http://dx.doi.org/10.14400/JDC.2015.13.12.171

애자일 제품 백로그 재사용 : 백로그 팩토링 접근

김지홍
가천대학교 IT대학 컴퓨터공학과

Agile Product Backlog Reuse : Backlog Factoring Approach

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

요 약 본 연구의 목표는 제품 백로그 항목의 재사용을 위해 일반화된 애자일 프로세스에 소프트웨어 재사용 기술의 도입이다. 소프트웨어 공학 커뮤니티에는 애자일 방법과 소프트웨어 재사용을 통합하려는 연구들이 있다. 대부분의 통합 연구는 프로덕트라인 공학에 애자일 프랙티스를 도입하는 연구들이며 애자일 방법에 재사용을 도입하는 연구는 많지 않다. 애자일 방법이 제공하는 활동과 산출물에는 재사용을 촉진하는 특징이 있다. 본 연구에서는 일반화된 애자일 프로세스에서 재사용을 위한 촉매활동과 특성을 식별하고 제품 백로그 항목을 위한 재사용 기술을 제안하였다. 본 연구의 결과를 통하여, 제품 백로그 항목의 재사용을 위해 일반화된 애자일 프로세스에 백로그 팩토링 기술을 통합할 수 있었다. 아울러, 제안된 방법을 식당 체인점 응용에 적용하고, 프로토타이핑을 통해 백로그 항목의 재사용을 얻을 수 있었다.

주제어 : 제품 백로그, 소프트웨어 재사용, 애자일 재사용, 애자일 방법, 소프트웨어 공학

Abstract The aim of the study is to introduce a software reuse technique into a generalized agile process for a reuse of a product backlog item. The software engineering community has made some efforts to integrate software reuse and agile methods. Most studies have emphasized the use of agile practices in software product line engineering, whereas reuse in agile methods has received little focus. Nevertheless, the activities and artifacts that provided by an agile method exhibit characteristics that promote reuse. In this paper, we identify catalytic activities and characteristics that promote reuse in a generalized agile process, and we suggest a reuse technique for a product backlog item. Based on our results, we integrate a backlog factoring technique into a generalized agile process for reuse. In addition, we apply the proposed technique and demonstrate a prototype of backlog item reuse in agile methods for a restaurant chain application.

Key Words : Product Backlog, Software Reuse, Agile Reuse, Agile Method, Software Engineering

1. 서론

급속한 기술 변화와 비즈니스 경쟁 환경으로, 개발자는 빠른 제품 출시를 요구받아 새로운 방법을 필요로 하기 때문에, 장기적인 계획주도의 개발 방식보다는 단기간의 빠른 제품 인도가 가능한 애자일 방식의 개발로 패

Received 10 October 2015, Revised 16 November 2015
Accepted 20 December 2015
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.

러다임 변화가 진행 중이다[1].

애자일 방법은 반복적이고, 코드 중심적이고, 문서를 최소화하면서 동작되는 제품의 빠른 인도를 통하여 고객에게 가치 전달을 강조하고 있어서 많은 인기를 얻고 있으며, 향후 10-15년간 성장이 예상되고 있다[2]. 한편 소프트웨어 재사용은 빠르고 저렴한 비용, 좋은 품질의 소프트웨어 개발에 오래 동안 사용된 기술로서 지난 수십년간 소프트웨어 개발의 효율을 향상시키는 솔루션으로 인식 되어왔다[3].

오늘날, 산업체에서는 애자일 방법과 소프트웨어 재사용 기술의 장점을 통합하는 연구가 공개적인 이슈이며 여러 연구들이 나타나고 있다[4]. 이들 노력들은 주로 소프트웨어 재사용을 추구하는 프로덕트 라인 공학에 애자일 방법의 도입에 집중되어있다[5,6]. 그러나 인기가 많은 애자일 방법에서의 재사용 통합 연구는 부족한 현실이다.

애자일 방법에는 재사용을 촉진하는 특징이 있어서 새로운 형태의 재사용이 가능하다. 이는 개발 초기 단계에서 백로그의 재사용뿐만 아니라 후속작업에서 추가적인 재사용 가능성을 높여준다.

본 논문에서는 애자일 방법이 제공하는 애자일 제품 백로그의 재사용을 위한 특성을 식별하고, 백로그 팩토링을 이용하는 백로그 수준의 재사용 방안을 제안하였다.

본 논문의 구성은 2장에서 기초 연구에 속하는 애자일 방법, 일반화된 애자일 프로세스, 그리고 백로그 팩토링에 관해 살펴본다. 3장에서는 백로그 팩토링을 이용한 애자일 제품 백로그 재사용을 지원하는 확장 프로세스를 제안하고, 4장에서는 프로토타이핑을 보인다. 5장에서는 관련 연구를 비교하고, 6장에서 결론을 기술한다.

2. 기초 연구

2.1 애자일 방법과 제품 백로그

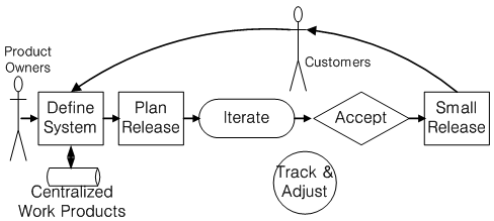
변화하는 비즈니스 또는 기술 환경에서 빠른 개발이 요구되는 오늘날, 애자일 방법은 폭포수형 개발의 엄격한 계획, 설계, 문서화 프로세스에 따른 문제점을 개선하기 위해 새로 제시된 경량급 소프트웨어 방법들을 총칭한다[7]. 애자일 방법 가운데 잘 알려진 방법으로는 익스트림 프로그래밍(XP)과 스크럼 등이 있다. 애자일 프로세스는 선제적이거나 사전의 상세한 계획을 억제하면서 요구사항 변경을 해결하고 있다. 모든 애자일 프로세스는 애자일 매니페스트에서 정의된 공통의 가치와 원칙을 공유하고 있다.

제품 백로그(product backlog)는 제품에 원하는 사항들의 우선순위화된 목록으로, 모든 프로젝트 참가자가 액세스 할 수 있고 애자일 개발활동의 중심에 있는 대표적인 산출물이다[8]. 이는 프로젝트에서 개발해야할 것과 개발의 우선순위에 대하여 일관되고 공통적인 이해를 제공한다. 제품 백로그는 제품 백로그 항목(backlog item)으로 구성되며, 각 항목은 새로운 기능, 개선사항, 수정사항 등을 모두 포함한다. 이들 항목의 표현은 일반적으로 사용자 스토리 형태로 작성되며, 우선적인 요구사항은 좀 더 상세한 내용으로 백로그 상위에 위치시키고 급하지 않은 요구사항은 덜 자세한 형태로 백로그 하단에 위치시킨다.

애자일 프로젝트의 이터레이션에서 나타나는 유사 또는 동일한 백로그 항목들은 재사용이 가능하다.

2.2 일반화된 애자일 프로세스

애자일 프로젝트는 작은 릴리스의 연속으로 완성되며, 각 릴리스는 일련의 이터레이션으로 이루어진다. 반복개발의 기본적 프랙티스와 인기 있는 애자일 방법의 공통적 요소를 기반으로 한 일반화된 애자일 프로세스는 [Fig. 1]과 같다[9,10].



[Fig. 1] Generalized Agile Process

시스템 정의 : 시스템에 요청되는 사항들은 프로젝트 기간 동안 언제든지 정의, 추가, 변경, 삭제 할 수 있다. 이들은 개발 참여자가 쉽게 접근할 수 있는 중앙 저장소에 유지되며 크기가 큰 요구 사항은 필요할 때 작은 크기로 분할될 수 있다.

릴리스 계획 : 릴리스 계획에서는 각 릴리스 주기의 시작 때 제품 소유자와 개발 팀이 함께 모여, 릴리스 내

에서 이터레이션을 어떻게 구성하고 작업할 것인지에 대한 범위와 일정활동의 계획을 결정한다. 릴리스 계획은 비즈니스 가치, 팀 개발 속도 등의 변동 요인에 의해 변경될 수 있다.

이터레이션 : 이터레이션은 계획, 개발, 인도의 3단계로 구성된다. 계획 단계에서는 현 이터레이션에서 개발할 항목을 선정한다. 제품 소유자는 이들 항목을 충분히 설명하고, 각 팀원은 필요한 작업(task)을 정의하고 추정한 후 개발을 서약한다. 개발단계에서는 각 팀원이 약속한 백로그 항목의 태스크가 개발된다. 인도단계에서 개발팀은 새로운 버전의 시스템을 인도한다.

작은 릴리스 : 릴리스 사이클의 절정은 새로운 버전의 출시이다. 애자일 방법의 반복적인 개발 방식은 더 작고, 더 빈번한 출시를 가능하게 한다. 이는 잦은 피드백을 통하여 실패 위험을 줄이고 시간 내 요구사항의 처리로 빠른 출시를 가능하게 한다.

추적과 조정 : 개발 프로세스에서 피드백, 처리량, 번차트 등을 이용하여 상태를 추적하고 과정을 조정하는 지속적인 과정이다.

2.3 백로그 팩토링

프로젝트를 함께 수행하는 팀원들은 여러 활동을 통해 이해와 경험을 공유하게 되므로 다른 문제 해결에도 이전과 비슷하거나 동일한 접근이 자주 나타날 수 있다.

백로그 팩토링이란 각 이터레이션의 수행에 필요한 요구사항이나 작업을 결정할 때, 이전 백로그와 동일 또는 유사 유무를 간단하게 발견하여 재사용 가능한 백로그 항목을 제시하는 기술이다[11].

백로그의 재사용을 허락하는 백로그 팩토링 절차는 착수-발견-결정의 반복적 활동으로 구성되며 모든 팀원에 의해 수행된다. 백로그 착수는 백로그 팩토링에 필요한 초기 작업을 준비하는 활동이다. 이는 백로그 항목을 충분히 이해하고, 스프린트 기간에 작업을 마칠 수 있도록 여러 개 작은 크기의 항목으로 분해하여, 간단한 이름을 할당한다. 백로그 발견은 최우선순위로 개발을 위해 제시된 후보 백로그 항목과 동일하거나 비슷한 기 개발된 백로그 항목이 있는지를 검색하는 활동이다. 백로그 결정은 발견 활동이후 재사용 가능성 여부를 검토하고 항목을 결정하는 활동이다.

3. 애자일 제품 백로그 재사용과 처리

3.1 애자일 제품 백로그 재사용과 특성

3.1.1 애자일 제품 백로그 재사용 (Agile Product Backlog Reuse : APBR)

애자일 제품 백로그 재사용이란 애자일 방법으로 소프트웨어 프로젝트를 하면서 재사용을 위한 별도의 노력과 투자를 피하고, 매 이터레이션을 통해 얻어지는 이해와 산출물을 활용하여 수행하는 제품 백로그 항목의 재사용을 말한다. 이는 소프트웨어 개발 초기에 재사용 지원의 특징을 가지며, 이어지는 분석, 설계, 코딩, 테스팅의 작업에서 또 다른 재사용의 기회를 열어준다.

3.1.2 APBR을 위한 촉매활동과 특성

애자일 프로세스가 기본적으로 제공하는 이터레이션, 계획회의, 시스템 정의활동들의 반복을 통해 얻게 되는 <Table 1>과 같은 지속적인 배경, 요구사항, 산출물에 대한 파악과 이해는 재사용을 촉진하는 촉매활동이다.

〈Table 1〉 Reuse Catalyst for APBR

	Reuse Catalytic Activity
1	Continuous Context Understand
2	Continuous Requirement Understand
3	Continuous Artifacts Understand

이터레이션 마다 거듭되는 촉매활동에 의해서 재사용을 촉진하는 애자일 방법은 <Table 2>와 같은 애자일 제품 백로그 재사용을 지원하는 특성을 갖는다.

백로그 중심 : 백로그는 프로젝트 기간 동안 개발 참여자에게 접근이 용이하고 매 이터레이션 마다 처리의 중심에 위치하여 개발의 작업을 주도하는 역할을 한다. 백로그에 있는 백로그 항목은 재사용 가능한 자산으로 사용될 수 있다.

분할과 정복 : 애자일 방법이 제공하는 분할은 일의 추정, 이해, 단위 기간 내 완료를 가능하게 한다. 분할은 적절한 자산의 발견, 이해 및 재사용을 촉진한다. 예를 들면, 애픽과 같은 큰 요구사항을 여러 개의 작은 스토리로 분할을 들 수 있으며, 이는 재사용의 가능성을 높여준다.

한 김에 재사용 : 매번의 개발 주기마다 시스템 정의,

계획, 이터레이션의 순위화와 같은 활동으로 얻어진 각종 이해와 결과물들은 해당 요구사항 구현뿐만 아니라 재사용 가능한 자산의 발견과 재사용에도 쓰일 수 있다. 이는 중복되는 개발 노력의 낭비를 줄이고 이미 획득한 개발의 노력과 결과물을 필요한 부분에 최대로 활용하려는 적극적 재사용 접근이다.

〈Table 2〉 Characteristics of APBR

Characteristics	Example
Backlog Centric	Product backlog, Iteration backlog
Divide & Conquer	Epic/Stories
By-product Reuse	Backlog Item, Catalyst activity

이와 같은 특성은 제품 백로그 수준의 재사용을 촉진하고 애자일 방법에서 경제적인 비용으로 효율적인 재사용을 지원한다.

3.2 APBR을 위한 역할 및 협동

3.2.1 APBR을 위한 역할

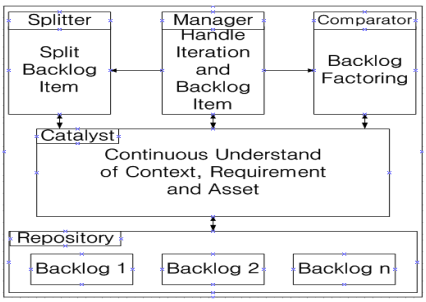
식별된 애자일 방법의 촉매활동과 특성을 활용하는 재사용에는 관리자, 분리자, 비교자, 촉매, 저장소의 5개 역할이 있다.

관리자(Manager)는 반복 처리를 주도하면서 이터레이션의 기반이 되는 백로그 항목을 관리한다. 분리자(Splitter)는 한 이터레이션 동안 수행 가능한 적절한 크기의 백로그 항목으로 분할을 담당한다. 비교자(Comparator)는 기존 산출물과 처리중인 백로그 항목의 재사용성을 검사하고 처리한다. 촉매(Catalyst)는 반복 활동에 따른 지속적 이해를 통해 효과적인 관리자, 분리자, 비교자의 재사용 활동을 촉진한다. 저장소(Repository)는 개발에 관련된 각종 정보를 저장하고 필요에 따라 다른 구성요소에 제공한다.

3.2.2 APBR을 위한 협동

백로그 중심, 분할과 정복, 한 김에 재사용을 최대로 활용하는 저비용의 효과적인 애자일 백로그 재사용의 달성은 [Fig. 2]와 같이 크게 수행자, 촉매, 그리고 저장소 역할들의 협동으로 이루어진다.

수행자는 애자일 백로그의 재사용을 수행하는 분리자, 관리자, 비교자를 말한다. 관리자는 각 반복을 주도하며 반복의 기본 산출물인 백로그 항목의 생성, 추가, 우선순위화, 삭제 및 선택을 관리하며 재사용을 유도한다. 분리자는 촉매가 제공하는 배경 및 요구사항의 이해를 바탕으로 적절한 크기나 주제에 의하여 작업 또는 재사용이 용이한 백로그 항목으로 분할을 담당한다. 비교자는 촉매에 의해 통상적으로 얻어진 요구사항과 산출물의 지속적 이해를 바탕으로 백로그 팩토링에 의해 백로그 항목의 동일, 유사, 상이성을 결정 및 관리한다. 저장소는 개발을 위한 계획, 분석, 설계, 테스트 및 구현에 관련된 산출물의 저장과 제공을 담당한다. 저장소는 프로젝트의 규모나 특징에 따라 여러 종류의 백로그가 지원되며, 이 가운데 제품 백로그의 백로그 항목은 재사용 가능한 자산이다. 촉매는 저장소와 수행자 중간에서 거듭되는 이터레이션으로 얻게 된 자산에 대한 지속적인 이해를 통하여 수행자의 작업을 촉진 한다.



[Fig. 2] Collaboration for APBR

3.3 APBR을 지원하는 애자일 프로세스

3.3.1 APBR 특성을 활용하는 재사용

경제적이고 효과적인 백로그 수준의 재사용을 얻기 위해서는 애자일 방법의 장점과 특징의 활용이 필수적이다. 매 반복에서의 핵심적 처리는 [Fig. 3]과 같이 새로 개발할 항목을 선정하고, 작업하기에 적절한 크기로 분할하면서 재사용성을 점검한 후 구현 작업에 연결시킨다.

제일 먼저, Get_Backlog()는 매 이터레이션의 작업을 주관하는 관리자의 한 부분으로서 현 비즈니스 가치나 환경 변화에 따른 백로그 항목의 업데이트와 우선순위 조정을 하고 최우선순위의 항목을 제공한다. 이어서, Split_BacklogItem()은 분리자의 한 부분으로서 백로그 항목을 이터레이션에 맞는 적절한 크기로 분할한다. 끝

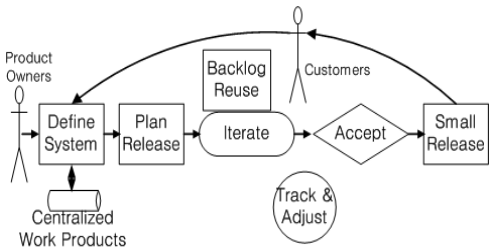
으로, BacklogFactoring()은 비교자의 한 부분으로 새로 개발할 백로그 항목과 동일 또는 유사한 이전의 산출물을 식별하고 재사용된 항목을 제시한다.

```
Backlog_Reuse() {
  TopItem = Get_Backlog();
  Split_BacklogItem(TopItem, count, Items);
  for i=1 to count;
    BacklogFactoring(Items[i]);
  }
}
Get_Backlog() {
   /* get TopItem from backlog */
}
Split_BacklogItem(Item, N, Items){
  /* split item into N Items */
}
BacklogFactoring(item) {
  /* perform backlog factoring */
}
```

[Fig. 3] Steps for APBR

3.3.2 애자일 프로세스의 확장

본 논문에서 제안한 애자일 방법과 소프트웨어 재사용 기술을 통합하는 애자일 제품 백로그 재사용을 위한 확장 애자일 프로세스는 [Fig. 4]와 같다.



[Fig. 4] Extended Agile Process for APBR

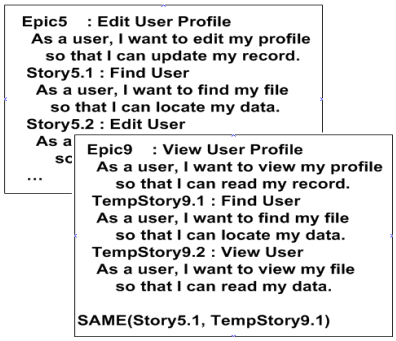
기존의 일반화된 애자일 프로세스에 추가된 백로그 재사용(Backlog Reuse)은 3.2절과 3.3절에 제시된 APBR을 위한 활동이다. 이는 재사용을 위한 촉매활동을 바탕으로 APBR 특성을 활용하는 애자일 방법과 백로그 팩토링으로 수행된다.

4. 적용과 프로토타이핑

본 장은 3장에서 제안한 애자일 제품 백로그 재사용 기술을 식당 체인점 응용에 적용하고 백로그 수준에서 재사용의 프로토타이핑 보인다.

• 동일 스토리의 적용

새로운 이터레이션에서 개발할 백로그 항목에 대한 애자일 백로그 재사용을 살펴본다. [Fig. 5]의 상위 부분은 기 수행된 여러 스토리 가운데 재사용 대상이 되는 Epic5(회원 파일 편집)와 적절한 크기로 분해된 2개의 스토리인 Story5.1(회원검색)과 Story5.2(회원편집)을 보여준다.



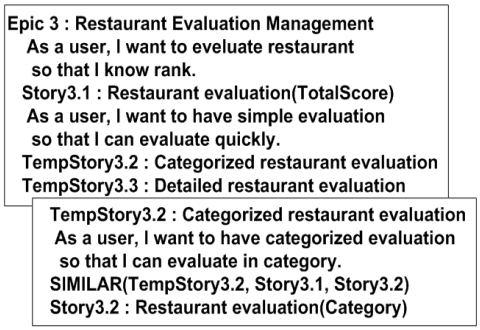
[Fig. 5] Backlog Reuse(Same Item)

2번째 화면은 3.3.1절에 소개한 Backlog_Reuse()를 통해 시작되는 재사용을 보인다. 새로운 이터레이션에서 처리중인 임시 스토리인 TempStory9.1(회원검색)과 이전 이터레이션에서 이미 수행된 Story5.1(회원검색)의 동일성이 발견되어 SAME(Story5.1, TempStory9.1)의 적용을 통한 재사용을 보이고 있다.

• 유사 스토리의 적용

여러 이터레이션을 끝내고 유사 스토리가 적용되는 제품 백로그 항목의 재사용을 위한 백로그 팩토링을 살펴본다. [Fig. 6]의 상단 화면은 이전 이터레이션에서 처리된 Epic3(식당평가)의 3개 분할(Story3.1(총점평가), TempStory3.2(범주별평가), TempStory3.3(상세평가))과 첫 번째 스토리 Story3.1(총점평가)의 수행을 보이고 있다.

2번째 화면은 새로운 이터레이션에서의 TempStory3.2 처리를 보이고 있다. TempStory3.2(범주별평가)는 여러 단수항목들로 구성되는 일종의 복수항목 평가이다. 범주별 평가의 각 항목은 이미 처리된 단일항목평가인 Story3.1과 유사성이 식별되어 SIMILAR(TempStory3.2, Story3.1, Story3.2)가 적용되고, 재사용성이 있는 Story3.2(Restaurant evaluation(category))의 할당을 보이고 있다.



[Fig. 6] Backlog Reuse(Similar Item)

5. 관련 연구 비교

애자일 방법과 소프트웨어 재사용의 통합은 다음과 같은 2가지 측면으로 볼 수 있다.

첫 번째 측면은 애자일 개발에 소프트웨어 재사용 지원을 통한 결합이다. McCarey는 애자일 개발환경에서 개발자에게 재사용 가능한 컴포넌트의 추천을 통한 통합을 제시하였다[12,13]. Hummel은 테스트 주도의 애자일 개발환경에서 필요한 컴포넌트를 웹에서 수확하여 제공하는 연구를 하였다[14]. 이들 연구는 개발 후반부 코딩에서의 재사용을 지원하고 있으며, 본 연구는 개발 초기에서의 재사용을 추구하고 있다.

두 번째 측면은 소프트웨어 재사용에 애자일 방법의 지원을 통한 결합이다. Tian과 Cooper[6]는 애자일 방법과 프로덕트라인 공학을 비교하고 프로덕트라인 공학에 애자일 원칙의 도입이 더 유리하다고 결론을 내었다[15]. Carbon[5]은 PulSE-I을 이용하여 애자일 원칙을 프로덕트라인 공학에 적용하였다. 이들 연구는 소프트웨어 재사용 가운데 대규모적 소프트웨어 패밀리의 재사용을 지원하는 소프트웨어 프로덕트라인에 집중되어있다. 본 연구에서는 애자일 방법에 재사용을 제안하였다.

최근에, Pacheco는 요구사항 카탈로그에 있는 소프트웨어 요구사항의 재사용을 위한 모델을 제안하였다[16]. 이 모델은 요구사항 프로세스 수행의 가이드라인을 정의하고 RequisitePro를 통한 재사용의 타당성을 보이고 있다. 이는 IEEE 표준 명세로부터의 요구사항 재사용을 다루고 있는데 비하여, 본 연구는 애자일 개발에서의 백로그 재사용을 목표로 하고 있다. 재사용과 애자일 통합의 연구는 여러 관점에서 꾸준히 계속되고 있다.

6. 결론

본 논문은 일반화된 애자일 방법과 제품 백로그 수준의 재사용과의 통합을 연구하였다. 먼저 재사용을 촉진하는 촉매활동과 애자일 방법이 제공하는 재사용 특성을 식별하였다. 그리고 일반화된 애자일 프로세스에서 백로그 팩토링을 이용하는 백로그 재사용 방안을 제안하였다.

프로토타이핑을 통해, 식당 체인점 응용에 확장 프로세스를 적용하고, 제품 백로그 항목의 재사용을 보일 수 있었다.

본 기술은 애자일 개발 초기 단계에서의 재사용과 후속적인 작업에서의 재사용과 연계 사용될 수 있다고 사려 된다.

앞으로의 과제는 애자일 개발에서 산출되는 여러 재사용 가능한 결과물간의 연관 개발 방안을 다룰 것이다.

REFERENCES

[1] Martini, A., Pareto, L., Bosch, J., Enablers and Inhibitors for Speed with Reuse, Proceedings of the 16th International SPLC, pp. 116-124, 2012.

[2] Farid, W.M., Mitropoulos, F.J., NORPLAN: Non-functional Requirements Planning for Agile Processes, In 2013 Proceedings of IEEE Southeastcon, pp. 1, 2013.

[3] Yuri Chernak, Requirements Reuse: The State of the Practice, 2012 IEEE International Conference on Software Science, Technology and Engineering, 2012.

[4] Martini, A., Pareto, L., Bosch, Communication factors for speed and reuse in large scale agile

software development, Proceedings of the 17th International SPLC, pp. 42-51, 2013.

[5] R. Carbon, M. Lindvall, D. Muthig, P. Costa, Integrating product line engineering and agile methods: flexible design up-front vs. incremental design, 2006.

[6] Tian K., Cooper K., Agile and software product line methods: Are they so different? APLE '06, 2006.

[7] Dina Salah, A Systematic Literature Review for Agile Development Processes and User Centred Design Integration, EASE, 2014.

[8] K. S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison-Wesley, pp. 99-103, 2012.

[9] Dean Leffingwell and Dave Muirhead, Tactical Management of Agile Development: Achieving Competitive Advantage, Rally Software Development Corporation, 2004.

[10] Dean Leffingwell, Scaling Software Agility, Addison Wesley, pp. 116, 2007.

[11] Ji-Hong Kim, Backlog Factoring : Extension of Task Factoring for Reuse in Scrum Method, The Journal of Digital Policy and Management v.10, n.10, pp. 339-345, 2012.

[12] McCarey et al., RASCAL: A Recommender Agent for Agile Reuse, Artificial Intelligence Review, Volume 24(3-4), pp. 253-276, 2005.

[13] McCarey, F., Cinneide, M., Kushmerick, N., An Eclipse Plugin to Support Agile Reuse, LNCS Volume 3556, pp. 1298-1301, 2005.

[14] Oliver Hummel, Colin Atkinson, Supporting Agile Reuse Through Extreme Harvesting, LNCS Volume 4536, pp. 28-37, 2007.

[15] Noor M., Rabiser R, Gr¨unbacher P., Agile product line planning: A collaborative approach and a case study. Journal of Systems and Software, 81(6), pp. 870, 2008.

[16] C. L. Pacheco et. al., A proposed model for reuse of software requirements in requirements catalog, Journal of Software: Evolution and Process, 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 ~ 89년 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