http://dx.doi.org/10.14400/JDC.2016.14.11.297

절약형 애자일 재사용을 향한 입력 대기열 항목의 재사용

김지홍
가천대학교 IT대학 컴퓨터공학과

Reuse of Input Queue Item Towards Economical Agile Reuse

Ji-Hong Kim
Dept. of Computer Engineering, College of IT, Gachon University

요 약 본 연구의 목적은 애자일 개발 초기 단계에서 재사용의 지원을 통하여 애자일 방법과 소프트웨어 재사용의 통합이다. 애자일 방법과 소프트웨어 재사용은 각기 다른 원칙과 프랙티스를 가지고 있지만, 2가지 방법은 모두 개발 시간 및 비용 단축과 생산성 향상과 같은 공통된 목표를 가지고 있다. 이 방법들은 급변하는 환경에서 요구사항 변경의 수용이나 빠른 개발 요구에 대한 해결책으로 기대되고 있다. 본 연구에서는 절약형 애자일 재사용과 유형을 식별하고, 혼합 애자일 방법 초기 단계에서 칸반 보드의 입력 대기열을 위한 재사용 기술을 연구하였다. 본 연구의 결과를 통하여, 스크럼과 칸반 혼합 방법에서 입력 대기열 항목에 백로그 팩토링 기술을 사용하여 애자일 방법과 소프트웨어 재사용을 결합할 수 있었다. 아울러, 제안된 방법을 이클래스 응용에 적용하여 입력 대기열 항목의 재사용과 2가지 접근 방법의 통합을 보였다. 본 연구는 애자일 개발 초기에서의 재사용에 기여를 의도하였다. 앞으로, 절약형 애자일 재사용 도구를 개발할 계획이다.

주제어 : 소프트웨어 공학, 소프트웨어 재사용, 스크럼, 칸반, 애자일 방법

Abstract The aim of the study is to combine software reuse with agile methods through reuse in the early stage of agile development. Although agile methods and software reuse have different practices and principles, these methods have common goals, such as reducing development time and costs and improving productivity. Both approaches are expected to serve as viable solutions to the demand for fast development or embracing requirement changes in the rapidly changing environments. In the present paper, we identify economical agile reuse and its type and study a reuse technique for input queue in Kanban board at the early stage of hybrid agile methods. Based on our results, we can integrate software reuse with agile methods by backlog factoring for input queue item in the hybrid Scrum and Kanban method. The proposed technique can be effectively applied to e-class applications and can reuse the input queue items, showing the combination of the two approaches. With this study, we intend to contribute to reuse in the early stage of agile development. In the future, we plan to develop a software tool for economical agile reuse.

Key Words : Software Engineering, Software Reuse, Scrum, Kanban, Agile Methods

Received 29 September 2016, Revised 1 November 2016
Accepted 20 November 2016, Published 28 November 2016
Corresponding Author: Ji-Hong Kim(Gachon University)
Email: wiskjh@gachon.ac.kr

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

오늘날 소프트웨어 개발자는 급변하는 기술 및 요구 그리고 경쟁에 대처하기 위해 끊임없이 새로운 방법을 추구하고 있다. 이를 위한 새로운 접근에는 빠른 시장진입, 개발 비용 절약, 생산성 향상을 지원하는 애자일 방법과 소프트웨어 재사용 기술이 있다[1].

애자일 방법은 반복 점증적 방법을 기반으로 잦은 검사, 적응, 협동 작업, 자기 조직화를 통하여 요구사항 변경의 수용과 신속한 고품질의 소프트웨어 개발을 허락하기 때문에 많은 산업체에서 채택하고 있다[2]. 여러 애자일 방법 가운데 스크럼은 오늘날 가장 인기 있는 방법론이며 칸반 방법은 작업의 시각화와 최적화 지원으로 적용이 늘어나고 있다[3]. 이에 따라, 스크럼 개발에서 칸반으로 전환이 점차 등장하고 스크럼과 칸반의 장점을 혼합한 애자일 방법의 사용이 나타나고 있다[3,4].

소프트웨어 재사용은 이전에 작성한 산출물 또는 경험을 새로운 프로젝트에 다시 사용함으로서 빠른 개발과 비용 절감을 허락하는 기술로, 꾸준히 채용되고 있는 소프트웨어 기술이다. 이는 코드뿐만 아니라 요구, 분석, 설계, 테스트, 어플리케이션에서도 가능하다[5,6,7]. 아울러, 칸반의 입력 대기열 항목은 칸반 방법에서의 요구사항으로 적극적인 재사용을 위한 새로운 대상이다.

애자일 방법과 소프트웨어 재사용의 통합은 서로의 장점을 공유하고 단점을 보완해주는 접근으로 간주되고 있다[2]. 이들을 통합하는 연구들은 주로 소프트웨어 프로덕트라인 공학에 애자일 도입의 연구들이고, 애자일 방법에서의 재사용 지원은 강조되지 않고 있다. 애자일 방법과 칸반이 가지고 있는 장점의 활용은 낭비의 억제나 작업의 최적화를 지원하기 때문에 애자일 개발에서 절약형 재사용이 가능하다.

본 연구는 절약형 소프트웨어 재사용과 유형을 식별하고 스크럼과 칸반의 혼합 애자일 방법에서 칸반 보드의 입력 대기열을 위한 재사용 방안을 제안하였다.

본 논문의 구성은 2장에서 기초 연구에 속하는 애자일 방법, 소프트웨어 재사용과 백로그 팩토링, 스크럼과 칸반에 관해 알아본다. 3장에서는 절약형 애자일 재사용을 향한 칸반 보드의 입력 대기열에서 재사용 기술을 제안하고, 4장에서는 프로토타이핑을 보인다. 5장에서는 관련 연구를 비교하고, 6장에서 결론을 기술한다.

2. 기초 연구

2.1 애자일 방법

애자일 방법은 폭포수형 개발 방법의 경직성을 개선하고 급변하는 요구에 대응하기 위해 나타났다. 이들은 도구보다는 개인의 상호작용, 문서보다는 동작하는 소프트웨어, 계약협상보다는 고객협력, 계획대로 수행하는 것보다는 변화에 적응과 같은 애자일 원칙에 동의하는 방법들을 말한다[5]. 대표적인 애자일 방법에는 익스트림 프로그래밍(XP), 스크럼, 칸반, 린 등이 있으며, 국내에서도 적용이 나타나고 있다[8]. 오늘날 이 가운데 스크럼이 가장 많이 채택되고 있으며 빠른 시장 진입시간, 품질과 생산성의 향상, 변경의 수용, 우선순위 재조정의 관리와 같은 장점으로 소프트웨어 개발 업계에서 사용이 증가하고 있다[9]. 한편 애자일 방법은 급변하는 환경에서, 미래의 재사용을 위한 투자를 위험으로 간주하기 때문에 재사용 가능한 산출물 개발에는 소극적이다[1].

2.2 소프트웨어 재사용과 백로그 팩토링

소프트웨어 재사용이란 기존의 소프트웨어 산출물을 새로운 소프트웨어 시스템 개발의 각종 단계에서 다시 사용하여, 개발 기간과 비용을 줄이고 고품질의 소프트웨어 인도를 가능하게 하는 핵심적 기술로 인식되고 있다[5,6]. 이런 접근 가운데 프로덕트라인 공학(Product Line Engineering : PLE)은 기능적으로 유사한 제품이 공유하는 아키텍처 또는 컴포넌트들의 공통점을 재사용하는데 기반하고 있다[10].

백로그 팩토링은 애자일 소프트웨어 개발에서 요구사항이나 작업을 결정할 때, 이전에 수행된 동일 또는 유사 스토리나 태스크를 간단하게 발견하여 재사용 가능한 항목을 제시하는 활동 및 기술이다[11].

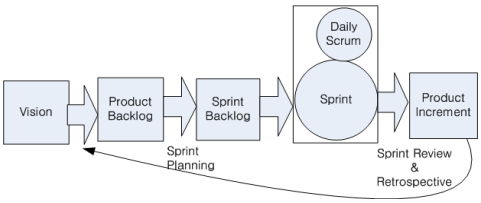
그동안 애자일 방법과 소프트웨어 재사용의 통합의 대부분 노력은 제품군의 재사용을 추구하는 프로덕트라인 공학에서 재사용의 도입에 집중되어있다[1,12]. 반면에 스크럼이나 칸반 같은 애자일 방법에 재사용의 도입은 많지 않다.

애자일 방법과 소프트웨어 재사용은 각각 새로 주목받는 기술과도 자주 통합 형태를 보이고 있다. 헬스케어, 홈케어 등 다양한 분야에 적용되고 있는 사물인터넷(IoT)[13,14]에도 애자일 접근이 나타나고 있으며[15], 안

드로이드 관련 연구[16,17]에서도 여러 재사용과의 결합이 있다[18,19]. 소프트웨어 재사용과 애자일 방법들은 소프트웨어 개발을 향상시키는 중요한 기술이다.

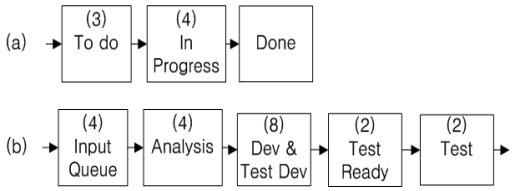
2.3 스크럼과 칸반

스크럼은 [Fig. 1]과 같은 반복 활동을 통해 진행된다[20]. 먼저, 비전(vision) 실현에 필요한 각종 요구사항을 제품 백로그에 추가하면, 스프린트 계획회의를 통해 개발할 항목을 제품 백로그로부터 선택한다. 팀은 선택된 항목을 작업 수준의 태스크로 나누어 스프린트 백로그를 작성하고 작업을 할당한다. 스프린트를 통해 작업이 수행되고 제품 증분을 갖게 된다. 매회의 스프린트 종료 후, 스프린트 검토(review)를 하고 회고(retrospective)를 통해 개선의 시간을 갖는다.



[Fig. 1] Scrum process

칸반은 스크럼과 달리 반드시 따라야할 규범이 적은 방법이다. 이는 스프린트와 같은 이터레이션, 스크럼 마스터나 제품 소유자와 같은 역할, 일일 스텐드업이나 회고, 데모와 같은 회의를 규정하지 않고 있지만, 필요하다면 사용할 수 있다[21]. 칸반에서는 소프트웨어 개발 작업의 시각화를 위해 [Fig. 2]와 같은 칸반 보드가 활용된다.



[Fig. 2] Sample Kanban Board

할 일(To do), 작업 중(In progress), 완료(Done)의 단순한 형태로 부터 입력 대기열(Input Queue), 분석, 개발 및 개발 테스트, 테스트 준비, 테스트와 같이 좀 더 세분화된 작업 단계의 컬럼(column)을 가질 수 있다[21,22].

칸반 방법에서 우선적 처리가 필요한 사항은 칸반 보드의 맨 왼쪽 열에 있는 할 일 또는 입력 대기열 컬럼으로 옮겨진다. 이들은 일반적으로 사용자 스토리 양식의 카드나 포스트잇에 작성되고, 작업은 각 컬럼에 숫자로 표시되는 진행 중 작업 제한(Work In Progress(WIP) Limit) 범위 내에서 당겨지는 방식으로 운영된다. 입력 대기열, 분석, 설계와 같은 작업 단계에서는 각각의 특화된 업무 및 재사용이 수행된다.

한편, 스크럼의 프랙티스와 칸반의 원칙을 혼합한 애자일 방법은 유지보수와 제품 개발을 함께하는 개발 환경에서 2가지 방법의 장점을 취하면서 유연성을 허락한다. 때때로 이런 혼합형 애자일 방법을 스크럼반(Scrumban)이라고 부르기도 한다[23,24]. 이들이 제공하는 반복 개발이나 칸반의 지속적 흐름 및 작업 제어는 쓸모없게 되는 작업 또는 출시에 배제되는 구현과 같은 각종 노력들의 허비를 억제하며[23] 절약형 재사용을 촉진한다.

3. 절약형 애자일 재사용과 입력 대기열의 지원

3.1 절약형 애자일 재사용

애자일 방법과 칸반이 제공하는 반복적 개발과 작업의 제어를 적절히 이용하고 개발에 나타나는 유사한 노력과 산출물을 재활용하면, 자원 및 비용을 절약하는 개발이 용이하다. 절약형 애자일 재사용이란 급변하는 인터넷 시대에 재사용을 위한 별도의 투자나 불필요한 노력 그리고 낭비를 최소화하면서 애자일 개발활동을 하고 있거나 한 김에 재사용하는 적극적 기술을 말한다. 이는 작업 상태의 관점에서 <Table 1>과 같이 할일 재사용, 진행 중 재사용, 완료 재사용의 유형을 갖는다.

할일 재사용(Todo Reuse)은 애자일 개발 초기에 할 일을 다루면서 재사용의 추진을 의미한다. 대화식 및 면대면 방식의 애자일 개발에서 작업의 시작을 대기하고 있는 요구사항에 대한 재사용 활동이 수행된다. 이는 주로 사용자 스토리 또는 입력 대기열 항목 수준의 재사용을 통해 이루어진다. 진행 중 재사용(In-progress Reuse)은 애자일 개발을 진행하면서 재사용을 위한 개발과 재사용을 통한 개발이 행하여지며, 공통적 사용이 가능한

표준 및 프래임워크 지원과 패턴 또는 라이브러리의 재사용이 장려된다. 이는 빠른 개발뿐만 아니라 일관성 있는 어플리케이션의 사용을 허락한다. 완료 재사용(Done Reuse)은 가장 많이 알려진 재사용 형태로서 애자일 방법을 통하여 이미 개발된 다양한 산출물에 대한 재사용을 말한다.

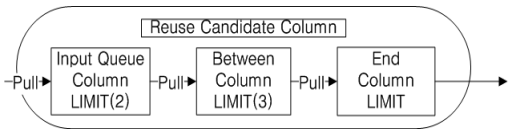
〈Table 1〉 Types of Economical Agile Reuse

Type	Description
Todo Reuse	Promote reuse for requirements
In-Progress Reuse	Take advantage of current work for reuse
Done Reuse	Reuse with various artifacts

3.2 애자일 보드와 재사용

3.2.1 절약형 애자일 재사용을 촉진하는 애자일 보드

애자일 보드는 절약형 애자일 재사용 개발의 중심에서 각 단계의 작업과 전체 작업을 파악하고 흐름의 제어와 최적화를 하면서 3.1절에서 식별한 3가지 유형의 재사용을 촉진하는 확장된 칸반 보드이다. 이 보드는 [Fig. 3]과 같이 입력 대기열 컬럼, 중간 컬럼, 종료 컬럼, 재사용 후보 컬럼의 기본 형태를 갖는다. 각 컬럼에서는 특화된 작업과 재사용이 가능하며 가용 범위 내에서만 당김 방식으로 작업을 끌어올 수 있다.

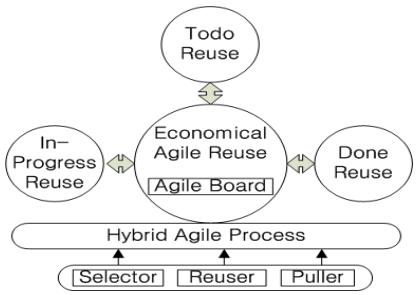


[Fig. 3] Agile Board

입력 대기열 컬럼(Input Queue Column)은 현 비즈니스 가치에 따라 가장 우선적으로 요구되는 항목들을 포함하고 있으며, 입력 대기열 항목은 적시 처리를 위해 제품 백로그로 부터 선택된 항목 또는 입력 대기열 컬럼으로 옮겨진 항목을 말한다. 이들은 준비의 정의(definition of ready)가 만족되는 일감들이다. 중간 컬럼(Between Column)은 입력 대기열 컬럼과 종료 컬럼 사이에 나타날 수 있는 여러 작업 단계 변형들의 단순 표현이다. 조직의 워크플로우에 따라 [Fig. 2](b)처럼 여러 개로 세분된 컬럼을 가질 수 있다. 이 컬럼은 프로젝트의 실질적 개발이 행해지는 부분이므로 설계, 구현, 테스트와 같은 작업이 나타난다. 종료 컬럼(End Column)에는 완료의 정의(definition of done)가 만족되고 출시를 기다리는 완료된 작업들이 옮겨진다. 재사용 후보 컬럼(Reuse Candidate Column)은 입력 대기열 컬럼이나 중간 컬럼에서의 재사용 지원을 위하여 개발 기간 동안의 여러 회의를 통하여 재사용 가능한 후보 자산이 추천되는 영역이다.

3.2.2 애자일 보드를 이용하는 재사용의 지원

절약형 애자일 재사용은 [Fig. 4]와 같이 스크럼과 칸반의 하이브리드 애자일 프로세스에서 애자일 보드를 중심으로 선택자, 재사용자, 당김자의 협동으로 이루어진다.



[Fig. 4] Component of Economical Agile Reuse

선택자(Selector)는 즉시 시작 할 준비가 된 최 우선순위의 작업을 결정한다. 이를 위하여 작업, 비즈니스 가치, 프로젝트에 대한 설명과 충분한 이해 및 고려가 필수이다. 재사용자(Reuser)는 입력 대기열 컬럼과 중간 컬럼에 있는 작업 항목을 위한 재사용을 수행한다. 이는 애자일 개발을 하고 있거나 한 김에, 여러 형태의 산출물 재사용을 지원한다. 당김자(Puller)는 선택자가 선정한 작업을 새로운 컬럼의 작업 제한 개수가 허락하는 범위 내에서만 당겨와 병목현상 또는 자원 낭비의 발생을 제어하는 역할을 한다.

3.3 입력 대기열을 위한 백로그 팩토링과 재사용 프로세스

애자일 개발의 모든 요구는 3.2.1에서 식별한 입력 대기열 컬럼으로부터 시작된다. 이 단계에서는 요구사항의 재사용뿐만 아니라 이어지는 분석, 설계, 구현에서도 높은 재사용 가능성을 열고 있다.

3.3.1 입력 대기열을 위한 백로그 팩토링

대화중심의 애자일 방법으로 소프트웨어 개발 작업 수행 중에 발견되는 이전과 동등 또는 유사한 요구사항에 대하여 2.2절에 소개된 백로그 팩토링의 적용이 가능하다. 이 기술은 3.2.2절에서 식별한 재사용자에 의해 수행된다.

입력 대기열을 위한 백로그 팩토링의 재사용 처리는 [Fig. 5]와 같다. 먼저, GetPriorityItem(Backlog)을 통해 최우선 순위로 수행될 후보 입력 대기열 항목(QueItem)을 백로그로부터 당겨오고, 이에 대해 BacklogFactoring()을 수행한 결과인 UpdateItem을 AddInputQ()를 통해 업데이트하고 백로그 팩토링을 마친다.

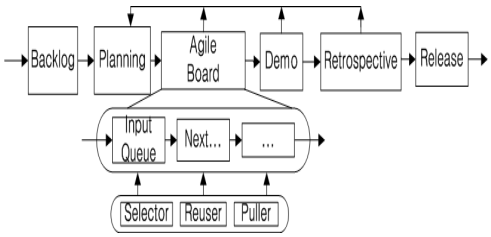
```
BacklogFacoringForInputQ() {
  QueItem = GetPriorityItem(Backlog);
  BacklogFactoring(QueItem, UpdateItem);
  AddInputQ(UpdateItem);
}
QueItem=GetPriorityItem(Backlog) {
  /* Retuen top priority QueItem from Backlog */
}
BacklogFactoring(QueItem, UpdateItem) {
  FindAsset(QueItem, OldItem);
  Decide(QueItem, OldItem, UpdateItem);
}
FindAsset(Tmp, Old) { /* Find Old simlar to Tmp */ }
Decide(Tmp, Old, Update) {/* Update if reusable */ }
AddInputQ(Item) {  /* Pull Item into InputQ */ }
```

[Fig. 5] Input queue item with Backlog Factoring

세부적으로, 임시 입력 대기열 항목(QueItem)을 입력받아 백로그 팩토링을 수행하는 BacklogFactoring()은, FindAsset()을 통해 이전에 수행된 유사 항목(OldItem) 유무를 간단히 검사하고, Decide()을 통해 재사용 가능한 항목 여부를 결정하여 입력 대기열 항목(UpdateItem)을 제시한다.

3.3.2 입력 대기열 항목을 위한 재사용 지원 프로세스

프로젝트에 따라 프로세스의 변형이 가능한 스크럼과 칸반 방법의 혼합 애자일 개발에서 절약형 애자일 재사용을 지원하는 프로세스의 예는, [Fig. 6]과 같이 백로그로 부터 시작하여 출시로 끝나기까지 중간에 계획회의, 데모, 회고의 반복 작업으로 구성된다.



[Fig. 6] Reuse Process example for Input Queue Item

계획회의(Planning)를 통하여 프로젝트의 목표 및 범위가 파악되고 이해되는 상황에서 새로운 입력 대기열 항목의 보충이 필요한 경우, 3.2.2절에 소개된 선택자(Selector)에 의하여 백로그(Backlog)로 부터 최우선 순위의 후보 입력 대기열 항목이 선택된다.

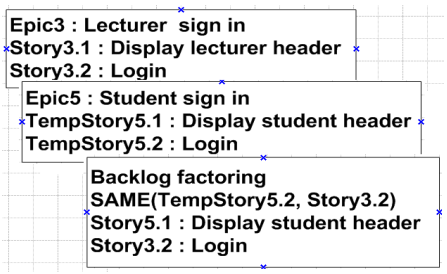
이어서 재사용자(Reuser)에 의해서 유사 항목이 발견되면 재사용 가능한 항목이 제시되고 당김자(Puller)에 의해 애자일 보드(Agile Board)의 입력 대기열(Inout Queue) 컬럼으로 옮겨진다. 이후, 이들 항목은 필요할 때 중간 컬럼에 해당되는 다음 단계(Next)로 당겨지고 처리된다.

4. 적용과 프로토타이핑

본 장은 3장에서 제안한 입력 대기열 항목의 재사용을 위해 백로그 팩토링 기술을 이클래스(e-class) 응용에 적용하고 백로그 수준의 프로토타이핑을 보인다.

- 동일 스토리의 적용

재사용 가능한 입력 대기열 항목의 제시를 위하여, 선행되는 몇 개의 항목의 작업을 마친 상황에서 동일 스토리에 대한 팩토링을 먼저 살펴본다.

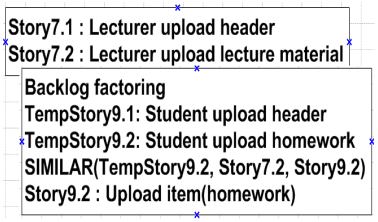


[Fig. 7] Backlog Factoring(Same Story)

[Fig. 7]의 상위 부분은 기 수행된 여러 스토리 가운데 재사용 대상이 되는 Epic3가 적절한 크기로 분해된 2개의 스토리인 Story3.x를 나타내고 있다. 중간 화면은 새로운 작업을 위해 최 우선순위로 선정된 후보 항목으로 백로그 팩토링 대상이되는 Epic5와 2개로 분해된 임시 스토리 TempStory5.x가 있다. 3번째 화면은 3.3.1절에 소개한 백로그 팩토링의 단계를 보인다. 먼저, GetPriorityItem()을 통해 우선순위로 처리할 TempStory5.2를 얻어 BacklogFactoring()을 수행한다. 세부적으로 FindAsset()과 Decide()에 의하여 단순 공통성 검사 결과인 SAME(TempStory5.2, Story3.2)를 수행하고 재사용 가능한 항목인 Story3.2의 할당을 보인다.

• 유사 스토리의 적용

여러 선행되는 항목의 작업을 끝낸 후 새로운 작업 선정에 유사 스토리가 적용되는 백로그 팩토링을 살펴본다. [Fig. 8]의 상단은 이미 수행된 Story7.x를 보여준다. 2번째 화면에는 백로그 팩토링을 위한 Epic9에 대한 임시 스토리인 TempStory9.1과 TempStory9.2가 제시되었다. TempStory9.2와 Story7.2는 유사성(SIMILAR)이 식별되어 재사용되고, TempStory9.2와 유사한 스토리인 Story9.2의 할당을 나타내고 있다.



[Fig. 8] Backlog Factoring(Similar Story)

5. 관련 연구 비교

애자일 방법과 재사용 기술을 통합하는 최근의 연구는 주로 PLE와 연관된 노력들이다. 대표적으로, PLE에 애자일 원칙을 통합한 Carbon[1]의 연구와, Tian과 Cooper[12]의 연구가 있다. Diaz[25]와 Kircher[2]는 PLE와 스크럼의 통합을 연구하였다. 이들은 PLE를 위한 재사용인데 비하여, 본 연구는 애자일 방법을 위한 재사용 연구이다. Martini[26]는 같은 통합을 위한 다른 접근으로서 애자일의 속도와 재사용을 촉진하는 요인과 억제하는 요인을 식별하고, 영향 인자의 가이드를 통한 프로세스 개선을 도모하였다. 이어지는 연구에서는[27] 빠른 속도와 재사용을 위한 의사소통에 영향을 끼치는 요인을 보고하였다.

최근, Silva[28]는 여러 애자일과 PLE 통합의 노력들을 복수 방법으로 연구하였다. 이들 발견 가운데 XP와 스크럼이 대표적 재사용 기술인 PLE와의 통합에 가장 많이 사용되는 애자일 방법으로 재확인 되었다. 이는 본 연구에 사용된 애자일 접근 방향과 함께한다.

6. 결론

본 논문은 애자일 방법과 소프트웨어 재사용의 결합을 위해 스크럼과 칸반의 혼합 애자일 방법에서 재사용을 연구하였다. 먼저, 절약형 애자일 재사용을 위한 활동과 유형을 식별하고, 칸반 보드 중심의 애자일 개발 초기 단계에서 재사용 방안을 제안하였다. 그리고 혼합 애자일 방법에 백로그 팩토링을 이용하는 입력 대기열 항목의 재사용을 통합하였다.

프로토타이핑을 통해, 이클래스 응용에 백로그 팩토링 기술을 적용하고 애자일 방법에서의 입력 대기열을 위한 재사용과 통합을 보일 수 있었다.

본 기술은 계획에 없는 요청이 자주 발생하거나 신제품 개발과 유지보수의 동시 작업이 필요한 중소 애자일 개발 환경에서의 재사용에 도움이 되리라 사료 된다.

앞으로의 연구 과제로, 스크럼과 칸반의 혼합 애자일 방법에서 절약형 애자일 재사용을 지원하는 도구의 개발을 다룰 것이다.

REFERENCES

[1] R. Carbon, M. Lindvall, D. Muthig, P. Costa, “Integrating product line engineering and agile methods: flexible design up-front vs. incremental design”, 1st International Workshop on Agile Product Line Engineering(APLE06), 2006.

[2] Kircher, M., Hofman, P., “Combining Systematic Reuse with Agile Development Experience Report”,

Proceedings of the 16th International Software Product Line Conference-Volume 1. ACM, pp.215-219, 2012.

[3] Mahnic, V., "Improving Software Development through Combination of Scrum and Kanban". Recent Advances in Computer Engineering, Communications and Information Technology, Espanha, 2014.

[4] Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J., "Transition of software maintenance teams from Scrum to Kanban", Hawaii International Conference on System Sciences (HICSS), 2016.

[5] Ian Sommerville, "Software Engineering, 10th Ed.", pp.75-76, 425-428, Pearson, 2016.

[6] Shari L. Pfleeger, Joanne M. Atlee, "Software Engineering 4th Ed.", pp.627-636, Pearson, 2010.

[7] Hee-Soo Kim, Hae-Sool Yang, "A study on the utilizing of automation migration tool through existing system reuse of enterprise", Journal of Digital Convergence, Vol. 12, No. 11, pp.317-327, 2014.

[8] D.H. Kim, Koh Chan, D.S. Kim, H.W. Kim, "A Study on the Agile-based Information System Audit Model", The Journal of Digital Policy and Management, Vol. 11, No. 8, pp.95-108, 2013.

[9] Campanelli, A. S., "A Model for Agile Method Tailoring", Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento, 2014.

[10] Klaus Pohl, van der Linden F., "Software Product Line Engineering", pp.13-14, Springer, 2005.

[11] Ji-Hong Kim, "Backlog Factoring : Extension of Task Factoring for Reuse in Scrum Method", The Journal of Digital Policy and Management, Vol. 10, No. 10, pp.339-345, 2012.

[12] Tian K., Cooper K., "Agile and software product line methods: Are they so different?", 1st international workshop on agile product line engineering, 2006.

[13] MyounJae Lee, "A Game Design for IoT environment", Journal of the Korea Convergence Society, Vol. 6, No. 4, pp.133-138, 2015.

[14] Seong-Hoon Lee, Dong-Woo Lee, "Actual Cases for Smart Fusion Industry based on Internet of Thing", Journal of the Korea Convergence Society, Vol. 7, No. 2, pp.1-6, 2016.

[15] Dhakshinamoorthy R., Thirunavukarasu S., "Turbo Charging IoT Projects with Agile Scrum Methodology", http://www.tcs.com, 2016.

[16] Uyeong Jeong, Youngkwan Ju, Joongnam Jeon, "A Porting Technique of WiFi Device on Android Platform", Journal of IT Convergence Society for SMB, Vol. 2, No. 1, pp.51-58, 2012.

[17] B. Speckmann, "The Android mobile platform", Master Thesis, Eastern Michigan University, 2008.

[18] Jin-Soo Park, Jang-Jin Kwon, Jang-Eui Hong, Min Choi, "Software Architecture Recovery for Android Application Reuse", Journal of IT Convergence Society for SMB, Vol. 3, No. 2, pp.9-17, 2013.

[19] Ruiz, I. J. M., Nagappan, M., Adams, B., Hassan, A. E., "Understanding reuse in the android market". Program Comprehension (ICPC), IEEE 20th International Conference, IEEE, 2012.

[20] Ken Schwaber and Jeff Sutherland, "Software in 30 days", p.61, Wiley, 2012.

[21] Kniberg, H., & Skarin, M., "Kanban and Scrum-making the most of both", pp.16, Lulu.com, 2010.

[22] Anderson, D. J., "Kanban: successful evolutionary change for your technology business", Blue Hole Press, 2010.

[23] Ladas, C., "Scrumban-essays on kanban systems for lean software development", Lulu.com, 2009.

[24] Wang, X., Conboy, K., & Cawley, O.,""Leagile" software development: An experience report analysis of the application of lean approaches in agile software development", Journal of Systems and Software, 85(6), pp.1287-1299, 2012.

[25] Diaz, J., Perez, J., Yague, A., & Garbajosaz, J., "Tailoring the Scrum Development Process to Address Agile Product Line Engineering", Proceedings of Jornadas de Ingeniería del Software y base de Datos, 2011.

[26] Martini, A., Pareto, L., Bosch, J., "Enablers and Inhibitors for Speed with Reuse", Proceedings of the 16th International Software Product Line Conference-Volume 1, ACM, pp.116-125, 2012.

[27] Martini, A., Pareto, L., Bosch, J., "Communication

factors for speed and reuse in large scale agile software development", Proceedings of the 17th international software product line conference, ACM, pp.42-51, 2013.

[28] da Silva, I. F., Neto, P. A. D. M. S., O'Leary, P., de Almeida, E. S., & de Lemos Meira, S. R., "Using a multi-method approach to understand Agile software product lines", Information and Software Technology, pp.527-542, 2015.

김 지 홍(Kim, Ji Hong)

- 1974년 2월 : 경희대학교 전자공학과(공학사)
- 1982년 8월 : California State University (Fullerton) Computer Science(이학석사)
- 1995년 8월 : 경희대학교 전자계산공학과 (공학박사)
- 1982년 5월 ~ 1989년 2월 : 미국 Interpac Software, 소프트웨어 엔지니어
- 1989년 3월 ~ 현재 : 가천대 컴퓨터공학과 교수
- 관심분야 : 소프트웨어공학, UML, 소프트웨어 재사용, 애자일 개발
- E-Mail : wiskjh@gachon.ac.kr