Journal of the Korea Academia-Industrial cooperation Society
Vol. 15, No. 8 pp. 5227-5232, 2014

http://dx.doi.org/10.5762/KAIS.2014.15.8.5227
ISSN 1975-4701 / eISSN 2288-4688

# 소프트웨어 제품품질 개선을 위한 위험분석 시스템 개발 및 활용에 대한 연구

장진욱[1*]
[1]건국대학교 정보통신대학교 인터넷미디어공학부

# Development and Application of a Risk Analysis system for Software Product Quality Improvement

**Jin-Wook Jang**[1*]
[1]**Division of Internet & Multimedia Engineering, Konkuk University**

**요 약** 소프트웨어 제품 위험을 초기단계에서 제거함으로써 품질이 높은 소프트웨어를 개발 할 수 있다. 이를 위해 위험을 효과적으로 분석하고 관리하는 위험 분석 시스템의 도입이 필요하다. 제품 위험 분석 및 도구의 관심이 확대 되고 있으나 기존의 위험 관련 도구는 위험 추적 수준으로 개발되었으며 분석 및 전략 수립은 지원하지 않아 조직에서 제품 위험 관리에 많은 어려움을 격고 있다. 이에 본 논문에서는 조직에서 소프트웨어 개발과정에서 야기 될 수 있는 제품 위험의 문제점을 초기단계에서 해결하고 축적된 품질 위험 데이터를 통하여 정형화할 수 있는 소프트웨어 위험 분석 시스템을 제안하였다. 또한 어떤 제품 위험을 분석하고 관리해야 하는지에 대한 가이드라인을 제공하고자 한다. 위험 분석 협의과정을 통하여 도출된 위험 아이템에 대하여 위험 분석정보와 전략 등을 제공하는 위험 분석 시스템을 구현하고 활용방안을 제시하였다.

**Abstract** Better quality software can be made by eliminating the software product risk in the early stages. To achieve this task, a risk analysis system that can effectively track and manage risks that have severe effects on software quality is needed. The existing risk analysis systems have some weaknesses as they are applied to organizations. The major problems of those systems are that they require organizations to collect as much risk data at a time without providing a proper explanation and even without the support of a risk management process. This paper resolves those problems by developing a risk analysis system that offers methods of managing risks. In addition, the system provides the guidelines of which risks should be gathered for each step. The system also has functions to generate a range of strategy and analysis information on risks.

**Key Words :** Risk Analysis, Product Quality, Product Risk Management, Software Quality, Quality Management

## 1. 서론

소프트웨어와 하드웨어의 결합이 이루어지는 과정에서 그 동안 예상하지 못한 다양한 품질 결함이 발생하고 있으며 그 심각도가 높아지고 있는 추세이다. 이에 다양한 품질관리 프로세스와 도구들이 적용되고 있으며 더욱 효과적인 품질 관리 전략에 대한 연구가 이루어지고 있다. 그 중 품질관리 및 개선방법 중 제한된 인력과 기술을 가장 효율적으로 적용할 수 있는 위험 기반 품질관리 접근법을 조직전체에 체계화시키고 프로세스화 시키기 위한 도구를 연구하였다.

위험관리 관련도구로는 HP Mercury, IBM Rational, MicroFocus를 중심으로 개발되어 왔으며 소프트웨어 및 하드웨어 리스크 완화 활동을 정량화하고 체계적인 데이터베이스를 통하여 관리되지 않고 있다.

또한 기존 연구에서 소프트웨어 개발 프로젝트 차원

*Corresponding Author : Jin-Wook Jang(Konkuk Univ.)
Tel: +82-2-450-0547 email: jwjang@konkuk.ac.kr
Received March 31, 2014 Revised (1st April 21, 2014, 2nd May 12, 2014, 3rd May 20, 2014) Accepted August 7, 2014

에서 프로젝트 수준의 기술적, 자원적 위험은 프로젝트 계획단계에서 분석되어 관리되고 있으나 소프트웨어 제품(product)에 대한 위험 관리는 제품 개발 마지막 단계에서 관리됨에 따라 프로젝트의 지연 및 품질을 저하시키는 원인이 되고 있다. 따라서 이러한 위험을 초기에 발견하여 제거해야만 품질이 높은 소프트웨어를 생산 할 수 있다. 이를 위해 품질에 많은 영향을 미치는 제품 위험을 효과적으로 분석하고 관리할 수 있는 도구의 연구가 필요하다. 기존 위험 관리 도구는 프로젝트 계획단계에서 위험을 식별하는 수준으로 연구되었으며 사용자에게 직접적으로 노출되는 제품의 위험을 식별하고 대응하는 전략을 수립하고 자산화 하는데 어려움이 있다[1].

본 논문에서는 제안하는 위험 분석 도구는 소프트웨어 프로젝트 초기 단계에서 제품 위험을 식별하고 분석 및 조정하여 소프트웨어 제품의 품질을 확보하는데 목적을 두고 있다. 그리고 위험 분석 데이터를 기반으로 위험 완화를 지원한다. 제안 시스템을 통해 제품 위험을 식별하고 분석하여 전략적인 소프트웨어 제품품질 향상 계획 수립에 기여할 수 있으며 조직차원의 품질인식을 제고시키는데도 기여할 수 있다[2].

## 2. 관련연구

### 2.1 SFMEA

SFMEA (software failure mode and effect analysis)은 시스템의 고장 분석 기법 중에 하나로 장애가 제품에 어떠한 영향을 미치는 가를 정성적으로 분석하여 장애유형 및 시스템 자체의 영향을 평가하는 바텀업(bottom-up) 방식의 분석 기법이다[2,3]. 가능한 장애 모드를 식별하여 해당 장애의 발생을 예방하는 위험 식별 및 분석에 대한 체계적 접근방법이 있다. 절차로는 첫째, 소프트웨어 품질관련 이해당사자를 식별하고 이해당자가 품질 위험 분석에 참여하도록 요청한다. 둘째, 품질 위험 분석을 진행하는 기법을 참여 이해 당사자에게 사전에 인지시킨다. 여러 가지 품질 위험 분석 기법을 적용할 수 있다면 이를 제안하고 참여 이해 당사자와 충분히 토의한다. 셋째, 품질 위험 분석을 수행하며 테스트 대상 시스템의 고장 모드(failure mode)를 도출한다. 또한 각 고장 모드(failure mode)가 발생할 경우 파급효과를 분석하여 위험 우선순위(RPN : Risk Priority Number)를 도출한다. 최종적으로 각 위험요소에 대한 해소전략(recommended action)을 수립한다. 위험 완화를 위한 우선순위 산출은 위험의 심각성(severity)과 우선순위(priority) 그리고 발생가능성(likelihood)을 고려하여야 한다. 넷째, 누락된 요구사항이 있는 지를 검토하고, 디자인 및 개발관련 문서들을 재검토하여 추가적인 고장 모드가 존재하는지 식별하고 이해당사자에게 리포팅 한다. 다섯째, 품질 위험 결과를 이해당사자에게 배포하여 이상 유무를 검토한다. 이상 항목이 식별될 경우 셋째부터 다시 반복한다. 마지막으로 조직 내 형상관리 시스템에 해당 문서를 등록하여 문서에 대한 버전관리를 수행한다[3,4].

### 2.2 위험 관리 프로세스

위험 관리는 Fig. 1와 같이 대상 시스템의 위험 대상이 무엇이고 어디에 위험이 있는지 확인하는 위험 식별 단계가 있으며 잠재적으로 결함이 많은 부분을 분석하는 위험 분석단계 위험 정보를 근거로 대처방안을 수립하는 위험 계획단계가 있다. 마지막으로 위험을 관리하는 추적단계로 구별할 수 있다[3-5].


[Fig. 1] Risk Management Process

## 3. 제품품질 위험관리 시스템 정의

### 3.1 소프트웨어 제품위험의 정의

제품 위험은 품질 관련 모델 및 전문가들에 의해 다양하게 정의 내려지고 있다. 위험(risk)는 소프트웨어 개발 이후 사용자에게 부정적 결과로 끝날 수 있는 위험으로 제품 사용자에게 미치는 기능 및 비기능적 영향력(impact)과 제품적인 결함의 발생가능성(likelihood)으로 정의할 수 있다[9]. 또한 위험은 결함(defect)이나 장애(failure)의 상위 개념으로 소프트웨어의 제품품질(product quality)과 직접적인 관련이 있다. 여러 의미에

서 위험은 결함 및 장애 데이터보다 중요하다[9]. 결함 데이터는 우선 프로젝트 관리를 위해 필요하다. 대규모 프로젝트의 경우 제품적인 위험뿐만 아니라 인력적인 위험을 포함하여 수 천 개의 위험을 포함할 수 있으며 이 위험은 각기 다른 이해관계자(stakeholder)와 프로젝트의 계획, 설계, 개발, 테스트 등의 각각 다른 단계에서 다양한 형태로 발견하게 된다[4,5].

### 3.2 위험 관리 시스템

위험 관리 시스템(risk management system)은 하나의 제품을 개발하는 과정에서 발생되는 기술적 및 기능적 위험을 관리할 수 있도록 도구 형태로 제공되는 것으로 기획자, 개발자, 품질관리자 등의 이해관계자간의 의사소통을 원활히 할 수 있도록 도와준다[3]. 이러한 도구는 데이터베이스(database) 시스템을 근간으로 식별된 제품 위험을 데이터베이스에 레코드(record) 형태로 입력, 수정, 삭제하면서 위험을 관리한다[4]. 여기서 위험은 소프트웨어 제품 자체의 제품 위험을 의미한다. 단 프로젝트에서 운영하고 있는 프로젝트 관리 시스템, 형상관리 시스템, 테스트 관리 시스템과는 별도로 운영할 수 있으나 연동하여 운영하는 경우 더욱 효과적이다.

위험 관리 시스템의 주요 기능에는 위험 식별, 위험 분석, 위험 대응계획, 위험 추적 등이 있으며 이를 통하여 어느 정도의 위험이 경감되었는지 분석하고 가시적으로 보여 줄 수 있다. 이때 다른 프로젝트 관리시스템과 데이터가 공유된다면 더욱 효과적인 제품품질 관리를 할 수 있다[6,7].

## 4. 위험관리 시스템 설계

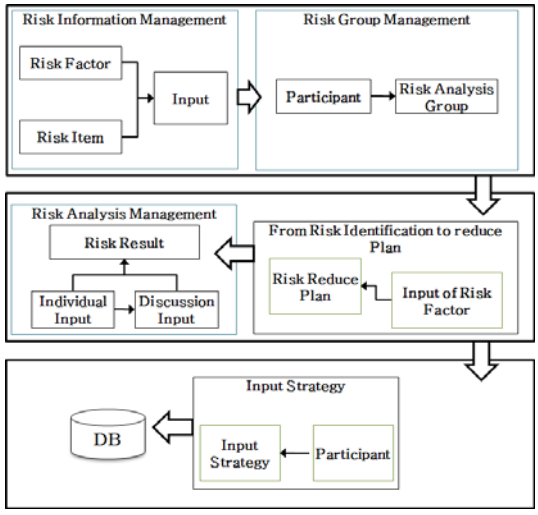
위험 분석 시스템의 설계는 소프트웨어 개발 조직에서 제품개발 계획단계에서부터 위험을 식별, 분석하고 전략을 수립 할 수 있도록 하는 도구이다.

### 4.1 시스템 구성설계

위험 분석기는 개발할 제품의 위험을 식별하여 분석하고 그 결과를 관리하는 '위험 관리', 위험 관리의 결과를 바탕으로 위험 완화 전략을 수립하고 관리하는 '위험 완화 전략 관리', '위험 관리기' 및 '위험 완화 전략'을 토대로 도출된 결과를 제공하는 '위험 정보 분석'으로 구성된다.

위험 분석 시스템의 세부 기능으로는 위험 기준 정보 관리 기능, 위험 분석 그룹관리 기능, 위험 분석 결과관리 기능으로 구분 된다. 각 기능별 세부내용은 다음과 같다[8,9].

1. 위험 관리 기능 : 위험 아이템 및 위험의 팩터 입력, 위험 분석 참여자 호출, 위험 분석, 개별 분석 입력, 위험 분석 협의, 분석 결과 조회 기능
2. 전략 관리 기능 : 전략 그룹 관리, 전략 팩터 및 아이템 입력, 전략 수립 참여자 호출, 전략 수립, 전략 조회 기능
3. 지원 기능 : 온라인 의견조정 기능, 데이터 내보내기 및 가져오기, 환경 설정




[Fig. 2] Risk Strategy System Structure

### 4.2 위험 분석 데이터

위험 분석 시스템에 입력되는 데이터는 Table 1에 정리된 내용과 같이 위험 분석 대상시스템과 대상시스템의 위험 아이템, 위험 팩터 등의 기본정보를 입력하고 위험 분석을 시작한다[10,11]. 이 중 1, 2, 3, 4, 5, 9, 10번 데이터는 이해관계자자 각각이 입력을 하게 되며 7, 10번 데이터는 각각이 입력한 데이터를 기반으로 제품품질 담당자가 수치를 취합하여 정하게 된다. 7, 8번의 데이터의 경우 소프트웨어조직에서 선임 개발자나 품질관리 담당자가 수행하게 되며 고객을 직접적으로 대응하는 고객서비스 담당자가 참여하여 객관적으로 진행되어야 한다.

[Table 1] Risk Analysis Data

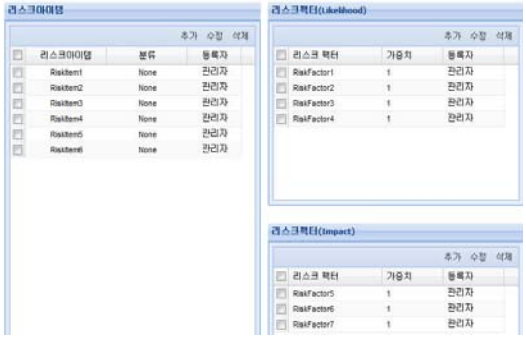
| No | Data | Data Description | Division |
|---|---|---|---|
| 1 | System Name | system under test | Input |
| 2 | Risk Item | function item of System | Input |
| 3 | Risk Factor | Aim of Project after Development | Input |
| 4 | Risk Type | Risk category | Input |
| 5 | Risk Value | input value of Individual and discussion | Input |
| 6 | Participant | The person in charge of product risk | Input |
| 7 | Risk Metrix | discussion result of risk | Analysis |
| 8 | Risk Management | From Risk identification to reduce Plan | Analysis |
| 9 | Likelihood | Development Experience, level of technology, Complexity, Degree of New Development, etc | Input |
| 10 | Business Impact | Treatment of user, Visibility, economics, etc | Input |

## 5. 위험 분석 시스템 구현

### 5.1 시스템 구현

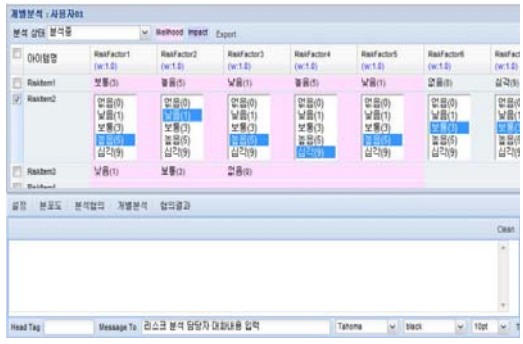
위험 관리를 위한 대상시스템이 결정되면 위험 분석을 위하여 시스템의 위험 아이템, 위험 팩터, 위험 분석 참여자 등을 입력 받는다. 오프라인 미팅을 통하여 충분한 협의 후 진행하는 것이 효율적이나 본 시스템에서는 온라인으로 참여하고 결과를 확인 할 수 있는 장점이 있다.

다음 Fig.3, 4, 5, 6은 주식회사에스티에이컨설팅 솔류션 프로젝트에서 구현한 화면이다.



[Fig. 3] Risk Analyzer input - Risk Item, Factor, User

제품 위험 분석 담당자는 채팅창을 통하여 위험 분석을 위한 정보로 제품의 개발 산출물과 사용자 유지보수 이력, 사용자에게 노출된 이슈 등을 선 공유함으로서 참여자에게 충분한 정보를 제공한다. 참여자는 각 위험 펙터를 기준으로 정성적인 중요도를 “0, 1, 5, 9” 중 각 사용자별 Fig. 4와 같이 입력한다.



[Fig. 4] Risk Analyzer & Chatting

### 5.2 위험 분석 및 협의

위험 분석 및 협의를 위하여 개인별 입력된 위험 수치를 기반으로 위험 분석 협의 값을 입력 받는다. 다음은 참여자별 개별입력한 중요도를 조율하는 과정이다. 위험 아이템의 선정과 중요도는 참여자의 역할에 따라 차이가 크다. 개발자가 보는 중요도와 기획자 및 품질관리자가 보는 중요도는 차이가 있다. 개발자는 기술적인 중요도를 상대적으로 높게 보는 경향이 있으며 기획자는 시스템이 사용자에게 노출되는 인터페이스와 같이 노출되는 부분에 좀 더 중요도를 높게 입력할 수 있으므로 조율을 통하여 의견의 방향을 일치시킨다.

[Fig. 5] Risk Analysis Screen

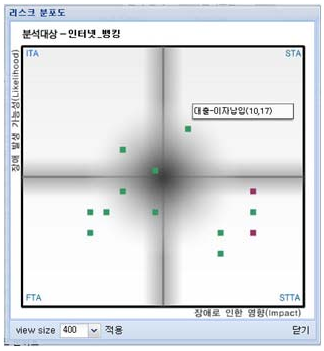
### 5.3 위험 관리 시스템의 주요 분석정보

각각의 이해관계자는 조정 값을 입력하게 된다. 입력

값은 편차가 있을 수 있으며 이를 하나로 귀결시키는 방법으로 평균값이나 중앙값을 선택하는 것도 하나의 방법이다. 본 연구에서는  참여자의 최대, 최소, 평균값을 이용한 3점법(3 Point Method)을 활용하여 조정 값을 계산하였다.

$$3\,Point\ Method = \frac{\Sigma(MAX + (4 \times AVG) + MIN)}{6}$$

다음 Fig. 6은 위험 분석 결과 화면으로 각각의 참여자 개별입력을 바탕으로 조정된 값이 반영된 화면이다. 해당 사례를 인터넷뱅킹 시스템을 대상으로 위험분석을 하였으며 장애로 인한 영향과 장애발생 가능성을 중심으로 볼 때 “대출-이자납입”에 대한 위험 아이템이 우선수선 순위가 높도록 배치되었다. 이러한 정보는 프로젝트 이해 당사자에게 제품풀질 우선순위의 가이드를 제공한다.




[Fig. 6] Risk distribution chart

이를 통해 프로젝트 관리자는 해당 프로젝트에 어느 위험의 발생가능성 및 사업적인 영향도가 높은지 파악할 수 있다. 프로젝트 관리자는 이러한 위험 분석결과를 근거로 이후 유사 프로젝트 진행 시 이러한 유형의 위험에 대비, 개발자에게 부족한 기술 교육, 공수의 조정, 장비 추가 등 사전에 프로젝트 차원의 위험을 낮출 수 있다.

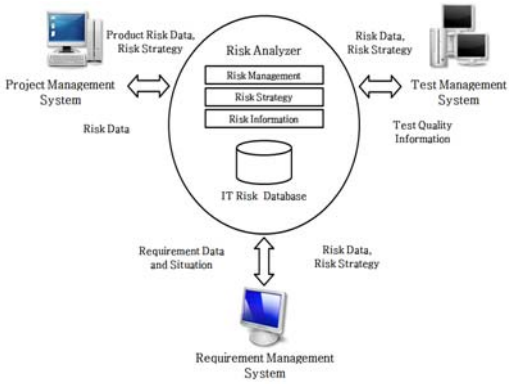
### 5.4 활용 분야

위험 분석 시스템은 프로젝트 관리 시스템과 연동하여 위험 분석기에 저장된 위험 정보와 위험 완화 전략정보를 프로젝트 관리 시스템으로 전달하여 프로젝트 관리 시스템에서 프로젝트 위험 관리에 활용한다.

위험 분석기는 테스트 관리 시스템과 연동하여 위험 분석기에 저장된 위험 정보와 위험 완화 전략정보를 테스트 관리 시스템으로 전달하여 위험 기반 테스트 관리를 지원한다. 상대적으로 위험이 높은 기능과 요구에 테스트에 집중될 수 있도록 하여 효과적인 테스트가 가능하다.

위험 분석기는 요구관리 시스템과 연동하여 위험 분석기에 저장된 위험 정보와 위험 완화 전략정보를 요구관리 시스템으로 전달하여 요구사항을 위험 우선순위에 따라 효과적으로 관리할 수 있다.

위험 분석기는 프로젝트 관리 시스템, 테스트 관리 시스템, 요구관리 시스템 등과 연동하여 독립적인 위험 분석 전문 도구로 활용된다.




[Fig. 7] Risk Analyzer utilize - Connecting with the Other Systems

## 6. 결론

본 논문에서 제안한 위험 분석기는 역할에 따라 다양한 효과를 볼 수 있다.

프로젝트 관리자는 다음과 같은 효과를 볼 수 있다.

첫째, 위험 분석 결과를 통해 소프트웨어 개발 초기단계에서 제품 품질 위험을 파악하고 이를 프로젝트 관리 및 계획에 반영할 수 있다.

둘째, 영향력이 큰 제품 위험의 재발 가능성을 막기 위해 예방 차원의 조치를 취할 수 있다.

셋째, 프로젝트 관리자는 위험을 제거하는 방법이 어떤 것이 효율적인지에 대한 정보를 정량적으로 파악할 수 있다.

넷째, 소프트웨어 품질 또는 프로젝트 관리 조직에서 위험 분석기를 이용하여 품질 및 프로젝트 위험 기반의 관리 프로세스를 적용함으로써 해당 조

직의 품질 관리 역량이 강화되고 이는 결국 소프트웨어 품질 향상으로 이어지게 된다.

조직 차원에서는 다음과 같은 효과를 볼 수 있다.

첫째, 조직의 개발성숙도 수준 별로 위험 수집을 적용할 수 있다.

둘째, 조직 차원에서는 어느 단계, 어느 프로세스에 위험이 많이 유입되는지 파악함으로써 조직의 표준 프로세스 개선 정보에 도움을 줄 수 있다.

셋째, 소프트웨어 개발과정에서 위험 분석기를 활용함으로써 의사 결정권자들이 위험 정보를 시스템으로 공유하게 온라인을 통하여 참여 참으로서 좀 더 많은 사용자참여가 가능하며 관련 업무의 의사소통 비용을 줄일 수 있다.

위험 분석기는 위험 분석 및 협의기능을 통해 제품 위험 분석 활동을 지원하고 조직의 위험을 개발 성숙도 수준에 따라 관리할 수 있도록 하였다. 각 수준마다 필요한 결함 관리에 대한 가이드라인도 제시하였다. 이를 통해 조직이 위험을 정량적으로 관리할 수 있는 토대를 마련하였다. 또한 위험 분석기는 조직에서 운영 중인 PAL(process asset library)과 연동하여 활용될 수 있다.

그리고 위험 분석기는 소프트웨어 개발 조직에 적합하게 개발되어 위험 식별 및 분석 위주의 활동을 수행하는 프로세스에 초점을 맞추었다. 더욱 더 높은 품질의 소프트웨어를 개발하기 위해서는 위험 예방에 관한 연구가 선행되어야 한다. 이를 위해 위험 예측 모델 및 위험 원인들 간의 상관관계 분석이 이루어져야 한다. 또한 위험 분석기를 통해 분석된 위험은 프로젝트 관리 시스템 및 테스트 관리 시스템과 연동하여 소프트웨어 개발 라이프사이클 전체와 연계하여 운영 하는 것이 필요하다. 이를 통해 소프트웨어 제품품질 향상에 기여 할 수 있다.

## References


[1] Ray C. Williams, George J. Pandelios, Sandra G. Behrens, "Software Risk Evaluation (SRE) Method Description (Version 2.0)", TECHNICAL REPORT, CMU/SEI-99-TR-029, ESC-TR-99-029, December 1999,

[2] Glenford J. Myers, Corey Sandler, "The Art of Software Testing", JohnWiley & SonsInc 2004.

[3] Erik Van Veenendaal, "Test Maturity Model Integration(TMMi)", Version 1.0 Produced by The TMMi Foundation, 2008.

[4] ISTQB, "Certified Tester Foundation Level Syllabus", 2007.

[5] James Bach, "James Bach on Risk-Based Testing", 1999.

[6] ISTQB, "Certified Tester Advenced Level Syllabus", 2007.

[7] NIPA, STA Testing, "Software test practice guide", STA. 2012.

[8] Kwon, Yong Rae, "software testing", Saeng Neung Press, 2010.

[9] Kwon, Wonil , "Software testing practice", 2007.

[10] Ron Patton, "Software Testing", Global press. 2003.

[11] ISO/IEC 29119 Test Process, 2008.



**장 진 욱(Jin-Wook Jang)** **[정회원]**

- 2013년 2월 : 건국대학교 신산업융합학과 (경영공학박사)
- 1999년 2월 ~ 2005년 2월 : 국방부 정보사령부 전산장교 (대위)
- 2008년 10월 ~ 2010년 12월 : 주식회사에스티에이컨설팅 책임
- 2011년 4월 ~ 2013년 2월 : SK communications PMO Manager
- 2013년 2월 ~ 현재 : 건국대학교 정보통신대학교 인터넷미디어공학부 산학교수

<관심분야>

Software Quality & Testing, Project Management