the development workload for bug fixing is accompanied by (b) the development workload of new content, which exhausts development teams. This paper examined a case of a company where (a) some developers suffered exhaustion while responding to bugs pointed out by users. However, many of the bugs were related to specific events and items. Thus, the company chose not to include the bug fixes in quick updates as long as they were not emergencies related to the game system. In addition, (b) initially PvE (Player versus Environment) content comprised 70% of the new content, whereas PvP (Player versus Player) content comprised 30%. However, the consumption of PvE content was faster than originally estimated, resulting in exhaustion of the development team. By contrast, PvP content was generated primarily by users, and additions to content were not frequent, although considerable time was needed to validate

^{a)} Graduate School of Economics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan, lec@outlook.jp

A version of this paper was presented at the ABAS Conference 2019 Spring (Huang, 2019).

© 2019 Wei Huang. This is an Open Access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

the content. For validation, simulations of portions of user fights were conducted, which indicated reduction of time by approximately 35%; in addition, the ratios of PvE and PvP content were changed to 40% and 60%, respectively. Through (a) and (b), the feeling of exhaustion experienced by the development teams was ameliorated, and by increasing the proportion of PvP, which had a high social component, the company succeeded in acquiring new users and monetizing existing paid users.

Keywords: workload reduction, online game industry, continuous development process

Introduction

In recent years, as services for products are gaining more importance, product development processes have shifted, as shown in Figure 1, from (A) a string of stand-alone projects to (B) continuous activities where products undergo quick updates based on feedback from users. Particularly, in the case of software and game products, the usage information of users is created automatically as a byproduct of product use, and this information flows into companies continuously via their products. Products with this type of characteristics are becoming complex artifacts constituting user usage data and the product itself. Data-rich environments create opportunities to understand multiple customer aspects, and it has been suggested that the effective use of data is associated with good product performance (e.g., Johnson, Friend, & Lee, 2017; Troilo, De Luca, & Guenzi, 2017).

To maintain competitive advantage, companies must adapt their products to changes in external environments (e.g., Fukuzawa, 2015; Kuwashima, 2013). Adaptations to changes in product development include (a) the design of flexible products, and (b) the flexible design

Figure 1. Comparison of traditional and continuous development activities

of products. Studies regarding (a) have focused on modularization of product architecture and the creation of core technology platforms (Baldwin & Clark, 2000; Ulrich, 1995), whereas studies on (b) have focused on effects and the cost of adaptation activities in development processes where flexibility is a key concept (e.g., Thomke, 1997; Thomke & Reinertsen, 1998).

However, continuous product development makes management of the development process and workflows more difficult, as development organizations are confronted with vast amount of information to be processed (Buganza & Verganti, 2006). As the same development organization is forced to develop over a long period of time, there is an increased risk of exhaustion of developers in the organization. In the case of Company X considered in this paper, unstable development schedules and work for processing and analyzing user information continued as (a) the workload of fixing bugs and (b) the workload of developing new content were added, which exhausted the development team. With modifications made in the measures for correcting bugs and increasing the percentage of user-generated content, the development team was able to rid itself of exhaustion and respond to the demands of users.

Case Study

This paper discusses Project A of a game development firm, Company X, a subsidiary of the leading Chinese net service Company Y. Originally, Company X was the mobile game development organization of Company Y; however, the entire organization became a subsidiary with the rapid growth of the mobile game industry. Most of the developers in Company X belonged to Company Y's development organization prior to becoming independent and were veteran developers of mobile games. The company has sufficient capital with periodic investments from venture capital and Company Y.

The main products of Company X are game applications for Android and iOS that target both domestic and overseas markets, although the company prioritizes the domestic market, and develops at a rate of approximately three games per year. As of March 2019, the company had produced 14 game applications, of which six were released in overseas markets. The profit model of the company revolves around the sale of in-game items, and 30% of mobile game revenues come from overseas. Company X's game products are mostly of the role-playing gaming (RPG) genre and incorporate into the character development play systems that are unique to the RPG genre, which are highly praised by both domestic and overseas users. The game developed in Project A discussed in this paper maintained a high ranking in its genre in various application stores after its release and showed excellent performance (i.e., the acquisition of new users and monetization of existing paid users) in comparison with products from other companies.

Interviews were conducted from January to March 2019 with subjects including one executive, one project leader, and three game developers. In addition to the interviews, an observational survey of the development team was conducted for approximately one hour.

Development organization and product

Company X has two unreleased projects in development projects and four released projects. A dedicated team is organized for each project, and the development teams are mid-sized on average, comprising 60 to 70 developers. Seven to nine specialized developers are assigned to work on game planning, character design, graphics, UI design, and so forth, and engage in their development work under the management of a project leader. Project leaders basically manage the entire project, including budgets, development schedules, deployment of team members, and game quality. In addition, an operations team is created for each project, with its primary work being system maintenance as well as visualization and analysis of post-release user data. In advance of subsequent updates based on such analyses, these teams build validation models. The operations team provides periodic feedback to the development team on user trends and make proposals to coordinate items added in upcoming updates and balance. Thus, after formal services begin, operations teams and development teams work together to analyze user play data and to continually develop refinements in maps, items, and other additional content and graphics, as they further evolve the product.

Development workload and exhaustion in Project A

The development workload in game development can be represented by the sum of tasks in the development of game content. The composition of tasks for Project A is of two types: development of updated content and bug fixing. Moreover, as shown in Figure 2, multiple development groups are responsible for tasks such as development planning, implementation, testing, and user data analysis. Prior to release, they are responsible for development planning, testing, and implementation. After release, analysis of user

Figure 2. Main work processes of game development (after release)

data is added to existing development organizations, and these act as the interface for feedback of user trends in the game to other organizations, and for development of updated content and responding to bugs.

As part of their long-term plan for updates, the goal of the development team for Project A was to conduct in-game events once in a week and add new updates once in a month. However, if a bug appeared in the game development project, the development team concentrated on fixing the bug by investing development resources. In this process, development would slow down as the number of resources allocated to other work in parallel development decreases. To maintain the current schedule, related groups (primarily development planning and testing) were used outside their regular work hours to help with temporary increases in the development workload. Because of many emergencies, overtime was easily incurred even in the implementation of bug fixes. These are indicated by red portions in Figure 2.

In the case of Project A, after the game was released, a certain number of users were secured, thereby gradually moving to profitability. However, six months after the release, the project schedule frequently became tight, with the average work time for project members going up to eleven hours per day. Exhaustion of the development team members became a clear problem. Team members did their best to fix bugs and maintain product performance. However, as four development team members became ill, certain development work stagnated. In the development of new content, which was being performed in parallel, sufficient time for planning and testing could not be provided; this resulted in users beginning to point out that the volume of the updated content provided monthly was paltry.

In fact, the team's development schedule for new content in updates was frequently delayed while prioritizing allocation of limited development resources to bug fixes; therefore, the overall project progress fell into a state of disarray. This resulted in the inability to produce new content updates according to plan, and the attrition of both new and paid users spiked temporarily, harming monetization and adversely impacting the company's development resource allocation in the subsequent period.

Breaking out of fatigued development sections

To relieve the exhaustion of the development team, Project A analyzed the reasons for the chaos in schedule management. Game development demands a high level of understanding of a product, and even though new members were added in a development team, considerable time was required for them to become effective. Thus, most bug fixes were performed by existing development team members working overtime. However, careful categorization of the status of bugs in Project A showed that many bug fixes are related to specific events and items, and few are related to the core game

system. After understanding this, the company took different measures based on the causes of bugs. Specifically, in emergencies related to the core game system, such as the inability of users to log in or mid-game dropouts, the team would respond as they always had, but all other bugs did not need to be urgently fixed. If a bug fix was not completed on time for the upcoming update, the team would not overextend itself; instead they would add the fix in a subsequent update. This change in bug fix responses caused a major decrease in average overtime per developer from 90 to 20 hours per month.

In addition, the game content of Project A is of two types: PvE and PvP. After examining Project A's teams where overtime easily occurred and carefully analyzing the details of the development workload, it was found that the time spent in both production and consumption differed by the type of game content.

PvE refers to "Player versus Environment," or games where users accomplish tasks given in the game environment and fight against set enemies. PvE is meant for new users, giving them the experience of fighting set enemies in a set environment, which would increase users' understanding of the game system. However, in the case of PvE, users advance in levels by accomplishing tasks that are given to them, thus consuming content rapidly, which tends to decrease the attractiveness of the game. The average development time for PvE is a total of five weeks—four weeks for planning and implementation and one week for testing.

By contrast, PvP refers to "Player versus Player," or games where users fight each other under the rules of the game environment, competing for ranking scores in the game. If the basic fighting rules and environment are set skillfully, users can fight various other players even after they clear through PvE, and the attrition rate for existing users is expected to decrease. The impact of paid items and newly added items within the game system is great, so maintaining balance in fighting with any number of users requires careful

Table 1. Product configuration of project A

	Planning and implemen- tation period	Test duration	Consump- tion period	Target players	Profitabil- ity
PvE	4 weeks	1 week	2 weeks	New users	Low
PvP	4 weeks	2–3 weeks	8–10 weeks	Existing users	High

validation of the number of rules and items. The average development time for PvP content is a total of six to seven weeks—four weeks for planning and implementation and two to three weeks for testing. These development times are summarized in Table 1.

Nine months after release, Project A allocated much of the new content to PvE, where consumption of content is fast. In the original game systems, 70% of content was PvE and 30% was PvP. However, an analysis of user play data showed that most users accomplish PvE content in under ten hours. The actual play time is much shorter than the predicted play time of 30 hours.

By contrast, consumption of PvP content is slow, and the frequency of content planning and implementation is less than that of PvE, although validation of content takes considerable time. In other words, an increase in the ratio of PvP content implies that the development workload for planning and implementation decreases due to slower content consumption, although the workload of the testing group increases.

In addition, because PvP has a high social component, creating and maintaining a healthy game ecosystem is extremely important. If content is focused on specific user groups, long-term performance of a product could drop (Huang, 2018); thus, validation will be

required. To validate efficiently, Project A began automatic testing, and simulation of portions of user fights resulted in an approximate 35% decrease in testing time. As of March 2019, the composition of game system content had reversed, with 40% PvE and 60% PvP.

Discussion

Many existing studies have assumed that flexibly adapting to changes in products and external environments leads to greater product performance (e.g., Buganza, Dell'Era, & Verganti, 2009; Buganza & Verganti, 2006; MacCormack, Verganti, & Iansiti, 2001). However, as with the case of Project A in game development firm Company X discussed herein, the cumulative workload in ongoing development activities exhausts development teams, making long-term maintenance of product performance difficult.

To maintain a high level of product performance even with limited development resources, Project A changed their response to bugs, thereby reducing the frequency of concentrated investment of development resources, and successfully reduced the development workload. In addition, by proactively using user-generated content such as PvP content, the company succeeded in acquiring new users and monetizing existing paid users.

Acknowledgments

This work was supported by JSPS Grant-in-Aid for Publication of Scientific Research Results, Grant Number JP16HP2004.

References

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of*

modularity: Vol. 1. Cambridge, MA: MIT Press.

- Buganza, T., Dell'Era, C., & Verganti, R. (2009). Exploring the relationships between product development and environmental turbulence: The case of mobile TLC services. *Journal of Product Innovation Management*, 26(3), 308–321.
- Buganza, T., & Verganti, R. (2006). Life-cycle flexibility: How to measure and improve the innovative capability in turbulent environments. *Journal of Product Innovation Management*, 23(5), 393–407.
- Fukuzawa, M. (2015). Dynamic capability as fashion. *Annals of Business Administrative Science*, 14, 83–96. doi: 10.7880/abas.14.83
- Johnson, J. S., Friend, S. B., & Lee, H. S. (2017). Big data facilitation, utilization, and monetization: Exploring the 3Vs in a new product development process. *Journal of Product Innovation Management*, 34(5), 640–658.
- Huang, W. (2018). Management of flexibility in continuous product development: The case of the online game industry. *Annals of Business Administrative Science*, 17, 215–225. doi: 10.7880/abas.0180921a
- Huang, W. (2019, June). *The management of exhaustion in continuous product development*. Paper presented at ABAS Conference 2019 Spring, University of Tokyo, Japan.
- Kuwashima, K. (2013). Followers of *Harvard Study*. *Annals of Business Administrative Science*, 12, 31–44. doi: 10.7880/abas.12.31
- MacCormack, A., Verganti, R., & Iansiti, M. (2001). Developing products on “Internet time”: The anatomy of a flexible development process. *Management Science*, 47(1), 133–150.
- Thomke, S. H. (1997). The role of flexibility in the development of new products: An empirical study. *Research Policy*, 26(1), 105–119.
- Thomke, S., & Reinertsen, D. (1998). Agile product development: Managing development flexibility in uncertain environments. *California Management Review*, 41(1), 8–30.
- Troilo, G., De Luca, L. M., & Guenzi, P. (2017). Linking data-rich environments with service innovation in incumbent firms: A conceptual framework and research propositions. *Journal of Product*

Innovation Management, 34(5), 617–639.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24(3), 419–440.