

# 소프트웨어 결함 추이분석을 통한 테스트 프로젝트 개선방안에 대한 연구

장진욱\*

<sup>1</sup>건국대학교 인터넷미디어공학부

## Study of the Improvement Measurement of Test project through Software Defect trend analysis

Jin-Wook Jang<sup>1\*</sup>

<sup>1</sup>Division of Internet & Multimedia Engineering, Konkuk University

**요약** 소프트웨어 결함에 대한 관심이 높아지고 있다. 기업은 소프트웨어를 개발하는 조직, 고객대응 조직, 품질관리 조직 등 다양한 방법과 채널을 이용하여 개선을 위한 노력을 하고 있다. 이렇게 수집된 결함 데이터 들을 그 조건과 용도에 맞게 분석하여 소프트웨어 결함을 최소화하고 나아가 제품품질 향상에 기여할 수 있다. 본 연구에서 기업용 소프트웨어의 결함 추이를 테스트 성숙도 모델을 기반으로 분석하여 결함 예방 프로세스를 수립하고 프로젝트에 적용하여 106개의 결함이 16개로 감소하였다.

본 연구는 소프트웨어 품질활동 중 기본이 되는 기존결함에 관심을 가짐으로서 최소한의 자원을 활용하여 소프트웨어 제품 품질을 개선할 수 있는 방안을 제시하는데 의의를 가진다.

**Abstract** The interest in software defects is growing. Companies are working on reducing them by various ways with a software development part, customer service part and quality management part. The defect data collected from them will be analyzed according to the condition and purpose to minimize the defect, and it will contribute to improving the quality of products. This study analyzed the software defect progress for companies based on the test maturity model, and set up the defect preventing process. They were applied to the project, and the number of defects decreased from 106 to 16.

This study focused on the existing defect, which is a basic requirement for software quality management, and have importance by presenting a way to improve the software product quality with minimum resources.

**Key Words** : Software Test project, Product quality, Defect trend, Test improvement, Defect Prevention Process

### 1. 서론

소프트웨어 테스트 프로젝트에서 결함에 대한 관심이 높아지고 있다. 기업은 소프트웨어를 개발하는 조직, 고객대응 조직, 품질관리 조직 등을 통하여 다양한 방법과 채널을 이용하여 개선을 위한 노력을 하고 있다. 이때 가장 기준이 되는 데이터는 개발과정에서 발견된 결함과 고객들이 사용 중에 발견된 결함 데이터이다[1].

대부분의 조직에서 이러한 데이터를 수집하여 결함을

수정하는데 사용되고 있다. 그러나 결함 데이터는 축적되고 관리되어 소프트웨어 결함을 줄이고 나아가 품질향상을 위한 프로세스 개선에 활용되지 못하고 단순히 발견된 결함을 재현정도로 활용되고 있다.

본 연구에서 소프트웨어의 결함 추이를 분석하였다. 테스트 성숙도 모델 테스트 활동을 기반으로 소프트웨어 테스트 프로세스를 수립하고 실행하는 방안에 대하여 연구하였다.

\*Corresponding Author : Jin-Wook Jang(Konkuk Univ.)

Tel: +82-2-455-4071 email: [jwjang@konkuk.ac.kr](mailto:jwjang@konkuk.ac.kr)

Received August 25, 2014

Revised (1st October 20, 2014, 2nd November 10, 2014)

Accepted January 8, 2015

소프트웨어 품질활동 중 기본이 되는 기존결함의 최소화하여 최소한의 노력으로 소프트웨어 제품품질을 개선할 수 있는 결함예방 프로세스를 제시하고 프로젝트에 적용하여 실행방안을 제시하였다는 점에 의의를 가진다.

## 2. 관련연구

### 2.1 소프트웨어 신뢰도 및 예방 프로세스

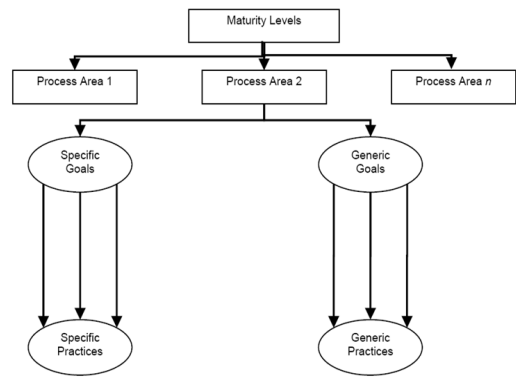
소프트웨어 신뢰도는 실제 운영 환경에서 소프트웨어가 일정 기간 동안 실패하지 않고 안정적으로 작동할 확률을 의미한다. 그래서 소프트웨어 신뢰도 측정은 시스템 인도 이후 결함을 측정함으로써 판단이 가능하다. 일반적으로 소프트웨어 신뢰도 측정의 대상은 시스템의 최종 인도이후 소프트웨어 개발 생산성과 결함발견의 정도로 판단할 수 있다[2].

시스템의 최종 인도 이후 발생하는 결함 정보의 수집 및 분석은 현실적으로 매우 어려운 상황으로 다음과 같은 활동이 필요하다. 첫째, 소프트웨어 신뢰도 예측, 둘째, 소프트웨어 개발 생명주기 결함 예측, 셋째, 전체 프로젝트 수행단계 중 요구사항 분석단계에서 테스트단계까지 개발 단계별 결함예측 및 결과 분석, 넷째, 프로젝트 수행 시 필연적으로 발생하는 결함 예측, 마지막으로 프로젝트 초기에 보다 정확히 예측하기 위한 모델 등이 필요하다. 본 연구에서는 이러한 점을 고려하여 결함예방 프로세스를 통하여 결함정보 수집, 보고, 분석을 통한 예방 프로세스를 확립하였다.

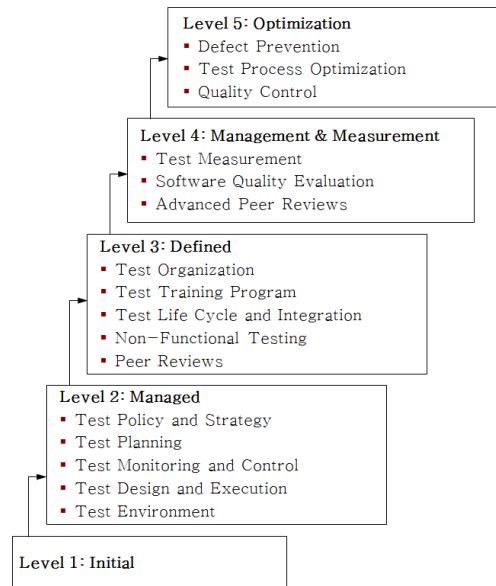
### 2.2 TMMi(Test Maturity Model integration)

TMMi는 일리노이공대 교수팀에 의해 최초 개발되었으며 소프트웨어 개발활동의 전 범위를 다루고 있는 CMMI(Capability Maturity Model integration)과 호환될 수 있도록 설계된 테스트 프로세스 모델이다. 이 모델은 테스트 성공 사례를 기반으로 개발되었으며 레벨에 따라 [Fig. 1]과 같이 성숙도의 목표와 하부활동으로 프로세스 영역이 존재하다[3].

테스트 프로세스 모델인 TMMi는 5레벨로 구성되어 있다. 효과적인 테스트 프로세스 핵심요소를 기술하는 5레벨은 단계적 평가 프레임워크로 테스트 프로세스 개선을 목적으로 사용된다. TMMi는 [Fig. 2]와 같으며 특히, 본 연구에서는 레벨 4, 5의 활동을 기준으로 테스트 결함데이터를 수집하여 개선안을 도출하였다.



[Fig. 1] TMMi Framework



[Fig. 2] TMMi Model

#### 1) 레벨 1 : 초기단계(initial)

레벨 1은 테스트는 정의되지 않거나 테스트와 디버깅의 한 부분으로 인식되고 조직은 일반적으로 프로세스를 지원하기 위한 안정적인 환경을 제공하지 못하는 수준이다. 테스트는 구현이 완료되고 비정형화된 형태로 이루어지며 테스트와 디버깅은 시스템에서 버그를 제거하기 위하여 수행된다.

#### 2) 레벨 2 : 테스트 수명주기 관리(managed)

레벨 2는 테스트와 디버깅이 구분되며 테스트가 소프트웨어 생명주기에서 하나의 독립된 단계로 정의되고, 결함 발견 활동에 집중한다. 상대적으로 중요하고, 의미

있는 레벨인데, 이를 달성하려면 테스트 정책을 별도의 문서로 두거나 품질정책이나 개발정책의 일부에 정의 하고 있어야 한다. 또한 테스트 목표를 정확히 하면서 이를 반영하여 테스트 전략 또는 접근법에 근거하여 테스트하고 있다는 것을 보여줘야 한다. 테스트 계획의 구체성을 높게 수립하고 이에 따라 테스트가 진행되어야 하며 테스트 규모 산정도 현실적으로 설득력 있게 작성 되어야 한다.

**3) 레벨 3: 소프트웨어 수명주기정의(defined)**

레벨 3은 테스트가 개발 생명 주기와 통합되는 단계로, 레벨 2에서 테스트를 포함하고 개발되었는지 검증하는 테스트 활동을 수행해야 한다. 이 레벨을 달성하기 위해서는 테스트 프로세스가 소프트웨어 개발 수명주기에 통합되어 있어야 한다. 또 별도의 테스트 조직을 갖추고 있어야 하며, 그렇지 않는 경우 테스트 조직을 독립적으로 유지하지 않는 명확하고 타당한 이유를 명시하고 있어야 한다. 레벨 2가 프로젝트 레벨에서 내재화하는 수준이라면 레벨 3은 조직차원에서 테스트를 내재화하는 수준이다.

**4) 레벨 4 : 운영과 측정(management and measurement)**

레벨 4 달성을 위해서는 ‘발전된 동료검토’ 활동이 수행되고 있어야 하는데, 이는 테스트 시각에서 볼 때 테스트 케이스를 요구사항 분석 단계부터 설계하고 작성하는 것을 통해 개발 중간 산출물의 결함을 조기에 발견하는 예방적인 테스트를 의미한다. 테스트를 관리하고 측정하는 단계로 소프트웨어 품질 평가와 매트릭스를 이용한 테스트 측정을 통해 테스트를 수치화하고 이를 기반으로 정량적으로 테스트를 관리하고 있어야 달성이 가능하다. 개발 제품뿐만 아니라 테스트 관련 문서 등도 검토의 대상이며 검토 활동을 측정의 필수요소로 인식한다.

**5) 레벨 5 : 최적화(optimization)**

결함예방과 품질제어 활동에 초점, 테스트 프로세스가 정의되고 관리되며, 비용과 효과가 추적되고 감시된다. 테스트 프로세스가 지속적으로 개선되고 조정되며 결함 예방과 품질제어 활동이 수행된다. 이들 활동들이 통계적 방법과 다양한 평가 기준에 의해 측정되고, 관리자는 지속적인 개선을 유도하기 위해 기반을 지원하고 동기를

부여한다. 이와 같이 TMMi은 단계별 테스트활동에 대해 가이드 하는 모델로서 높은 일관성과 완성도 높은 모델 구조를 가지고 있다. 테스트 성숙도의 가장 기본이 되는 레벨 2는 테스트 프로세스를 구축하는 최초의 단계로서 [Table 1]과 같이 5개의 프로세스와 18개의 세부 프로세스로 정의할 수 있다. 단, TMMi에서 제시하는 프로세스는 모델로서 그 의미를 가지며 실행방안을 적용할 때는 조직의 요구사항과 필요성을 고려하여 단계적 및 선택적으로 적용하여야 한다. 각 단계별 세부 내용은 다음과 같다.

[Table 1] Basic Process Area

Process Area	Specific Process	
Test Policies & Strategies	1	Establish testing policy
	2	Develop test strategies
	3	Establishing test metrics
Test Planning	4	Risk Analysis
	5	Test access
	6	Test Estimation
	7	Test plan
	8	Test planning consent
Test Monitoring and Control	9	Test progress monitoring
	10	Planning and preparation expected product quality monitoring
	11	Test Correction and finishing
	12	Test analysis and design using test design techniques
Test design and execution	13	Test implementation runs
	14	Test execution
	15	Test incident management
Test environment	16	Development test environment requirements
	17	Implement a test environment running
	18	Management and control of the test environment

**3. 테스트 결함분석 프로세스 수립**

**3.1 테스트 성숙도 향상을 위한 활동 개선**

TMMi의 기본 프로세스의 각 영역 중 테스트 정책 및 전략, 테스트 계획, 테스트 모니터링, 테스트 설계, 테스트 환경 등의 강화를 통하여 결함예방 프로세스의 효과를 증대 시킬 수 있다. 기본 프로세스 중 결함예방 활동을 위한 프로세스 개선범위로 [Table 2]는 테스트 정책 및 전략, 테스트 계획, 테스트 모니터링 및 제어, 테스트 설계 및 실행, 테스트 환경 부분의 주요 개선사항이다.

[Table 2] Apply Process Area

Process Area	Addition Specific Process	
Test Policies & Strategies	1-1	Add to defect prevention policy
Test Planing	4-1	Analysis of the defect occurrence probability
	4-2	Frequency analysis of defects
	6-1	Resource Analysis for the reproduction of
Test Monitoring and Control	9-1	analysis of Defect trend
Test design and execution	13	Test implementation runs
	14	Test execution
	15	Test incident management
Test environment	17-1	Build a reproducible defect environment
	18-1	Maintenance repair defect reproduce the environment

\*Remarks : 1-1 is Specific Process number of [Table 1] and improved Specific Process number

3.2 결함 예방 프로세스

결함원인과 유형을 분석함으로써 상당수의 결함을 예방하고 방지할 수 있다. 대부분 결함을 축적하고 분류하여 공유된 데이터가 존재하고 있으나 결함의 유형을 분석하고 분류하여 대응 및 예방하는 활동은 부족한 현실이다. [Fig. 3]과 같이 결함이 보고되고 수집되면 결함해결 조직에 의해 결함원인이 분석되고 개선계획이 도출된다. 개선계획은 결함의 재발견 및 중복결함을 줄이기 위한 예방활동이다. 결함원인 분석은 조직 및 프로젝트 레벨단위로 이루어지며 특히, 사용자환경에서 발생된 결함을 체계적으로 관리하고 대응할 수 있어야 한다. 또 결함 분석은 단순결함의 원인분석과 프로세스 준수뿐만 아니라 수립된 예방활동이 소프트웨어 개발 및 품질개선 활

동으로 지속적으로 실행될 때 효과가 있다. 수립된 예방 활동은 테스트 정책 및 전략, 테스트 계획, 테스트 설계 및 실행활동 등과 연계하여 지속적으로 관리되고 실행되어야 하며 주기적으로 갱신되고 공유되어야 한다.

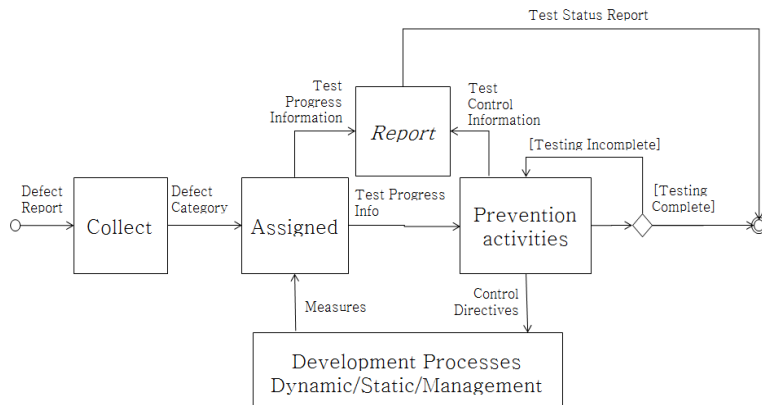
발견된 결함은 개발프로세스의 요구사항, 분석, 설계, 구현 등의 단계와 연계되어 분석되고 수립된다. 예방계획은 반복적으로 수집된 결함분석을 통하여 예방활동이 이루어진다[4,5].

제시된 결함예방 프로세스는 테스트 정책, 계획에 반영하고 테스트 모니터링 및 제어를 통하여 지속적으로 분석된다. 또한 이러한 활동을 통하여 분석된 내용은 테스트 설계 시 테스트 케이스에 반영되고 실행을 위한 테스트 환경에 적용된다.

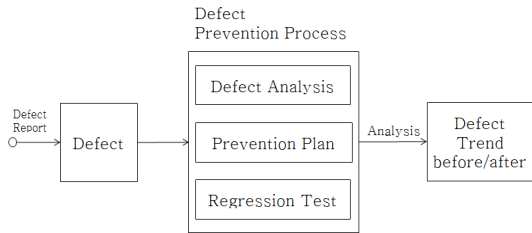
4. 적용 및 개선

4.1 적용 방법 및 연구모형

결함예방 테스트 프로세스 적용을 통하여 테스트 프로젝트의 결함예방은 [Fig. 4]과 같이 발견된 결함을 기준으로 결함분석 결함예방, 회귀테스트를 중심으로 적용하였다. 결함이 관련 고객대응, 개발, 품질조직을 통하여 수집되면 결함 예방 프로세스의 활동에 따라 결함의 유형과 고객에 따른 분석을 통하여 테스트 계획과 환경에 대해 정의한다. 이때 테스트 케이스 기존 개발과정의 테스트 케이스를 포함한 고객 환경 테스트를 실시하여 재현한다. 재현된 결함을 기반으로 관련 테스트를 강화하여 결함을 최소화 한다.



[Fig. 3] Defect Prevention Process



[Fig. 4] Research Model

### 4.2 결함분석 및 예방활동 적용

결함분석 및 예방활동을 위하여 인터넷 서비스 개발 프로젝트를 대상으로 20회에 걸쳐 테스트 결함 예방 프로세스에 따라 테스트 적용 및 보완을 반복 하였다. 주요 내용으로 결함 예방계획의 요구사항 정의, 테스트에 필요한 테스트 결과, 개발 프로그램, 개발 기획서, 개발 및 고객 결함보고 내용 구체화 등을 기반으로 테스트 정책, 전략, 계획, 설계, 환경을 포함한 테스트 케이스 수정과 결함 환경의 보존 등을 통하여 결함재연 비율이 증가되었다. 20회의 반복적인 프로세스 실행 및 회귀 테스트를 수행하였다.

269건의 결함을 대상으로 결함예방 프로세스를 실행한 결과 이 적용전과 비교하여 16건으로 결함이 감소하였으며 특히, 결함추이의 변화의 폭이 규칙적이고 안정화를 보였다.

[Fig. 5]와 같이 결함추이가 감소하는 경우 제품출시에 대한 위험이 낮으며 제품품질에 대한예측과 자신감을 확보할 수 있다. 그리고 일부분의 확인테스트 활동이 필요함을 확인할 수 있으며 6에서 9차 사이 적용전후 결함

변화에 차이를 보이기 시작하였으며 10차 이후 일정수준으로 결함이 유지되는 것으로 보아 안정적인 품질수준을 예상할 수 있다. 결함 예방 프로세스 적용 전후를 요약하면 [Table 3]과 같이 결함이 106으로 감소하였으며 회귀 테스트 포함 테스트 케이스 100개 감소로 실행시간이 400분 감소하였다.

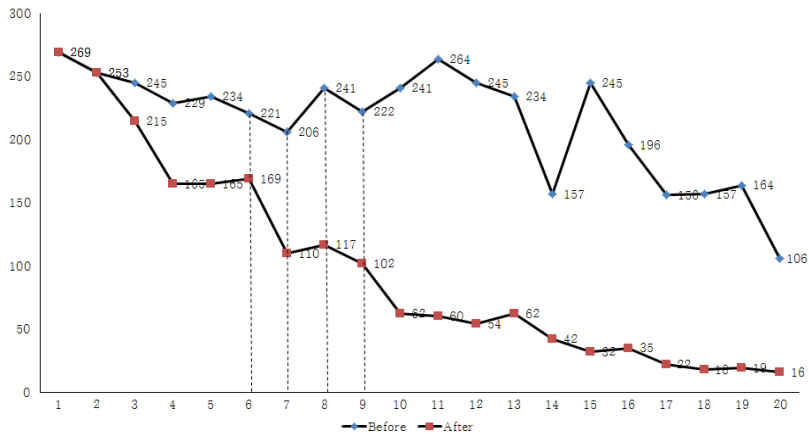
[Table 3] Before/After

Division	Before	After
Defect	106	16
Test Case Number	120	20
Time(min)	450	30

### 4.3 결함예방 및 개선방안

소프트웨어 결함은 요구사항 분석단계와 설계과정에 단순한 실수나 잘못된 정의가 구현된 후 발견되는 경우와 구현과정에 개발자의 불완전한 로직 구현 및 단순코딩 실수로 야기되는 경우도 있으며 때로는 잘못 정의된 테스트 케이스로 인하여 결함으로 보고되는 경우도 있다.

요구사항이나 개발범위에 정의된 내용을 테스트 케이스로 정의하고 소프트웨어를 실행한 결과 기대결과의 차이를 보고하여 수정을 요청하는 과정에 다양한 의사소통이 생기고 의견충돌이 일어나기도 한다. 대부분 출시 및 업데이트시기에 영향을 받은 기획, 개발팀의 의견과 소프트웨어의 결함을 발견한 테스트 팀이나 품질관리 부서의 프로세스와의 차이로 야기된다. 그러나 사업적인 목표로 인하여 결함을 수정하고 출시하는 시간을 줄이고 시장상황에 맞게 신제품을 출시하는 것이 더 큰 이익이



[Fig. 5] Defect Trend Analysis

되기도 한다. 품질관리 부서의 정책과 전략도 중요한 부분이나 경영자 및 책임자의 품질에 대한 인식과 의지가 중요하다.

소프트웨어 결함을 객관적으로 바라보고 판단하는 관리자는 두 가지 관점에서 결함을 이해하고 대응하는 자세가 필요하다. 첫째, 적시 시장출시를 통하여 사업적 목표를 달성하여 수익확보와 시장선점을 달성하여야 한다. 둘째, 요구사항 및 기타 품질이슈로 인한 고객의 불만과 기업의 이미지저하에 따른 장기적 경쟁력 상실에 대한 위험이다. 제품의 빠른 출시와 품질에 대한 이슈대응은 모든 조직의 프로젝트 관리자 및 경영자의 가장 큰 고민이다[6,7].

이런 경우 두 가지 관점의 의사결정을 통하여 판단하고 관련자를 설득한다면 상대적으로 합리적인 의사결정을 할 수 있다. 첫째는 사업적인 의사결정의 요소로 사용자에게 노출정도, 사업의 매출발생, 전략적 제품의 시장선점, 사용자의 제품에 대한 기대만족 등을 고려 할 수 있다. 둘째는 기술적인 요소로 제품의 출시를 위한 기술적인 복잡도, 연관성이 높은 모듈의 구성, 개발자의 기술적 대응능력 등이다. 이 두 가지 요소로 볼 때 사업적인 목표의 출시가 우선 된다면 주어진 자원을 최대한 투자하여 출시이후 품질적인 이슈를 극복하는 방법이 필요하며 만약 기술적인 요소가 부족하고 추가적인 위험요소가 발견된다면 출시시점을 조정하더라도 기술적인 위험을 최소화하고 품질확보를 선행하는 것이 적절하다. 품질관리 부서는 이 과정에서 경영자가 정확한 의사결정을 할 수 있도록 개발과정에 발견된 결함의 발견과 해결과정 데이터의 흐름을 제시하여 이 시점에서 제품을 출시한다면 어느 정도의 사용자 결함이 발견될 수 있으며 어느 정도의 해결시간이 필요한지 객관적인 데이터를 근거로 의견을 제시하여야 한다.

결함을 수정하고 개선하는 개발조직이나 결함을 발견하고 관리하는 품질관리 부서에서도 위와 같은 우선순위로 결함의 중요도를 결정한다면 최소한의 의사소통을 통하여 소프트웨어 품질을 확보할 수 있을 것이다.

## 5. 결론 및 향후 연구방향

본 논문은 소프트웨어 품질 중 제품품질 개선에 대하여 연구하였다. 소프트웨어 제품품질 개선을 위하여 가장 반복적이고 활용이 용이한 결함데이터를 분석하여 결

함 예방 프로세스를 수립하고 이를 테스트 프로젝트에 적용하기 위한 개선방안을 연구하였다. 이는 결함데이터의 분석을 통하여 테스트 프로세스를 개선하고 테스트 성숙도를 향상시키는데 기여할 수 있다. 나아가 소프트웨어 개발프로세스 개선을 위한 활동으로 발전이 가능하다. 소프트웨어 개발 프로젝트의 규모나 도메인에 관계없이 결함데이터를 통한 테스트 활동의 개선은 적용이 용이하며 반복적인 재작업을 최소화 할 수 있다.

본 연구는 테스트 프로젝트에 결함 예방 프로세스를 적용하러 전후 검증 하였으며 차후 다양한 도메인에 적용을 통하여 일반화 및 표준화 될 수 있도록 연구가 필요하다.

## References

- [1] Muhammad Javed, Bashir Ahmad, Zaffar Abbas, Allah Nawaz, Muhammad Ali Abid, Ihsan Ullah, "Decreasing defect rate of test cases by designing and analysis for recursive modules of a program structure: Improvement in test cases", International Journal of Computer Science and Information Security, Vol. 10, No. 8, August 2012.
- [2] SK C&C, Youngki Hong, "Improve of IT projects productivity through Defect predictive", 2009.
- [3] Erik Van Veenendaal, "Test Maturity Model Integration(TMMi)", Version 1.0 ( February, 17 th 2008) Produced by The TMMi Foundation, 2008.
- [4] NIPA, STA Testing, "Software test practice guide", STA, 2012.
- [5] ISO/IEC 29119 Test Process, 2008.
- [6] ISTQB, Certified Tester Advanced Level Syllabus, 2007.
- [7] Ron Patton, "Software Testing", Global press. 2003.

## 장 진 옥(Jin-Wook Jang)

[정회원]



- 2013년 2월 : 건국대학교 신산업융합학과 경영공학박사
- 1995년 2월 ~ 2005년 2월 : 국방부 정보사령부 전산장교 (대위)
- 2011년 4월 ~ 2013년 2월 : SK communications Manager
- 2013년 2월 ~ 현재 : 건국대학교 정보통신대학 인터넷미디어공학부 산학교수

<관심분야>

Software Quality & Testing, Project Management