Journal of Industrial Information Integration 14 (2019) 16–23

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.elsevier.com/locate/jii

An agile approach for managing requirements change to improve learning and adaptability

reminds me of how startups target changes so RE goals move with that. Here, agile development with continuous iteration and feedback can be used as input in RE and adapted lessons learned.

This reading solidifies my learning about RE changes with more examples make me notice that RE, along with agile, is such a continuous feedback and target-changing process that needs to be embraced.

changes not just because of agile, but also because of the initial need to satisfy customers, transforming to long-term internal company goals, so the target will be shifting constantly to work its way to feed customer goals into company goals. Learnt requirement engineering should allow space to change.

Woogon Shim[a], Seok-Won Lee[b,*]

[a] Software Engineering Lab, LG Electronics, Seoul, South Korea
[b] Department of Software & Computer Engineering, Ajou University, San 5 Woncheon-dong, Youngtong-gu, Suwon-si, Gyeonggi-do 443-749, South Korea

Question: achieve goals by customers' behavioral change - what behavior change is the company expecting and how does this change achieve their goal? Is it because in Fig 4, steps 1-3 the customer is having their hypothesis validated, and from that, in step 4 it shows how they will achieve the company goal? Behavioral changes expected is from customer feedback?

Question: would u say step 1 - 3 in Fig 4 is like a feasibility analysis?

ABSTRACT

Nowadays agile development has become a mainstream development methodology. Yet agile is still lacking in how in-depth it deals with Requirement Engineering (RE) compared to the other development stages. There have been attempts to apply agile RE techniques to traditional development and conversely to apply traditional RE techniques to agile development. But the biggest problem is that it is difficult to maintain the essence of agile which is to "*collaborate*", "*(iterative and frequent) inspect and adapt*", and "*deliver customer value*". In this paper, we define six criteria for an agile requirement management approach and propose a lightweight agile approach that maintains the characteristics of agile while combining the RE techniques of academia, business analyst, and agile & lean community. Furthermore, we show an industrial case study which we've applied our approach.

1. Introduction

As with the traditional development approach, RE is crucial in agile development [1]. RE researchers have attempted to incorporate radical practices from agile to address the limitations of traditional RE [2]. In contrast, to overcome the limitation of agile RE, people in the agile community have tried to combine proven traditional methods [3,4]. Also, in the agile community, there has been some attempts to remedy its shortcomings by themselves [5,6]. As a result of these studies, the recognition has improved a lot but academia approach to supplement the agile requirement engineering unfortunately does not understand or reflect the characteristics of agile RE [7], or it is difficult to learn easily because of formal methods.

Agile development is known to be suitable for complicated or complex zone projects with requirements and technology changes based on the Stacey Matrix [8]. The agile method emphasizes that you do not resist change, but rather embrace it [9]. But, as we all know, change management is not easy. Problems often arise in development due to requirement changes: How to evaluate the impact of changes easily? How to update requirements related to the changes efficiently? And so forth. Unfortunately, there is no clear, practical guide to handling these issues.

Before addressing this problem, to maintain the essence of agile, we have identified some important criteria that the new approach should have based on our 10 year experiences as an agile coach.

Agile development doesn't begin with all requirements perfectly defined and validated. In addition, it often cannot involve nor be on-site with all the different kinds of customers, leaving no choice but to elicit requirements from a few customer proxies (e.g., Product Owner in Scrum). Thus the requirements should be regarded as a hypothesis which is to be validated continuously [10,11]. By validating facts from the most critical hypotheses, you should avoid wasting unnecessary resources in implementing the requirements due to unconfirmed hypotheses. It is all about the "Validated Learning" in Lean Startup [12]. To do more effectively, each requirement must be able to distinguish whether it is a hypothesis or a fact easily (C1). "Delivering the highest value to customers quickly [9]" is one of the core agile principles. And Steven Blank pointed out that lots of startups fail not product development failure but from a lack of customers [11]. Therefore, the requirements should be structured to being customer-centered, and we must be able to find out what requirements belong to the specific customer (C2). On the other hand, the development company must create profits, so they must not only achieve the customer (external) goals, but the internal goals of the development company as well [13]. Customer satisfaction is essential, but the development company is not a charity. The satisfied customer should change their behavior to what the development company expected. So, it must be able to consider both the internal goals of the development company as well as the customer goals (C3). In agile development, RE activities are executed in parallel with the product development [14], it is therefore necessary to receive feedback of the product development periodically, in order to develop the requirements and to adapt the lessons learned [15] (C4). Agile development tends to specify concrete scenarios as requirements for deriving test cases from them and validating them easily [16] (C5).

reminds me of how startups target changes so RE goals moves with that. here agile development with continuous iteration and with that feedback, can use that feedback as input in RE and adapt lessons learnt. More examples making me notice RE along with agile, is such a continuous feedback and target changing process that needs to be embraced.

changes in not just because of agile, but also because of inital need to satify customer but future success of internal compnay goals, so target will e shifting constantly

* Corresponding author.
E-mail addresses: woogon.shim@lge.com (W. Shim), leesw@ajou.ac.kr (S.-W. Lee).

https://doi.org/10.1016/j.jii.2018.07.005
Received 10 November 2017; Received in revised form 6 February 2018; Accepted 31 July 2018
Available online 01 August 2018

Ultimately, the goal of agile RE approach is to optimize the stakeholder's understanding and to enrich the collaboration and communication. Therefore, the RE approach should be easy to learn, facilitate communication, and be able to proceed using low tech tools (C6).

Some criteria that the agile RE approach should support are as followed:

- C1. All requirements should be regarded as hypotheses, and it is necessary to be able to distinguish between hypothesis and non-hypothesis (fact).
- C2. It must be clear what the requirements are for which customers, and what goals the customer is going to achieve.
- C3. Customer satisfaction is essential, but the satisfied customer should change their behavior to what the development company expected. Both outside-in (external) and inside-out (internal) goal modeling should provide.
- C4. It must support dual track; concurrent with development and reflect feedback regularly.
- C5. Requirements should be able to specify as concrete scenarios to derive test cases from them and validate them efficiently.
- C6. It should support low tech tools and be easy to prompt collaboration and communication.

In this paper, we define six criteria for an agile requirement management approach and propose a lightweight agile approach that maintains the characteristics of agile while combining the RE techniques of academia, business analyst, and agile & lean community.

In what follows, Section 2 summarizes related studies to handle agile requirements and presents the result of comparison among those studies based on six criteria for Agile RE which we've proposed above. Section 3 describes major artifacts and procedure of our approach. The industrial case study we've experimented is addressed in Section 4. Finally, Section 5 is providing our concluding remarks and highlighting future work.

overcoming gathering the right requirements with a hypothesis vs fact model by left impact mapping connection lines and below HYPEX or QCD by validating hypothesis progressively thro experiment results

2. Related works

2.1. Academic community

In academia, similar to the traditional one, they knew that agile RE was carried out at the early stage of the project [17] or before the start of every iteration [18]. However, in agile development, RE do in parallel with development throughout the project period [14], and development starts even when the requirements are not perfect. In the goal sketching technique [19], by allowing TBD (to be determined) in goal modeling, it can provide a level of abstraction of Goal-oriented RE (only as needed when necessary) and also give a few notations that can be easily used by various stakeholders. On the other hand, RE-KOMBINE [20], which tolerates the presence of inconsistencies and conflicts in requirements to be incurred when undertaking development with incomplete requirements. But it also provides paraconsistent logic to detect and resolve inconsistencies early on instead. It's quite flexible but difficult to learn and use it, since it's a formal method.

Olsson and Bosch note that a small number of decision makers tend to determine features because it takes too long to get feedback from customers and collecting and analyzing data is inefficient. In order to overcome this, they proposed HYPEX (Hypothesis Experiment Data-Driven Development) [4] or QCD (Qualitative/quantitative Customer-driven Development) [10] model which can check the customer's value and product feature by receiving prompt and accurate feedback data from customers. This model considers the requirements as hypothesis and validates them progressively based on the results of the experiment and the results.

2.2. Business analyst community

Discover to Deliver [21] presents agile product planning and analysis methods. The discovery phase, which uncover new requirements, and the deliver phase, which develops and delivers the needs of the identified customers, are constantly repeated. It provides predefined 7 dimensions for stakeholders to guide collaborative. Structural communication is made for each dimension. After pouring out various ideas (divergence), screening the items with the highest ROI (convergence) and determining the product requirements of the next iteration. Although this approach is concerned with iterative development, it does not deal with how to manage change of requirements based on what you have learned.

Value Proposition Design [22] is a follow-up study of Business Model Canvas [23]. It is a separate extension of 'Value Proposition' and 'Customer Segment' among the 9 blocks. In-depth analysis of customers can help us identify needs and pain and separately consider the features that the company will offer to provide value to customers. During each iteration, you can review whether the product prepared ('value proposition') by the company matches the customer analysis ('customer segment'), and visualize it with the whiteboard and post-it.

2.3. Agile & Lean community

Impact Mapping [5] presented a simple and effective requirements management system to escape the "*user story card hell*[24]". Impact mapping is a strategic planning technique. It prevents organizations from getting lost while building products and delivering projects, by clearly communicating assumptions, helping teams align their activities with overall business objectives and make better roadmap decisions. Actually, It is a typical user story format "*As a ⟨type of actor⟩, I want ⟨deliverable⟩ so that <impact>*" is restructured into a mindmap form. It rearranges user stories as a form of Why-Who-How-What. It acts as the goal- and customer-centered map which guides us to not be lost during development. And the connection line between each item is the assumption which needs to be validated. However, there is a limitation in that despite showing the entire requirement map it is difficult to see how features are connected and how they flow from a customer's point of view.

Instead, User Story Mapping [6] creates narrative end-to-end backbone scenarios that deliver value to the customer and then attaches the specific work and exception handling tasks below each stage of the skeleton scenario according to its priority. Then group them so that can provide an outcome of the customer's perspective and can establish a release plan. It is easy to understand the overall functional flow, easy to plan releases, but it is difficult to grasp the product context from an individual customer perspective.

2.4. Coverage analysis for the defined criteria

Goal Sketching [19] provides an assumption type of goal (/a/). However, it is not intended to regard all requirements as a hypothesis and but to specify the domain assumptions only which are identified. It focuses on system goal tree rather than a customer. We can identify related customers by finding its agents which implement the goal. Since TBD is allowed, updating requirements are possible during the development. Based on the article, supporting to use concrete scenarios is unclear. Moreover, by providing only a few simple notations, anyone can quickly learn and use.

RE-KOMBINE [20] also supports notation for domain assumption, but cannot distinguish requirements as hypotheses that require validation. The modeling result is in the form of a goal and task tree for the target system and doesn't expose customers directly. While it is not explicitly modeling the goals of the customer and the development company, it is possible to model each goal separately, since different goal trees can be considered simultaneously to find conflicts. It also

remind of customer goals target slowly change to company goals and this can model it separately - which help recognize this need allows conflicts and contradictions instead of completeness so that it can be updated and managed in parallel with development. The notation itself is simple, but it is difficult to learn because it uses formal methods internally to find contradictions.

HYPEX [4] and QCD [10] suggests the model for continuously accepting customer feedback from implementation incremental while considering requirements as hypotheses and concurrent with development. Case studies and notations cannot be found, so we cannot judge whether C2 to C6 criteria support. Besides, we cannot tell that it supports C3 because we could not distinguish between customer and development company goals.

Discover to Deliver [21] emphasizes infinite discover and deliver cycle during agile development and proposes seven predefined dimensions and tools for each dimension to elicit requirements comprehensively. It is not possible to determine whether criteria are supported from C2 to C6 because it is challenging to identify data on how to model and organize requirements. After each cycle, we can refine requirements by accepting feedback on implementation which based on the hypothesis of the previous cycle, and it suggests the customer perspective as a significant dimension, so we can conclude that it supports C1 and C2 implicitly. Besides, we cannot tell that it supports C3 because we could not distinguish between customer and development company goals.

Value Proposition Design [22] analyzes the customer's needs and pains and considers the product candidates to be satisfied with the analyzed customer to determine whether a solution fits by the set-based approach. In this process, analyzing needs and pains from customer perspectives and structuring them for each customer satisfy C2. Because it is a way of matching and incrementally improving solutions that are supposed to satisfy the customer, It meets C1 and C4. However, we considered C3 as an implicit because it is difficult to understand the goal or strategy of the development company only considering customer satisfaction. Scenarios and examples are not specified, but they can be used, and the notation is simple and easy to use.

Impact Mapping [5] consists of a mind-map of Why-Who-How-What format, and the connection line of each node is an assumption. In the center are the customers who play an essential role in achieving the goal, and the requirements are laid out in a tree shape for each customer. However, because the customer is considered as a means of achieving goals of the development company, the goal of the customer perspective is not evident. Because it came from the agile community, it can be done concurrently with development, it is simple to write, and it uses example-based requirements because the author suggested "Specification by Example" [16].

User Story Mapping [6] is also a way created by the Agile community and can be run concurrently with the development. The notation is simple, and each user story is expanded based on backbone end-to-end scenario. In this method, the particular scenario plays a significant role. Besides, the map can be modified for each iteration, and it can be examined whether the existing hypothesis was correct or the strategy should be modified. Each step of the backbone scenario can identify which customers are involved and which customers' requirements, it is not easy to see only the requirements for that customer based on a particular customer. Also, it is difficult to coexist with the goal of the development company perspective because it deals only with the goal of the customer perspective.

We analyzed how well each study covered the defined criteria which we have mentioned before and summarized the result into Table 1.

3. Proposed approach

To improve learning and adaptability, continuous and regular requirement changes must be handled efficiently. It is a complicated problem, so we need to combine multiple tools appropriately.

Table 1
Coverage of the defined criteria.
Legend: O (Explicit), △ (Implicit), ? (Unknown), X (Not supported).

Category	Related work	C1	C2	C3	C4	C5	C6
Academic	Goal Sketching [19]	△	△	X	O	?	O
	RE-KOMBINE [2]	△	?	△	△	?	△
	HYPEX [4], QCD [10]	O	?	X	O	?	?
Biz. analyst	Discover to Deliver [21]	△	△	X	O	?	?
	Value Proposition Design [22]	△	O	△	O	△	O
Agile & Lean	Impact Mapping [5]	△	O	△	O	△	O
	User Story Mapping [6]	△	△	X	O	O	O
	Proposed approach	O	O	O	O	O	O

3.1. Two main artifacts

There is no one size fits all approach. In our approach, essentially, we combine and use two tools. One is the Extended Impact Map for a visual representation of requirements, and the other is the Kanban board for visualization of progress.

3.1.1. Two types of Extended Impact Map

According to Table 1, we took Impact Mapping whose coverage is highest as a foundation of our new approach. Of course, Value Proposition Design also has the same coverage, but their target domain is not just RE but business analysis.

Impact Mapping is a useful tool, but there are some shortcomings: (1) It mixes two perspectives on Impacts. First is the customer perspective and the second is the development company perspective; (2) Cannot distinguish if a hypothesis was validated or not by just looking at the map; (3) Cannot keep track of changing progress. A map is just the static requirement snapshot at a specific time; (4) Cannot readily see how stories flow as a meaningful backbone story for customers.

To overcome these shortcomings, we extended or supplemented as following for each: (1) We separated the map into two, "Customer Impact Map" and "Development Company Impact Map", so that we could focus more on each point of view (in Fig. 1); (2) We added some styles to provide rich meanings; (3) The transition between snapshots needs to be managed and visualized. The use of a Kanban board can be an excellent choice to track the progress [25,26]; (4) User Story Mapping is a pivot tool which complements the Impact Map to show and build the backbone story.

Because agile development is customer-centered, in most cases we always use the "Customer Impact Map" ((a) in Fig. 1). At first, we identify customers (or stakeholders) by listing our product usage scenarios. After customer definition, the relative importance of the listed customers is to be analyzed according to their level of interest and influence [27], and the higher the priority, the thicker line between the goal and the customer becomes. During customer analysis, profiles of the customers are created and linked to their needs and pain [22]. Needs, pain, and solutions are described in specific scenarios. Using scenarios is much more lively and useful in deriving goals [16,28]. A dashed line is a hypothesis, and a solid line is a fact. Also, a thickness of line represents the importance of items at the same level. Customer Impact Map consists of three layers: goal, customer, and solution. Customers' goal groups all items, so customers are well exposed. Need and pain are other names of customer goal. We can quickly grasp their relationships.

We should consider the cost-effectiveness of delivering value to customers by using the resources of the development company. From the development company's perspective, the objective of achieving the customer's goals is an expectation that the customer might contribute to the development company's sub-goals. We can draw goal tree in a similar way of Customer Impact Map. The only difference is that the third layer is not needs, pain, and solutions for a customer but is sub-goal which is expected to a customer by the development company ((b) in

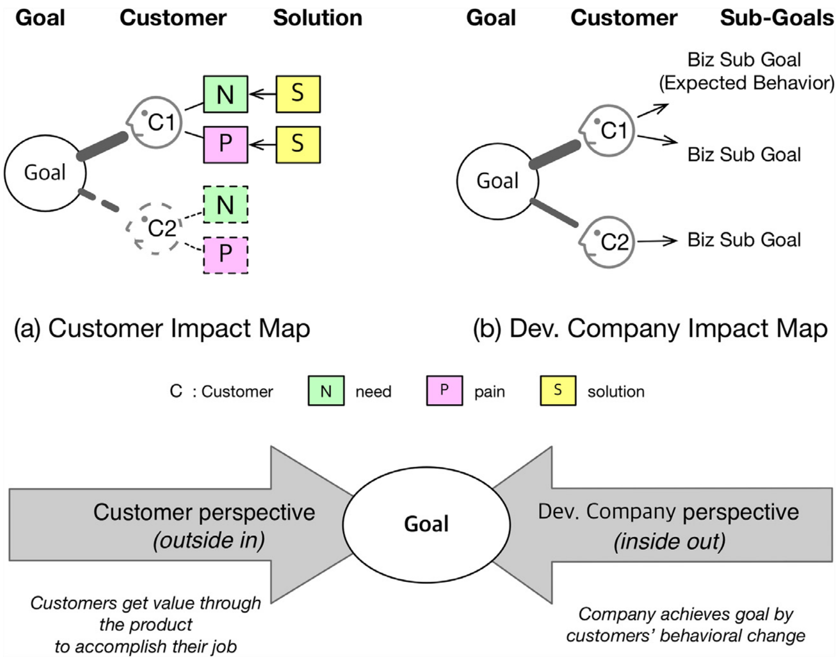


Fig. 1. Two types of extended Impact Map.

what does it mean that company achieve goals by customers behavioral change - what behavior change is company expecting and how does this change achieve their goal?

Fig. 1).

3.1.2. *Two types of process kanban board*

At the beginning of each iteration, we move high priority hypotheses (by customer value, risky, and so on) from the “Customer Impact Map” to the “Backlog” column in the Kanban board (Fig. 2). To more focus on customers, we took swim lanes for each customer in the Kanban board. Regardless of which agile process follows, when an iteration begins, the Validated Learning process [12] starts. After an iteration ends, depending on the results of the experiment, an item is added, modified, expanded or deleted in the “Customer Impact Map.” It is why the “Done” column in Fig. 2 is divided into three.

We need to maintain two Kanban boards for each track: the Discovery track focuses on requirements (customer layer), and the Delivery track focuses on implementation (solution layer) in parallel (Fig. 4). Business people and requirement analyst mainly use the Discovery Kanban, And a development team mainly uses the Delivery Kanban. One crucial thing is that only validated items from the Discovery Track should enter to the Delivery Track preferably (Fig. 3).

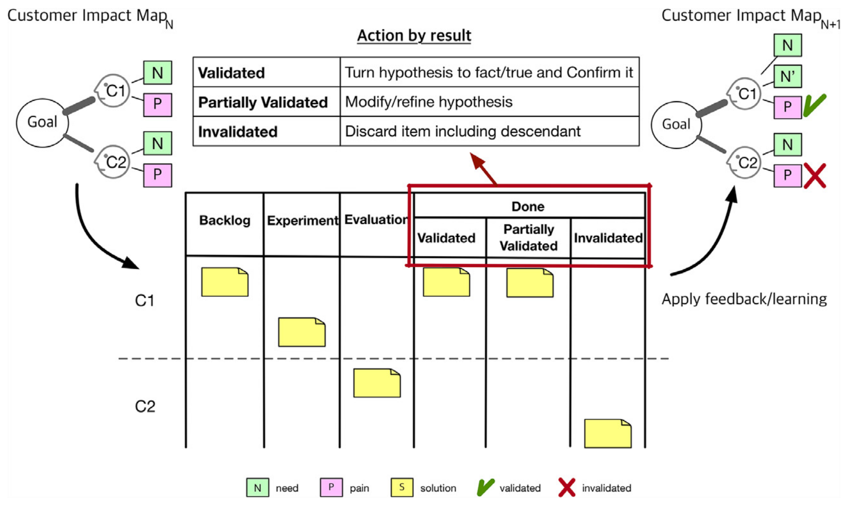
3.2. *Process of our approach*

We start with the Customer Impact Map. At the center, we set a goal and identify customers. Then, we identify need and pain of customers and connect our solutions. In these steps, we validate hypotheses and track their progress in the Discovery Kanban board. For identified customer and customer goals, the development company implements a solution in the Delivery Track and uses the Delivery Kanban board to track progress.

After each iteration, the development company watches the customer's response to the product and checks that company goals are progressing well. At that time, the Development Company Impact Map will be validated and updated. Fig. 4 depicts the simplified process of our approach which consists of five steps.

4. Case study

To help you understand, we provide a web-based digital signage monitoring system at the ACME (name of an arbitrary company) airport


Action by result

Validated	Turn hypothesis to fact/true and Confirm it
Partially Validated	Modify/refine hypothesis
Invalidated	Discard item including descendant

Fig. 2. Extended Impact Map and Kanban board.

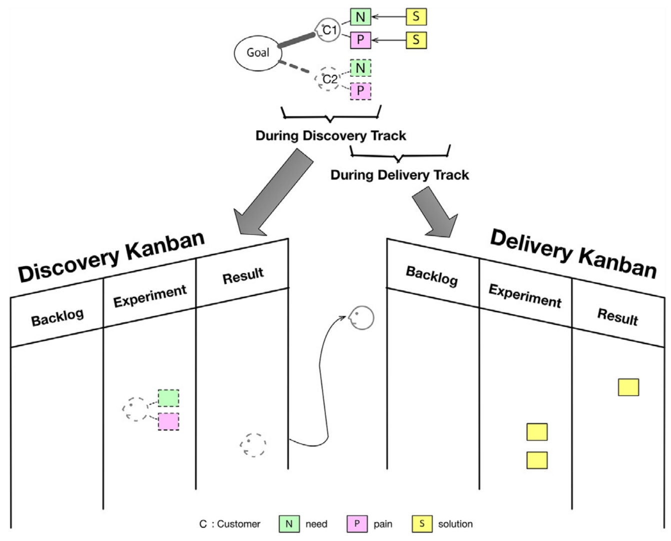


Fig. 3. Relationships between Customer Impact Map and two Kanban boards.

as an example, and the Fig. 5 is a brief description of the system:

The customer presents only the abstract requirements of the monitoring and control system and doesn't provide the detailed requirements yet. The customer seems to be keeping in mind the past solution of our company as the final system. However, for our business, an airport is the first domain to enter, so it is urgent to identify essential customer requirements considering domain characteristics.

On the other hand, the User Experience and Marketing departments within the development company provide additional needs based on their plans and independent of this project. The development team must work in tandem with the needs of uncertain and uncertain customer needs and the needs of the internal departments. Requirements are continually changing in the process of refining customers and requirements. Development team decides to handle this problem, a scrum of two weeks iteration (sprint) is applied.

A unit of analysis is a scrum team which consists of eight developers, four of them had experience in agile development, and the rest is the first time. The objective of this case is to verify that the proposed approach is useful considering six criteria. We will investigate whether the proposed approach can manage uncertain and rapidly changing requirements, whether it is easy to track progress, customer satisfaction as well as managing the internal goal of the development company.

4.1. Step 0. Define project goal

The first thing to do is to define the development company's goals for this project. In this example, By coordinating the opinions of various stakeholders within the company, we defined "*Implement all basic features of the ACME airport signage monitoring system until the 1st milestone*" as the goal.

4.2. Step 1. Identify customer

Identify the customers involved in achieving project goals. Customer identification can use predefined viewpoints [29] and can be derived by creating an end-to-end scenario that spans the lifecycle of the product [6]. Before the primary monitoring activities, customers can be further derived by creating scenarios at the early phase where the signage is installed and registered in the system for the first time, or at the resolution phase at the time of failure occurrence. The signage monitoring operator can be identified in the monitoring phase, and the field operator can be identified during the installation phase. Besides, it is possible to identify the parent contractor at the development phase before publishing the system. After the customers are identified, customer prioritizing should be performed [27] (See Fig. 6).

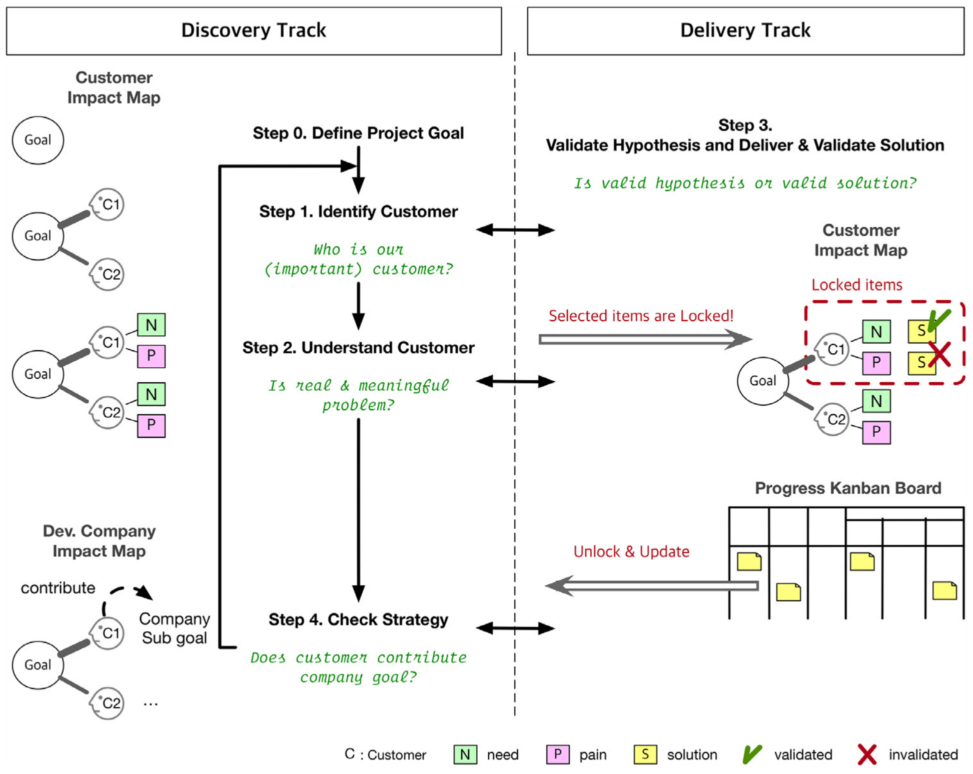
If any of the identified customers are unimportant or unrelated, delete them. To visually reveal which customer should focus more, high-priority customers will thicken the line between the central goal and the customer in the Customer Impact Map.

requirement allow space to change

would u say step 1 - 3 is like a feasibility analysis

what does it mean that company achieve goals by customers behavioral change - what behavior change is company expecting and how does this change achieve their goal?

is this like fig 4 where in step 3 the customer is having their hypothesis validated, and from that decide in ste 4 if they will achieve the company goal?



Discovery Track
Delivery Track
Customer Impact Map
Goal
C1
C2
N
P
S
Step 0. Define Project Goal
Step 1. Identify Customer
Who is our (important) customer?
Step 2. Understand Customer
Is real & meaningful problem?
Step 3. Validate Hypothesis and Deliver & Validate Solution
Is valid hypothesis or valid solution?
Customer Impact Map
Locked items
Selected items are Locked!
Progress Kanban Board
Unlock & Update
Dev. Company Impact Map
contribute
Company Sub goal
...
Step 4. Check Strategy
Does customer contribute company goal?
C : Customer
N need
P pain
S solution
validated
invalidated

Fig. 4. Process of our approach in dual track context.

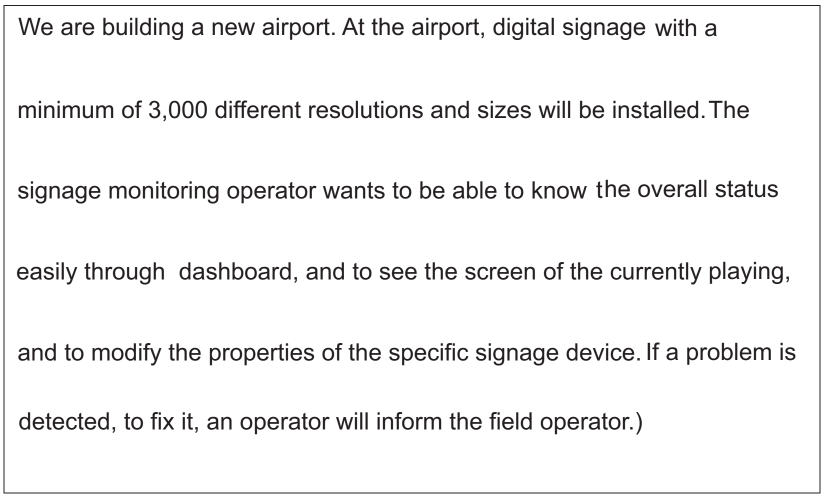

We are building a new airport. At the airport, digital signage with a minimum of 3,000 different resolutions and sizes will be installed. The signage monitoring operator wants to be able to know the overall status easily through dashboard, and to see the screen of the currently playing, and to modify the properties of the specific signage device. If a problem is detected, to fix it, an operator will inform the field operator.)

Fig. 5. Digital signage monitoring system at the ACME airport.

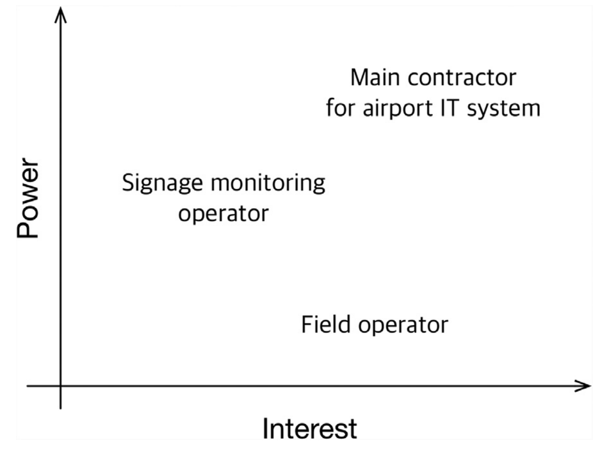


Fig. 6. Customer prioritization (using Mendelow's Matrix).

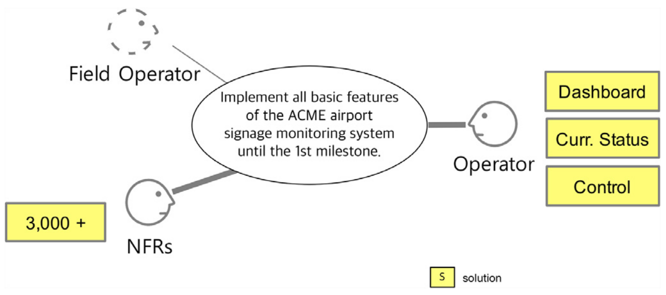

Field Operator
Implement all basic features of the ACME airport signage monitoring system until the 1st milestone.
Operator
Dashboard
Curr. Status
Control
3,000 +
NFRs
S solution

Fig. 7. Customer Impact Map; identified customers (with solutions).

We can identify two customers (actors): Signage Monitoring Operator (shortly 'Operator') and Field Operator. Until now, we have not known if one of the customers, the 'Field Operator' is under system boundary or not. So, we connected the Field Operator with a dashed line. The existence of the Field Operator itself is a hypothesis. However, the Operator is a vital customer of this project. She wants to have a dashboard, to see current status, and to control devices. Moreover, "more than 3,000 devices" is a quality criterion. So, we can create a abstract customer who named "NFRs" (nonfunctional requirements) for assigning quality criteria to it (Fig. 7).

4.3. Step 2. Understand customer

It can be said that it provides functional value to customers when a product help them to accomplish their work in the situation they are in [30]. The goal of defining the job and analyzing needs and pain is to understand customers' tasks and situations. You can also use persona in this step. This step is similar to the "Customer Segment" analysis of the Value Proposition Canvas [22].

From above brief description (Fig. 5), we can find three needs or solutions of the Operator: provide a dashboard, monitor current status, and control signage remotely. To know why they want and to validate if these needs are essential to them or not, we have an interview with some operators. An interview is a good tool for validating hypotheses during the Discovery track. After having the interview, we can add needs and pains and link with corresponding solutions (Fig. 8). We can identify need and pain from solution, and vice versa.

Let's assume that the pain point is not valid that the Operator cannot recognize a failed device lately (after observation or interview). If so, we can remove all descendant items including the pain point. It is a feature which has inherited from the original 'Impact Map' (Fig. 9).

This set of procedures is performed in the Discovery Track, so tracking and visualizing the progress using the Discovery Kanban board (Fig. 10).

Before the end of this step, we need to verify whether the customer has the needs and pain we identified and whether it makes sense for us to solve the problem. It is a waste to solve unreal and meaningless problems.

4.4. Step 3. Deliver and validate solution

Until Step 2, we could validate the hypotheses through interviews, observations, prototyping, and others without product development, but now we have a solution for actual problem solving and receive feedback from the customer about the result. Begin developing solutions only if they are essential customers or significant issues.

For example, let's look at the need to make it easy to find thousands of signage. There may be various alternatives such as "search by device

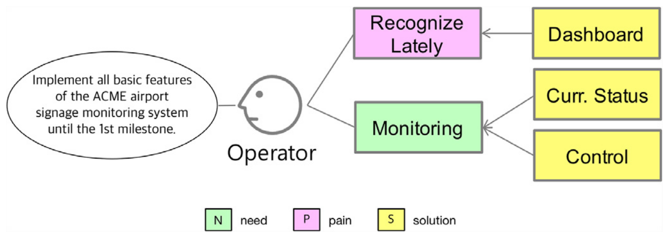

Implement all basic features of the ACME airport signage monitoring system until the 1st milestone.
Operator
Recognize Lately
Monitoring
Dashboard
Curr. Status
Control
N need
P pain
S solution

Fig. 8. Customer Impact Map; need and pain of the Operator.

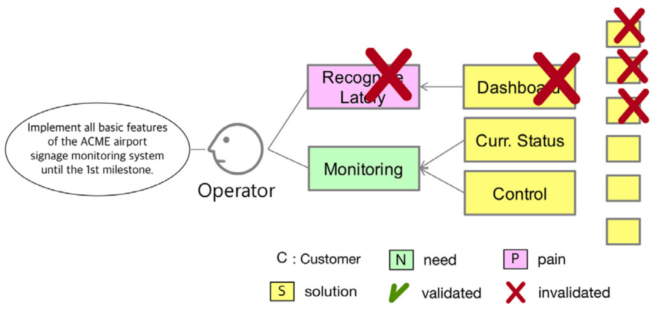


Fig. 9. Customer Impact Map; invalidate all descendant items.

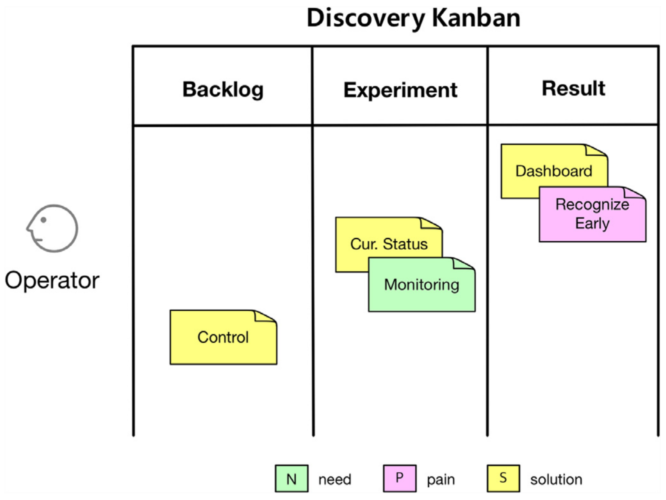


Fig. 10. Discovery kanban board.

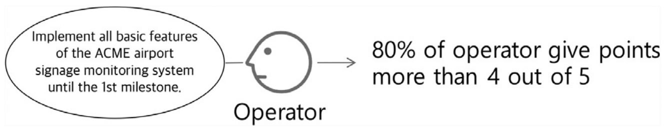

Implement all basic features of the ACME airport signage monitoring system until the 1st milestone.
Operator
80% of operator give points more than 4 out of 5

Fig. 11. Dev. company Impact Map; expected behavior of the operator.

name," "search by attribute (i.e., resolution, size, location)" to meet the needs. After thinking about which of the alternatives to implement, and after actual implementation, we receive feedback from the customer (or customer agent) to maintain, change or discard the solution. This set of procedures is performed in the Delivery Track, so tracking and visualizing the progress using the Delivery Kanban.

4.5. Step 4. Check strategy

The final step is to see if our clients contribute to the company's goals and how they will modify their strategies if they have problems contributing. It may take a long time for customers who get the value to make changes in the behavior company wants, and this step is skipped because it is difficult to measure. However, in the case of a startup, there is no time and money to repeat the iteration, so this step is crucial in determining the company's durability. toDO?? what does it mean^?

As described before, from the development company's perspective, the objective of achieving the Operator's goals is an expectation that the Operator might contribute to the development company's sub-goals. Key customers and company's sub-goals are depicted in the Development Company Impact Map. We should deliver the value the Operator can feel and satisfy the Operator. Fig. 11 shows the target of satisfaction which the development company wants to take quantitatively.

After several iterations, we can get Fig. 12. i.e., the requirement that only authenticated users should be able to access the system, the authorization and access controls are shared with both the signage monitoring operator and the field operator. To eliminate redundancy, we have created an abstract customer named "User." Besides, system-wide constraints or NFR attributes are categorized to abstract customers as "Constraints" or "NFRs." On the other hand, the map of expectations for customers of the development company is shown in Fig. 13.

mention of accommodating changes feel unavoidable and remind me of startups where their target changes frequent and so RE goals are changing frequent too - maybe we do need to develop better system that can accomodate to changes

5. Conclusions and future work

Agile development, which has to accommodate continuous change, must be able to organize the requirements better and efficiently accommodate the changes. In this paper, we defined six criteria that are missing in the agile REs that have been experienced in past agile projects. According to these criteria, we can tell that existing agile RE approaches could be suboptimal. Even the approach proposed by the authors did not completely satisfy the six criteria, but the authors tried to contain all criteria, it was a meaningful new attempt which could be applicable in the field.

After investigating case study, we conducted interviews with participants whether the new approach is useful for managing requirements or not (first-degree method). The qualitative result we had is: The hypothesis and validated hypothesis were visible at a glance, making it easy to grasp the progress; how many hypotheses are remaining. It was nice not to waste resources developing features based on validated hypotheses. Requirements were defined based on specific scenarios to facilitate verification. Working with User Story Mapping was good.

However, also, various applications of the industry should be identified, and application examples and effectiveness of other practitioners should be validated for the same approach. Besides, since the case study is only for the first milestone for a limited period, there is a limitation that the issue in the whole process of production is not revealed. Moreover, customers of the Development Company Impact Map are the same as customers of the Customer Impact Map, and it seems necessary to replace them with the internal stakeholder of the

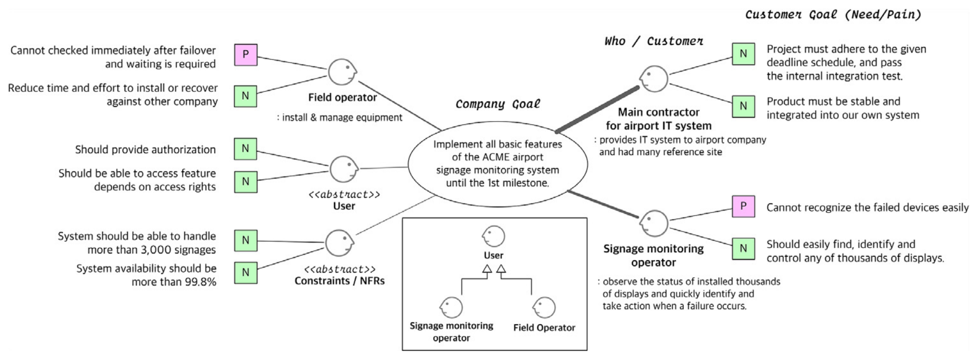

Cannot checked immediately after failover and waiting is required
P
Reduce time and effort to install or recover against other company
N
Field operator
: install & manage equipment
Should provide authorization
N
Should be able to access feature depends on access rights
N
<<abstract>>
User
System should be able to handle more than 3,000 signages
N
System availability should be more than 99.8%
N
<<abstract>>
Constraints / NFRs
Company Goal
Implement all basic features of the ACME airport signage monitoring system until the 1st milestone.
User
Signage monitoring operator
Field Operator
Who / Customer
Customer Goal (Need/Pain)
Main contractor for airport IT system
: provides IT system to airport company and had many reference site
N
Project must adhere to the given deadline schedule, and pass the internal integration test.
N
Product must be stable and integrated into our own system
Signage monitoring operator
: observe the status of installed thousands of displays and quickly identify and take action when a failure occurs.
P
Cannot recognize the failed devices easily
N
Should easily find, identify and control any of thousands of displays.

Fig. 12. Customer Impact Map.

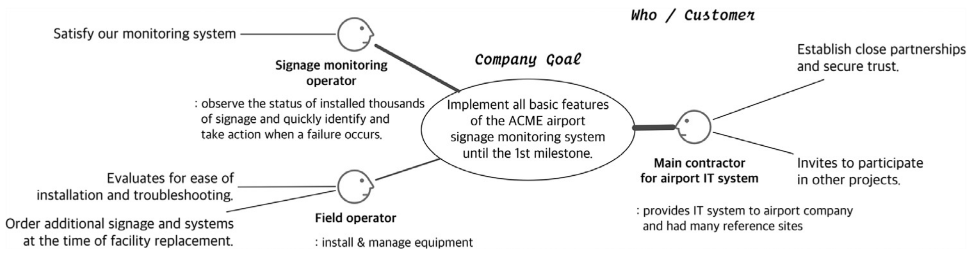

Who / Customer
Company Goal
Implement all basic features of the ACME airport signage monitoring system until the 1st milestone.
Satisfy our monitoring system
Signage monitoring operator
: observe the status of installed thousands of signage and quickly identify and take action when a failure occurs.
Evaluates for ease of installation and troubleshooting.
Order additional signage and systems at the time of facility replacement.
Field operator
: install & manage equipment
Establish close partnerships and secure trust.
Invites to participate in other projects.
Main contractor for airport IT system
: provides IT system to airport company and had many reference sites

Fig. 13. Development Company Impact Map.

development company.

In future research directions, it is necessary to acquire more practical examples of the industry applying the proposed method and further refine step-by-step activities. Also, when applied to a large organization, we expect to be able to shorten the time and effort of interview to various customers' voices directly through the Crowd RE method and verifying the hypothesis through crowd review.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2017R1D1A1B03034279).

References

[1] T.S. Mendes, M.A. de F Farias, M. Mendonça, H.F. Soares, M. Kalinowski, R.O. Spínola, Impacts of agile requirements documentation debt on software projects, Presented at the the 31st Annual ACM Symposium, Pisa, Italy, 2016, pp. 1290–1295.
[2] I. Inayat, S.S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, *Comput. Hum. Behav.* 51 (2015) 915–929.
[3] K. Boness, R. Harrison, K. Liu, Goal sketching: an agile approach to clarifying requirements, *J. Adv. Softw.* (2008).
[4] H.H. Olsson, J. Bosch, From opinions to data-driven software R&D, Presented at the Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on, 2014, pp. 9–16.
[5] G. Adzic, *Impact Mapping.* Provoking Thoughts, 2012.
[6] J. Patton, P. Economy, . User Story Mapping, O'Reilly Media, Inc, 2014.
[7] M. Lee, Just-in-time requirements analysis—the engine that drives the planning game, Presented at the Proceedings 3 rd International Conference Extreme Programming and Agile Processes in Software Engineering, 2002, pp. 138–141.
[8] R.D. Stacey, Complexity and Creativity in Organizations, Berrett-Koehler Publishers, 1996.
[9] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas, Principles behind the Agile Manifesto, [Online]. Available: http://agilemanifesto.org/principles.html.
[10] H.H. Olsson, J. Bosch, Towards continuous customer validation - a conceptual model for combining qualitative customer feedback with quantitative customer observation, ICSOB 210 (13) (2015) 154–166.
[11] S.G. BlankCustomer Development Methodology, Stanford Technology Ventures Program's Roundtable on Enterpreneurship Education, (2008).
[12] E. Ries, The Lean Startup, Crown Business, 2011.
[13] E. Lagerstedt, Business strategy: are you inside-out or outside-in? *INSEAD Knowl.* (August-2014) [Online]. Available: https://knowledge.insead.edu/blog/insead-blog/business-strategy-are-you-inside-out-or-outside-in-3515.
[14] S. Lerén, I. Domingues, Impact-driven scrum delivery, Presented at the Global SCRUM Gathering, Phoenix, 2015.
[15] J. Patton, Agile Development and Scrum Quick Reference, [Online]. Available: http://jpattonassociates.com/agile-development-and-scrum-quick-reference/.
[16] G. Adzic, Specification by Example. Manning Publications, 2011.
[17] Ben Linders, "Delivering Software with Water-Scrum-Fall," Nov. 2015.
[18] Z. Racheva, M. Daneva, A. Herrmann, A conceptual model of client-driven agile requirements prioritization: results of a case study, Presented at the Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010, p. 39.
[19] K. Boness, R. Harrison, Goal Sketching: towards agile requirements engineering, Presented at the International Conference on Software Engineering Advances (ICSEA 2007, 2007, p. 71.
[20] N.A. Ernst, A. Borgida, I.J. Jureta, J. Mylopoulos, Agile requirements engineering via paraconsistent reasoning, *Inf. Syst.* 43 (C) (Jul. 2014) 100–116.
[21] E. Gottesdiener, M. Gorman, Discover to Deliver, EBG Consulting, Inc., 2012.
[22] A. Osterwalder, Y. Pigneur, G. Bernarda, A. Smith, Value Proposition Design, John Wiley & Sons, 2014.
[23] A. Osterwalder, Y. Pigneur, Business Model Generation, John Wiley & Sons, 2010.
[24] J. Shore, Beyond Story Cards: Agile Requirements Collaboration, [Online]. Available: http://www.jamesshore.com/Presentations/Beyond%20Story%20Cards.html.
[25] D.J. Anderson, Kanban: Successful Evolutionary Change For Your Technology Business, Blue Hole Press, 2010.
[26] H. Kniberg, Lean from the Trenches: Managing Large-Scale Projects with Kanban, Pragmatic Bookshelf, 2011.
[27] A. Mendelow, Stakeholder mapping, Presented at the Proceedings of the 2nd International Conference on Information Systems, Cambridge, MA, 1991.
[28] C. Rolland, G. Grosz, R. Kla, Experience with goal-scenario coupling in requirements engineering, IEEE (1999) 74–81.
[29] I. Sommerville, P. Sawyer, Viewpoints: principles, problems and a practical approach to requirements engineering, *Ann. Softw. Eng.* 3 (1997) Jan.
[30] A. Klement, Replacing the User Story With the Job Story – Jobs to be Done, . [Online]. Available, November-2013. https://jtbd.info/replacing-the-user-story-with-the-job-story-af7cdee10c27#.i6750x1yu.