

요구명세의 테스트 가능성 검토와 측정 방법

서 광 익⁰ · 최 은 만
동국대학교 컴퓨터멀티미디어공학과
{bradseo⁰, emchoi}@dgu.ac.kr

A Method of Testability Review and Measurement of Requirements Specification

Kwang Ik Seo⁰ · Eun Man Choi
Department of Computer Multimedia Engineering, Dongguk University

요 약

이 논문에서는 소프트웨어 프로젝트 수행 중에 비용이나 일정 측면에서 비중이 큰 테스트 작업의 위험도를 줄이기 위하여 테스트 가능성 검토 방법과 이를 객관적으로 측정하는 방법을 제시하였다. 막대한 예산이 소요되는 대규모 소프트웨어 프로젝트에서 정확한 ROI(Return of Investment)를 예측하고 테스트 작업에 걸림돌이 될만한 요소들을 파악한다면 소프트웨어 개발 작업이 더욱 효율적인 엔지니어링 작업이 될 것이다. 이 연구에서는 블랙박스 테스트의 기준이 되는 산출물들 중에 먼저 요구명세서의 테스트 가능성에 대한 검토 방법을 제시하였다. 사례연구를 통하여 요구명세서의 어떤 요소들이 테스트가능성을 높이는지 파악하였고 이를 항목으로 만들어 객관적인 측정이 가능하도록 하였다. 연구 결과는 테스트 작업에서 케이스 설계만이 아니라 테스트 실행, 결과의 분석, 결함 위치 발견 및 수정 작업까지도 비용을 줄여주는 효과를 보이고 있다.

1. 서 론

소프트웨어 테스트 작업은 비용이 많이 드는 작업으로 알려져 있다. 전체 예산의 40 내지 50 퍼센트를 차지한다. 테스트 케이스를 설계하고 테스트 환경을 구축하며 테스트를 수행하고 결함의 위치를 파악하여 수정하기 까지는 많은 인력과 장비 그리고 시간이 필요하다.

테스트 비용을 줄이려는 노력은 다양하다. 테스트 케이스를 줄이면서도 결함을 잘 발견할 수 있도록 테스트 기준을 향상시키는 방법이 있다[1]. 또한 테스트 케이스를 만드는 과정을 자동화 하든지 테스트 결과를 무인으로 판단할 수 있도록 테스트 오러클을 자동화하는 방안도 있다[2]. 여러 가지 방안이 소개되고 있으나 테스트 비용을 획기적으로 줄이는 보편적인 방법으로 실용되고 있지는 않다.

테스트 작업의 효율성을 확보하는 또 다른 방법은 테스트 가능성을 꼼꼼히 점검해 보는 것이다. 소프트웨어는 그 시스템이 가지는 고유 특성에 따라 테스트 작업을 용이하게 할 수도 있고 어렵게 만들어 비용과 자원이 예산을 초과하여 프로젝트의 실패를 초래할 수도 있다. 이를 테스트 가능성(testability)이라고 하며 테스트 작업이 효과적으로 수행되기 위하여 미리 체크하여야 할 중요한 요소(factor)다.

이제까지 테스트 가능성이라는 주제의 연구는 주로 테스트 가능성의 정의나 테스트 용이하게 하는 특성(observability, controllerability)을 찾아내는 것[3], 테스트를 용이하게 하기 위하여 설계하는 기법을 제안한 것[4], 설계서나 프로그램에서 테스트 가능성을 따져보는 방법을 제시한 것[5] 뿐이다. 실제 대

규모 소프트웨어의 테스트 작업을 결정짓는 것은 블랙박스 시험에 중요한 기준이 되는 요구명세서이다. [5]에서 제시한 설계나 프로그래밍 단계에서 테스트 가능성을 결정짓는 요소들은 소프트웨어 시스템 자체가 가지는 고유한 성질을 나타낼 뿐이며 테스트 작업에 소요되는 자원 변동에 크게 영향을 주지는 않는다. 그러나 요구분석 명세가 테스트 작업을 위하여 구체적인 인스턴스가 제시된 사용사례(use-case)가 포함되어 있다면 테스트 케이스 작성에는 큰 도움이 된다.

따라서 이 논문에서는 기능 시험 작업의 적정성을 예측하기 위하여 요구분석 명세를 근거로 한 테스트 가능성을 검토하는 방법을 제시한다. 또한 요구명세서에서 어떤 요소가 테스트 가능성을 높여주는 것인지 파악하여 객관적으로 측정할 수 있는 틀을 제시한다.

2. 테스트 가능성

IEEE에서 발행한 소프트웨어 관련 용어집[6]에는 테스트 가능성(testability)을 “시스템이나 컴포넌트가 테스트 기준을 확립하고 테스트를 실행하기가 쉬운지를 나타낸 정도”라고 정의하고 있다. 소프트웨어가 쉽게 테스트 가능한가라는 특성은 원시코드로만 결정되는 특성은 아니다. 소프트웨어의 설계 구조에 의하여 좌우될 수도 있으며 요구분석 명세도 잘 작성된다면 테스트 가능성을 높일 수 있다. 따라서 테스트 가능성을 더 정확히 정의한다면 ‘소프트웨어 산출물들이 테스트를 용이하게 하는 정도’라고 할 수 있다.

테스트가능성은 프로그램을 테스트하는데 예상 비용을 예측하기 위해 필요한 프로그램의 속성 중 하나이다. 테스트가능

성을 측정한다는 것은 곧 테스트 할 프로그램의 속성들을 측정하는 것이다. 테스트가능성이 높다는 것은 테스트를 수행하는 동안 결함들이 쉽게 발견할 수 있음을 의미한다. 반면, 낮다는 것은 테스트하기 어렵고 테스트를 마친 후에도 신뢰성이 결여된다는 것을 의미한다. 따라서 프로그램의 필요한 테스트 비용은 테스트가능성 측정으로 파악할 수 있다[1].

[표 1] Collard가 제시한 테스트가능성 요소

가독성 (Readable)	테스트 엔지니어와 같은 다른 사용자가 판독할 수 있어야 한다.
이해성 (Understandable)	기록된 중요한 문맥, 의도 및 의미들을 쉽게 이해할 수 있어야 한다.
정확성 (Correct)	소프트웨어에 구현할 요구 사항이 정확하게 기술되어야 한다.
완전성 (Complete)	구현할 요구사항이 빠짐없이 모두 기술되어 있어야 한다.
수정가능성 (Up-to-date)	명세서에 대해 수정이 가능해야 한다.
안정성 (Stable)	변화가 적어야 한다.
문서화 (Documented)	이해를 돕기 위해 그래픽과 같은 모델의 문서화가 필요하다.

3. 요구분석 명세의 테스트 가능성

요구분석 명세서에는 개발할 소프트웨어의 기능적 요구와 기능 외적 요구를 기술한다. IEEE Std. 830이나 TTA에서 모델로 제시한 요구분석 명세서에는 명세서를 직접 생성하는데 사용되거나 좀더 구체적인 표준을 위한 모델로 이용될 수 있다 [7]. 하지만 테스트 가능성을 높이는 것에 대한 언급은 없다.

그러면 테스트가 쉽도록 작성된 요구명세서와 그렇지 않은 명세서를 비교해 보자.

[표 2] 테스트가 용이한 요구명세서의 예[8]

Scenario name	RegisterACustomer	
Participating actor instances	john, judy: Customer	
Flow of events	1. User runs the program. 2. If there already exist registered customers, first record will be shown in windows. 3. Press the 'add' button in registry window. 4. Erase all input fields for another customer registration. 5. Enter the input fields for another customer	2. Number : 1 Name : john Home phone : 279-1245 Home address: 1500 S. Wabash Chigaco, IL60616 Company : ABC electronics Office phone: 323-9824 Office address:2010 31st. Chicago, IL60612 Mobile phone: 999-2351 5. Number : 2 Name : judy Home phone :333-2475 Home address: 1500 S. Michigan Chigaco, IL60615 Company: Home Insurance Office phone: 256-4964 Office address:20 W. 50st. Chicago, IL60544 Mobile phone: 560-7156

	6. Press the 'move' button for retrieving certain customer.
--	---

위의 요구 명세서는 시스템에서 고객 등록 절차와 예외 사항에 대해 구현되어야 할 기능에 대해 구체적으로 보이고 있다. 이와 같이 구체적인 인스턴스를 갖는 시나리오가 요구명세서에 포함되어 있다면 테스트 케이스 작성을 용이하게 할 것이다.

4. 테스트 가능성 검토와 측정

[표 1]에서 제시한 테스트가능성의 요소를 메트릭에 적용하기에는 측정하려는 내용과 범위가 모호하기 때문에 메트릭의 요소로 다루기에는 다소 문제점이 있다. 따라서 정량적 측정을 위해서는 메트릭 가능한 요소를 추출하여야 한다.

측정 가능한 요소로는 요구 사항이 산출물로 문서화 되어 있고 소프트웨어를 구현 한 후 산출물이 요구명세서의 기술된 내용과 동일한지 측정하기 위한 정확한 목록이 있어야 한다.

다양한 객체지향 프로세스들이 동의하는 객체지향 분석 과정에서는 기능 모델을 표현하기 위해 [표 3]에서 나타내듯이 사용 사례를 이용한다.

[표 3] 작업 단계와 UML 다이어그램의 관계

모델 종류	다이어그램	분석	설계	구현
기능모델	사용 사례	○		
정적모델	객체	○	○	
	클래스	○	○	○
	컴포넌트		○	○
동적모델	배치			○
	인터랙션	○	○	
	액티비티	○	○	
	상태	○	○	

따라서 기능 중심 요구분석을 기술하기 위해 작성하는 사용 사례에 대한 메트릭 가능 요소를 추출하였다.

[표 4] 기능적 메트릭 요소

내용	세부 내용	설명
사용사례	사용사례	사용사례 작성 여부
인스턴스	시나리오	구체적인 시나리오
	시나리오 순차도	시나리오의 시간 흐름에 따라 진행되는 메시지 전달 과정[6]
완전성	각 세부기능도	세부적인 기능에 대한 기능도
	시스템 기능도	전체적인 시스템 레벨의 기능을 표현하는 사용사례와 외부 사용사례와의 연관성 표현
예외처리	각 세부기능도	완전성에 나타난 세부기능도의 예외 사항을 사용사례로 표현
	시스템 기능도	완전성에 나타난 시스템 기능도의 예외 사항 표현

하지만 요구명세서에의 비기능적 요소 또한 테스트할 수 있는 요소들을 추출하여 테스트가능성을 높일 수 있다. 비기능적 요소로는 성능요구, 하드웨어 요구, 사용자 인터페이스와 관련

된 요구사항들을 점검한다.

[표 5] 비기능적 메트릭 가능 요소

내용	세부 내용	설명
성능 요구	반응 시간	정확성 및 신뢰성에 대해 기술
	처리 요구 시간	처리 시간의 한계 기술
	처리율	예상 자료, 사용자 규모 기술
H/W 요구	시스템 규모	구동하기 위한 제약사항 기술
	통신 규모	통신에 대한 제약사항 기술
사용자 인터페이스	인터페이스	사용자 인터페이스 작성 여부
	데이터 타입	입력될 자료에 대한 정의
	예외 사항 정의	입력 후 발생 가능한 예외사항 정의
	사용자 권한	인터페이스에 접근 가능한 사용자 정의

또한 요구명세서는 시스템의 인수를 위한 테스트 기준을 제공하여야 한다[8]. 시스템 개발을 의뢰할 때 시스템에 대해 기대 수준이 있기 때문에 원하는 기능, 특성, 품질 등에 대해 정확하고 정량적인 설명이 필요하다.

[표 6] 인수 조건 메트릭

내용	세부 내용	설명
기능시험	인수 기준	인수를 위한 정량적 기준 정의
	세부 사용사례 기능	인수 후 세부적인 사용사례에 대한 예상 결과 정의
	시스템 기능	인수 후 전체 기능에 대한 예상 결과물 정의
성능시험	반응 시간	인수 후 예상 반응 시간 정의
	처리 소요 시간	인수 후 예상 처리 소요 시간 정의
	처리율	인수 후 예상 처리율 정의

위에서 제시한 메트릭의 요소들을 중심으로 요구명세서를 검토해 보기로 한다.

본 논문에서 케이스 스터디로 사용할 요구명세서의 자료는 무작위로 인터넷에서 검색하였으며 개인 정보의 보호를 위해 요구명세서의 제목만 밝히고 출처는 생략하기로 한다.

- 사용된 요구명세서 목록

- ① 웹기반 사이트 개발 문서: 텍스트 위주의 기능 설명만 되어 있다.
- ② 유치원 관리 프로그램: 기능은 텍스트 위주의 설명이다.
- ③ 밴딩 머신 시스템: 사용 사례가 있지만 정보가 불충분하다.

[표 7] 케이스 스터디

내용	세부 내용	사례 ①	사례 ②	사례 ③
사용사례	사용사례	X	X	○
인스턴스	시나리오	X	X	X
	시나리오순차도	X	X	X
완전성	각 세부 기능도	X	○	X
	시스템 기능도	X	○	X
예외처리	각 세부 기능도	X	X	X
	시스템 기능도	X	X	X
성능 요구	반응 시간	X	○	○
	처리 요구 시간	X	○	○

H/W 요구	처리율	X	○	X
	시스템 규모	X	○	○
	통신 규모	X	X	X
사용자 인터페이스	인터페이스	X	○	○
	데이터 타입	X	X	○
	예외 사항 정의	X	X	X
	사용자 권한	X	X	X
기능 시험	인수 기준	X	○	○
	세부 사용사례 기준	X	○	X
	시스템 기능	X	X	X
성능 시험	반응 시간	X	X	X
	처리 소요 시간	X	X	X
	처리율	X	X	X
접근범위(좌용개수/전체개수)		0%	41%	32%

위의 표에서처럼 각 메트릭 요소에 해당 사항을 점검한 후 전체 접근범위를 측정하여 테스트에 소요될 비용들을 예측해 볼 수 있다. 적용 결과 요구명세서의 작성 내용에 따라 테스트 가능성, 즉 테스트에 필요한 자원과 노력을 예상 할 수 있다.

5. 결론 및 향후 과제

객체 지향 기법으로 시스템을 구현할 때 테스트의 효율성을 높이기 위한 테스트가능성을 측정하는 기준으로 요구명세서로부터 여러 항목들을 추출해 보았다. 이러한 항목들은 프로그램을 기능 위주로 테스트할 때 필요한 필수 요소이며 특히 요구명세서에 반드시 목차로 들어가야 할 요소들이다. 그리고 제안한 메트릭 요소를 중심으로 명세서를 구성한다면 요구명세서 자체가 테스트케이스 설계로 이어질 수 있다.

향후 구체적인 각각의 메트릭 요소와 테스트가능성에 대한 요구명세서의 완벽성을 구하기 위한 연구가 필요하겠다.

참고문헌

- [1] Pu-Lin Yeh, "Software Testability Measurements Derived from Data Flow Analysis", *Proceedings of the Second Euromicro Conference*, 1998.
- [2] D. Graham, B. Matrick, *Software Test Automation: Effective Use of Test Execution Tools*, ACM Press, 1999.
- [3] Roy S. Freedman, "Testability of Software Components", *IEEE Trans. of Software Engineering*, Vol. 17., No. 6, June 1991. pp.553-564.
- [4] R. Binder, "Design for Testability in Object-Oriented Systems", *Communications of the ACM*, Sep. 1994, pp. 87-101.
- [5] S. Jungmayr, "Reviewing Software Artifacts for Testability", <http://www.informatik.fernuni-hagen.de>
- [6] IEEE Std 610. 12-1990, IEEE Standard Glossary of Software Engineering Terminology, 1990.
- [7] IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications, 1998.
- [8] E, Choi, "Use-Case Driven Test for Object-Oriented System", *Proceedings of the IASTED International Conference*, ACTA Press, 2001.