

기민한 컴포넌트 개발 프로세스

한정헌*, 인호**

*고려대학교 컴퓨터정보통신대학교 소프트웨어공학과

**고려대학교 컴퓨터학과

e-mail : scant,hoh-in@korea.ac.kr

Agile Component Development Process

Jeong-Heon Han*, Hoh-In **

*Graduate School of Computer and Information Technology, Korea University

**Dept of Computer Science & Engineering, Korea University

요 약

아직도 많은 프로젝트에서 지속적으로 변화하는 요구사항과 불가능한 납기일정 그리고 품질을 무조건 만족해야 하는 상황에 많은 개발자들이 괴로워하고 있다. 또한 최근 들어 프로젝트들이 그 개발의 방법으로 컴포넌트 기반 개발(CBD)를 많이 사용하고 있다. 이러한 상황에서 재사용성을 강조하는 품질 좋은 컴포넌트를 식별하고, 개발하기란 매우 어렵다. 또한 기존의 CBD 개발 방법의 프로세스가 요구사항을 초기에 고정하고, 그 기반 위에 여러 단계를 거치기 때문에, 납기기간이 짧거나, 요구사항이 고정되지 않았을 경우 많은 어려움에 직면하고 있다. 본 논문에서는 요구사항이 불명확하고, 납기일정이 짧은 프로젝트에서 유용하도록, 작은 단위의 반복과 그 피드백을 통해, 고객의 요구사항 변경에 빠르게 반응할 수 있고, 기존의 컴포넌트 개발 방법을 단순화한 기민한 컴포넌트 개발 프로세스를 제시한다. 이러한 제안된 기법의 실용성을 검증하기 위해 기존 식별 방법과의 비교, 평가를 제시한다.

1. 서론

여전히 많은 프로젝트에서, 과당경쟁으로 인한 저가수주 관행과 불가능한 일정, 비용초과로 인해 ‘불가능을 가능케 하라’는 구호아래 오늘도, 밤을 지새우고 있다. 또한 현대의 더욱 더 복잡해진 비즈니스와, 고객의 모호한 요구사항 및 빠른 속도로 변하는 요구사항이 프로젝트를 더 어렵게 하고 있다.

이러한 소프트웨어 공학의 오래된 문제들에 대해, 많은 은총알(Silver Bullet)[1]들이 제시되어 왔다. 또한, 이러한 요구사항 변경에 대응해서 전통적인 방법론들은 요구사항의 변경을 재 작업, 또는 프로젝트의 위험으로 간주하고, 요구사항을 초기에 확정하고, 확정된 요구사항을 추적하고, 그 변경을 관리함으로써, 변경 자체를 억제하고자 노력해 왔다.

그러나 최근 들어, 이러한 요구사항의 변경이나, 불확실한 요구사항을 소프트웨어개발에서 필연적이고, 자연스런 것으로 보고, 이에 적응하고자 노력하는 애자일 개발 프로세스가 주목 받고 있다. 대표적인 방법으로, 익스트림 프로그래밍(Extreme Programming)[6]이 있다. 애자일 개발 프로세스는 특성상 정형한 프로세스를 제시하지는 않지만 원칙, 상세한 실천지침을 제시함으로써 임기응변 식 개발과는 구별된다.

그리고, 애자일 개발 프로세스와는 별도로, 프로그래밍 방식은 더욱더 복잡해진 비즈니스 문제의 해결을 위해, 좀더 이해 가능하고, 쉽고, 빠르게 프로그래밍 할 수 있는 방법을 찾아 구조적 방법에서 객체지향(OO)방법으로 발전해 왔다. 특히 최근 들어 객체지향 기술에서 좀더 효율적인 재사용 및 생산성을 향상시킬 방안으로 컴포넌트 개발 방식이 선호되어 왔다, 그렇지만 이러한 컴포넌트 개발 방

식은 여전히 요구사항을 초기에 고정하고, 이 고정된 요구사항을 중심으로 체계적인 여러 단계에 걸쳐 우수한 컴포넌트를 식별 개발하는 데에만 초점을 맞추어 왔다. 그러나 문제는 아직도 많은 프로젝트에서, 요구사항은 고정되지 않고, 고정하기도 힘들다는데 있다.

본 논문에서는 현실 프로젝트에서 일반적으로 많이 발생하는 불완전한 요구사항에 불가능한 납기일정 그리고 품질을 무조건 만족해야 하는 어려운 CBD 개발 프로젝트 상황에서의 실용적인 방법을 찾고자 한다. 이러한 상황에서, 빠른 피드백으로 고객의 요구사항을 명확히 확정시킬 수 있는 애자일 방법은 매우 유용하다. 그리고 그러한 빠른 반복을 가능하게 할 컴포넌트 개발 방식도 필요하다.

기존의 컴포넌트 개발 방법을 살펴 보면, CBD 에서 핵심은 기능적으로 재사용이 가능하고 유지보수 단계의 효율적인 시스템 관리를 위해서 상호 의존성이 적은 독립적인 컴포넌트를 잘 식별하는 것인데, 기존의 방법들은 그 재사용성과 그 메커니즘을 너무 강조한 나머지 그 절차가 복잡하고, 무거운 경향이 있어서 애자일 개발 프로세스에 활용하기에는 어려운 점이 있다.

본 논문에서는 빠른 반복, 피드백을 바탕으로, 단순하고, 가벼워서 구성원들이 이해하기 쉽고, 기민한 컴포넌트 개발 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2 장에서는 논문의 배경을 살펴본다. 3 장에서는 기민한 컴포넌트 개발 프로세스를 제시한다. 4 장에서는 기존 컴포넌트 개발 방법과의 비교 분석 및 평가를 제시한다. 5 장에서는 결론 및 향후 연구과제를 제시한다.

2. 배경 및 문제점

2.1 XP(Extreme Programming)

애자일 개발 프로세스의 대표적인 방법론으로 의사소통, 피드백, 단순성, 용기, 존중 같은 가치에 바탕을 둔 소프트웨어 개발 철학이다.[6] 특히 소프트웨어 개발의 성공 요소의 하나로, 인적 요소를 강조하며, 다음과 같은 범위 통제 원칙을 제시한다. 초기에 요구사항 기능(스토리)의 우선순위 및 첫 번째 반복에 포함되는 스토리의 범위를 추정하고, fix 된 일정에, 이를 구현함으로써, 실제로 구현된 범위의 소요시간을 경험적으로 획득하게 된다. 이러한 DATA 를 가지고, 다음 이터레이션의 일정을 제어하기 위해, 고객 스스로 가장 중요한 기능구현을 우선순위에 놓도록 유도함으로써, 범위를 제어하여, 실제로 구현 가능한 일정을 산출할 수 있도록 한다. 설계방법으로, 진화적인 설계 및 구두 전달 방법을 선호하고, 점진적 설계 및 지속적인 설계 개선(refactoring)를 강조한다. 그러나 실제로 적용 시, 기본적으로 자율적인 실천방안들을 제시하기에, 고도로 숙련된 자 외에는 실천하기 힘들고, 수행자들이 미숙할 시 그리고, 대규모 프로젝트일 경우 이해의 혼란으로 인해, 주먹구구식인 개발이 될 위험성이 있다. 그리고 아직까지 컴포넌트 개발에 대한 내용은 고려하지 않고 있다.

2.2 컴포넌트 식별 방법

[2]에서는 기존의 컴포넌트 식별 절차를 다음과 같이 분류하였다.

2.2.1 Domain Engineering based Methods

이 방법은 초기에 컴포넌트 식별을 위해 도메인 엔지니어링 과정을 거친다. 컴포넌트 설계자는 도메인 분석을 통해, 하나의 비즈니스 도메인 안에서 비슷한 요구사항의 그룹을 식별하고, 거기서 commonality 와 variability 를 찾아 낸다. 주로 도메인분석을 통한 설계자의 경험과 직관을 중요시하기에, 그 체계적 방법이 모호한 점이 있다.

2.2.2 Cohesion-Coupling based Clustering Analysis Methods

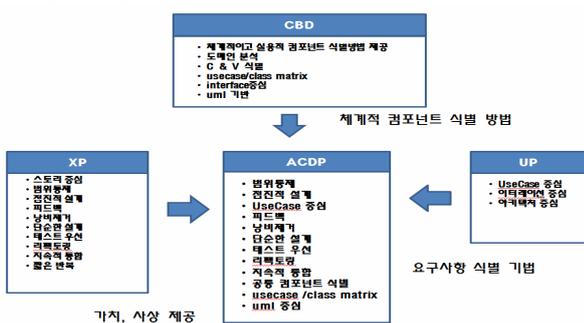
대표적인 방법으로, [4]에서는 컴포넌트의 granularity 를 가장 중요한 요소를 강조하고, 그 기준으로 high cohesion 과 low coupling 를 제시한다. 구체적인 방법으로, Function Type 에 의한 Key Class 의 식별, coupling 측정 알고리즘, Component Clustering 알고리즘, UseCase/UseCase Table 등을 제안한다.

2.2.3 CRUD based Methods

2 의 방법을 기반으로 더욱 체계화한 방법이 주를 이루며, [3]에서는 클래스의 C&V 속성식별을 중심으로, 구체적인 단계 및 Class/Class matrix 를 제시하였으나, 검토해야 할 요소가 많고, 다소 복잡하여 많은 시간 및 검토가 필요하다.

3. 기민한 컴포넌트 개발프로세스 제안

3.1 ACDP 의 개념



(그림 1) ACDP 의 개념도

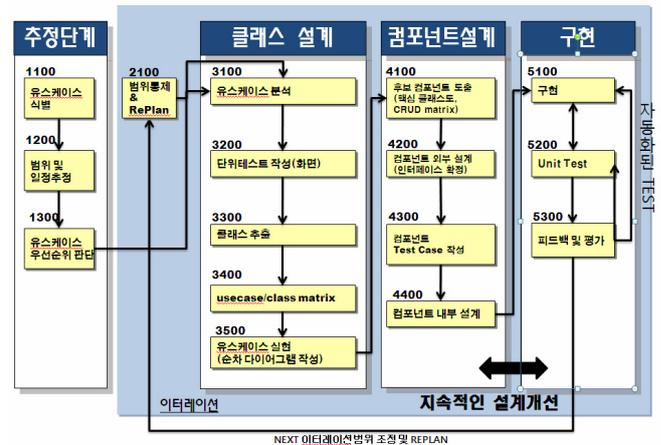
ACDP(Agile Component Development Process)는 애자일 방법론 하나인 XP 의 가치, 원칙 기반 위에, 요구사항 추출에 대한 기법으로 RUP 의 UseCase 를 활용하였고, 기존 CBD 추출방법의 절차적 방법을 간소화하여 하이브리드하게 결합하였다.

3.2 ACDP(Agile Component Development Process) 의 원칙

- 예측 가능한 일정 - 이터레이션 수행 경험을 통해 생산성을 측정하고 가능한 범위 및 일정을 세운다.
- 고객과 개발자의 관계를 WIN-WIN 으로 설정한다.
- 단순한 컴포넌트 추출방법 - 빠른 개발을 위해, 단순하고 명확하고, 이해가 쉬운 방법을 제시한다.
- 단순한 설계- 불필요한 복잡성을 제거한다.
- 중복제거- 중복된 코드를 포함하지 않는다.
- 낭비제거- 불필요한 산출물은 과감히 생략한다.
- 최소화 - 최소한의 클래스와 메소드를 가진다.
- 지속적인 설계개선(점진적, 반복적)- 구현을 통해 설계를 개선하고, refactoring 한다.
- 설계와 동시에 TEST CASE 를 작성한다. - 100% 다시 가동되는 단위테스트를 보유한다.
- 고객을 프로젝트 팀에 포함시킨다.

3.3 ACDP 의 프로세스

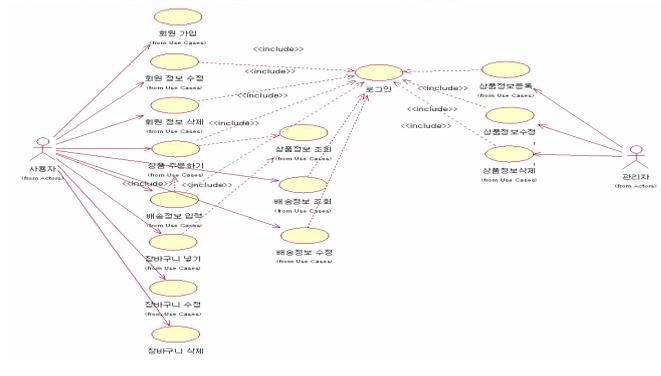
본 장에서는 프로세스 설명과 함께 사례로 인터넷 전자상거래 시스템을 설계하였다.



(그림 2) 전체 프로세스

3.3.1 추정 단계

- 1100 유스케이스 식별
- 고객과 함께 전체 업무 범위를 설정하고, 액터의 역할에 따른 시스템의 기능을 UseCase 로 식별한다.



(그림 3) 유스케이스 식별

■ 1200 이터레이션 범위 및 일정 추정

고객과의 미팅을 통해, 고객에게 각각의 UseCase 에 대한 업무기능에 대해 설명을 듣는다. 기능설명을 근거로, 설계자 및 개발자가 직접 자신의 일을 추정하게 한다.

■ 1300 유스케이스 우선순위 판단

1200 에서 추정된 DATA 를 INPUT 으로, 가장 중요한 구현 대상을 첫 번째 이터레이션 범위로 확정하게 한다. 일정조정을 위해, 첫 번째 이터레이션시 일정상, 포기해도 되는 분량도 일부 포함한다.

■ 2100 범위통제 & REPLAN

각각의 이터레이션의 초기 PLAN 를 작성하고, 이터레이션 후, 이전 이터레이션의 결과를 바탕으로, 계획변경을 할 수 있는 단계이다.

3.3.2 클래스 설계 단계

■ 3100 유스케이스 분석

고객 업무담당자와 함께 UseCase 흐름에 따라, 글로 명세 한다. (UseCase 명세서) 고객 업무담당자가 직접 작성하거나, 고객과 설계자와 함께 작성하는 것이 가장 좋다.

■ 3200 단위테스트 작성(화면)

UseCase 명세서를 기반으로 단위테스트를 작성하여 구현 후 요구사항 구현 검증의 기준으로 활용한다.

■ 3300 UseCase 에서 후보 클래스 추출

UseCase 명세서를 분석하여 명사중심의 후보 클래스를 추출한다.

■ 3400 UseCase/Class 매트릭스 작성

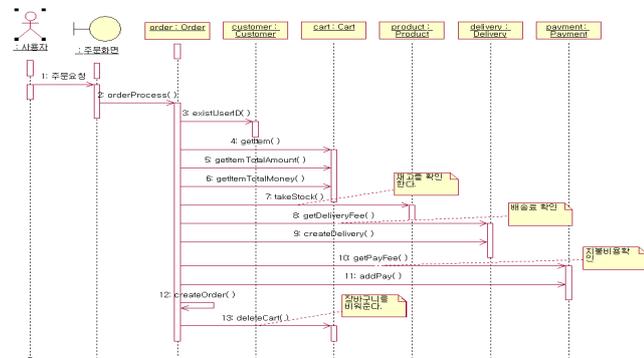
UseCase/Class 매트릭스를 작성하여 유스케이스에 유용한 클래스가 도출되었는지, 후보 클래스를 검증한다. 또한 공통으로 사용되는 클래스를 식별한다. 이 공통 클래스는 1 차 후보 컴포넌트가 된다.

| | Class1 | Class2 | Class3 | Class4 |
|----------|--------|--------|--------|--------|
| UseCase1 | ● | | | |
| UseCase2 | | ● | ● | |
| UseCase3 | ● | | | ● |

(표 1) 유스클래스/클래스 매트릭스

■ 3500 UseCase 실현

UseCase 명세서를 기반으로 기본흐름과 보조흐름 별로, 3400 에서, 추출한 클래스를 이용하여, 순차 다이어그램을 작성한다. UseCase 명세서상의 행위를 오퍼레이션으로, 항목을 속성으로 추출하고 흐름상 필요한 요소를 추가한다.



(그림 4) 유스케이스 실현 시퀀스 다이어그램

3.3.3 컴포넌트 설계 단계

■ 4100 후보 컴포넌트 도출

3500 에서 추출한 클래스의 연관관계를 모두 표현하는 '핵심 클래스도' 를 작성한다. 'Class CRUD 매트릭스' 도 작성하여, 후보컴포넌트를 추출한다. 후보 컴포넌트 추

출의 기준은 다음과 같다.

- 핵심 클래스도 의 연관성(응집도)[3]
- CRUD 매트릭스의 C>D>U>R 의 기준으로 응집 성을 파악하여 후보 컴포넌트를 도출한다. [5]

그림 4 의 핵심 클래스 도를 보면, 그 연관관계의 느슨함과 응집도를 살펴서 클래스들을 분류할 수 있고, 그림 5 의 매트릭스의 C, D, U 관계를 살펴봐도 그 응집도를 확인할 수 있다.

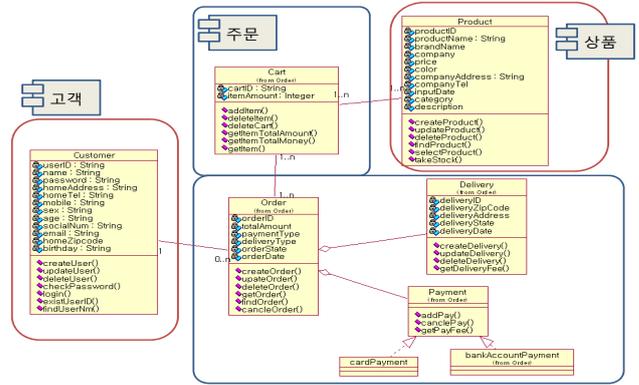


그림 5) 핵심 클래스도

| | 고객 | 장바구니 | 상품 | 배송 | 주문 | 지불 |
|----------|----|------|----|-----|----|-----|
| 회원가입 | C | | | | | |
| 회원수정 | U | | | | | |
| 회원삭제 | D | | | | | |
| 장바구니 넣기 | | C | | | | |
| 장바구니수정 | | U | | | | |
| 장바구니삭제 | | D | | | | |
| 상품등록 | | | C | | | |
| 상품수정 | | | U | | | |
| 상품삭제 | | | D | | | |
| 상품주문하기 | R | R,D | R | | C | |
| 주문변경 | R | | R | R,U | U | R,U |
| 주문삭제 | R | | R | U | D | R,U |
| 배송입력 | | | | C | R | |
| 배송수정 | | | | U | R | |
| 배송조회 | | | | R | R | |
| 로그인 | R | | | | | |
| 상품결제(지불) | R | | R | | U | C |

(표 2) 클래스 CRUD 매트릭스

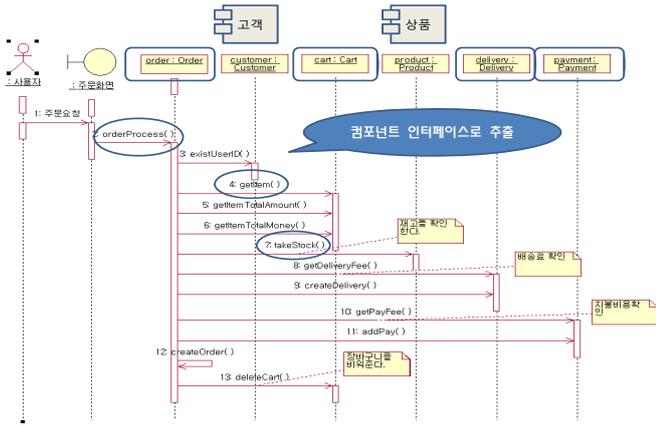
본 사례에서는 그 결과 다음과 같이 컴포넌트를 식별하였다.

| 컴포넌트 | 포함 클래스 |
|------|--|
| 고객 | Customer |
| 상품 | Product |
| 주문 | Order, Cart, Delivery, Payment, CardPayment, OnlinePayment |

(표 3) 식별된 컴포넌트

■ 4200 컴포넌트 외부설계

3500 에서 작성하였던 순차 다이어그램상의 클래스를 후보 컴포넌트로 대체하여, 그 첫 진입 오퍼레이션을 컴포넌트 인터페이스로 추출한다. 그림 6 과 같이 기존 유스케이스 실현 시의 시퀀스 다이어그램의 클래스를 도출된 컴포넌트로 대체하면, 그 첫 연관이 있는 오퍼레이션이 각각의 컴포넌트의 인터페이스가 된다.



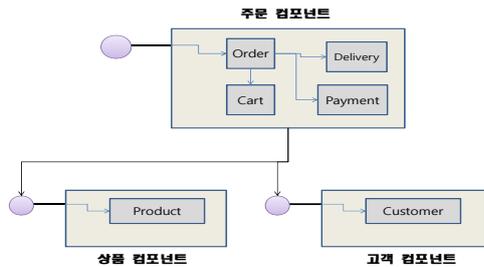
(그림 6) 컴포넌트 인터페이스 추출과정

■ 4300 컴포넌트 테스트 케이스 작성

4200의 작업 완료와 동시에, 확정된 인터페이스의 INPUT과 RETURN 값을 기반으로 컴포넌트 Test Case를 작성한다.

■ 컴포넌트 내부 설계

확정된 컴포넌트 내의 구조를 기존 확보한 클래스로 구조화한다.



(그림 7) 획득된 컴포넌트의 연관관계

3.3.4 구현단계

■ 5100 구현

구현 시 문제로 인해 설계가 변경되어야 할 경우 지속적으로 설계를 개선한다.

■ 5200 Unit Test

테스트는 4300에 작성된 컴포넌트 Test Case를 우선 수행한 후, 3200에 작성된 단위테스트(화면)케이스를 수행한다. 테스트는 모든 기능 구현완료 후 진행되는 것이 아니라, 구현과 동시에 병렬적으로 진행된다.

■ 2100 범위통제 & REPLAN

다시 2100으로 돌아와서, 이터레이션 종료 후 소요된 일정과 추정된 일정의 차이를 비교하여 판단한다. 진행 시 문제점을 분석하고, 이전 이터레이션의 범위대비 소요시간을 반영하여, 차기 이터레이션의 구축 범위와 일정을 재추정한다. 이로써, 팀의 해당 기능 구현의 정확한 소요기간 산정이 가능할 것이다. 또한 추정결과 정해진 일정 대비, 일정지연이 예상되면, 구현 범위를 고객과 합의 하에 기능의 우선순위를 판단하여, 범위를 재조정할 수 있다.

또한 기존의 컴포넌트 식별방법보다 그 절차를 간소화하고, 단순화 하였기 때문에, 빠른 컴포넌트 개발이 가능하고, 빠른 피드백을 통해, 고객의 요구를 명확히 하고, 다시 빠른 반복을 실행할 수 있다.

4. 평가

본 논문에서 제안된 기법은 애자일 프로세스 하에서의 컴포넌트 식별 및 그것을 포괄하는 개발 프로세스이다. 따라서, 기존컴포넌트 개발방법과의 비교, 평가도 애자일

개발 프로세스의 원칙[7]을 충실히 따르고 있는가 하는 점과, 제안된 간소한 컴포넌트 식별방법이 기존방법의 컴포넌트 식별의 중요 요소들을 반영하고 있는가 하는 점이다.

| 비교항목 | | 식별방법 | | | |
|---------------------|----------------|---------------------|-----|-----|-----|
| | | [3] | [4] | [5] | 본제안 |
| 애자일 요소 | 빠른 인도를 통한 고객만족 | | | | 고려 |
| | 점진적 반복 설계 | | | | 고려 |
| | 변경포용 | | | | 고려 |
| | 빠른 반복통한 빠른 인도 | | | | 가능 |
| | 피드백 중시 | | | | 고려 |
| | 범위통제 | | | | 고려 |
| | 테스트 우선 | | | | 미흡 |
| | 지속적 통합 | | | | 미흡 |
| | 단순성 | | | | 고려 |
| | 컴포넌트 요소 | Coupling & Cohesion | 반영 | 반영 | 반영 |
| Interface Design | | 반영 | 반영 | 반영 | 반영 |
| Understanding(easy) | | | | | 고려 |
| Reusability | | 반영 | 반영 | 반영 | 고려 |

<표 4> 비교 평가표

5. 결론

본 논문에서는 소프트웨어 공학의 만성적인 문제인 불명확한 요구사항 및 일정지연을 빠른 피드백을 통해, 요건을 확정하고, 범위통제를 통해 효율적으로 관리할 수 있는 기민한 개발 프로세스를 살펴보았다. 본 제안 방법은 다음과 같은 요소를 내포하고 있다.

- ✓ 빠른 반복, 피드백을 통한 고객 요건 확정
- ✓ 빠른 반복 경험을 통한 생산성 측정 및 그에 기반한 범위 통제(범위통제를 통한 일정관리)
- ✓ 빠른 반복에 적합한 단순한 컴포넌트 식별방법

그러나 본 논문은 애자일 프로세스에 맞는 기민한 컴포넌트 개발 프로세스중 컴포넌트 식별절차에 초점을 맞추었기 때문에, 지속적 통합방법 이라던지, 테스트 우선 개발 등, 애자일의 다른 요소의 그 구체적 구현방법에는 다소 부족한 점이 있다. 이후 연구과제로, 이 부분을 포함할 필요가 있고, 많은 적용사례가 필요하다. 또한, 본 논문에서 제안한 이러한 절차를 만족시킬 기반이 되는 범위통제가 가능한 계약문제 및 제안된 절차의 대규모 프로젝트 적용가능성에 대한 연구도 필요하다.

참고문헌

[1]프로덕트 브룩스, "THE MYTHICAL MAN-MONTH"피어슨 에듀케이션 코리아, 2007.
 [2]Zhongjie Wang, Xiaofei Xu, and Dechen Zhan, "A Survey of Business Component Identification Methods and Related Techniques," INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY VOLUME 2 NUMBER 4, 2005 ISSN.
 [3]E.S.CHO,S.D.KIM,S.Y.RHEW, "A DOMAIN ANALYSIS AND MODELING METHODOLOGY FOR COMPONENT DEVELOPMENT", International Journal off Software Engineering and Knowledge Engineering Vol. 14, No. 2,2004.
 [4]J.K.Lee,S.J.Jung,S.D.Kim, "Component Identification Method with Coupling and Cohesion," in Proc. of 8th Asia-Pacific Software Engineering Conf. Macau, China, 2001.
 [5]최미숙, 조은숙, 박재년, 하중성, "클래스들 간의 정적 동적 관계에 의한 2 단계 컴포넌트 식별방법", 정보과학회논문지: 컴퓨팅의 실제 제 111 권 제 1 호, 2005.2.
 [6]켄트벡, 신시아 안드레스, "익스트림 프로그래밍" 인사이드, 2005.
 [7]Kent Beck, James Grenning, Martin Fowler 외, "Principles behind the Agile Manifesto", http://agilemanifesto.org/.