

# 공통성과 가변성 분석을 활용한 프로세스 라인 개발 방법

최승용\*, 김순태\*\*, 김정아\*

\*관동대학교 컴퓨터교육과

\*\*전북대학교 소프트웨어공학과

e-mail: {boromi, clara}@kd.ac.kr, \*\*stkim@jbnu.ac.kr

## A Process Line Development Method based on Commonality & Variability Analysis

Seung-Yong Choi\*, Sun-Tae Kim\*\*, Jeong-Ah Kim\*

\*Dept of Computer Education, Kwan-Dong University

\*\*Dept of Software Engineering, Chon-Buk National University

### 요 약

수시로 바뀌는 소프트웨어 프로세스 수립자의 지식과 경험에 의존하여 소프트웨어 프로세스 수립자가 좋아하는 임의의 방법으로 소프트웨어 프로세스를 정의하는 환경에서는 소프트웨어 프로세스의 품질을 높이는 데 한계가 있다. 그러므로 보다 효율적으로 프로세스 테일러링을 하는 기술이 필요하다. 본 논문에서는 소프트웨어 프로세스의 재사용을 가능하게 하는 프로세스 라인 구축 방법을 제안한다. 본 방안을 프로세스 테일러링에 적용하면 프로세스 재사용 기회를 높이고 프로세스 재적용 노력을 줄일 수 있다.

### 1. 서론

소프트웨어 프로세스 정의 활동은 소프트웨어 프로세스 수립자의 지식과 경험에 상당히 의존하는 경향이 높다. 수시로 바뀌는 소프트웨어 프로세스 수립자의 지식과 경험에 의존하여 소프트웨어 프로세스를 정의하는 환경에서는 소프트웨어 프로세스의 품질을 높이는 데 한계가 있다. 그 이유를 들면 다음과 같다.

- ① 소프트웨어 프로세스 수립자의 지식과 경험 수준이 서로 다르기 때문에 소프트웨어 프로세스 품질이 소프트웨어 프로세스 수립자의 역량 수준을 따라가는 경향이 있다.
- ② 소프트웨어 프로세스 수립자의 역량 수준에 의존하다 보니 소프트웨어 프로세스의 주요 요소 (활동, 역할, 양식, 점검표, 가이드라인 등)가 정확히 정의되지 않는 경우가 있다.
- ③ 급히 개발해야 하는 특정 개발 영역에만 맞춰 소프트웨어 프로세스를 정의하다 보니 소프트웨어 프로세스 정의 수준이 포괄적이지 않는 경향이 있다.
- ④ 소프트웨어 프로세스 정의 수준이 포괄적이지 않다 보니 맞춤화 과정을 통해 재정의한 소프트웨어 프로세스를 다른 개발 영역에 활용하기 힘든 경우가 생긴다.

이를 해결하기 위해 본 논문에서는 소프트웨어 프로세스의 재사용을 가능하게 하는 프로세스 라인 구축 방법을 제시한다.

### 2. 관련 연구

#### 2.1 소프트웨어 제품 라인

소프트웨어 제품 라인 (Software Product Line) [1,2]은 소프트웨어 설계 단계부터 재사용을 강조하기 때문에 구축된 핵심자산의 재사용을 통해 소프트웨어 개발 및 유지보수

비용을 감소시킬 수 있으며 시장의 요구사항에 대한 즉각적인 반영과 동시에 개발 생산성과 제품 품질을 향상시킬 수 있는 소프트웨어 재사용 기술이다. 본 논문에서는 소프트웨어 프로세스의 구성요소에 대한 공통성과 가변성을 식별하는 기술로 휘처 중심 (feature-oriented approach) 도메인 분석 방법 [3]을 적용하고 휘처 중심 도메인 분석 방법에서 사용하는 휘처 모델링 방법을 재정의하여 소프트웨어 프로세스의 구성요소에 대한 공통성과 가변성을 식별하도록 한다.

#### 2.2 프로세스 테일러링

CMU/SEI에서는 프로세스 테일러링 (Process Tailoring)을 “개발 조직의 표준 프로세스를 특정한 비즈니스나 프로젝트에 필요한 기술적인 요구에 적합하도록 조절하는 절차”라고 정의하고 있다. 소프트웨어 프로세스도 소프트웨어 제품이 갖고 있는 특징을 갖고 있다 [4]. 소프트웨어 프로세스도 군을 형성하고 있으며 해당 군에 속하는 프로세스들 간에는 공통점과 가변성을 식별하여 특정 프로젝트에 특화된 프로세스를 만들어 낼 수 있다. 따라서 프로세스 테일러링에 소프트웨어 제품 라인 기법을 적용하면 프로세스 재사용 기회를 높일 수 있다.

#### 3. 공통성과 가변성 분석을 이용한 프로세스 테일러링 방안

본 장에서는 재사용 가능한 소프트웨어 프로세스의 핵심자산을 활용하여 특정 프로젝트에 특화된 소프트웨어 프로세스를 생성하는 데에 필요한 소프트웨어 프로세스 라인 구축 방법을 기술한다. 소프트웨어 프로세스 라인 (Software Process Line, 이하 프로세스 라인) 정의는 “특

정 도메인 (소프트웨어 제품 개발 영역)에 속하는 소프트웨어 프로세스들의 공통된 핵심자산을 공유하는 소프트웨어 프로세스들의 집합”으로 소프트웨어 제품 라인 정의를 재정의하여 사용한다. 본 제안 방법에서는 소프트웨어 프로세스 구성요소를 ‘활동, 기법, 역할, 산출물’로 정의하고 소프트웨어 프로세스 구성요소를 포함하고 있는 소프트웨어 프로세스를 소프트웨어 프로세스 휘처 (Software Process Feature, 이하 프로세스 휘처)라 칭한다. 또한, 정의한 소프트웨어 프로세스 구성 요소인 ‘활동, 기법, 역할, 산출물’은 ‘활동 휘처, 기법 휘처, 역할 휘처, 산출물 휘처’로 명한다.

### 3.1 프로세스 영역 선정

프로세스 영역 선정 단계에서는 일반화하고자 하는 프로세스 목적을 파악하고 프로세스 목적을 달성하기 위해 수행해야 하는 활동들을 식별한다. 활동 식별을 위해 선정한 프로세스 영역과 관련된 프로세스 참조모델들을 참조하여 활용한다.

<표 1> 활동 목록표 양식

프로세스 이름	/* 프로세스 이름 기술 */		
프로세스 목적	/* 프로세스 목적 기술 */		
참조모델 이름	참조모델 A	참조모델 B	참조모델 C
활동 내용	/* 활동 목록 기술 */	/* 활동 목록 기술 */	/* 활동 목록 기술 */

<표 2> 활동 목록표 작성 예

프로세스 이름	요구사항 개발 프로세스		
프로세스 목적	고객 기대사항과 제품 및 컴포넌트들에 대한 요구사항을 도출하고 분석하기 위함.		
참조모델 이름	Davis 모델 [5]	Elizabeth 모델 [6]	Pressman 모델 [7]
활동 내용	<ul style="list-style-type: none"> <li>요구사항 추출</li> <li>요구사항 분류</li> <li>요구사항 명세</li> </ul>	<ul style="list-style-type: none"> <li>요구사항 동의</li> <li>요구사항 분석과 모델</li> <li>요구사항 상세 및 전략 결정</li> <li>상세 요구사항 동의</li> </ul>	<ul style="list-style-type: none"> <li>프로젝트 개시</li> <li>요구사항 추출</li> <li>요구사항 상세</li> <li>요구사항 협의</li> <li>요구사항 명세</li> <li>요구사항 확인</li> </ul>

### 3.2 활동 정의 일반화

활동 정의 일반화 단계에서는 프로세스 영역 선정 단계에서 도출한 프로세스 참조모델의 활동 목록들을 서로 비교하여 참조모델 간 활동들의 공통성과 가변성을 식별하고 일반적 활동으로 정의한다.

<표 3> 활동 비교표 양식

참조모델 A \ 참조모델 B	(b) C & V			(d) 활동분할 개수	(h) C & V	(f) 일반적 활동
	활동명					
(a) C & V		(c)	(c)			
		(c)	(c)			

(e) 활동분할 개수					
(i) C & V					
(g) 일반적 활동					

Pressman	C&V	M	M	M	M	M			
일반적 활동							활동분할 개수	C&V	일반적 활동
C&V	활동명	프로젝트 개시	요구사항 추출	요구사항 분석	요구사항 상세	요구사항 협의	요구사항 명세	요구사항 인수 기준 수립	요구사항 확인
M	요구사항 추출		(1,1)					0	M
M	요구사항 분석			(1,1)				0	M
M	요구사항 명세					(1,1)		0	M
O	요구사항 테스트 전략 수립							null	O
O	요구사항 인수 기준 수립							null	O
O	상세 요구사항 동의						(1,1)	0	O
	활동분할 개수	null	0	0	null	0	0		
	C&V	O	-	-	O	-	-		
	일반적 활동	프로젝트 개시	요구사항 추출	요구사항 분석	요구사항 상세	요구사항 협의	요구사항 명세	요구사항 인수 기준 수립	요구사항 확인

(그림 1) 요구공학 프로세스 참조모델 비교 예

더 이상 비교할 참조모델이 없다면 <표 3> 활동 비교표를 이용하여 도출한 일반적 활동 목록을 <표 4> 일반적 활동 목록표 형태로 정리하여 일반적 활동으로 정의한다.

<표 4> 일반적 활동 목록표 양식

프로세스 이름	/* 일반화된 프로세스 이름 기술 */
프로세스 목적	/* 일반화된 프로세스 목적 기술 */
공통성/가변성 구분	일반적 활동 목록
/* 필수, 선택, 태일 여부 기재 */	/* 일반화된 활동 목록 기술 */

<표 5> 일반적 요구사항 개발 활동 목록표 작성 예

프로세스 이름	일반적 요구사항 개발 프로세스
프로세스 목적	고객 기대사항과 제품 및 컴포넌트들에 대한 요구사항을 도출하고 분석하기 위함.
공통성/가변성 구분	일반적 활동 목록
선택	프로젝트 개시
필수	요구사항 추출
필수	요구사항 분석
선택	요구사항 협의
필수	요구사항 명세
선택	요구사항 인수 기준 수립
선택	요구사항 테스트 전략 수립
선택	요구사항 확인

### 3.3 활동 목적 달성에 필요한 기법 식별

기법 식별 단계에서는 활동 목적 달성을 위해 필요한 다양한 기법들을 조사하여 목록화한다. 일반적으로 개발 방법론에는 작업 절차, 작업 방법, 산출물, 기법, 도구에 대한 내용을 포함하고 있기 때문에 여러 개발 방법론을 참조하여 활동 목적 달성에 유용한 기법들을 식별하여 목록화하는 것이 효과적이다.

<표 6> 기법 목록표 양식

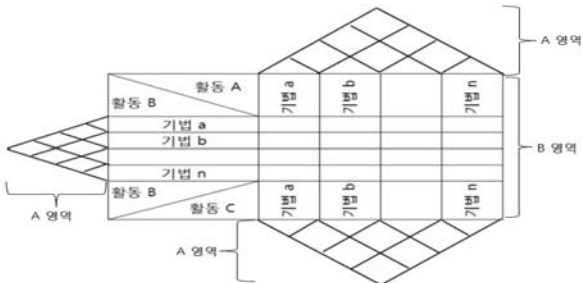
프로세스 이름	/* 일반화된 프로세스 이름 기술 */	
프로세스 목적	/* 일반화된 프로세스 목적 기술 */	
활동 목록	공통성/가변성 구분	기법 목록
/* 일반화된 활동 이름 기술 */	/* 필수 또는 선택 여부 기재 */	/* 기법 이름 기술 */

<표 7> 기법 목록표 작성 예

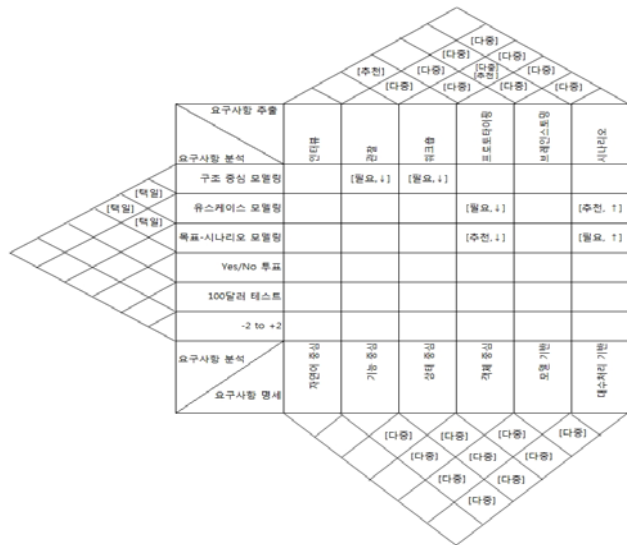
프로세스 이름		일반적 요구사항 개발 프로세스
프로세스 목적		고객 기대사항과 제품 및 컴포넌트들에 대한 요구사항을 도출하고 분석하기 위함.
활동 목록	공통성/가변성 구분	기법 목록
요구사항 추출	필수	인터뷰
	선택	관찰
	선택	브레인스토밍
	선택	워크숍
	선택	시나리오
	선택	프로토타이핑

3.4 활동 간 기법 상호관계 정의

예를 들어 개발 방법론으로 UP (Unified Process) 방법론을 선택하여 일련의 요구사항 개발 프로세스를 진행한다는 것은 요구사항을 식별하고, 정의하는 데 있어서 유스케이스 모델링 기법을 사용하여 요구사항 개발 프로세스를 진행한다는 의미와 같다고 볼 수 있다. 그러므로 개발 방법론 관점에서 개발 전/후 활동에서 사용하는 기법들 사이의 상호관계를 정의할 필요가 있다. 요구사항 추출 활동에서 요구사항 추출 기법으로 워크숍 (workshop) 기법을 선택하는 경우, 일반적으로 워크숍은 워크숍 준비 -> 워크숍 수행 -> 워크숍 종료의 단계로 진행되는데 워크숍 수행 단계에서 진행 방식에 따라 인터뷰 (interview) 기법이나 브레인스토밍 (brainstorming) 기법을 이용하기도 한다. 그러므로 활동 내에서 사용하는 기법들 사이의 상호관계도 정의할 필요가 있다.



(그림 2) 기법 간 상호관계 정의표 양식



(그림 3) 기법 간 상호관계 정의 예

3.5 활동 반복 속성 정의

소프트웨어 개발 프로세스 모델은 소프트웨어 개발을 위해 따라야 하는 활동들을 단계적으로 식별하고 정의하기 위한 기본적인 틀 (framework)을 제공하는 모형 (model)이다. 소프트웨어 개발 프로세스 모델은 크게 폭포수 모델 (Waterfall Model), 점진형 모델 (Incremental Model), 진화형 모델 (Evolutionary Model)로 나눌 수 있으며 개발 방법론과 밀접한 관계가 있다. 그러므로 추진하고자 하는 프로젝트의 특성 및 위험 요인에 따라서 소프트웨어 개발 프로세스 모델을 선정하게 된다. 특히, 점진형 개발 프로세스 모델이나 진화형 개발 프로세스 모델을 따르는 개발 방법론을 선택하게 되면 증가분 (increment)을 만들어 내는 활동을 반복 (iteration)하게 된다. 그러므로 소프트웨어 개발 프로세스 모델 관점에서 활동의 반복 속성을 정의할 필요가 있다.

<표 8> 활동 반복 속성 정의표 양식

일반적 활동	개발 주기			활동 반복 최솟/최댓값 합계
	단계 a	...	단계 n	
활동 a				
...				
활동 n				

<표 9> 활동 반복 속성 정의 예

일반적 활동	개발 주기				활동 반복 최솟/최댓값 합계
	도입	정련	구축	전이	
프로젝트 게시	[1 : 1]				[1 : 1]
요구사항 추출	[1 : 1]	[1 : 2]			[2 : 3]
요구사항 분석	[1 : 1]	[1 : 3]	[1 : 1]		[3 : 5]
요구사항 협의		[1 : 1]	[1 : 1]		[2 : 2]
요구사항 명세		[1 : 2]	[1 : 1]		[2 : 3]
요구사항 인수 기준 수립			[1 : 2]		[1 : 2]
요구사항 테스트 전략 수립			[1 : 2]		[1 : 2]
요구사항 확인			[1 : 2]		[1 : 2]

3.6 프로세스 휘처 모델 정의

활동 일반화 과정을 통해 도출한 결과는 <표 10> 기준으로 일반화한 프로세스 휘처 모델로 정의한다 (그림 4).

<표 10> 프로세스 휘처 모델 표기법

모델 요소	의미	표기법
필수	소프트웨어 프로세스 정의 시 필수적인 휘처임을 의미한다.	●
선택	소프트웨어 프로세스 정의 시 선택 또는 비선택이 가능한 휘처임을 의미한다.	○
관계	부모 휘처와 자식 휘처로 연결됨을 의미한다.	—
태일 선택	여러 휘처 중 휘처 하나만 선택 가능한 것으로 여러 휘처 중 반드시 휘처 한 개는 선택해야 함을 의미한다.	A
다중 선택	여러 휘처 중 적어도 휘처 한 개는 선택해야 함을 의미한다.	△

추천 관계	기법 휘처 간의 인과 관계가 필요 관계보다 약하여 관계된 기법 휘처를 선택하지 않아도 된다. 즉, 관계가 없는 다른 기법 휘처를 선택할 수 있음을 의미한다.	추천 ----->
필요 관계	기법 휘처 간의 인과 관계가 강하여 관계된 기법 휘처를 반드시 선택해야 한다. 즉, 관계가 없는 다른 기법 휘처를 선택 할 수 없음을 의미한다.	필요 ----->
반복성	활동 휘처가 반복 속성이 있음을 의미한다.	[최소 반복 횟수 : 최대 반복 횟수]
프로세스	프로세스 휘처를 의미한다.	<<Process>> 프로세스 이름
그룹	그룹 휘처를 의미한다.	<<Group>> 그룹 이름
활동	활동 휘처를 의미한다.	<<Activity>> 활동 이름
기법	기법 휘처를 의미한다.	<<Method>> 기법 이름
역할	역할 휘처를 의미한다.	<<Role>> 담당역할 이름
산출물	산출물 휘처를 의미한다.	<<Work Product>> 산출물 이름

4. 평가

본 논문에서 제시한 프로세스 라인 구축 방법을 통해 핵심 자산으로 구축해 본 요구사항 개발 프로세스와 기존 요구사항 개발 모델과 비교한 결과는 다음과 같다.

<표 11> 요구사항 개발 프로세스 비교 평가

비교 항목	Davis 모델	Elizabeth 모델	Pressman 모델	본 연구 모델
활동 세분화	+	+	++	+++
활동 반복 속성	없음	지원	없음	지원
기법 간 상관 관계	없음	없음	없음	지원
구성요소 선택 유/무	없음	없음	없음	가능
재사용성	없음	없음	없음	지원
테일러링 가능 여부	없음	없음	없음	가능

\* 개수 비교 기준 : -(없음), +(1개 이상), ++(5개 이상), +++(7개 이상)

비교 평가 결과, 활동 세분화 항목에서는 활동 정의 일화 과정을 통해 참조 모델보다 활동 개수가 늘어남에

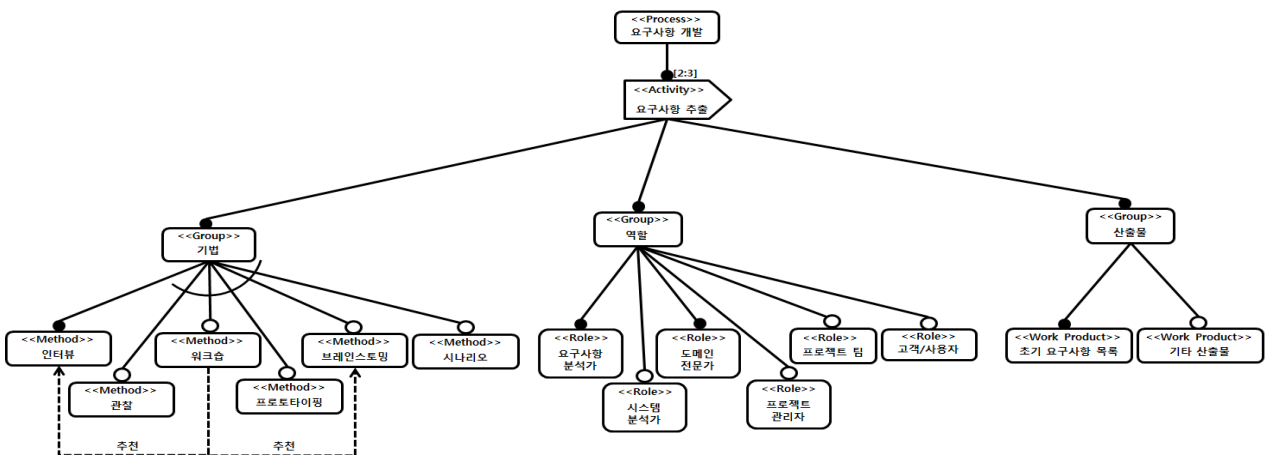
따라 본 연구 모델이 비교 우위를 나타냈다. 활동 반복 속성 항목에서는 Elizabeth 모델과 본 연구 모델이 다른 참조 모델보다 비교 우위를 나타냈다. 또한, 기법 간 상관관계 항목, 구성요소 선택 유/무 항목, 재사용성 항목, 테일러링 가능 여부 항목에서도 다른 참조 모델보다 비교 우위를 나타냈다. 따라서 본 연구 모델을 활용해 소프트웨어 프로세스를 구축하는 것이 유용함을 알 수 있다.

5. 결론

본 논문에서는 소프트웨어 프로세스의 재사용을 가능하게 하는 프로세스 라인 구축 방법을 제안했으며 요구사항 개발 프로세스를 예제로 선정하여 프로세스 라인 구축 방법에 따라 소프트웨어 라인을 구축하는 사례를 체계적으로 제시했다. 향후, 본 연구 성과를 기초로 프로세스 라인 구축 과정에서 사용되는 각종 양식 작성을 자동화할 수 있는 도구, 프로세스 휘처 모델 표기법을 지원하는 프로세스 휘처 모델 저작 도구 등을 개발해야 한다.

참고문헌

- [1] Paul Clements, Linda Northrop, "Software Product Lines: Practices and Patterns", Addison-Wesley, 2001.
- [2] Colin Atkinson et al., "Component-Based Product Line Engineering with UML", Addison-Wesley, 2001.
- [3] K. Kang et al., "Feature Oriented Product Line Software Engineering: principles and guidelines", Domain Oriented Systems Development: Perspectives and Practices, Taylor & Francis, pp. 29-46, 2003.
- [4] L. Osterweil, "SOFTWARE PROCESSES ARE SOFTWARE TOO", ICSE '87, pp. 2-13, 1987.
- [5] Alan Davis, "Just Enough Requirements Management: where software development meets marketing", Dorset House Publishing, 2005.
- [6] Elizabeth H, "Requirement Engineering, 2nd Edition", Springer, 2005
- [7] R. S. Pressman, "Software Engineering: A Practitioner's Approach", McGraw-Hill, 1997.



(그림 4) 요구사항 추출 활동에 대한 휘처 모델 예