

릴리즈 플랜의 적응적 요구사항 우선순위 프로세스

성 재 석[†]·강 동 수^{**}·송 치 양^{***}·백 두 권^{****}

요 약

요구사항에 대한 우선순위는 릴리즈 플랜을 위한 핵심적 활동이기 때문에 요구공학에서 특히, 오픈 시장(Open Market)을 고객으로 하는 시장 주도형 제품개발에 있어서 중요하다. 또한, 요구사항 우선순위는 주어진 요구사항 간의 상호의존 관계를 바탕으로 프로세스 모델, 제품 종류 및 우선순위 프로세스에 대한 경험 등을 사전에 고려하여 우선순위화를 위한 방법과 관점 등을 선택하는 활동이 중요하다. 그러나, 기존 연구들은 요구사항간의 상호의존 관계를 정적 관계만 고려하였고, 고려된 관점들이 비용/가치 등으로 한정적이고 체계적인 우선순위 프로세스를 제공치 못하고 있다. 따라서 본 논문에서는 우선순위화를 위한 모델을 설계하고 개발 제품의 목표와 조직에 적합하도록 우선순위 방법과 관점 등을 선택할 수 있는 적응적 요구사항 우선순위 기법 및 프로세스를 제안한다. 특히, 요구사항간의 정적/동적 상호의존 관계 유형을 정의하고, 다양한 관점에 의한 우선순위화를 통해 릴리즈 플랜의 완성도를 높였다. 이로써 상호의존 관계 및 다양한 관점을 고려한 우선순위 모델기반의 체계적인 우선순위 프로세스를 정립하여 유연하고 충족스러운 우선순위화와 릴리즈 플랜을 통하여 합리적으로 의사결정을 도모할 수 있다.

키워드 : 요구사항 우선순위, 릴리즈 플랜, 상호의존, 요구 공학, 시장 주도형 제품개발

An Adjustable Process of Requirements Prioritizing for Release Plan

Jaeseok Seong[†]·Dongsu Kang^{**}·Cheeyang Song^{***}·Dookwon Baik^{****}

ABSTRACT

The priority of requirement is important because the priority is a critical activity of release plan especially in software development which has an open market customer. Also, it is important for stakeholders to select a method and aspects to prioritize requirements. The selection is based on the organizational experience of a priority process, the process model of the product, goals and a type of products, and dependencies between requirements. But, the current researches considered only static dependency between requirements and did not suggest a systematic priority process. In addition, the current researches only suggest limited aspects to prioritize requirements, such as cost and value. Therefore, this paper proposes an adjustable priority process based on a priority model to select a method and aspects for the suitable priority for product and organization. Especially, this paper enhances the completeness of a release plan by a definition of static and dynamic dependency types between requirements. This paper suggests a priority model, which considers the dependencies between requirement and various viewpoint of software development. Based on the priority model, the paper suggests a systematic priority process to promote reasonable decisions to the priority and release plan of requirement.

Keywords : Requirements Prioritization, Release Planning, Dependency, Requirements Engineering, Market-Driven Development

1. 서 론

요구사항 분석 단계의 다양한 활동 중에 특히 추출된 요구사항을 우선순위화하여 릴리즈 플랜(release plan)을 수립하는 활동의 중요성이 증가하고 있다. 왜냐하면, 패키지 소프트웨어 시장의 등장으로 시장 주도형 소프트웨어 개발(Market Driven Software Development)에 대한 관심이 고객 지향 소

프트웨어 개발(Customer Specific Software Development)에 비해 증가하고 있기 때문이다[1]. 그에 따라 패키지 소프트웨어 개발에 주로 사용되는 반복적이고 점증적인 소프트웨어 개발 프로세스에서 후보 요구사항에 대해 반복(iteration) 별로 어떤 휘처(feature)를 구현할 지 결정하여 계획을 수립해야 하는 릴리즈 플랜이 시장 주도형 요구 공학(Market Driven Requirement Engineering)에서 가장 중요한 활동 중의 하나가 되고 있다. 요구사항을 우선순위화 하는 가장 큰 이유는 제한된 프로젝트의 자원으로 일정 안에 소프트웨어 제품을 출시하기 위함이다[2]. 즉, 보유하고 있는 자원과 시간의 제약으로 인하여 모든 요구사항을 만족시킬 수가 없기 때문에 요구사항은 여러 릴리즈로 분산되어야 하며, 그에 따라 우선순위와 릴리즈 플랜의 중요성이 점차 증가하고 있다[3].

※ 이 연구에 참여한 연구자는 '2단계 BK21 사업'의 지원을 받았다.

[†]준 회원 : 고려대학교 컴퓨터·전파통신공학과 석사과정

^{**}준 회원 : 고려대학교 컴퓨터·전파통신공학과 박사과정

^{***}정 회원 : 경북대학교 소프트웨어공학과 조교수

^{****}종신회원 : 고려대학교 컴퓨터·전파통신공학과 교수

논문접수: 2008년 10월 21일

수정일: 1차 2008년 11월 25일

심사완료: 2008년 11월 25일

우선순위와 릴리즈 플랜 관련하여 다양한 연구가 이루어지고 있지만 요구사항간의 상호의존 관계를 고려하고 또한, 우선순위에 대한 방법(Methods)과 관점(Aspects) 등을 소프트웨어 개발 제품의 다양한 상황에 맞게 선택할 수 있는 릴리즈 플랜에 대한 연구는 미흡하다. 관련된 기존 연구를 크게 두 가지로 나눌 수 있는데, 첫째는 우선순위에 중점을 둔 연구들[3-9]이다. 이들은 각 연구별로 개별적인 우선순위 방법론을 제안하며 특정 관점만을 고려도록 하여 요구사항 우선순위 관점에 대한 선택의 여지가 부족하다. 또한, 각기 상이한 우선순위 프로세스를 갖고 있으며 요구사항이 상호 독립적이라는 가정하에 상호의존 관계를 다루고 있지 않다. 이로 인해 한 개의 릴리즈에 제한되어 우선순위 계획을 수립한다는 것이다. 둘째는 릴리즈 플랜에 초점을 맞춘 연구들[10-15]이다. 이들 연구는 요구사항간 또는 기존 시스템과 새로운 요구사항간의 정적 상호의존 관계 및 영향 분석에 관심을 갖고 릴리즈 플랜을 생성하는 방법 또는, 수학적 예측을 기반으로 개발 역량과 요구사항을 비교함으로써 릴리즈 플랜을 생성하는 기법을 제안하고 있다. 그러나, 이러한 기법들의 공통점은 릴리즈 플랜 프로세스에 포함되어 있는 우선순위 방법이 체계적이지 못하고 그 관점이 비용, 가치, 중요도, 위험 등으로 제한적이라는 문제점을 갖고 있어 충족스러운 우선순위 결과를 제공하지 못하고 있다. 결국, 기존 연구는 요구사항간의 정적/동적 상호의존관계와 다양한 관점이 결합된 일원화된 우선순위 프로세스를 제공하지 못한다.

릴리즈 플랜의 핵심은 고객을 만족시키는 제품을 적기에 출시할 수 있도록 요구사항에 대해 제품 상황에 적절한 관점을 선정하여 우선순위화하며 요구 휘저간의 상호의존 관계의 정의 및 적용을 요구한다. 그러나, 기존 연구에서는 이러한 것들을 충분히 만족시키지 못하고 있다.

따라서, 본 논문은 소프트웨어 개발 제품의 다양한 상황에 적응적으로 요구사항 우선순위 기법을 적용하여 릴리즈 플랜을 수립할 수 있도록 방법과 관점을 유연하게 선택할 수 있는 우선순위 프로세스를 제시한다. 먼저, 우선순위에 직간접적으로 영향을 미치는 요소를 식별하여 우선순위를 위한 모델을 설계하고, 요구사항 간의 상호의존 관계는 정적 관계에 동적 관계를 추가하여 4가지 유형으로 정의한다. 우선순위화 하기 위해 기존의 다양한 우선순위 방법의 특징을 이해하여 개발 조직의 특성에 부합하는 방법을 선택하며, 제품의 개발 목표에 기반하여 요구사항 우선순위의 관점을 우선순위 모델을 기반으로 선택하도록 한다. 선정된 관점에 적절한 가중치를 부여하고, 우선순위화에 참여할 이해관계자를 확정하며, 관점별 기준을 명확히 정의한 후 우선순위화를 실시한다. 이로써 개발 제품과 조직에 적응적이고 유연한 요구사항 우선순위 프로세스를 정립하여 좀 더 충족할 만한 우선순위 결과를 기대할 수 있다.

본 논문은 아래와 같이 구성되어 있다. 2장은 관련연구로 시장 주도형 제품 개발의 특징에 대해 설명하며, 우선순위와 릴리즈 플랜 및 상호의존 관계에 대한 기존 연구의 장단점을 비교 분석한다. 3장은 릴리즈 플랜에 대한 전체 우선순위화 프로세스를 제시하며, 4장에서는 요구사항 우선순위를

위한 모델을 설계하여 이것을 기반으로 한 적응적 요구사항 우선순위 기법과 상호의존 관계 종류를 정의한다. 5장에서는 핸드폰 소프트웨어 개발에 본 논문의 제안을 적용하고 그에 대한 분석 및 평가를 실시하며, 마지막 6장에서는 결론 및 향후 과제를 기술한다.

2. 관련 연구

본 장에서는 시장 주도형 소프트웨어 제품 개발 관점의 요구공학과 반복 점증적 개발방법론이 요구사항 우선순위와 릴리즈 플랜 측면에서 고려해야 하는 특징은 무엇인지 알아 보며, 기존의 우선순위와 릴리즈 플랜 기법에 대해 비교 분석하고 끝으로 기존 연구된 상호의존 관계 유형에 대해 기술한다.

2.1 시장 주도형 요구공학과 반복 점증적 개발

시장 주도형 요구공학과 고객 지향 요구공학의 특징을 <표 1>과 같이 다양한 측면에서 비교할 수 있다. 고객 지향 개발 제품의 고객은 주로 군, 정부와 같은 거대 조직인 경우가 많은 반면, 시장 주도형 제품은 대개 오픈 시장을 대상으로 한다. 시장 주도형 요구 공학의 특징은 여러 연속적인 릴리즈를가진 생명주기(Life Cycle)를 갖게 되고 시장이 있는 한 지속된다는 것으로 릴리즈 플랜이 중요한 활동이 된다. 고객 지향 제품의 요구공학은 협상과 갈등의 해결에 많은 노력을 기울이지만, 시장 주도형 요구 공학은 우선순위, 비용 예측 및 릴리즈 플랜에 좀 더 집중하며 이러한 활동은 개발 조직에 의해 대부분 이루어지며 요구사항이 매우 많고 지속적으로 증가한다[16].

위와 같은 특징으로 인하여 반복 점증적 개발에서 릴리즈 플랜 및 우선순위 추출기법이 가져야 할 특징은 첫째, 신기술, 아키텍처 관련 위험, 중요 요구사항의 누락 및 성능과 관계된 위험의 축소가 반복 계획의 중심이라고 할 수 있다. 즉, 초기 반복에서 심각한 위험은 제거 또는 완화되어야 함으로 초기 반복에 위험이 크거나 또는 아키텍처와 연관된

<표 1> 시장 주도형 개발과 고객 지향 개발 특징 비교[4]

| Facet | Market-driven Development | Customer-specific Development |
|-------------------------|--|--|
| Main Stakeholders | Developing Organization | Customer Organization |
| Users | Unknown, may not exist until product is on market | Known or identifiable |
| Requirements Conception | Invented(by market pull or technology push) | Elicited, analyzed, validated |
| Lifecycle | Several release as long as there is a market demand | One release, then maintenance |
| Specific RE issues | Steady stream of requirements, prioritization, cost estimating, release planning | Elicitation, modeling, validation, conflict resolution |
| Primary Goal | Time-to-Market | Compliance to specification |

〈표 2〉 릴리즈 플랜과 우선순위 기법 특징 비교

| Method Factor | 릴리즈 플랜 중심 기법 | | | 우선순위 중심 기법 | | | | | |
|------------------|----------------------------|------------------------|-------------------------|-----------------------------------|------------------------|--------------------------------------|------------------------|--|-----------------|
| | EMFEM | IFM | EVOLVE | Cost-Value Diagram | \$100 Test | Wiegiers' Method | Numerical Assignments | Planning Game | Ranking |
| Stakeholders | Developer, Project Manager | All Major Stakeholders | All Major Stakeholders | Project Manager, Customers, Users | All Major Stakeholders | Project Manager, Customer, Developer | All Major Stakeholders | Customer, Developer | One Stakeholder |
| Aspects | Effort | Cost, Time-to-Market | Business Value, Urgency | Cost, Value | Importance | Value, Cost, Risk | Importance | Importance, Cost, Risk | Importance |
| Method | No Defined | Heuristics | Optimization heuristics | AHP | Cumulative Voting | Absolute value | Absolute value | Sorting to 3 Categories by customer, developer | Ordering |
| Scales | None | Ordinal | Ordinal | Ratio | Ratio | Ratio | Ordinal | Ordinal | Ordinal |
| Dependency | None | Precedence | Coupling, Precedence | None | None | None | None | None | None |

요구사항을 위치하도록 해야 한다. 둘째, 요구사항의 선후 관계에 따라 우선순위가 되어야 한다는 것이며[17], 셋째로는 특정한 릴리즈에서 구현하기 위해 선택된 요구사항은 하나의 완전한 요구사항의 집합이어야 한다는 것이다. 즉, 상호의존 관계 관점에서 나누어진 각 요구사항 집합 간에는 가능한 서로 독립적이어야 하고 각각 릴리즈될 수 있어야 한다. 많은 상호의존 관계를 가진 요구사항들은 위험을 최소화하고 구현을 쉽게 하기 위해 동일한 릴리즈로 스케줄되어야 한다[5].

2.2 우선순위와 릴리즈 플랜

요구사항 우선순위에 관한 모든 요구사항의 중요성은 동등하지 않으며, 이해관계자 사이에서도 가치와 중요도 등에서 차이가 있기 때문에 또한 제한된 프로젝트 자원으로 정해진 일정에 소프트웨어 제품을 릴리즈하기 위한 방법으로 요구사항에 대해 중요도 또는 구현 순서 등에 따라 우선순위를 하는 것을 말한다[2]. 또한, 릴리즈 플랜이란 시장 주도형 요구 공학에서 시장의 필요를 만족시키기 위해 요구사항 우선순위와 자원 예측을 기반으로 연속적인 제품 출시를 위해 휘처를 어떤 릴리즈에 할당할지 의사 결정하는 활동이라고 정의할 수 있다[18].

우선순위와 릴리즈 플랜 기법에 대한 많은 연구가 이루어져 왔는데 릴리즈 플랜에 중심을 두거나, 단순히 요구사항을 우선순위가하여 선택하는 기법에 초점을 두고 있으며 대표적 기존 기법들은 다음과 같다.

- Estimation-Based Management Framework for Enhance Maintenance (EMFEM)¹⁾ [10]
- Incremental Funding Method(IFM) [11][12]
- Evolutionary and iterative approach (EVOLVE) [13]
- Cost-Value Diagram [6]

- \$100 Test [4]
- Wiegiers' Method [7]
- Numerical Assignments [8][19]
- Planning Game [9]
- Ranking [4]

〈표 2〉는 기존에 연구된 릴리즈 플랜과 우선순위 기법들에 대해서 상호의존 관계를 다루고 있는지, 우선순위에 대해서는 어떤 이해관계자들이 참여하며 우선순위 방법, 관점 및 스케일(Scale)은 어떤 특징이 있는지 비교 분석하였으며, 일부 내용은 [14]를 참조하였다. 〈표 2〉에서 보듯이 릴리즈 플랜에 중심을 두고있는 기법들은 우선 순위 방법이 경험에 의존하거나 명확하게 정의되어 있지 않으며, 상호의존 관계를 다루고 있지는 않지만 정적 상호의존 관계만을 일부 다루고 있다. 반면에 요구사항 우선순위에 초점을 맞춘 기법들은 상호의존 관계를 고려하고 있지 않으며 우선순위를 위한 관점 역시 소프트웨어 개발 제품의 목표 및 다양한 상황을 고려함이 없이 단순한 형태의 우선순위 관점을 정하고 있다.

2.3 상호의존 관계

상호의존 관계란 요구사항간 또는 구현된 시스템과 새로운 휘처간의 기능 측면의 정적, 동적 상호의존 관계를 말하며, 요구사항에 대한 우선 순위는 점증적 개발 계획에서 중요한 활동이지만 요구사항이 상호 연관되어 있기 때문에 우선 순위 기반만으로 일정을 생성하는 것은 어렵다. 일반적으로 요구사항 중에 20%만이 다른 요구사항에 독립적이라고 한다[15]. 이런 배경에서 릴리즈 플랜 즉, 특정 릴리즈에 구현할 요구사항을 선택하는 것은 상호의존 관계의 관점에서 보면 요구사항 휘처의 집합을 분할하는데 있어 그 릴리즈간에 최소의 상호의존 관계가 있도록 선택하는 것이다.

기존 릴리즈 플랜 관련 연구에서 상호의존 관계 종류로는 연결(Coupling)과 선후 관계(Precedence)의 정적 상호의존

1) 이 약어는 단지 편의를 위해 사용된 것이며 최초에 그와 같이 명명된 것은 아니다.

관계만을 고려하고 있으며[13], 릴리즈 플랜과 관계없이 요구사항 또는 휘처간의 상호의존 관계에 관한 연구는 제품 계열(Product Line), 설계 등 다양한 관점에서 이뤄지고 있는데[20-25], 이 연구들과 릴리즈 플랜에서 다루고 있는 상호의존 관계를 정적, 동적, 가치 상호의존 관계(Static, Dynamic, Value dependency)와 같이 세 종류로 나누어 볼 수 있다.

- 정적 상호의존 관계는 휘처들 사이에 고유하게 존재하는 관계로 계층 구조화된 레벨 상에서 휘처간의 의존 관계로 부모-자식과 같은 구조적 관계를 의미한다. 구현 측면에 있어 선후 관계 및 요구사항 간에 불일치(Inconsistency)와 같이 동시에 존재할 수 없거나, 하나의 요구사항을 만족시키면 다른 요구사항의 만족도는 떨어지는 관계가 존재한다.
- 동적 상호의존 관계는 동작 운영 시에 휘처들 사이에 연동되는 의존관계로 계층 구조화된 휘처에서 서로 다른 레벨에서 나타나며 순차적으로 또는 동시에 동작 운영되는 관계로 하나의 휘처가 다른 휘처에 의해 영향을 받아 상태, 행위, 데이터, 코드 등이 변경되는 관계를 의미한다.
- 가치 상호의존 관계는 하나의 휘처가 다른 휘처의 가치나 비용을 증감시키는 관계로[23] 릴리즈 플랜 관점에서 요구사항 또는 휘처간의 상호의존 관계 종류로 보기보다는 제품기획 또는 전략 측면에서의 우선 순위 또는 트리아지(Triage)개념에 포함되는 것이 타당하다고 생각되는 관계이다.

요구 공학에서 피하기 어려운 10가지 함정 중의 하나가 '요구사항에 대해 우선순위화하는 것은 필요 없다.'고[26] 할 정도로 중요한 활동임에도 Ad hoc하게 진행되고 있는 것이 현실이다. 기존에 우선순위와 릴리즈 플랜에 관련된 연구가 많이 있었지만 요구사항간에 정적 상호의존 관계만을 고려하고 있으며 개발 제품과 조직의 특성을 고려하여 우선순위 방법과 관점 등을 선택하도록 하지 못하고 있다. 시장 주도형 소프트웨어 제품의 요구사항은 지속적으로 추출 반영되며 제품을 적기에 출시해야 하기 때문에 초기 반복에 위험을 최소화시킬 수 있도록 우선순위화 및 릴리즈 플랜되어야 한다.

3. 릴리즈 플랜 프로세스

본 논문은 요구사항 우선순위에 영향을 주는 이해관계자, 개발 제품의 목표, 유형 및 제품 개발 방법론에 따라 일원화된 적응적 요구사항 우선순위 기법을 기반으로 릴리즈 플랜 프로세스를 체계화하는 것이다. 이를 위해, 우선순위를 부여함에 있어 선택적이고 관점별 가중치를 부여하는 기법을 적용한다. 제시하는 프로세스의 특징은 다음과 같다.

- 우선순위 모델을 통한 접근
 - 요구사항의 우선순위에 관계하고 영향을 주는 요소

들을 설계하여 이에 기반한 우선순위 프로세스를 구축한다.

- 우선순위 관련 요소를 모델화하여 가변적 우선순위 요소 등의 확장 및 접근이 용이하다.
- 다양한 영향 요소를 고려한 우선순위화
 - 개발 조직의 경험, 요구사항의 규모, 개발 제품의 목표, 애플리케이션 유형, 개발 방법론 등 다양한 영향 요소와 특성에 부합하는 유연하고 충족스러운 우선순위화를 실시한다.
 - 비즈니스/고객/개발 관점을 기반으로 응용의 관점별로 적용적인 우선순위화를 할 수 있다.
- 요구사항 간의 정적 및 동적 상호의존 관계 분석
 - 요구사항 간의 관계는 정적인 관계 즉, 구조적 관계만이 아닌 동작 운영 시에 상호 연동되는 관계 역시 존재하기 때문에 동적 상호의존 관계 분석을 사용한다.
 - 동적 상호의존 관계에 있는 휘처의 바인딩을 통해 릴리즈별로 기능 결합 발생을 방지할 수 있다.

시장 주도형 소프트웨어 제품 개발의 가장 중요한 요소는 적기에 제품을 출시하는 것이다. 따라서, 릴리즈 플랜의 목적은 정해진 출시 일정에 활용 가능한 자원으로 고객에게 최대의 가치를 줄 수 있도록 많은 후보 요구사항들로부터 우선순위화를 통해 최적의 요구사항을 선정하는 것이다[27]. 그를 위해 요구사항 추출부터 릴리즈 플랜 생성까지의 프로세스를(그림 1)과 같이 제안하며 특히, 적응적 요구사항 우선순위 기법(PA6)은 본 논문을 통해 정립한 프로세스로 요구사항 우선순위 모델을 활용한다. 요구사항 추출(PA1)부터 상호의존 관계 정의(PA4) 또는, 우선순위 방법 선택(PA61)과 관점 선택 및 가중치 부여 (PA62)단계와 같이 수행 과정 및 결과에 따라 상위 단계의 수정 및 보완이 필요한 경우도 있다.

3.1 요구사항 추출 (PA1: Process Activity 1)

시장 주도형 제품 개발에서 요구사항의 추출은 전통적인 요구사항 추출 방법과 차이가 있다. 소프트웨어 제품이 시장에 최초로 출시되기 이전에는 명확한 사용자 또는 고객이 존재하지 않으며 다만 잠재적인 고객이 있을 뿐이다. 따라서 최초에는 제품의 전략적 목적에 기반하여 관련 조직에 의해 요구사항은 만들어 진다. 전통적으로 사용되는 요구사항 추출 기법이 적절하지 않으며 일반적으로는 시장 주도형 요구공학에서는 시장조사가 가장 보편적으로 사용된다. 시장에 제품이 출시된 이후에는 기술보고서, 결점보고서 등 다양한 경로를 통해 요구사항이 지속적으로 추출된다[27].

3.2 요구사항 계층 구조화 (PA2)

추출된 요구사항을 그룹핑 및 체계화하여 상위수준에서 하위수준으로, 전체에서 부분으로 계층적으로 구조화하는 것은 구현할 전체 시스템을 표현하는데 효과적인 방법이다. 기본적인 방법으로 Affinity Diagram 또는 KJ(Kawakita Jiro)법을 활용할 수 있으며[28] 가장 일반화된 요구사항이 상위

레벨에 위치하게 되며 가장 구체적인 요구사항이 하위레벨에 놓이게 된다. 이 과정을 통해 개발할 전체 시스템의 모습과 고객 요구사항이 구체적으로 무엇인지에 대해 명확하게 이해하게 된다. 또한, 다양한 이해관계자들에 의해 발생할 수 있는 요구사항간의 모순, 갈등 또는 일관성이 결여되는 부분을 파악, 제거할 수 있는 장점이 있다.

3.3 휘처 식별 (PA3)

요구사항이 갖고 있는 여러 문제점 중 하나는 요구사항이 상당히 많을 수 있다는 것과 추상화 레벨이 상이함에 따른 우선순위화, 상호의존 관계 및 자원 예측 등을 수행할 때 어려움이 발생할 수 있다는 것이다[4]. 따라서, 본 논문에서 릴리즈 플랜 프로세스를 설계하는데 있어 기본적인 개체로서 개별적인 요구사항을 다루지 않고 휘처를 사용하도록 한다[20,29]. 여기서 사용할 휘처라는 용어의 정의는 이해관계자의 관점에서 강하게 연결된 요구사항의 집합[22], 또는 결합력 있는 식별 가능한 기능 유닛(Unit)으로 표현되는 개별 요구사항의 집합화로 정의할 수 있다[30].

휘처에 대한 개념 및 휘처에 대한 식별방법에 대한 것은 [30-35]를 참고하도록 하며, 이하 논문에서 사용되는 '요구사항', '요구 휘처'는 '휘처'와 동일한 의미로 간주하도록 한다.

3.4 상호의존 관계 정의 (PA4)

2.3절에서 언급한 바와 같이 릴리즈 플랜에서 가장 핵심적인 요소 중의 하나가 바로 휘처간에, 또는 기존 시스템과 새로 요구된 휘처간의 상호의존 관계를 파악하여 릴리즈 플랜에 반영하는 것이다. [34]에 의하면 릴리즈 플랜을 실시하면서 겪는 어려움에 대한 설문 조사 결과 요구사항에 대한 상호의존 관계에 대한 만족할 만한 전체상을 얻을 수 있는 쉬운 방법이 없다는 것이 가장 많은 응답률을 받기도 하였다.

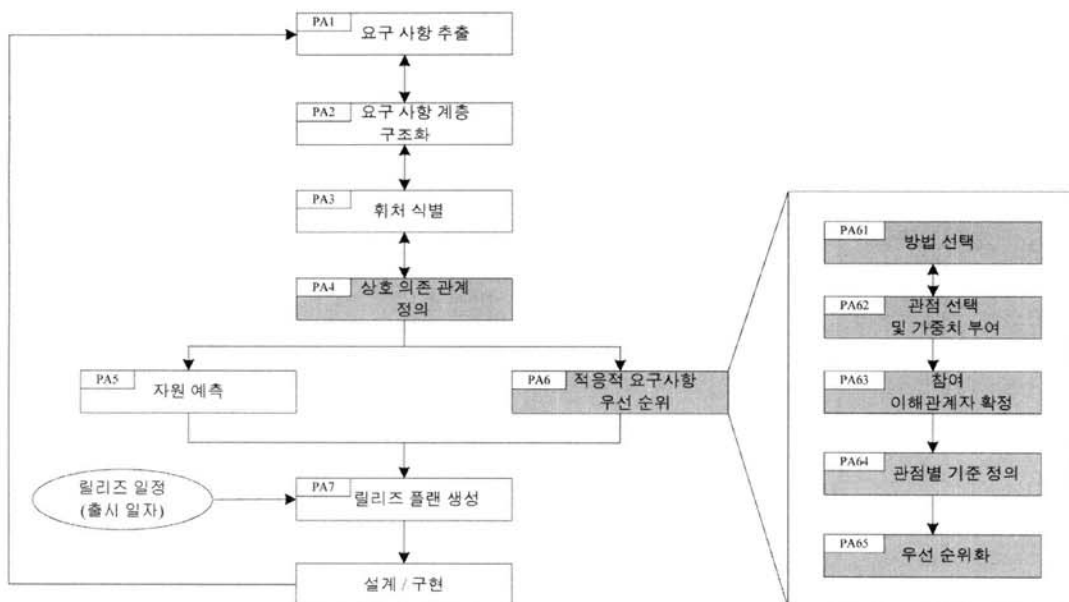
상호의존 관계를 확인하기 위해서는 파악하고자 하는 관계 종류를 정의하고 휘처간의 어떤 관계가 있는지 상호 비교하는 것이 일반적인 방법이다. 이 과정에 문제가 될 수 있는 것이 두 가지인데 첫째로는 관계 종류에 대한 명확한 이해가 필요하다는 것이다. 따라서 휘처간의 상호의존 관계의 종류는 다양할 수 있지만 릴리즈 플랜 관점에서 반드시 고려해야 하는 관계 유형을 식별하여 정의하는 것이 중요하다. 둘째, 복잡한 대형 시스템 개발의 경우에 휘처의 수가 많아서 확장성(Scalability)이 문제가 될 수 있다. 이와 같은 경우, 상호 비교의 수를 줄일 수 있는 방법은 상호의존 관계가 대단히 많은 휘처를 먼저 식별하여 이 휘처를 중심으로 상호 비교를 수행하는 것이다. [5]에서 20%의 요구사항이 상호의존 관계 전체의 67~79%를 차지한다는 결과가 있기도 하다. 본 논문에서 관련 연구를 기반으로 제안하고자 하는 상호의존 관계 종류에 대한 내용은 4.3절에서 세부적으로 다룬다.

3.5 자원 예측 (PA5)

자원을 예측하는 것은 소프트웨어 공학의 초기부터 중요한 이슈중의 하나이며, 특히 요구사항 단계에서 예측을 하고 정확도를 높인다는 것은 쉬운 일이 아니다. 그러나 이러한 자원에 대한 추정이 없이는 계획을 위한 밑바탕을 제공할 수 없으므로 적절한 예측 방법의 선택과 지속적인 예측 훈련을 통해 정확도를 올리는 것이 중요하다.

3.6 적응적 요구사항 우선순위 (PA6)

요구사항에 대한 우선 순위화는 요구 공학에서 특히, 시장 주도형 제품 개발에서 상호의존 관계와 함께 중요한 활동 중의 하나이다. 우선 순위화를 위해 기존 우선순위 방법에 대한 이해와 제품과 조직의 특징 등에 따라 우선순위 방



(그림 1) 릴리즈 플랜의 적응적 우선순위화 프로세스

법과 관점을 선택하며 그에 대해 적절한 가중치를 부여하고, 활동에 참여할 이해관계자를 확정된 다음 관점 별 기준을 명확히 정의하게 된다. 적응적 요구사항 우선순위 프로세스에[36] 대해서는 4.2절에서 자세하게 기술한다.

3.7 릴리즈 플랜 생성 (PA7)

릴리즈 플랜의 일반적인 목적은 많은 후보 요구사항으로부터 가용한 자원을 활용하여 정해진 출시 일정 안에 제품의 목표 달성 및 고객에게 가치를 최대화할 수 있는 요구사항의 집합을 선택하는 것으로 매우 복잡한 활동이다. 릴리즈 플랜을 생성하는 이 단계에서는 앞 프로세스에서의 결과를 바탕으로 시장 상황 등 전략적 측면에서 결정된 릴리즈 별 출시 일정이 함께 고려되어 각 릴리즈별로 구현 되어야 할 회차를 선정하게 된다. 상호의존 관계, 요구사항 우선순위 및 출시 일정 등 제약 사항을 만족하는 여러 가지의 릴리즈 플랜이 나올 수 있으며, 그 결과를 프로젝트관리자와 요구분석가가 위험 측면 또는 자원의 효율적 사용 등을 검토하여 결정하게 된다[12].

4. 우선순위 모델 기반 적응적 요구사항 우선순위 프로세스

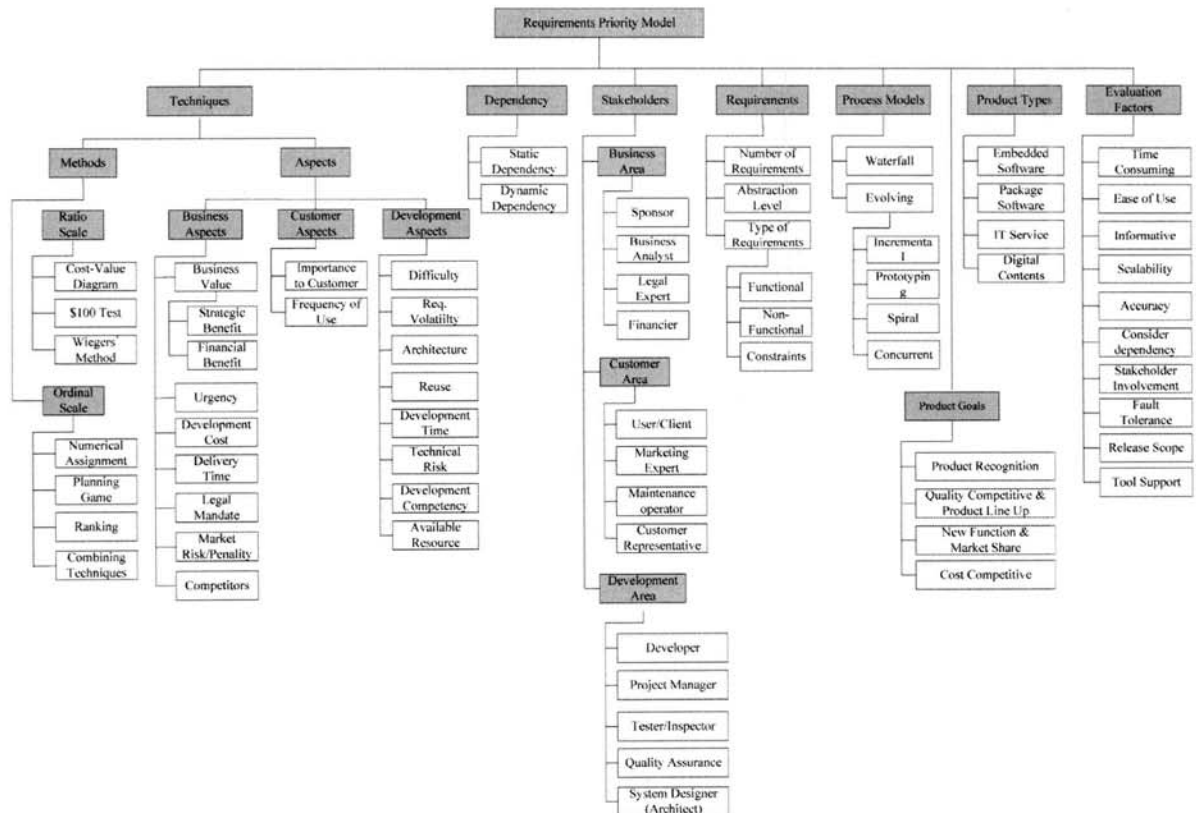
본 장에서는 요구사항 우선순위 모델의 내용에 대해 기술한 이후에, 그에 기반한 적응적 요구사항 우선 순위 프로세

스와 상호의존 관계 종류를 기술한다.

4.1 우선순위 모델

요구 회차에 대한 우선순위는 시장 주도형 소프트웨어 개발에서 중요한 활동 중의 하나로 다양한 이해관계자의 관점에 따라 중요도, 긴급도, 위험 등과 같은 관점에 기반하여 요구 회차에 대해 순위를 매기거나 중요 정도를 나타내는 활동이다. 그러나, 기존 연구에서는 2.2절에서 논의한 바와 같이 개별적인 요구사항 우선순위 기법과 기법간의 비교 연구가 이루어지고 있으며 또한, 우선순위 전반에 대해 다루는 연구가 이루어 지고 있지만[35,37], 우선순위화 이전에 개발 제품의 상황과 여건에 맞도록 유연성을 갖고 우선순위 관점과 방법 등을 선택 적용하지 못하고 있다.

따라서 본 연구에서는 기존에 연구된 요구사항 우선순위 방법으로부터 구성요소를 추출, 관계성을 식별하여 (그림 2)와 같이 요구사항 우선순위 모델을 설계하여 방법과 관점 등을 제품 상황에 적합하도록 선택할 수 있도록 제공한다. 우선순위 모델은 기법(Techniques), 상호의존 관계(Dependency), 이해관계자(Stakeholders), 요구사항(Requirements), 프로세스 모델(Process Model), 제품 목표(Product Goals), 제품 종류(Product Types), 평가 요소(Evaluation Factors)로 구분되어 있으며, 각 요소에 대해 우선순위 시에 영향을 주는 특징에 대해 아래에 기술한다.



(그림 2) 요구사항 우선순위 모델

4.1.1 우선순위 기법(Techniques)

요구사항 우선순위 기법은 방법과 관점으로 나눌 수 있으며 요구사항, 프로세스 모델, 제품 종류 및 목표, 조직의 경험에 따라 적절한 방법과 관점을 선택하는 것이 필요하다.

먼저, 이러한 우선순위 방법들의 가장 두드러진 차이점은 어떤 척도(Scale)와 우선순위 관점을 사용하는가 이다. 그 척도에는 명목(ordinal)과 비율(Ratio) 척도로 나눌 수 있으며, 명목 척도는 숫자 간격의 크고 작음에는 의미가 없으며 숫자의 순서에만 의미가 있게 된다. 대표적인 방법으로 Numerical Assignment, Planning Game 등이 있다. 비율척도는 숫자의 간격에 의미를 부여할 수 있게 되지만, 복잡한 계산을 필요로 한다. 이 척도로 구분되는 대표적인 방법으로 AHP법에 기반한 비용-가치 다이어그램(Cost-Value Diagram), \$100 테스트, Wiegers' Method 등이 있다[38]. 이러한 척도에 대한 이해와 더불어 우선순위 기법의 장단점을 충분히 이해하고 있는 것이 방법 선택에 있어 중요하다.

요구 회차를 우선순위화하기 위해서는 다양한 관점들 중에서 어떤 기준에 의해 선택된 관점으로 순위화하여야 한다. 개발 제품이 어떤 목표를 갖고 있는냐에 따라 관점을 선택하는 것이 합리적임으로 다양한 관점을 체계화하기 위해 우선순위 모델에서는 비즈니스, 고객, 개발의 세 가지의 뷰(View)로 관점을 모델링 함으로써 개발 제품의 목표에 부합하는 관점을 선택한다.

- 비즈니스 관점(BA: Business Aspects)은 사업의 관점에서 요구사항에 대해 우선순위화하는데 있어 가질 수 있는 관점으로 전략적/재정적 이득에 영향을 주는 정도와 같은 비즈니스 가치, 긴급도, 개발 비용과 시간, 법적 의무, 시장 위험 및 경쟁사 현황(기능보유/개발) 등이 될 수 있다.
- 고객 관점(CA: Customer Aspects)은 고객의 중요도와 사용 빈도와 같이 사용자 관점을 말한다.
- 개발 관점(DA: Development Aspects)은 개발자의 관점에서 관심이 가는 인자들로 난이도, 요구사항 안정도, 아키텍처 영향도, 재 사용성 등의 관점을 말한다.

4.1.2 상호의존 관계(dependency)

많은 연구 결과 요구사항 각각은 독립적이지 않으며, 그들 사이에 다양한 종류의 상호의존 관계가 있다는 것이 알려져 있다[22]. 2.3절에서 언급한 바와 같이 그 종류는 크게 정적, 동적 상호의존 관계로 분류할 수 있는데 기존 기법에서는 동적 상호의존 관계를 고려하고 있지 않다.

4.1.3 이해관계자(Stakeholders)

이해관계자는 우선순위에 영향을 미치는 중요한 인자이다. 요구사항에 대해 누가 어떤 입장으로 선택된 관점에 대해 선호도를 갖느냐가 중요하기 때문이다. 여기서 이해관계자는 새로운 시스템 또는 애플리케이션의 구현에 의해 실질

적으로 영향을 받는 사람 또는 조직을 말한다[31].

우선순위화를 위해 이해관계자를 우선순위 관점에 대응하도록 비즈니스 영역(Business Area), 고객 영역(Customer Area) 및 개발 영역(Development Area)의 3가지 뷰(View)로 분류하였다. 비즈니스 영역의 이해관계자는 수익과 전략적 측면에, 고객 영역의 이해관계자는 제품 사용자의 가치에 그리고, 개발은 기술적이 부분에 주로 관심을 갖는다.

4.1.4 요구사항(Requirements)

요구사항의 규모, 추상화 수준 및 종류가 우선순위 방법 선택에 영향을 주는 인자로 볼 수 있다[39]. 추상화 수준을 결정하는 것은 어려운 일이며 요구사항의 숫자와 그 복잡도에 달려있다. 따라서 본 논문에서 제안하는 계층 구조화를 통한 회차 도출이 유용한 방법이 될 것이다. 요구사항의 종류 역시 우선순위에 영향을 줄 수 있으며 특히, 비 기능 요구사항 또는 제약사항에 대한 우선순위를 어떻게 할 것인가에 대한 연구는 비교적 수행되지 않은 영역이지만 점차 비 기능적 요구사항이 중요해 지고 있기 때문에 연구가 필요한 부분이다.

4.1.5 프로세스 모델(Process Model)

프로세스 모델은 릴리즈 플랜의 범위 및 릴리즈 일정 수립에 영향을 줄 수 있다. 프로세스 모델이 폭포수 모델이라면, 하나의 릴리즈 플랜 만 수립되게 되며 정해진 출시 일정에 맞추기 어려운 우선순위가 낮은 요구사항에 대해서는 고객 또는 내부적 협상이 주요 이슈가 된다. 진화적 개발 모델이라면 하나 이상의 릴리즈 플랜이 수립되게 될 것이며, 기존 컴포넌트의 재사용, 수정 또는 기존 구현 시스템과의 영향을 고려해야 한다.

4.1.6 제품 목표(Product Goals)

소프트웨어 제품 개발의 목표는 다양할 수 있지만 (그림 2) 요구사항 우선순위 모델에서와 같이 일반적으로 상품 인지도(Product Recognition), 품질 경쟁력과 제품 구성(Quality Competitive & Product Line Up), 신규기능 개발과 시장 점유율(New Function & Market Share) 및 가격 경쟁력(Cost Competitive)과 같이 4가지로 분류해 볼 수 있다. 이것은 우선순위를 위한 관점, 즉 무엇에 기준을 두고 순위화할 것인가에 영향을 준다.

4.1.7 제품 종류(Product Types)

소프트웨어 제품 종류를 임베디드, 패키지, IT서비스 및 디지털 콘텐츠로 구분할 수 있다. 개발 제품이 임베디드 소프트웨어 제품이라면 하드웨어 관련 요구사항 및 그에 대한 일정이 함께 고려되어야 하며, 패키지 소프트웨어라면 시장 주도형 제품이기 때문에 우선순위, 비용 예측 및 릴리즈 플랜이 중요한 활동이 된다. IT 서비스 및 디지털 콘텐츠 제품은 고객 지향 제품으로 협상과 갈등의 해결이 우선순위보다 중요한 활동이 된다.

4.1.8 평가 요소(Evaluation Factors)

평가 요소는 요구사항 우선순위에 직접적인 영향을 주는 요소는 아니지만 우선순위 방법을 선택하기 위해 방법을 상호 비교할 때에 필요한 요소이다. 다양하게 제안된 기법들에 대해 비교 평가를 수행한 연구들에서 사용한 평가 요소들이다[3,4,40-43]. 평가 요소 별 객관적 기준이 정의되어 있지는 않으며 우선순위에 소요되는 시간 (Time Consuming), 편의성(Ease Of Use), 정보량 (Informative) 등이 있다.

4.2 적응적 요구사항 우선순위 프로세스

3장 (그림 1)의 적응적 요구사항 우선순위화 프로세스에서 보는 바와 같이 우선순위 모델을 활용하여 먼저 우선순위 방법과 관점을 선택하여 가중치를 부여한 후 참여할 이해 관계자를 확정하며 점수 부여 기준을 명확히 하는 활동이 적응적 요구사항 우선순위 프로세스의 핵심적 활동이다. (그림 2)의 요구사항 우선순위 모델은 기존 우선순위 기법들의 구성요소와 요구사항 우선순위화를 위해 고려해야 하는 요소들과 그 관계를 식별하여 설계하였다. 따라서, 우선순위를 수행할 때 제안된 우선순위 모델을 활용하여 개발한 소프트웨어 제품에 적합한 새로운 우선순위 기법을 설계하여 수행할 수 있다.

4.2.1 방법 선택 (PA61)

요구사항 우선순위 방법 선정을 위해서는 첫째, 기존 우선순위 방법의 특징을 이해하고 둘째, 개발 및 조직의 우선순위 적용 경험을 고려하며, 마지막으로 개발 프로젝트의 규모 즉, 요구사항의 규모를 고려하여 방법을 선택하도록 한다. 어떠한 우선순위 관점을 선택하느냐에 따라서도 방법 선정에 영향을 줄 수 있다. 왜냐하면, 선정된 관점이 기존 기법과 동일하다면 그 기법 그대로를 적용하는 것이 유용할 수도 있기 때문이다. 따라서, 방법과 관점의 선택은 상호 영향을 주게 됨으로 순차적으로만 진행되지 않는다.

1st 선정기준) 우선순위 방법의 특징 이해

2.2절에서 우선순위 방법들에 대해서는 간단히 특징 또는 장단점을 소개하였다. 우선순위 모델의 평가요소 중 직접적인 우선순위 활동과 별도로 수행되는 활동 관련 평가 요소인 상호의존 관계 고려(Consider Interdependency)와 릴리즈

범위(Release Scope)는 제외하도록 하며, 정확도(Accuracy)에 대한 평가는 쉽지 않고, 참고 할 비교 실험에 관한 연구가 부족하여 정보량(Informative)과 비례할 것으로 추정되어 제외하여 아래의 7개 인자로 기존 비교 연구를 참고하여[3, 4, 40-43] 우선순위 중심 기법 6개에 대해 <표 3>과 같이 평가를 하였다. 평가의 입자성(Granularity)은 높지 않으나 우선순위 방법 선택의 기준으로 사용할 수 있는 수준이라고 할 수 있다.

- Time Consuming: 우선순위화에 소요되는 시간 평가
- Ease Of Use: 사용 편의성 평가
- Informative: 우선순위의 분석 정보량 평가
- Scalability: 요구사항의 규모가 클 때 적용이 용이한가를 평가
- Stakeholder Involvement: 개발 제품에 관련된 이해관계자들의 체계적 식별 유무 평가
- Fault Tolerance: 우선순위가 제대로 수행되었는지 자체 점검 방법 유무 평가
- Tool Support: 우선순위를 위한 Tool지원 유무 평가

<표 3>의 평가 결과를 보면, Numerical Assignment, Ranking 및 Planning Game이 대체적으로 확장성(Scalability), 편의성(Ease Of Use), 시간소모(Time Consuming)면에서 좋은 평가를 받은 반면에 분석 정보량(Informative) 측면에서 좋지 못한 평가를 받고 있다. 그러나, AHP법에 기반한 비용-가치 다이어그램은 분석 정보량에서는 좋은 평가를 받고 있지만 확장성, 편의성, 시간소모 측면에서는 낮은 평가를 받고 있다. <표 3>를 참고하여 프로젝트 상황에 적절한 우선순위 방법을 선정한다.

2nd 선정기준) 개발 조직의 요구사항 우선순위 프로세스 적용 경험 고려

적용 경험이 있다면 사용했던 방법 또는 시간이 소모되고 복잡하지만 좀 더 정보량이 많은 방법을 선택할 수도 있을 것이다. 그러나, 우선순위 프로세스 적용 경험이 없다면 갈등과 저항을 줄이기 위해 사용하기 쉬운 간단한 방법부터 적용하는 것이 바람직하다.

3rd 선정기준) 요구사항의 규모

요구사항의 규모가 매우 크다면 확장성이 좋은 방법을 선

<표 3> 기존 우선 순위 기법 평가

| Factor | Method | Cost-Value Diagram | \$100 Test | Wiegiers' Method | Numerical Assignments | Planning Game | Ranking |
|----------------|-------------------------|--------------------|------------|------------------|-----------------------|---------------|---------|
| Prioritization | Time Consuming | 1 | 3 | 3 | 3 | 5 | 3 |
| | Ease Of Use | 1 | 3 | 3 | 5 | 5 | 3 |
| | Informative | 5 | 5 | 5 | 3 | 3 | 3 |
| | Scalability | 1 | 3 | 3 | 5 | 5 | 5 |
| | Stakeholder Involvement | 3 | 5 | 3 | 5 | 5 | 1 |
| | Fault Tolerance | 5 | 1 | 1 | 1 | 1 | 1 |
| Common | Tool Support | 3 | 1 | 1 | 1 | 1 | 1 |

[범례] 5: 매우 우수, 4: 우수, 3: 보통, 2: 미흡, 1: 매우 미흡

택 사용한다.

이상에서 살펴본 바와 같이 요구사항 우선순위를 위해 어떤 방법을 선택하느냐는 방법 간에 trade-off가 있기 때문에 매우 어려운 의사 결정이 될 수 있다. 분석의 정보량을 높이면서 시간 소모나 적용의 어려움을 희생할 것인지, 정보량이 다소 부족하더라도 간편하게 빠르게 우선순위를 수행할 것인지를 선택한다.

4.2.2 관점선택 및 가중치 부여(PA62)

요구사항 우선순위 관점은 다양할 수 있기 때문에 어떤 관점을 가지고 요구사항에 대한 우선순위를 수행해야 하는지 결정하는 것은 쉬운 일이 아니지만 목표와 위험 완화를 바탕으로 합리적인 관점을 선택하고 가중치를 부여할 수 있다. 이렇게 선정된 관점이 2.2절과 같이 기존에 연구 제안된 방법의 관점과 다르지 않다면 그 방법을 선정하여 우선순위화 하는 것이 바람직할 것이다. 그렇지 않다면 4.1절의 우선순위 모델과 기존 우선순위 방법을 응용하여 관점을 체계화하여 아래 제시한 3가지 기준에 따라 우선순위 관점을 선택 우선순위화하도록 하는 것이 바람직하다.

본 논문에서 제안하는 관점 선택의 기준은 첫째, 제품의 목표를 기반으로 선정하는 것이다. 요구공학에서 목표가 차지하는 비중은 크다고 할 수 있다. [44]에서 목표는 요구사항을 식별하고 구조화하고 타당성을 증명하는 논리적인 메커니즘이라고 정의하고 있다. 또한, [45]에서는 부적절한 요구사항을 피하는 것이 요구공학의 중요 관심사 중의 하나인데 목표는 요구사항의 적절성에 대한 분명한 기준을 제공한다 라고 목표의 역할에 대해 언급하고 있다. 따라서, 목표는 요구사항을 우선순위화하는 관점을 선택하는 중요한 기준이 된다. 개발할 제품의 목표에 따라 우선순위 모델에서 구분한 비즈니스(BA), 고객(CA), 개발(DA)의 세가지 측면에서 관점을 선택할 수 있는데, 그 목표는 다양할 수 있지만 일반적으로 그 목표를 아래와 같이 4가지 케이스(Case)로 나누어 볼 수 있다. 케이스별 목표에 따라 우선순위 모델의 관점 분류 체계를 활용하여 관점을 선택하고 케이스별 관점의 중요도(<표 4> 참조)에 따라 가중치를 부여한다.

- 1st Case) 시장에서 상품의 인지도를 높이는 것이 중요한 목표일 경우에는 비즈니스 가치, 전략적 이점, 출시 일정, 고객 중요도와 같이 비즈니스와 고객의 관점이 중요하다. (관점의 중요도 순서 BA>CA>DA)
- 2nd Case) 품질 경쟁력과 제품 구색이 중요한 목표일 경우에는 고객 중요도, 재 사용성, 기술적 위험 최소화,

경쟁사와 같은 고객과 개발의 관점이 중요하다. (관점의 중요도 순서 CA>DA>BA)

- 3rd Case) 새로운 기능 개발 및 시장 점유율 유지가 중요한 목표일 경우에는 고객 중요도, 출시 일정, 비즈니스 가치와 같은 고객과 비즈니스의 관점이 중요하다. (관점의 중요도 순서 CA>BA>DA)
- 4th Case) 가격 경쟁력이 중요한 목표일 경우에는 난이도, 요구사항 변동성, 재 사용성과 같은 개발과 비즈니스의 관점이 중요하다. (관점의 중요도 순서 DA>BA> CA)

두 번째 관점 선택의 기준은 반복 점증적 개발의 초기 반복에서 위험을 최소화해야 한다는 것이다. 2.1절에서 언급한 바와 같이 반복 점증적 개발에서 우선순위 추출 기법이 가져야 할 중요한 특징 중의 하나가 초기 반복에서 심각한 위험요소가 제거될 수 있도록 하는 것이 우선순위화의 중요한 관점이 된다.

셋째, 관점의 개수가 최대 5개 이하로 선정하는 것이다. 관점의 개수에 대한 명확한 기준을 정의하기는 어렵다. 그러나, 선택한 관점의 개수가 많을 경우 그에 따라 우선순위에 소요되는 시간이 증가하게 된다. 게다가 대상 휘처가 많을 경우 특히, AHP를 기반으로 한 우선 순위 방법에선 기하급수적으로 시간이 증가할 수 있기 때문에 관점을 선택할 때 그 개수에 주의해야 한다.

관점을 선택한 이후에 그에 대해 가중치를 부여하게 되는데 가장 일반적으로 사용하는 방법으로 선택된 관점에 대한 가중치의 합이 1이 되도록 가중치를 부여하며, BA, CA, DA의 세가지 측면 각각에서 관점이 선택되었다면 <표 4>의 제품 개발 목표에 따른 관점의 중요도 순서에 따라 가중치를 부여한다.

4.2.3 참여 이해관계자 확정(PA63)

개발 제품의 목표에 기반하여 선정된 관점에 대해 우선순위화 하기 위해 각 관점에 대해 가장 정확하게 요구사항에 대해 우선순위화 할 수 있는 이해관계자를 우선순위 모델을 활용하여 선정한다. 즉, 관점과 이해관계자의 세가지 분류체계가 서로 대응되도록 이해관계자를 복수로 선정한다. 예를 들면, 개발 관점에서 난이도(Difficulty)에 대해서는 개발 영역의 개발자를 복수로 선정한다.

4.2.4 관점별 기준 정의(PA64)

우선순위 방법과 관점을 어떻게 선정하느냐와 더불어 중요한 것은 참여한 이해 관계자들이 선정된 방법과 관점 특히, 관점에 대해 동일하게 이해할 수 있도록 관점을 명확히 정의하고 점수 부여 기준을 정의하는 것이다[38]. 선택한 관점으로 각 요구 휘처에 대하여 평가를 수행하여 점수를 부여하게 된다. 이 때에 그 기준이 명확하지 않으면 이해관계자 사이의 평가 기준이 달라 정확성이 저하되게 되며, 시간 소모도 증가하게 된다. 따라서 우선순위화에 참여하는 이해관계자 사이에 일치된 의견을 이룰 수 있도록 기준을 명확히 하는 것은 매우 중요하다.

<표 4> 제품 목표에 따른 BA, CA, DA의 중요도

| Case | Goals | BA | CA | DA |
|----------|---------------|----|----|----|
| 1st Case | 상품 인지도 | | | |
| 2nd Case | 품질 경쟁력과 제품 구색 | | | |
| 3rd Case | 신규 기능과 시장 점유율 | | | |
| 4th Case | 가격 경쟁력 | | | |

[법례] 색이 짙을수록 해당 Case에서 관점의 중요도가 높다

4.2.5 우선순위화(PA65)

식별 열거된 휘처에 대해 우선순위 방법과 관점의 선택 및 가중치를 부여한 이후 참여할 이해관계자를 확정하고 우선순위 작업을 수행하게 된다. 이해관계자를 우선순위 관점의 분류와 마찬가지로 비즈니스, 고객, 개발의 3가지 뷰(View)로 하였다. 따라서 이해관계자는 자신의 담당 영역에 해당하는 관점에 대해서만 우선순위화를 실시한다. 예를 들어, 고객 영역의 이해관계자인 마케팅 전문가(Marketing Expert)는 고객 관점의 고객 중요도와 사용빈도에 대해서 우선순위 활동을 하도록 한다. 이해관계자 영역 내에 여러 명이 참여할 수 있고 그로 인해 상호 작용이 있을 수 있기 때문에 익명으로 우선순위 활동을 수행한 후 통합하는 것이 바람직하며, 이 때 복수 개의 값을 처리하는 방법은 델파이 방법 또는 중앙값을 활용한다.

4.3 상호의존 관계 정의(PA4)

하나의 요구사항에 의해 수행된 동작은 의도하거나 기대하지 않았음에도 다른 요구사항에 영향을 줄 수 있기 때문에 요구사항은 복잡하게 관계되어 있고 서로 영향을 주기 때문에 독립적으로 다루어질 수 없다.

4.3.1 정적/동적 상호의존 관계 유형

2.3절 관련연구에서와 같이 상호의존 관계 종류를 정적, 동적 및 가치 상호의존 관계의 세 종류로 나누어 볼 수 있다. 기존 릴리즈 플랜 기법에서는 정적 관계의 일부만 고려하였으며, 가치 관계 종류의 경우 상호의존 관계 종류에서 고려하기보다는 우선순위의 관점에서 다루는 것이 타당하다.

휘처간 상호의존 관계 종류는 다양할 수 있지만 기존 연구를 바탕으로 하여, 특히 릴리즈 플랜에서 고려해야만 하는 최소한의 상호의존 관계를 동적 연결(Dynamic Coupling), 정적 연결(Static Coupling), 구현 선후 관계(Precedence by implement) 및 불일치(Inconsistency)의 4가지로 정의한다.

- 동적 연결은 구현 후 동작 운영할 때 연동되는 관계로 순차적으로 또는 동시에 동작 운영되거나, 하나의 휘처가 다른 휘처에 의해 영향을 받아 상태, 행위, 데이터, 코드 등이 변경되는 관계로 그 결합 정도에 따라 Mandatory와 Optional로 나눌 수 있다. 즉, 두 개의 휘처가 반드시 동일한 릴리즈에 포함되어야 하면 Mandatory, 우선순위와 출시일정에 따라 동일한 릴리즈로 묶여지지 않아도 된다면 Optional로 한다. 동적 상호의존 관계를 여러 가지로 세분화 할 수 있으나 실제적으로 파악이 쉽지 않고 릴리즈 플랜에 미치는 영향이 적을 것으로 판단되어 하나의 종류로만 정의하였다.
- 정적 연결은 구조적으로 부모-자식과 같은 관계로 동일한 릴리즈에 함께 구현되어야 하는 관계를 나타낸다. 그 결합 정도에 따라 Mandatory와 Optional로 나눌 수 있으며, Mandatory는 동일한 릴리즈에 포함되도록 하지만, Optional의 경우 우선순위와 출시일정에 따라 동

일한 릴리즈에 포함되지 않아도 된다.

- 구현 선후 관계는 구현 과정에 선후 관계로 $f(i)$ 가 구현되기 이전에 $f(j)$ 가 먼저 구현되어야만 하는 관계를 나타내며, 반드시 동일한 릴리즈에 포함되지 않아도 된다.
- 불일치는 휘처간 갈등(Conflicts) 또는 불일치 관계로 휘처가 동시에 존재할 수 없거나, 하나의 요구사항을 만족시키면 다른 요구사항의 만족도는 떨어지는 관계를 나타낸다. 요구사항을 계층 구조화할 때 1차 점검 및 수정 보완될 것이며, 상호의존 관계를 정의하는 단계(PA4)에서 2차적으로 확인되며 발생할 시에는 요구사항 및 휘처에 대한 수정 보완을 해야 한다.

5. 사례 연구 및 평가

3, 4장에서 제시한 우선순위 모델 기반 적응적 우선순위 프로세스의 타당성을 검증하기 위해 핸드폰 소프트웨어 개발의 일부 파트를 대상으로 사례를 적용하며 그에 대한 분석과 기존 기법과 비교 평가를 하였다. 사례연구 대상 핸드폰은 기본적인 기능에 Flash UI와 Game, 500만 화소 카메라를 강점으로 저장 메모리 공간을 확보를 위해 SD카드가 중요한 인자가 되고 있다.

5.1 사례 연구

제안된 프로세스를 핸드폰 소프트웨어 개발 일부 파트의 요구사항을 기초로 하여 상호의존 관계를 정의하는 단계(PA4)를 시작으로 하여 릴리즈 플랜 생성(PA7)까지 사례를 적용하였다. 사례 연구 대상에 대한 요구사항 추출은 이루어져 계층구조화를 통해 요구사항간 불일치 및 일관성이 없는 부분을 제거하였으며 10개의 휘처($f1 \sim f10$)를 식별하였다. 또한, 본 사례의 제품 개발 목표는 품질 경쟁력 확보와 제품 구색을 갖추기 위한 제품이며, 75일 이내에 첫 번째 릴리즈를 출시하여야 한다. 제품 종류는 임베디드 소프트웨어로 하드웨어 요구사항 및 개발 일정이 고려되어야 하나 본 논문에서는 고려하지 않기로 하며, 개발 조직은 이와 같은 프로세스를 통해 릴리즈 플랜을 수립해 본 경험이 없는 상태라고 한다.

5.1.1 상호의존 관계 정의(PA4)

휘처간의 상호 비교를 통해 4.3절에서 정의한 4가지 종류에 대해 식별한다. <표 5>의 결과에 따라서 Camera($f1$)과 SD Card($f2$)는 촬영 후 저장 공간 선택으로 연결됨으로 동적 연결(Dynamic Coupling) Mandatory가 되고, Flash UI($f6$)와 Cricket Game($f9$)은 정적 연결(Static Coupling) Mandatory이기 때문에 동일한 릴리즈에 포함되어야 함을 알 수 있다. 또한, $f10$ 구현 이전에 $f5$ 가 구현되어야 하는 관계도 파악되었다. 즉, $f1, f2, f6, f9$ 가 바인딩되고 $f5$ 는 $f10$ 이전에 구현되어야 한다.

5.1.2 자원 예측(PA5)

<표 6>과 같이 각 휘처에 대해 예상 Size를 LOC(Line

<표 5> 상호의존 관계 정의

| features | | f1 | f2 | f3 | f4 | f5 | f6 | f7 | f8 | f9 | f10 |
|----------|--------------|--------------|--------------|-------------|-------------|-----|--------------|-------------|------|--------------|-----|
| | | SD Card | Camera | Phonebook | PC Sync | SMS | Flash UI | Bluetooth | Zoom | Cricket Game | MMS |
| f1 | SD Card | | DC Mandatory | | | | | | | | |
| f2 | Camera | DC Mandatory | | | DC Optional | | | | | | |
| f3 | Phonebook | | | | | | | DC Optional | | | |
| f4 | PC Sync | | DC Optional | | | | | | | | |
| f5 | SMS | | | | | | | | | | P→ |
| f6 | Flash UI | | | | | | | | | SC Mandatory | |
| f7 | Bluetooth | | | DC Optional | | | | | | | |
| f8 | Zoom | | | | | | | | | | |
| f9 | Cricket Game | | | | | | SC Mandatory | | | | |
| f10 | MMS | | | | | P← | | | | | |

| | |
|--------------|-------------------------------|
| SC Mandatory | Static Coupling Mandatory |
| SC Optional | Static Coupling Optional |
| DC Mandatory | Dynamic Coupling Mandatory |
| DC Optional | Dynamic Coupling Optional |
| P→ | Precedence(f1 → f2 ; f1 선 구현) |
| P← | Precedence(f1 ← f2 ; f2 선 구현) |
| I | Inconsistence |

<표 6> Resource예측

| features | | Resource예측 | | | | | |
|----------|--------------|------------|-------------|----------|---------|--------------|-------|
| | | Size | 생산성(LOC/hr) | Time(hr) | 예상 개발 일 | 기용 자원 (일당인원) | 공수(일) |
| f1 | SD Card | 700 | 20 | 35 | 11.7 | 1 | 11.7 |
| f2 | Camera | 1500 | 20 | 75 | 25.0 | 1 | 25.0 |
| f3 | Phonebook | 600 | 20 | 30 | 10.0 | 1 | 10.0 |
| f4 | PC Sync | 350 | 20 | 17.5 | 5.8 | 0.5 | 11.7 |
| f5 | SMS | 500 | 20 | 25 | 8.3 | 0.5 | 16.7 |
| f6 | Flash UI | 650 | 20 | 32.5 | 10.8 | 1 | 10.8 |
| f7 | Bluetooth | 400 | 20 | 20 | 6.7 | 1 | 6.7 |
| f8 | Zoom | 100 | 20 | 5 | 1.7 | 1 | 1.7 |
| f9 | Cricket Game | 1200 | 20 | 60 | 20.0 | 2 | 10.0 |
| f10 | MMS | 900 | 20 | 45 | 15.0 | 1 | 15.0 |

Of Code)로 예측하였으며, 생산성은 20LOC/Hr로 가정하였다. 전일제(Full Time) 팀원인 경우, 한 주에 16~18Hrs 정도를 개발에 직접 관련된 일을 수행하므로 하루를 3Hr으로 계산하여 필요 일수를 계산하였다[46]. 예를 들어, SD Card의 예상 개발 일수는 35Hr/3 = 11.7일이다. 개발을 위한 총 공수는 119.2일로 예측되었다.

5.1.3 적응적 요구사항 우선순위(PA6)

요구사항 우선순위를 위해 먼저 개발의 목표와 우선순위 모델을 통해 우선순위 방법과 관점을 선택한다. 이로서, 어플리케이션의 특성에 따라 다양한 관점을 고려한 방법과 관점을 선택하고 가중치를 부여함으로써 적응적인 요구사항의 우선 순위화를 가져갈 수 있다.

- 제품 개발의 목표는 품질 경쟁력 확보와 제품 구색을 갖추는 것
- 개발 조직은 이와 같은 프로세스를 통해 릴리즈 플랜을 수립해 본 경험이 거의 없는 상태

- 요구사항의 규모 크지 않음

우선순위 방법(PA61)은 상기와 같은 상황을 고려하여 요구사항 숫자가 많지는 않지만 상대적으로 쉬운 방법을 선택하는 것이 실제 적용하기 용이함으로 Numerical Assignment 방법을 응용하기로 하였다. 관점의 선택(PA62)은 4.2절에서 논의된 2nd 케이스와 동일하며 고객 관점과 개발 관점이 중요함으로 CA에서 고객 중요도와 DA에서 재 사용성 및 기술적 위험 영향도를 선정하였다. 또한, 가중치 부여는 고객 관점이 현 상황에서 가장 중요함으로 50%, 제품 구색이 중요한 시기임으로 재 사용성에 30%를 부여하기로 하였다. 참여할 이해관계자(PA63)로는 선정된 관점에 대응하여 고객 중요도에 대해서는 마케팅 전문가 2명, 재 사용성 및 기술적 위험에 대해서는 개발자 2명과 프로젝트 매니저를 선정하였으며, 관점 별 정의 및 기준 명확화(PA64)를 위해 이해 관계자들이 함께 <표 7>과 같이 협의하여 정의하였다.

2) 출처: PMBOK Guide 2000 Figure 11-2

〈표 7〉 우선순위 관점 및 기준

| Scale | 고객 중요도 | 재 사용성 | 기술적 Risk |
|-------|--|----------------------------|--|
| 5 | Mandatory (고객은 그것이 없으면 안됨) | Very Much (반드시 재사용될 것임) | Very High (전체 일정의 20%이상의 지연 초래 예상) ² |
| 4 | Very Important (고객은 그것이 없는 것을 원하지 않음) | Very (재사용될 것임) | High (전체 일정의 10~20%의 지연 초래 예상) |
| 3 | Rather Important (고객은 그것을 감사할 것임) | Rather (재사용될 수 있음) | Moderate (전체 일정의 5~10%의 지연 초래 예상) |
| 2 | Not Important (고객은 그것을 없음을 인정할 것임) | Less (재사용되지 않을 것임) | Low (전체 일정의 5%미만의 지연 초래 예상) |
| 1 | Does not matter (고객은 그것을 필요로 하지 않음) | Not (재사용되지 않음) | Very Low (미미한 일정 지연을 지연 초래 예상) |

〈표 8〉 우선순위 결과

| 구분 | 고객관점 | | | 개발 관점 | | | | | | | | Priority | Rank |
|--------------|-------|----|--------|-------|----|----|--------|----------|----|----|--------|----------|------|
| | 고객중요도 | | | 재 사용성 | | | | 기술적 Risk | | | | | |
| Weight | 0.5 | | | 0.3 | | | | 0.2 | | | | | |
| Stakeholders | S1 | S2 | Median | D1 | D2 | PM | Median | D1 | D2 | PM | Median | | |
| f1_f2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 1 |
| f3 | 5 | 4 | 4.5 | 5 | 4 | 5 | 5 | 1 | 1 | 1 | 1 | 3.95 | 5 |
| f4 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 1 | 1 | 1 | 1 | 3.2 | 7 |
| f5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 4.4 | 3 |
| f6_f9 | 5 | 5 | 5 | 3 | 4 | 3 | 3 | 5 | 5 | 4 | 5 | 4.4 | 3 |
| f7 | 2 | 3 | 2.5 | 3 | 3 | 2 | 3 | 4 | 5 | 5 | 5 | 3.15 | 8 |
| f8 | 5 | 5 | 5 | 4 | 3 | 4 | 4 | 5 | 4 | 5 | 5 | 4.7 | 2 |
| f10 | 3 | 2 | 2.5 | 5 | 4 | 5 | 5 | 3 | 3 | 4 | 3 | 3.35 | 6 |

- 고객중요도: 각 회차를 고객이 중요하게 생각하는 정도
- 재 사용성: 개발 관점에서 각 회차가 향후 다음 제품 개발에 다시 사용될 정도
- 기술적 위험: 각 회차 개발에서 위험 발생시 개발 일정의 지연에 영향을 주는 영향 정도

〈표 8〉에서와 같이 상호의존 관계를 통해 식별된 요구 회차에 대해 선정된 관점으로 이해관계자들 각각 익명으로 점수를 부여하였으며, 복수 개의 응답에 대한 통계 값으로 중앙값(Median)을 취하였다. 우선순위화(PA65)는 각 회차의 관점별 이해관계자의 중앙값과 가중치를 곱한 후에 더한 값으로 점수의 순위를 구하였다. 그 결과 f1_f2가 1순위가 되었고 f7이 최하 순위가 되었다.

5.1.4 릴리즈 플랜 생성(PA7)

릴리즈 플랜 생성을 위해 요구사항 간의 정적/동적 상호 의존 관계를 식별하여 회차를 바인딩하고, 개발 목표와 요구사항 규모, 조직의 경험 등에 기반한 적응적 요구사항 우선순위화 프로세스에 따라 회차의 순위(Rank)를 결정한다. 또한, 각 회차별로 공수를 예측하여 릴리즈 일정(첫 번째 릴리즈 75일 내 출시)을 고려하게 되면 〈표 9〉과 같이 5가지 경우가 생성될 수 있다. 이 5가지 케이스에 대해 누적 공수와 위험관리 측면 등을 고려하여 선택하도록 한다. 릴리즈

1에 대한 누적 공수를 검토해 보면 플랜 3이 75.8일에 출시할 수 있어 0.8일이 초과되는 경우이며, 그 다음으로 플랜 5가 69.2일로 출시 일 내에 개발할 수 있는 경우이다. 플랜 3과 5의 차이점은 누적 공수 외에 포함되는 회차에서 차이가 난다. 플랜 3에는 우선순위가 3위인 f5가, 플랜 5에는 순위가 4위인 f3가 포함되게 된다. 누적 공수에서 플랜 3은 0.8일 초과되지만 우선순위가 3위인 회차가 포함되고, 플랜 5는 일정에는 문제가 없지만 우선순위가 4위인 회차가 포함되게 된다. 이러한 결과를 통해 합리적 의사 결정을 할 수 있다.

5.2 사례적용의 분석평가

본 프로세스를 적용한 5.1절의 결과를 3가지 측면에서 분석을 하였다. 첫째, 우선순위 방법 선택의 측면이다. 본 사례에서는 참여자 대부분이 우선순위 및 릴리즈 플랜에 대한 경험이 거의 없으므로 〈표 2〉에 의거하여 비교적 쉬운 방법인 Numerical Assignment를 선택하였지만, 이와 같은 것을 고려하지 않고 만약에 AHP에 기반한 비용-가치 다이어그램 기법을 선택했다면 새로운 프로세스 적용에 상당한 저항과 갈등을 유발했을 수 있다. 우선순위 모델에서 \$100 테스트와 랭킹방법이 선택되었을 경우에는 여러 관점을 선택하여 추진하기가 방법의 특성상 어려울 수 있지만 나머지 방법의 경우에는 선택된 관점을 적용할 수 있었을 것이다.

둘째, 관점 선택의 측면에서 분석해 보면, 프로젝트의 목

〈표 9〉 릴리즈 플랜 생성 결과

| feature | Rank | 공수 | Plan 1 | Plan 2 | Plan 3 | Plan 4 | Plan 5 |
|-----------------|------|------|--------|--------|--------|--------|--------|
| f1_f2 | 1 | 36.7 | 1 | 1 | 1 | 1 | 1 |
| f3 | 4 | 10.0 | 2 | 1 | 2 | 1 | 1 |
| f4 | 6 | 11.7 | 2 | 2 | 2 | 2 | 2 |
| f5 | 3 | 16.7 | 2 | 1 | 1 | 2 | 2 |
| f6_f9 | 3 | 20.8 | 1 | 2 | 1 | 2 | 1 |
| f7 | 7 | 6.7 | 2 | 2 | 2 | 2 | 2 |
| f8 | 2 | 1.7 | 1 | 1 | 1 | 1 | 1 |
| f10 | 5 | 15.0 | 2 | 2 | 2 | 2 | 2 |
| Release 1 누적 공수 | | | 59.2 | 65.0 | 75.8 | 48.3 | 69.2 |
| Release 2 누적 공수 | | | 60.0 | 54.2 | 43.3 | 70.8 | 50.0 |

〈표 10〉 기존 연구와의 비교 평가 기준

| 비교 항목 | | 내용 |
|------------------------|-------------------------|---|
| Prioritization | Time Consuming | 우선순위화 시간이 오래 소요되는가? |
| | Ease of use | 배우기 쉽고 사용하기 편리한가? |
| | Informative | 우선순위화 이후에 분석 정보량이 많은가? |
| | Scalability | 요구사항의 규모가 클 때 적용이 용이한가? |
| | Stakeholder Involvement | 개발 제품에 관련된 이해관계자들이 체계적으로 식별되는가? |
| | Fault Tolerance | 우선순위가 잘 못 수행되었는지를 자체 확인할 수 있는 방법이 있는가? |
| Considering dependency | Static dependency | 요구사항 간의 정적 상호의존 관계 파악을 위한 의존 관계 종류가 정의되어 있는가? |
| | Dynamic dependency | 요구사항 간의 동적 상호의존 관계 파악을 위한 의존 관계 종류가 정의되어 있는가? |
| Common | Tool Support | 우선순위와 릴리즈 플랜을 위한 Tool이 지원 되는가? |
| | Softness | 우선순위를 위한 방법과 관점 등의 선택에 유연성이 있는가? |

표를 기반으로 관점을 선택하지 않고 예를 들어, 중요도만을 선택했다면 <표 9>에서 중요도가 5인 위치가 f1_f2, f3, f5, f6_f9, f8로 8개 위치 중에 5개가 된다. 이렇게 된다면, 우선순위의 의미가 매우 축소될 수 밖에 없다. Ad hoc하게 관점을 선택하는 것보다 개발 제품의 목표에 부합하는 관점을 선택하여 요구사항이 우선순위화될 수 있도록 하는 본 기법이 좀 더 합리적이라고 판단된다.

마지막으로, 동적으로 상호의존 관계인 f1(SD Card), f2(Camera)를 파악하지 않았을 경우를 고려해 보자. 적용 사례에서의 방법과 관점을 그대로 사용하였고, <표 8>에서 f1의 기술적 위험 및 재 사용성이 f1_f2를 함께 고려할 때 보다 낮아서 순위가 10개 위치중에 9위가 된다면 릴리즈 플랜에서 f1은 2번째 릴리즈로 계획될 것이다. 동적 상호의존 관계로 함께 구현되어야 하는 f2와 별개 릴리즈로 구현되게 됨에 따라 f2의 구현 완료 후에 테스트시 문제가 생길 수 있으며, 고객에게 출시된 이후에도 f1의 미 구현에 따라 고객에게 불만족을 초래할 수 밖에 없게 될 것이다.

상기에서 분석한 바와 같이 제품 개발 목표와 조직에 따라 유연하게 요구사항 우선순위 방법과 관점을 선택하고 동

적 상호의존 관계를 고려하는 본 논문의 기법 우수성을 확인할 수 있다.

5.3 비교 평가

관련연구에서 언급된 9개의 우선순위 기법과의 비교 평가를 위해 우선순위 모델에서 제안한 10가지 평가 요소 중에서 정확성을 제외하고, 유연성(Softness)에 대한 평가 요소를 추가하였으며, 상호의존 관계 평가에 동적 부분을 반영 총 10개의 기준으로 비교 평가하기로 하고 그 기준을 <표 10>에 기술하였다. 정확성(Accuracy)에 대한 비교 평가 연구는 [41]에서 5개 기법에 대해 실시하였으나 통계적으로 의미 있는 차이가 나타나지 않아 제외하였다.

<표 11>의 우선순위(Prioritization)관련 6가지 항목에 대한 평가는 4.2절의 <표 3>과 동일하며, 그 평가 값은 기존 연구된 방법론에 대해 상호 비교한 연구들([3][4][40-43]) 참고하여 본 논문에서 상대적 평가를 통해 부여하였다. 참고할 연구가 없거나 평가가 어려운 부분에 대해서는 'No Ref.'로 표현하였으며, 본 논문에서 제안한 프로세스에 대한 평가는 5.1절의 적용 사례를 기준으로 평가하였다.

〈표 11〉 기존 연구와의 비교 평가 결과

| Method | EMFEM | IFM | EVOLVE | Cost-Value Diagram | \$100 Test | Wiegner's Method | Numerical Assignments | Planning Game | Ranking | 제안 프로세스 | |
|------------------------|-------------------------|---------|---------|--------------------|------------|------------------|-----------------------|---------------|---------|---------|----|
| Prioritization | Time Consuming | No Ref. | No Ref. | No Ref. | 1 | 3 | 3 | 3 | 5 | 3 | 3 |
| | Ease Of use | No Ref. | No Ref. | No Ref. | 1 | 3 | 3 | 5 | 5 | 3 | 3 |
| | Informative | No Ref. | 3 | 3 | 5 | 5 | 5 | 3 | 3 | 3 | 3 |
| | Scalability | No Ref. | No Ref. | No Ref. | 1 | 3 | 3 | 5 | 5 | 5 | 3 |
| | Stakeholder Involvement | 3 | 5 | 5 | 3 | 5 | 3 | 5 | 5 | 1 | 5 |
| | Fault Tolerance | 1 | 1 | 1 | 5 | 1 | 1 | 1 | 1 | 1 | 1 |
| Considering dependency | Static dependency | 1 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| | Dynamic dependency | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| Common | Tool Support | 3 | 1 | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | Softness | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5 |
| Total Summation | | 10 | 15 | 19 | 20 | 24 | 22 | 26 | 28 | 20 | 30 |

[범례] 5: 매우 우수, 4: 우수, 3: 보통, 2: 미흡, 1: 매우 미흡

〈표 11〉의 평가 결과를 보면, 평가 항목 중 우선순위화 측면에서 제안 프로세스의 점수는 기존 기법들과 달리 제품의 목표와 개발 조직의 경험 등에 따라 방법을 선택 또는 응용할 수 있는 강점이 있으므로 점수가 유동적일 수 있다. 상호의존 관계 측면은 정적/동적 관계를 모두 정의 제안하여 타 기법에 비하여 점수가 높으며, 우선순위 모델 기반 적용적 요구사항 우선순위 프로세스로 유연성 측면에서 가장 우수한 프로세스로 볼 수 있다. 전체적으로 보았을 때에도 총점이 30점으로 Planning Game, Numerical Assignment 등의 기법보다 우수한 것으로 판단된다.

6. 결론 및 향후 과제

요구사항에 대한 우선순위와 릴리즈 플랜은 요구공학에서 특히, 오픈 시장을 고객으로 하는 시장 주도형 제품 개발에 있어서 핵심적 활동이다. 기존 기법은 요구사항 간에 정적 상호의존 관계만을 분석 적용하였고, 제한적인 우선순위 관점을 제안하였으며 우선순위 프로세스가 체계적이지 못하다.

본 논문에서는 요구사항 우선순위 모델을 설계하고 그에 기반하여 우선순위화를 실시하기 전에 우선순위 방법의 특징을 이해하고 조직의 경험과 개발 제품의 목표 등을 고려하여 우선순위 방법, 관점 및 이해관계자 등을 합리적으로 선택하도록 하였다. 특히, 우선순위화 이전에 요구사항 간의 정적/동적 상호의존 관계를 정의하여 릴리즈 플랜의 완성도를 높였다. 본 논문을 통하여 첫째, 개발 제품과 조직에 따라 유연하게 적용할 수 있는 체계화된 요구사항 우선순위 프로세스를 정립하여 충족스러운 우선순위 결과를 도모할 수 있으며, 우선순위 및 릴리즈 플랜 프로세스가 요구 분석 단계에 정착할 수 있다. 둘째, 요구사항 간 동적 상호의존 관계 유형을 정의하여 우선순위화 이전에 휘쳐 바인딩에 의한 릴리즈 플랜의 수립으로 기능의 결함 발생을 방지한다. 셋째로는 요구사항 분석단계에서 패키지 소프트웨어 개발을

위해 반복적으로 사용가능하며, 여러 릴리즈에 의한 제품 개발 및 출시가 아닌 하나의 릴리즈로 제품을 개발하는 조직에서도 내부적으로 릴리즈 플랜을 수립 적용 가능하다.

향후 과제로는 본 연구에서 제안한 프로세스를 실 사례에 적용하여 나타나는 문제점을 보완하고 또한, 임베디드 소프트웨어 제품 개발에 초점을 맞춘 요구사항 우선순위와 릴리즈 플랜을 수행하는 방법과 요구사항 우선순위에 비 기능 요구사항을 고려하는 방법에 대한 연구를 진행하고자 한다.

참고 문헌

- [1] M Khurum, K Aslam, T Gorschek, "A Method for Early Requirements Triage and Selection Utilizing Product Strategies," Software Engineering Conference 2007, APSEC 2007. 14th Asia-Pacific, pp.97-104, 4-7 Dec., 2007.
- [2] Firesmith, D. G., "Prioritizing Requirements," Journal of Object Technology (JOT), 3(8), Swiss Federal Institute of Technology(ETH), Zurich, Switzerland, pp.35-47, September/October, 2004.
- [3] Joachkim Karlsson, Claes Wohlin, Bjorn Regnell, "An Evaluation of methods for prioritizing software requirements," Information and Software Technology, 39(14-15), pp.939-947, 1998.
- [4] Berander, P. and Andrews, A., "Requirements Prioritization," in Engineering and Managing Software Requirements, ed. Aurum, A., and Wohlin, C., Springer Verlag, Berlin, Germany, pp.69-94, 2005.
- [5] Carlshamre P, Sandahl K, Lindvall M, Regnell B, Natt och Dag J., "An industrial survey of requirements interdependencies in software Release Planning," Proceedings of the 5th IEEE international symposium on requirements engineering, pp.84-91, 2001.
- [6] Karlsson, J., and Ryan, K. "A Cost-Value Approach for Prioritizing Requirements," IEEE Software 14(5), pp.67-74, 1997.

- [7] Karl E. Wiegers, *Software Requirements*, second edition, pp.247-258, Microsoft Press, 2003.
- [8] Berander P, "Using students as subjects in requirements prioritization," *Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04)*, IEEE Computer Society, Los Alamitos, pp.167-176, 2004.
- [9] K. Beck, "Extreme Programming Explained: Embrace Change," Addison-Wesley, Reading, Mass., 2000.
- [10] Penny D. A., "An Estimation-Based Management Framework for Enhance Maintenance in Commercial Software Products," *In Proceedings of International Conference on Software Maintenance (ICSM)*, pp.122-130, 2002.
- [11] Denne, M. and Cleland-Huang, J., "The Incremental Funding Method: Data Driven Software Development," 21(3), pp.39-47, 2004.
- [12] Ruhe, G., Saliu, M. O., "The Science and Practice of Software Release Planning," *IEEE Software*, 2005.
- [13] Greer, D., and Ruhe, G., "Software Release Planning: an Evolutionary and Iterative Approach," *Information and Software Technology*, 46(4), pp.243-253, 2004.
- [14] Saliu O, Ruhe G., "Supporting software Release Planning decisions for evolving systems," *Proceedings of 29th IEEE/NASA software engineering workshop*, Greenbelt, MD, USA, 6-7 April, 2005.
- [15] Carlshamre P, Sandahl K, Lindvall M, Regnell B, Natt och Dag J. "An industrial survey of requirements interdependencies in software Release Planning," *Proceedings of the 5th IEEE international symposium on requirements engineering*, pp.84-91, 2001.
- [16] B. Regnell and S. Brinkkemper, "Market-Driven Requirements Engineering for Software Products," *Engineering and Managing Software Requirements*, A. Aurum and C., Wohlin (eds.), Berlin, Germany, Springer Verlag, pp.287-308, 2005.
- [17] Jacobson, I., Booch, G., Rumbaugh, J., *The Unified Software Development Process*, pp.94-98, Addison Wesley, Object Technology Series, 1999.
- [18] G. Ruhe and M.O. Saliu, "The Art and Science of Software Release Planning," *IEEE Software*, 22(6), pp.47-53, 2005.
- [19] IEEE Std 830-1998, *IEEE recommended practice for software requirements specifications*, IEEE Computer Society, Los Alamitos, 1998.
- [20] WeiZhang, Hong Mei, Haiyan Zhao, "A Feature-Oriented Approach to Modeling Requirements Dependencies," *Proceedings of the 2005 13th IEEE International Conference on Requirements Engineering (RE'05)*, 2005.
- [21] W. N. Robinson, "Requirements Interaction Management," *ACM Computing Surveys*, Vol.35, No.2, pp.132-190, June, 2003.
- [22] Y. Lee, C. Yang, C. Zhu, and W. Zhao. "An Approach to Managing Feature Dependencies for Product Releasing in Software Product Line," in *Proceeding of the 9th International Conference on Software Reuse: Reuse of Off-the-Shelf Components (ICSR 2006)*, pp.27-141, Turin, Italy, June, 12-15, 2006.
- [23] A.G. Dahlstedt, A. Persson, "Requirements Interdependencies-Moulding the State of Research Into a Research Agenda," *In Proceedings of Ninth International Workshop on Requirements Engineering: Foundation for Software Quality*, Klagenfurt/Velden, Austria, pp.55-64, June, 2003.
- [24] W. Zhang, H. Mei, and H. Zhao. "Feature-driven requirement dependency analysis and high-level software design," *Requir. Eng.*, 11(3):205-220, 2006.
- [25] A.G. Dahlstedt, A. Persson, "Requirements Interdependencies: State of the Art and Future Challenges," *Engineering and Managing Software Requirements*, Springer Berlin Heidelberg, pp.95-116, 2005.
- [26] Karl E. Wiegers, "Karl Wiegers Describes 10 Requirements Traps to Avoid," *Software Testing & Quality Engineering*, January/February, 2000.
- [27] Dahlstedt, Å. G., Karlsson, L., Persson, A., Natt och Dag, J. and Regnell, B., "Market-Driven Requirements Engineering Processes for Software Products - a Report on Current Practices," Submitted to *Development of Product Software*, DoPS-03, Velden, Austria, 20 and 21 June, 2003.
- [28] 성 재석, 강동수, 송치양, 백두권, "요구사항 우선순위 프로세스 설계", *한국정보과학회 학술발표논문집, 한국정보과학회 2008 종합학술대회 논문집, 제35권 제1호(B)*, pp.13-18, 2008.
- [29] Robertson, S. & Robertson, J. *Mastering the Requirements Process*. 2nd Ed., p.333, Addison-Wesley, 1999.
- [30] C. Turner, A. Fuggetta, L. Lavazza, A. Wolf, "A Conceptual basis for Feature engineering," *J. Syst. Software*. 49 pp.3-15, 1999.
- [31] Dean Leffingwell, Don Widrig, *Managing Software Requirements ; A Use Case Approach*, Second Edition, pp. 10-11, pp.95-100, pp.50, Addison Wesley, 2003.
- [32] K. Kang et al., "Feature-Oriented Domain Analysis(FODA) Feasibility Study," tech. report CMU/SEI-90-TR-21, *Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh*, 1990.
- [33] K.C.Kang, S.J.Kim, J.J.Lee, K.J.Kim, E.Shin, "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference," *Architectures; Annals of Software Engineering*, Vol.5, 1998.
- [34] J. Momoh, G. Ruhe: "Release Planning process improvement - an industrial case study," In: "Software Process: Improvement and Practice," Vol.11, Issue3, pp.295-307, 2006.
- [35] Frank Moisiadis, "The fundamental of Prioritizing Requirements, *System Engineering*," *Test & Evaluation Conference*, 2002.
- [36] 성재석, 강동수, 송치양, 안상선, 백두권, "적응적 요구사항 우선순위 기법", *한국정보처리학회 학술발표논문집, 제30회 한국정보처리학회 추계학술발표대회, 제15권 제2호*, pp.462-465, 2008.
- [37] P. Berander, K. A. Khan, and L. Lehtola, "Towards a Research Framework on Requirements Prioritization," *SERPS 06*, Umeå, Sweden, 2006.
- [38] Karlsson, L., Höst, M., Regnell, B., "Evaluating the practical use of different measurement scales in requirements prioritization," In: *ISESE 2006. Proceedings of the 2006 ACM/IEEE international Symposium on Empirical Software Engineering*, pp.326-335. ACM Press, New York, 2006.

[39] Karlsson, J., Olsson, S. and Ryan, K., "Improved Practical Support for Large-scale Requirements Prioritization," *Requirements Engineering Journal*, 2(1), p.51-60, 1997.

[40] Lehtola, L., and Kauppinen, M. "Empirical Evaluation of Two Requirements Prioritization Methods in Product Development Projects," *Proceedings of the European Software Process Improvement Conference (EuroSPI 2004)*, Trondheim, Norway, pp.161-170, 2004.

[41] Karlsson, J. "Software Requirements Prioritizing," *Proceedings of the 2nd International Conference on Requirements Engineering (ICRE'96)*, Colorado Springs, CO, pp.110-116, 1996.

[42] Karlsson, L., Thelin, T., Regnell, B., Berander, P., and Wohlin, C., "Pair-Wise Comparisons versus Planning Game Partitioning - Experiments on Requirements Prioritisation Techniques," *Journal of Empirical Software Engineering*, Vol 1.11 Nr2, 2006.

[43] Ahl, Viggo. "An Experimental Comparison of Five Prioritization Techniques - Investigating Ease of Use, Accuracy, and Scalability," Master Thesis No. MSE-2005-11, School of Engineering, Blekinge Institute of Technology, 2005.

[44] A. I. Antón, "Goal-Based Requirements Analysis," *Proceedings of ICRE'96*, pp.136-144, 1996.

[45] A. Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," 5th International Symposium on Requirements Engineering, IEEE Computer Society Press, pp.249-261, 2001.

[46] WATTS S. HUMPHREY, "Why Does Software Work Take So Long?," SEI Interactive, June, 1998.



성 재 석

e-mail : damuljss@gmail.com
 1995년 인하대학교 전자공학과(학사)
 1995년~1998년 SKC 근무
 2001년~현 재 LG전자 MC연구소 근무
 2007년~현 재 고려대학교 컴퓨터·전파통신공학과 석사과정

관심분야: 소프트웨어공학, 요구공학, Project Management, Six Sigma



강 동 수

e-mail : 2008010372@korea.ac.kr
 1997년 해군사관학교 전기공학(학사)
 2006년 국방대학교 전산학과(이학석사)
 2008년~현 재 고려대학교 컴퓨터·전파통신공학과 박사과정
 관심분야: 소프트웨어공학, SOA, System Engineering, 요구공학



송 치 양

e-mail : cysong@knu.ac.kr
 1985년 한남대학교 전산학과(학사)
 1987년 중앙대학교 전산학과(이학석사)
 2003년 고려대학교 컴퓨터학과(이학박사)
 1990년~2005년 한국통신 중앙연구소 책임연구원

2005년~2007 상주대학교 소프트웨어공학과 조교수
 2007년~현 재 경북대학교 소프트웨어공학과 조교수
 관심분야: UML 모델링 기술, 소프트웨어 개발방법, IP-TV 서비스



백 두 권

e-mail : baikdk@korea.ac.kr
 1974년 고려대학교 수학과(학사)
 1977년 고려대학교 대학원 산업공학과(공학석사)
 1983년 Wayne State Univ. 전산학과(이학석사)

1985년 Wayne State Univ. 전산학과(이학박사)
 2002년 고려대학교 정보통신대학 학장
 1986년~현 재 고려대학교 컴퓨터·전파통신공학과 교수
 1989년~현 재 한국정보처리학회 부회장
 1992년~현 재 ISO/IEC JTC1/SC32 국내위원회 위원장
 1999년~현 재 정보통신진흥협회 데이터 기술위원회 의장
 2005년~현 재 한국시물레이션학회 고문
 관심분야: 데이터베이스, 소프트웨어공학, 데이터 공학, 컴포넌트 기반 시스템, 메타데이터 레지스트리, 정보 통합, 프로젝트 매니지먼트