

## 요구사항 우선순위 프로세스 설계\*

성재석<sup>○\*</sup>, 강동수\*, 송치양\*\*, 백두권\*

\*고려대학교 컴퓨터 전파통신 공학과, \*\*경북대학교 소프트웨어공학과

{damuljss, greatkoko, baikdk}@korea.ac.kr, \*\* cysong@knu.ac.kr

### A Design of Process for Requirements Prioritization

Jaeseok Seong<sup>○\*</sup>, Dongsu Kang\*, Cheeyang Song\*\*, Dookwon Baik\*

\*Department of Computer Science and Engineering, Korea University,

\*\*Department of Software Engineering, Kyungpook National University

#### 요 약

소프트웨어 시스템 개발에 있어 요구사항 분석 단계는 개발 프로세스의 최 상위 단계로 프로젝트의 성공과 실패를 결정하는 중요한 단계이다. 특히, 반복 점증적 개발 프로세스를 적용하는 경우에 있어 고객 요구사항을 우선순위화하여 제한된 일정과 자원을 활용하여 제품을 개발하는 방법론이 필요하다. 기존 방법과 달리 본 논문에서는 SW QFD를 활용하여 고객관점 우선순위에 개발 관점을 추가하고 하위 수준(Low Level) 요구사항까지 우선순위화 하는 Enhanced SW QFD 방법론을 제안한다. 고 품질의 소프트웨어 제품 개발을 위해 고객 요구사항을 명확하게 이해하고 고객과 개발자 양방향 관점 우선순위 프로세스를 제시한다.

#### 1. 서론

소프트웨어 시스템 개발에 있어서 요구사항 분석 단계는 프로젝트의 성공과 실패, 개발 비용 및 유지보수 등의 측면에서 중요하다. 프로젝트 실패의 25%가 요구분석 오류 및 불충분한 사용자 참여에 기인하고 있으며[1], 요구사항의 누락, 모순, 오해, 또는 잘못된 해석으로 인해 발생한 결함 비율이 고객이 제기한 전체 결함의 60%이다[2]. 또한, 요구사항 단계에서 야기된 결함의 제거 비용은 평가 단계에서 제기된 결함에 비해 약 10~15배의 비용이 소요되고 있다[3].

요구사항 프로세스는 요구사항 관리와 개발 측면으로 나누어 볼 수 있는데 관리적 측면에는 협상, 베이스라인, 변경관리, 추적성 등이 포함되며, 요구사항 개발 측면에는 도출에서부터 시작하여 분석, 명세, 검증의 단계가 상호 보완적 활동으로 진행 된다. 특히, 분석 단계의 주요 활동으로 도출된 요구사항에 대한 중요도 또는 구현 순서에 따라 우선순위화를 하는 활동이 있다. 요구사항을 우선순위화 하는 가장 큰 이유는 제한된 프로젝트 자원으로 일정 안에 소프트웨어 제품을 출시하기 위함이다[4]. 즉, 명확한 우선순위를 갖게 되면 일정과 자원의 문제 또는, 요구 사항의 추가, 변경 등이 발생할 경우 그 우선순위에 따라 요구사항 선별 및 구현 순서를 조정할 수 있기 때문이다.

그런데, 도출된 요구사항을 우선순위화하기 위해서는 적절한 방법론과 우선순위화를 위한 기준이 명확해야 하는데 기존에 요구사항을 우선순위화 하기 위한 방법에는 AHP, QFD, Karlsson's Cost-value Approach, Wiegiers Prioritization Matrix, Value-oriented Prioritization 등이 있다[5][6]. 그러나, 이 방법론들은 주로 상위 수준의 요구사항에 대해 다루고 있으며 또한, 그 관점이 고객 중심으로 치우쳐 있다.

따라서, 본 논문에서는 도출된 요구사항에 대해 고객의 우선순위 관점에 개발자의 관점을 추가하여 함께 고려하는 2-way View기반으로 우선순위를 결정하고자 하며, 상위 수준(High Level) 요구사항에서 분화된 하위 수준(Low Level) 요구사항까지도 우선순위화 할 수 있도록 Enhanced SW QFD(Software Quality Function Deployment) Process 방법론을 제안하고자 한다. 특히, 이 방법론은 기존 SW QFD 방법론을 활용하였으며, 반복 점증적 개발 방법론의 요구사항 분석 단계에 우선순위화 방법론으로 적절하다.

본 논문의 구성은 2장 관련 연구로 일반적인 우선순위 프로세스와 SW QFD에 대해 살펴보면, 3장은 본 논문에서 제안하는 우선순위 프로세스를 세부적으로 기술하며, 4장에서는 적용 사례를, 마지막 5장에서는 결론 및 향후 과제를 기술한다.

\* 이 연구에 참여한 연구자는 '2단계 BK21 사업'의 지원을 받았음.

2. 관련 연구

2.1 우선순위 프로세스

요구사항에 대한 우선순위화 목적은 요구사항의 상대적 중요도 또는 필요성을 결정하여 불필요한 요구사항을 제거하고 또한, 요구사항의 구현 순서를 정하기 위함이다[4]. 일반적인 요구사항에 대한 우선순위화 프로세스는 먼저 이해관계자에게 요구사항 우선순위의 필요성과 중요성을 확신시키며, 둘째, 이해관계자에게 우선순위 프로세스에 대한 교육을 해야 하며, 셋째, 도출된 실제 요구사항을 분류 및 체계화 한다. 넷째 우선순위화 작업을 수행하고 다섯째, 각 요구사항을 구현하는 데 필요한 노력도를 예측하며 여섯째, 앞의 결과를 바탕으로 개발 일정을 수립하며, 마지막으로 요구사항 변화에 따라 우선순위를 유지하여야 한다[4].

2.2 SW QFD(Software Quality Function Deployment)

QFD는 1960년대 일본에서 제조업분야에서 시작되었으며 Software분야에의 활용은 1984년부터 제안되어 SW QFD라는 이름으로 적용되고 있다[7]. 기본적인 SW QFD의 모습은 그림 1과 같으며, 이를 활용하여 요구사항에 대한 우선순위화를 실시할 수 있다.

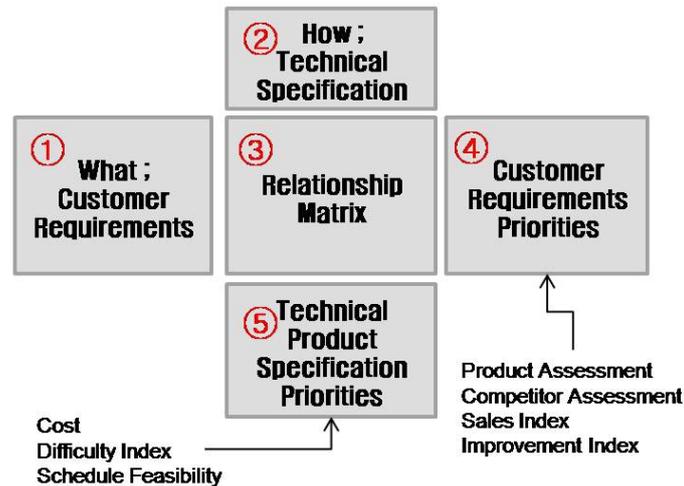


그림 1. Basic Software QFD

요구 분석 단계의 SW QFD 프로세스는 그림 1과 같으며, 첫 번째 단계에서는 고객 또는 이해관계자를 정의하고 고객 요구사항을 정의하며, 두 번째 단계는 요구사항을 기술적으로 측정 가능한 요소로 전환하며, 세 번째, 요구사항과 기술적 제품 명세간의 관계 정도를 표현한다. 네 번째로는 고객조사 data를 기초로 요구사항에 대해 우선순위화 하며, 마지막으로 기술적 명세에 대해 우선순위화를 진행한다[7].

기존 SW QFD를 활용한 우선순위 방법은 개발자 관점의 우선순위를 반영하지 못하며, 상세한 수준으로 전개한 요구사항에 대해 우선순위화하지 못하는 문제점이 있다.

3. 우선순위 프로세스

그림 2는 본 논문에서 제안하는 SW QFD를 활용한 Enhanced SW QFD에 대한 프로세스이다. 3.1절부터 각 영역에 대해 상세하게 기술할 것이다.

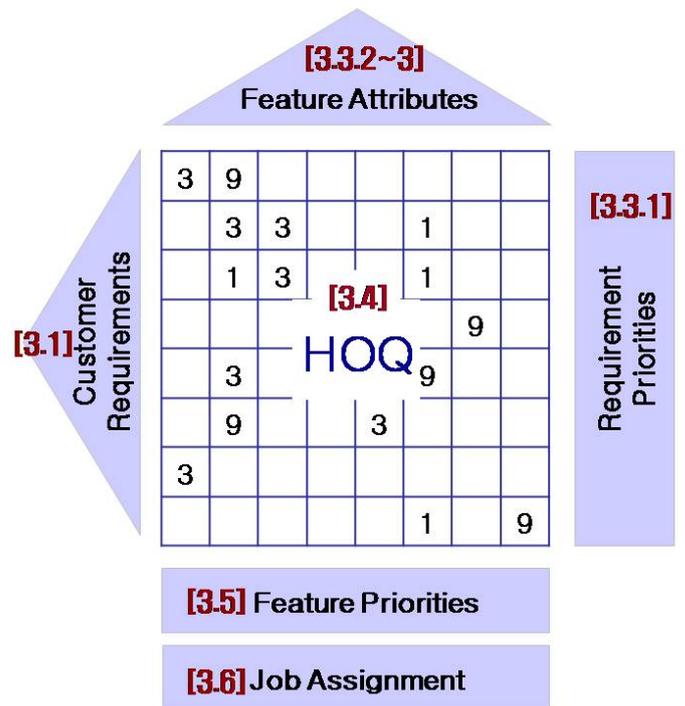


그림 2. Enhanced SW QFD

그림 3은 Enhanced SW QFD를 사용한 요구사항에 대한 우선순위화 프로세스를 프루차트로 표현한 것이다. 먼저, 하위 수준 요구사항에 대해 우선순위화 하기 위해 상위 요구사항을 하부 전개하며, 다음으로, 요구사항에 대한 우선순위를 정하기 위해서 명확한 관점과 우선순위 대상을 정립한다. 고객 관점에서는 편의성, 중요도 등이 될 수 있고, 개발 관점에서는 요구사항의 확정 정도, 난이도, 위험 정도 등이 될 수 있다. 어떤 요인의 관점과 기준으로 우선순위화를 할 것인지는 소프트웨어 개발 상황과 환경에 따라 다양한 인자를 고려하여 선정하여야 한다. 또한, 우선순위 대상 역시 필수 요구사항인지 선택 요구사항을 포함하는지 명확히 한 다음, 정해진 관점과 기준에 따라 우선순위화를 진행한다. 다음으로 일정계획 수립을 위한 사이즈를 예측하고, Feature 특성에 대해 우선순위화를 실시한 후 마지막으로 자원을 할당한다.

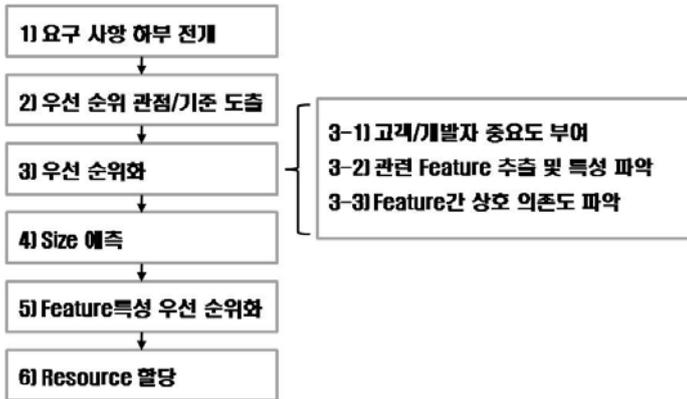


그림 3. 우선순위화 프로세스

3.1 요구사항 하부 전개

요구사항 도출 단계에서 도출된 상위 수준(High Level)요구사항에 대하여 그림 4 와 같이 Affinity Diagram 또는 KJ(Kawakita Jiro)법을 활용하여 대/중/소와 같은 분류 기준에 따라 그룹핑 및 하부 전개한다. 이 과정을 통해 고객 요구 사항이 구체적으로 무엇인지에 대해 개발자 관점에서 명확하게 이해하게 된다.

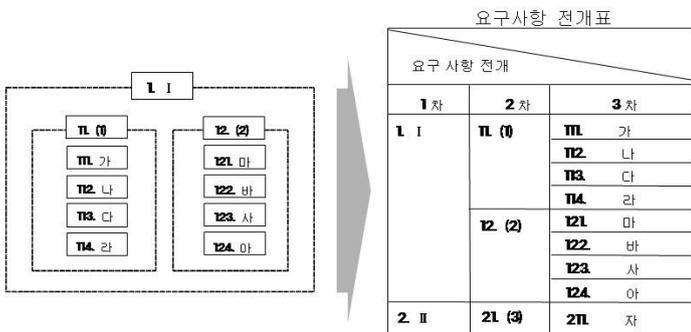


그림 4. 요구사항 전개표

3.2 우선순위 관점/기준 도출

요구사항을 우선순위화 하는데 있어 어떤 관점과 기준으로 수행할 지를 결정하는 것이 필요하다. 우선순위의 관점은 소프트웨어 제품 개발과 관련된 이해 관계자들의 입장에 따라 다양할 것이다. 고객 또는 고객을 대신하는 역할을 하는 조직의 입장에서는 sales point 가 될 수 있는 부분을 중요하게 여길 것이며, 개발의 관점에서는 요구사항의 난이도와 확정된 요구사항인지 향후 변경 가능성이 있는지, 어떤 제약 사항 또는 위험 요소가 있는지 등이 중요할 것이다. 반면에 경영층에서는 비즈니스 가치, 일정 및 비용 등이 중요한 요소가 될 것이다. 따라서 개발 프로젝트의 상황에 따라 적절한 우선순위 관점을 선택해야 한다.

본 논문에서는 상기 언급한 관점 중에서 고객과 개발의 관점에서 일반적으로 사용될 수 있는 관점으로 고객 중요도, 요구사항 확정도, 구현 난이도, 핵심 제약 사항을 선정한다.

3.3 우선순위화

3.3.1 고객/개발자 중요도 부여

요구사항 전개표					고객 관점		개발관점		우선순위	
요구 번호	1차	2차	3차	출처	요구사항 확정도	고객 중요도	구현 난이도	Key Constraints		
					(1)	(2)	(3)	(4)	(5)	(6)

그림 5. 고객 개발자 관점 우선순위표

그림 5 와 같이 하위 수준까지 전개된 요구사항에 대해 고객과 개발자 관점으로 분리하여 점수를 부여한다. 고객 관점으로는 고객 중요도와 [8] 요구사항이 확정된 정도로 점수를 주어 고객 관점의 가중치를 도출한다. 개발 관점에서는 구현 난이도와 핵심 제약 사항(Key Constraints)으로 개발 관점의 가중치 등급을 추출한다.

표 1. 고객 개발자 우선순위 기준 점수

(1)요구사항 확정도	(2)고객 중요도	(4)구현 난이도	(5)핵심제약사항
기본기능: 5 점	Mandatory: 5 점	현기술로 불가능:5 점	치명적:3 점
스펙확정: 4 점	Very Important: 4 점	신규:4 점	보통:2 점
스펙변경가능(소):3 점	Rather Important: 3 점	유사경험:3 점	미비:1 점
스펙변경가능(대):2 점	Not Important: 2 점	기존경험:2 점	
스펙미확정:1 점	Does not matter: 1 점	단순작업:1 점	

표 1 과 같은 기준으로 점수를 부여하며 그림 5 의 (3)고객 관점 가중치는 (1)요구사항 확정도와 (2)고객 중요도의 곱으로 1 점에서 25 점 사이의 요구 중요도를 갖게 되며. (6)개발 관점 가중치 등급은 (4)구현 난이도와 (5)핵심 제약사항(Key Constraints)의 값에 의해 그림 6 의 Block Diagram 으로 표현할 수 있으며 A 부터 E 등급까지 5 단계로 개발 관점 가중치 등급을 부여한다. 구현 난이도는 기존 개발의 경험을 근거로 하여 경험이 없으며 현 기술로 불가능한 경우를 5 점, 단순 작업은 1 점과 같이 점수를 부여하며, 핵심 제약 사항(Key Constraints)은 소프트웨어 개발 프로젝트 전반에 가장 치명적인 기술적 제약 사항으로 예를 들면, 메모리 사용 정도와 같은 것이다.

그림 6 과 같이 각 요구사항은 A 부터 E 등급까지 개발 관점 가중치 등급이 부여되며 각 등급이 의미하는 바에 따라서 다음과 같은 대응 방안이 있을 수 있다.

A 등급에 해당하는 요구사항은 구현 난이도가 매우 높은 경우로 요구사항 변경 협의, 구현 방법에 대한 연구 및 핵심 리스크 관리 대상이 될 것이다. B 등급 요구사항은 개발 경험이 없는 신규 요구사항이다. 이 경우 리스크 관리 대상이 되며, 경험 많은 개발자를 할당하는 것이 필요하다. C 등급 요구사항은 구현 난이도 측면보다는 제약 사항 해결을 위한 방안 연구와 관련 기술에 경험이 있는 개발자를 할당 하는 방안이 중요할 것이다.

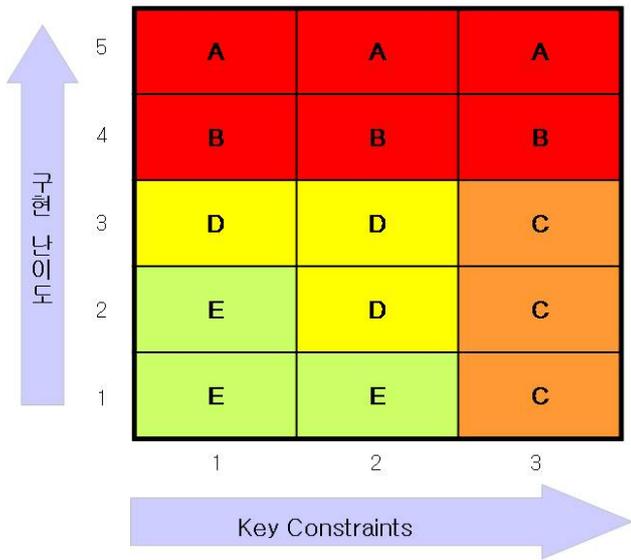


그림 6. 개발 관점 가중치 Block Diagram

기본적으로 요구사항에 대한 우선순위는 고객 관점 가중치 총점을 기준으로 하게 되며, 개발 관점 가중치 등급은 우선순위를 조정하는 역할을 하게 된다. 예를 들어, 고객이 강하게 요구하는 기능인데 개발 관점에서는 개발 난이도 및 핵심 제약 사항에 상당한 영향을 끼치는 A 등급이 있을 수 있다. 이런 경우 자원과 일정을 고려한 인도 계획을 수립할 수 있다. 또 다른 경우는 고객은 중요하게 생각하고 있지 않으나 개발 관점 가중치가 A 등급인 경우이다. 이런 경우 요구사항 삭제를 검토하는 것이 타당할 것이다.

3.3.2 관련 Feature 추출 및 특성 파악

3.1 절에서 전개된 요구사항에 대응된 SW Feature 를 추출하여 그림 7 의 3.3.2 영역에 열거한다. 반복 점증적 개발 프로세스를 적용함으로 기존에 사용되었던 유사 Feature 가 열거될 것이며 새로 필요한 Feature 역시 도출될 것이다.

열거된 Feature 들의 특성을 대변할 수 있는 특성치를 선정하여 각 Feature 별로 측정 및 예측을 한다. 이 특성치는 Feature 의 개선을 이끌 수 있는 인자가 되는

것이 바람직하다. 예를 들어, Feature 의 복잡도 또는 사이즈가 될 수 있다.

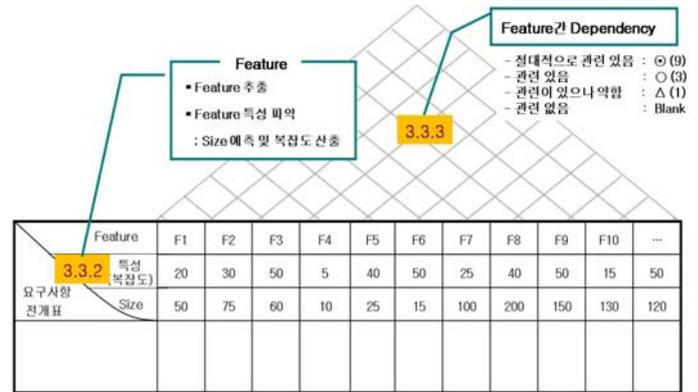


그림 7. 관련 Feature 추출 및 특성, 상호 의존도

이를 통하여 각 Feature 에 대한 분석을 실시하게 되고 3.5 절의 Feature 특성 우선순위화 결과에 따라 개선 대상 Feature 를 선정한다. 또한, 사이즈를 예측함으로 향후 SW 개발 상세 계획을 수립할 때 매우 유용하게 사용될 수 있다.

3.3.3 Feature 간 상호 의존도확인

상기 절에서 도출된 Feature 간 상호 의존도를 확인하는 단계로 이를 통해 필요한 경우 구현 순서를 조정해야 한다. Feature 간에 밀접하게 관련 있는 경우 9 점, 보통 관련이 있는 경우 3 점, 약하게 관련 있는 경우 1 점, 관련 없는 경우 공란으로 하여 상호 의존 관계를 파악한다.

3.4 사이즈 예측

Enhanced SW QFD 의 HOQ(House Of Quality)부분에 해당하며 그림 8 의 3.4 영역이다. 여기서는 전개된 요구사항과 관련된 Feature 들에 대해 개발해야 할 사이즈를 예측하게 된다. 유사 Feature 에 대해서는 정확한 양을 예측하는 것이 어려움으로 수정 정도를 퍼센티지(%)로 예측하여 10%정도의 수정 개발이 필요하다면 '1'로 표현토록 한다. 이런 과정을 통해 개발자들은 각 요구사항 별로 Feature 에 대한 구체적 설계 구상을 하게 되며 요구사항 별 예상 사이즈를 예측할 수 있다.

추가적인 결과로 HOQ 값 즉, 요구사항에 대한 Feature 에 대한 수정 정도가 작다면, 향후 재 사용을 위해 컴포넌트로 개발할 수도 있으며 공통성 Feature 와 가변성 Feature 로 구분할 수도 있다.

3.5 Feature 특성 우선순위화

Feature 의 개선을 이끌어 낼 수 있는 특성치에 대해 그림 8 의 3.5 영역에 순위를 주어 문제가 있는 Feature 를 선정하여 개선을 유도할 수 있도록 한다.

Feature	F1	F2	F3 (New)	F4	F5	F6	F7	F8	F9	F10 (New)	...	예상 생성 Size	등급
특성 (복합도)	20	30	10	5	40	50	25	40	50	20	...	...	A
요구사항 전개표	Size	50	75	60	10	25	15	100	200	150	15	...	B
Req. 1	9		20						8		...	250	3.6
Req. 2		8	30			3.4	7	3	4		...	398	
...											...	...	
Req. n	1		10					1		15	...	194	
Feature 특성(복합도)순위	5	3	5	6		3.5	9	1	7	10	12	...	N

그림 8. 사이즈 예측, Feature 우선순위 및 자원 할당

3.6 자원 할당

3.4 절의 사이즈 예측 결과를 바탕으로 요구사항에 대한 생성 사이즈를 예측할 수 있게 되며, 개발자의 생산성(사이즈/시간)과 요구사항에 대한 개발 관점 가중치 등급을 고려하여 요구사항 별 담당자를 그림 8 의 3.6 영역에 할당한다.

4. 적용 사례

	A	B	C	D	E	F	G	I	J	K	N	O	EW	EX	EY	
1																
2		매크로 Click														
3		Source File List per each Function							Requirement Number					파일 복잡도	파일별 라인수	file priority
4																
5																
6																
7								A1.1.1	A1.2.1	A1.2.2	A2.2.1	A2.3.1	STDIF	STTLN		
8	func	func2	func3	func4	func5	func6	func7									
10	Api	GuiApulib	GuiApulib.h					4						0	41	
11	Api	GuiApulib	StdApulib.h					1						0	16	
12	Api	GuiApulib	Include	AmPulib.h				1						0	63	
52	Api	GuiApulib	Include	CamGui	CamApulib.h			1						0	17	
59	Api	GuiApulib	Source	BaseCtrl.c				2						2	298	
68	Api	GuiApulib	Source	IconList.c				1						3	186	
83	Api	Inc	ColorDef.h					1						0	54	
93	Api	Polulib	Include	AppDef.h				1						0	63	
97	Api	Polulib	Include	AppDef.h				1						0	42	
883	Module	PhoneMenu	AmMenu.c					1						7	618	
900	Module	PhoneMenu	MenuDetail.c					2						3	893	
942	Module	PhoneMenu	MenuBaseCtrl.c					1						0	19	
2744																
2745																
2746					요구사항 확정도(1)			5	5	5	5	5				
2747					고객 요구 정도(2)			4	5	1	5	2				
2748					고객 가중치 총점[(1)*(2)]			20	25	3	25	10				
2749																
2750					구현 난이도			1	3	4	4	3				
2751					메모리 사용정도			1	2	3	3	3				
2752					개발자 가중치 총점			A	B	D	D	C				
2753																
2754																
2755					구현 노력도			2	0.1	0	0.8	2.8				
2756					예상 생성 라인수			362.4	8.9	116.3	180.9	505.2				
2757																
2758																

그림 9. SW QFD Tool

본 장에서는 Enhanced SW QFD 를 활용한 요구사항 우선순위를 산업체에 적용한 결과를 설명한다. Enhanced SW QFD 를 위한 tool 은 Microsoft Office Excel 프로그램을 활용하여 그림 9 와 같은 형태로 제작하였으며, IT 개발팀 30 명에게 본 프로세스에 대해 교육을 시킨 후 적용하였다.

적용 결과 참여했던 개발자의 90%가 Enhanced SW QFD 적용 결과에 만족하였으며, 힘들었지만 요구사항에 대한 체계적인 고민이 가능하였고, 요구 사항과 개발 우선순위를 명확히 할 수 있어 좋았다는 의견이 많았다.

5. 결론 및 향후 과제

본 논문은 요구사항 분석 단계에 요구사항에 대해 합리적으로 우선순위를 정하기 위한 방법으로 Enhanced SW QFD 를 제안하였다.

요구사항을 우선순위화하는데 있어 기존의 고객 관점에서만이 아닌 구현 난이도와 제약 사항과 같은 개발자 관점을 함께 고려하는 2-way View 기반 우선순위를 통해 합리적으로 우선순위를 결정할 수 있게 하였으며, 하위 수준(Low level) 요구사항까지 전개하여 우선순위화함으로써 요구사항 단계에서 가장 중요한 요소인 커뮤니케이션 향상을 가능하게 하였다.

적용 사례를 통해 제시한 프로세스의 효용성을 확인할 수 있었다.

향후 연구는 우선순위 인자 선정의 합리적 프로세스 도출과 비 기능 요구사항에 대한 우선순위화 하는 방법 및 요구사항 변동에 대한 따른 우선순위 유지 방안에 대한 연구를 진행하겠다.

참고 문헌

- [1] Standish Group International. The Chaos Report, 1995.
- [2] Boehm B.W., The High Cost of Software, in Horowitz E., Practical Strategies For Developing Large Software Systems, Addison Wesley, 1975.
- [3] Watts S. Humphrey, Terry R. Snyder, Ronald R. Willis, Software Process Improvement at Hughes Aircraft, IEEE Software Volume 8 , pp 11 – 23 , 1991.
- [4] Donald Firesmith, Prioritizing Requirement, Journal of Object Technology, vol 3, no. 8, pp35~47, 2004.
- [5] Frank Moisiadis, The fundamental of Prioritizing Requirements, System Engineering, Test & Evaluation Conference, 2002.
- [6] Jim Azar, Randy K. Smith, David Cordes, Value-Oriented Prioritization: A Framework for Providing Visibility and Decision Support in the Requirements Engineering Process, Technical Report Department of Computer Science University of ALABAMA, 2007.
- [7] Stephen Haag, M.K. Raja, and L.L Schkade, Quality Function Deployment Usage in Software Development, Communication of the ACM, vol. 39, 1996.
- [8] Joachim Karlsson, Software Requirement Prioritizing, Proceedings of the second International Conference on Requirements Engineering(ICRE '96), 110-116, 1996.