A New Business Model of Custom Software Development for Agile Software Development

Yoshihito Kuranuki
SonicGarden
6F, 1-3-4 Jinnan, Shibuya-ku
Tokyo, 150-0041, Japan
kuranuki@sonicgarden.jp

Tsuyoshi Ushio
SimpleArchitect
Japan
ushio@simplearchitect.com

Tsutomu Yasui
Eiwa System Management
Ueno HS Building 8F,
2-7-7, Ueno, Taito-ku,
Tokyo 110-0005, Japan
tsutomu.yasui@gmail.com

Susumu Yamazaki
University of Kitakyushu
1-1 Hibikino,
Wakamatsu-ku
Kitakyushu, Fukuoka,
808-0135,Japan
zacky@kitakyu-u.ac.jp

ABSTRACT

In this paper, we propose a new business model for custom software development industry utilizing agile software development to embrace changes. Traditional business model usually does not work well with agile software development because it mandate up-front man-month estimation based on a fixed scope. Our new model proposes a solution with two notable features; fixed monthly subscription and weekly agreement of what to provide. We successfully applied the new business model to more than 20 cases over 3 years and this paper describes one of them.

Categories and Subject Descriptors

D.2.9 [**Management**]: Software Process Models

General Terms

Management

Keywords

Agile Software Development, Lean Startup, Business Model, DevOps

1. INTRODUCTION

1.1 Background

Business environment where software are used is changing. Any type of service provider used via Internet must continually release new versions based on what users want. Lean Startup [1] states that startup must radically add and change its service with feedback from users.

Such environment requires software development to adapt to changes and keep doing that. It is Agile Software Development [2][3] that answers such requirements. In Agile Software Development, developers continually deliver a small scope to production environment in short cycles by doing analysis, design, coding and testing in parallel. This process enables developers to respond to mid-way changes.

Revised version of the article in Proc. InnoSWDev'14, November 16, 2014, Hong Kong, China.

Agile Software Development within a company comes off best when the company internally develops software with hired programmers. Whereas its merits diminish considerably when the company outsources the whole development to an outside organization — the traditional custom software development business enforces an up-front fixed-scope man-month estimation but it does not work well with Agile. Among many reasons, we must point out that it's because of a conflict of interest between the contractee and the contractor. The contractee's interest is in modifying and changing the software after the first launch with feedback from users whereas the contractor puts great effort to deliver the whole software at once within the estimated cost and schedule.

Many companies find it difficult to hire and motivate capable programmers and they must choose to outsource. Although Agile presents ways to adapt to change, the traditional custom software development business cannot benefit from them because the scope is fixed and due-date is estimated up-front. Hence those companies have challenges to benefit from utilizing Agile Software Development.

Our proposal is to introduce a new business model for custom software development to incorporate both outsourcing and changing software. This paper shows a new business model we established and proved based on our publication [4]. We also explain issues we found, their solutions, and essential approaches.

1.2 Purpose

In the traditional software development with fixed-scope estimation, it is hard to adopt Agile Development to adapt to changes. Our purpose is to present a new business model which can adapt to changes in custom software development businesses.

1.3 Approach

With our new business model we greatly improved effectiveness of Agile compared to the traditional model — an cost estimation based on fixed scope — in custom software development business. The improvement comes from fixed monthly payment without setting deadline which enables us to develop without up-front cost estimation. The model also makes development companies more profitable.

1.4 Organization

The rest of this paper is organized as follows: Section 2 proposes the new business model, and Section 3 shows its application. Section 4 discusses the impact to the industry which our business model will make. Finally Section 5 concludes this paper.

2. THE PROPOSED BUSINESS MODEL

2.1 Problem

We propose a new business model of custom software development suitable for Agile Software Development to embrace change. The business model of the traditional software development with fixed-scope estimation has the following three problems:

1. **Incentives to add more people**: Although an agile team should be small, elected and unified for efficiency, a traditional developer team tends to be oversized because it adopts estimation based on man-month and has an incentive to waste more man-months.
2. **Building useless functions**: To estimate a software with fixed-scope at once requires a prediction for whole function before developing it. However, the prediction tends to be wide of the mark. Thus, it causes a bunch of useless functions.
3. **Cost increase**: The maintenance cost can get exponentially higher after the first release because of overhead in estimations of each change.

These problems are caused by "Nouhin" and due-date. "Nouhin" is a way of delivery in Japanese business custom. If you perform Nouhin of a software product under the fixed price and scope contract, you must transfer the ownership to your customer at the time of delivery. Nouhin usually requires comprehensive documentation of the software. The fix-price and scope contract inherently dictates that the most important thing for custom software development companies are to complete the set scope within due date. Naturally, usefulness of the software tends to be overlooked although the customer's benefit will be damaged.

We propose a new business model based on a hypothesis that we can solve these problems by avoiding Nouhin and due-date in custom software development.

2.2 Solution

Our new business model has the following features:

1. **Monthly subscription** for the custom software development and operation service. Rather than the fixed-price and scope contract.
2. **Weekly agreement of what to provide** including developing features and operation. We never promise scope, work time and a due date for our customer. We never estimate resource using man-month in the long term.

In our new business model, we start with an MVP (Minimum Viable Product)[1]. We develop and deploy the MVP within two weeks for the customer to see the working software. The first MVP is followed by weekly sprints. Sprints are run with a meeting and development cycle until a launch to end users, usually around 3 months. We keep development and operation after the launch. Services are usually operating on a cloud by ourselves. It is not Nouhin —— we never transfer the ownership to our customer.

We assign one programmer for one customer. The programmers are in charge of entire activities, i.e., requirement engineering, design, implementation, testing and operation. Our programmer may be assigned for several customers. From the point of view of a customer, he/she just employs a capable programmer with a fixed monthly fee without any need of procurement or training. This is similar to the business model of a legal adviser, where a customer commissions him/her without employing.

Our customer can change the specification not only during developing but also after the initial release. We do not estimate cost but change priorities for changing the request because we adopt the monthly subscription model. Our programmer always keeps the software maintainable while incorporating changes. We do not need any documentation because there is no handoff as a single programmer is assigned for the entire development and operation for a single customer.

Programmers stay in our office and we make no promise of working time with customers. Instead, we promise to continue developing any new functions of and improvement on the software and operating it stably. Our customer confidently expects to receive outcomes without managing programmers by themselves. Our customer pays monthly subscription fee, hence, of course, he/she may stop the subscription anytime if they feel our output doesn't meet their expectation. This works as a penalty for us. Our programmer can be assigned to two or more customers if he/she improves development skills and productivity of software. This brings profit to us.

This approach solves the above-mentioned problems as follows:

1. **Incentives to add more people**: It does make no sense to us to increase the size of a team —— customers pay only monthly fixed fee, regardless of the size of a team. That leads programmers to be more productive and efficient, instead of putting more time and resource.
2. **Building useless functions**: We have more incentive to improve cost performance than to make many useless functions because of monthly subscription.
3. **Cost increase**: We always accept any changing requests from our customer even after the release to the end users because of monthly subscription.

Our new business model brings our customers and ourselves a common goal to develop software without any conflicts between our customers and us.

2.3 Software Development Process

Fig. 1 shows the software development process of our business model.

The following certain steps are required to implement our business model effectively.

1. Free Discussion Period - to share client's vision and goal. (No. 1 in Fig. 1)
2. Free Trial Development Period - to agree with the performance of development. (No. 2 in Fig. 1)
3. Weekly iterative development and meetings. (No. 3 to 6 in Fig. 1)
4. United development and operation. (No. 6 and 7 in Fig. 1)

2.3.1 Free Discussion Period

In our business model, we value small number of long-term subscription contracts than many short-lived contracts because we do not develop the software and receive fee entirely at once but continue to *grow* it with receiving monthly subscription, i.e., we have an incentive to sustain customer's business for long time. Thus, we start by consulting a prospective customer freely to share and assess his/her business model. In this free discussion

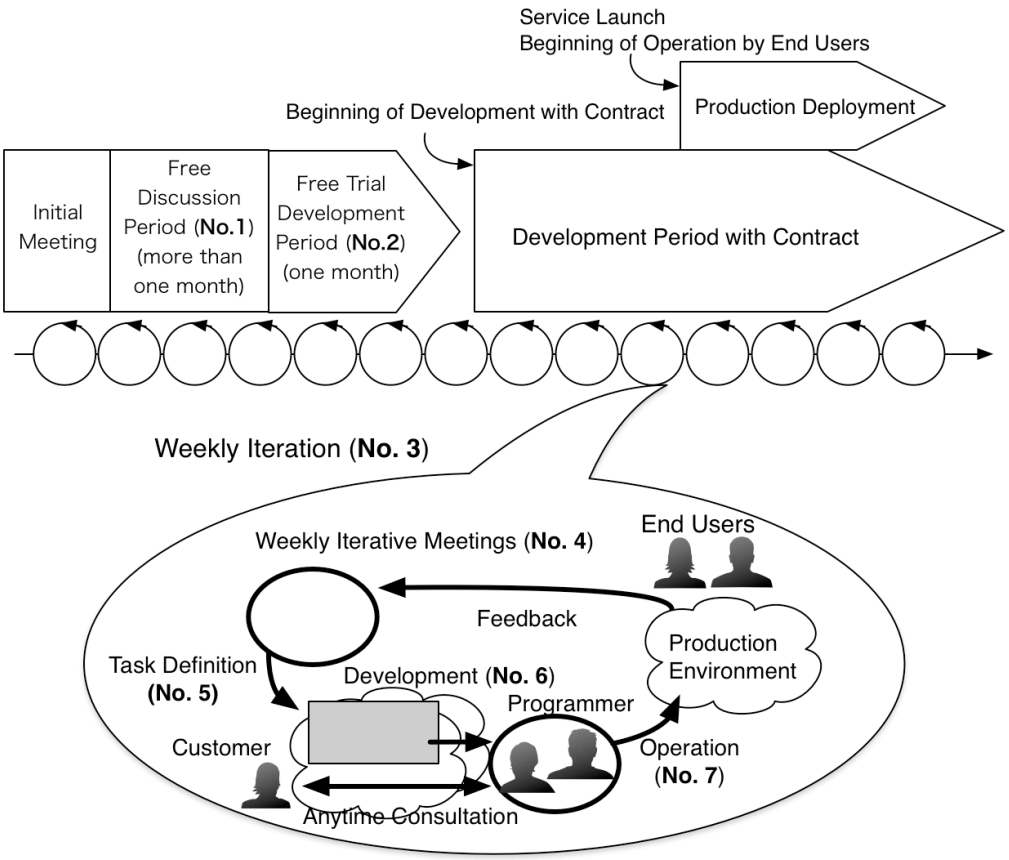

Fig. 1 The Overview of the Software Development Process of Our Business Model

period, we share what to develop essentially with the canditate. We assess his/her business plan in the following terms:

- **Mission**: Why to develop; and
- **Vision**: What to realize by the software.

Business which defines its mission and vision clearly will be successful. The goal of the free discussion period is to design the software in detail enough to release it for end users about three months later.

2.3.2 Free Trial Development Period

In our business model, our programmer promises to provide valuable software artifacts to our customer monthly. However, the customer cannot know how much the value will be before actual development. To release this anxiety, we provide free trial development service for one month. If he/she is satisfied with the artifacts developed in this period, he/she will make a contract with us starting from the next month.

The goal of the first two weeks of this period is a working software on a production environment with bare-minimum application features with which customer expects to test the software. Developer focuses his/her effort to that goal so usually no documents are written. After the goal is achieved, we add features in usual one-week development cycle (see the next section).

2.3.3 Weekly Iterative Development

Our customer and programmer have weekly meetings and finish the design of the software immediately. This designing focuses on only the iteration during the next one week. After the meeting, our programmer develops it at his/her own pace. At the next meeting, they have a meeting to test the artifacts in the week and design for the next week. This is repeated while the contract is valid.

The customer decides whether he/she accept functions of the software which have been developed weekly. We do not write any documents of specification but express specification with the working software itself because writing and managing consistently both such documents and software are redundant and troublesome.

We launch the software to the end users within about three months. It is easy for us to launch it because the customer always checks it weekly. We continue to develop it iteratively even after the launch. We discuss roadmap of the business and software and plan for up to next three months once every about three months. In our business model, we do not promise to complete the software. Rather, we discuss with the customer what is important to his/her business, plan to realize functions in the priority order, and provide valuable software to him/her.

2.3.4 United Development and Operation

After two weeks of the free trial development, we begin operation of an environment for the customer to test the software. After about three months, we launch to the end user with operation on

the production environment. We provide common testing environment for all customers, while we provide a production environment for each customer.

We develop and operate continuously in parallel. We apply some continuous integration methods: git and GitHub for source code management, automated testing, continuous production, *etc*.

3. CASE STUDY — AsMama

We have already developed more than twenty projects using our business model since three years ago. We describe a case, AsMama, Inc., which has built a good relationship with us for a long time.

3.1 Why Do They Need the Web Service?

AsMama Inc. provides the "Kosodate Share" (childcare sharing) matching service for mothers using PCs and mobile phones, in which mothers who are acquaintances rely on each other for taking to and fro or day-care of kids. The mission of AsMama is to provide a social platform of people both helping and being helped in raising kids.

Because this is a matching service via Internet, it requires software. However, there was no programmer among the founding member. Thus, AsMama considered the following choices:

- Recruitment; and
- Outsourcing.

AsMama could not employ programmers because they could not evaluate skills and competence of programmers because they are not software professionals. Even if it was possible, it also was hard to evaluate and train programmers.

Outsourcing needs whole scope definition because a traditional software company, which uses Nouhin, requires it upfront. Since AsMama is a new business, AsMama could not define all functions upfront.

AsMama needed a partner who willing to "grow" the service collaboratively by starting from a discussion of what to build first and continually developing functions incrementally, instead of developing and "Nouhin" the whole software at once. We thought it could fit for our business model. We proposed it and they accepted. Then, we started this project.

3.2 The Result

We started discussion at August, 2012. At first, AsMama proposed a kind of Social Networking Service (SNS). However, we found that the essence of their business is a matching service rather than SNS from the discussion what they want to do. Thus, we picked up essential matching functions to develop at first.

We began to develop the software as a trial from December, 2012. At first in this trial period, we implemented a supporter list function, which is important for an end user who needs childcare. During the trial period, we built the functions of the screen to select one of the supporters on a map visually.

AsMama had been satisfied with the process of meetings and the working software during the trial period, and has contracted us to continue developing it. In other cases, some customers were not satisfied at the end of trial period and choose not to contract us.

AsMama launched the matching service in April 2012. We labeled the service as beta to limit the number of users and encourage more feedback. We have continued to improve the software based on the feedback and to increase the number of end users gradually. Our business model brings us consideration of essential functions based on end users' voices.

As of June 2014, the matching service has acquired more than ten thousand end users. Even now, we continue to develop and operate the software. We assign one programmer to one customer basically, while we assign two programmers to one customer with additional fees if the customer requires larger development. In AsMama's case study, we assigned two programmers in the development, temporally.

This business model could help customer's business change from end-user's feedback. And we succeeded to grow with our customer's success.

4. DISCUSSION

The proposed business model will satisfy certain needs which are different from those the traditional model will satisfy. That means we changed the problem itself rather than mere solution.

- Needs of the traditional business model
  - Scope and delivery date are fixed up-front; there is strong force to develop everything at once.
  - Goal is to complete the requested software.
  - Waterfall development proess works well.
- Needs of the proposed business model
  - Requests are prone to change and hard to foresee; there is force to continue long-term development.
  - Goal is to grow business.
  - Agile development works well.

Those are two difference set of needs hence target markets also differ. Fig. 2 shows market segmented with two axes; custom made or not, and "Nouhin" delivery or not.

The market of our business model (No. 1 in Fig. 2) is under-developed in Japan. We believe it can be a promising market based on the fact that our effort is attracting strong interest in Japanese industry and the number of inquiries we receive from prospective customers.

We uncovered several risks in our business model. Following are some of the risks we effectively controlled.

- How to avoid customer who requires to fix scope up-front.
- Quality control in the environment where each developer works for different software and still all software need to achieve certain quality.
- Chances that a programmer can no longer work —— clients relied on the programmer could miss the whole development force.
- How to avoid unqualified programmers.

We would like to have a discussion about the following in the workshop.

- What other risks should be considered and how to control them.
- Prospects of the business model among other areas in the world.

We also would like to show how actually we controlled aforementioned risks in our 3 years of experience.

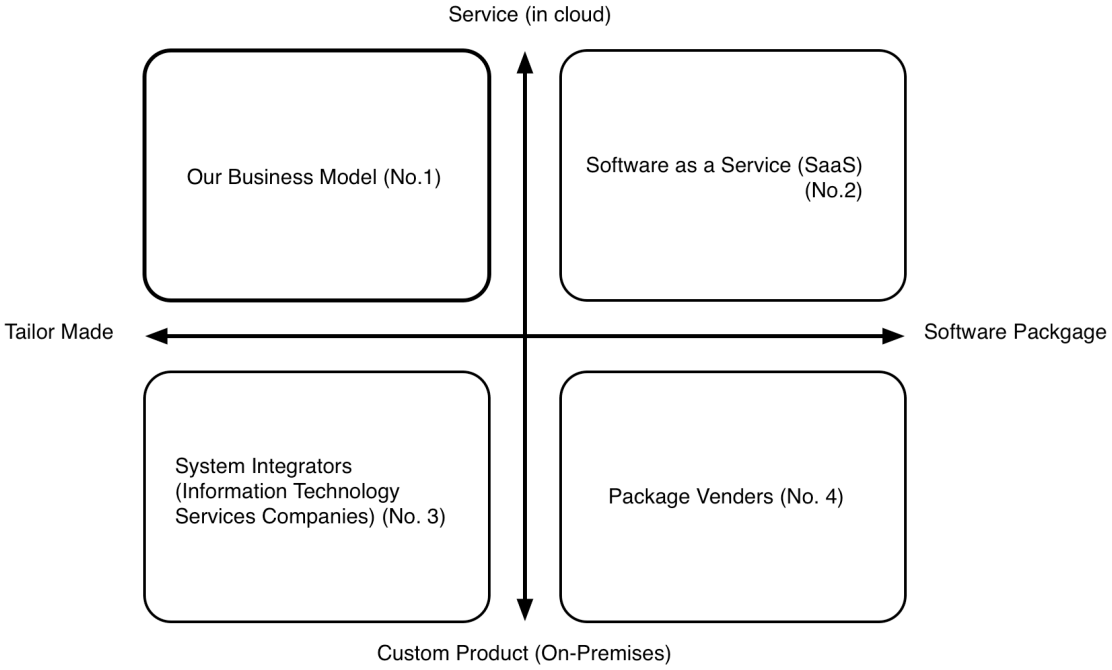

Fig. 2 The Four Categories of Software Industry

5. SUMMARY

In this paper, we have pointed out that we should break traditional business custom "Nouhin" (a way of delivery), and proposed a new business model of custom software development, to apply agile development method effectively, which was proposed to "embrace change", to Japanese traditional custom software development industry. We also describe one of case studies to which our business model is applied. We would like to discuss risk hedge and future prospects of our business model in this workshop.

6. ACKNOWLEDGMENTS

We thank all our customers for agreeing with and accepting the value of our new business model. We also thank all programmers in SonicGarden for conducting our business model.

7. REFERENCES

[1] Ries, E., 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Business*. Crown Business, New York.

[2] Beck, K. et al., 2001. *Manifesto for Agile Software Development*. available at http://agilemanifesto.org.

[3] Beck, K., 2000. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Reading, MA.

[4] Kuranuki, Y., 2014. *Nouhin wo Nakuseba Umaku Iku: Sofutouea Gyokaino "Joushiki" wo Kaeru Bijinesu Moderu. (No Deliver: A Business Model Changing Business Custom of Software Developer.)* (In Japanese),Nippon Jitsugyo Publishing, Tokyo, Japan.