

TMMi를 반영한 국방 소프트웨어 테스트 절차 발전방안

박연동
국방기술품질원

Defense software test procedure improvement measure reflecting the TMMi

Yeon-Dong Park
Defense Agency for Technology and Quality

요약 현대 무기체계는 전자장비 탑재의 비중이 늘어나고 있으며, 그에 따른 소프트웨어 탑재비중도 늘어나고 있다. 소프트웨어 비중이 높아감에 따라 그 품질관리와 개발방법에 대한 관심도 높아졌다. 물리적으로 가시화 된 기계체계와 달리 논리적으로 비가시화 된 소프트웨어는 결합에 노출될 경우 복구의 어려움, 재사용의 어려움, 자원의 낭비와 같은 위험에 놓여 질 수 있다. 미국의 경우 국방 소프트웨어의 개발과 품질관리를 위한 별도의 프로세스를 연구하고 적용하는 노력을 통해 CMM(Capability Maturity Model)을 만드는 성과도 거두었다. 우리나라 또한 국제표준을 바탕으로 국방 소프트웨어 개발 및 품질관리에 대한 기준을 만들어 활용하고 있다. 그러나 소프트웨어의 발전 속도에 맞추어 기준을 변경해야 함에도 아직 기초수준에 머무르고 있다는 지적이 있다. 소프트웨어 발전 속도에 비해 기준이 낮다면, 자원 재사용 비용 및 복구비용이 기하급수적으로 늘어난다. 본 논문에서는 CMM의 소프트웨어 테스트 파생형인 TMMi(Test Maturity Model-integration)을 통해 국방 소프트웨어 테스트 절차 발전 방안에 대해 논의하고, 국방 소프트웨어 품질 관리에 대한 청사진을 제시하고자 한다.

Abstract As the portion of modern weapons systems equipped with electronic components increases, the presence of embedded software has also increased, subsequently drawing interest in quality management and development methods. Unlike mechanical systems, software is logically intangible, hard to troubleshoot, and hard to reuse, making it hard to utilize limited resources when exposed to failures. The United States had successful results in establishing the Capability Maturity Model (CMM) by studying and applying separate processes for development and quality management of defense software. South Korea has also established, and utilizes, a standard of development and quality management for defense software based on international standards. However, some say those standards still remain at a basic level, and should be modified along with the progress in software. If the standard stays at a basic level, compared to software progression, the cost to reuse and restore resources will increase exponentially. This paper discusses improvement in the test processes for defense software through the Test Maturity Model-integration (TMMi)-derived from the CMM-and presents a blueprint for defense software quality management.

Keywords : Software Quality, Software Testing, Component Based Development, CMMi, TMMi

본 논문은 국방기술품질원 연구과제로 수행되었음.

*Corresponding Author : Yeon-Dong Park(Defense Agency for Technology and Quality)

email: ydpar@dtaq.re.kr

Received September 4, 2020

Revised October 23, 2020

Accepted January 8, 2021

Published January 31, 2021

1. 서론

1.1 TMMi와 국방 소프트웨어의 현 주소

TMMi(Test Maturity Model-integration)은 기존에 성숙도 측정으로 사용되던 CMMi(Capability Maturity Model-integration)의 소프트웨어 테스트 부문 파생모델이다. CMMi는 제품 수명주기에 따른 기획-설계-개발-양산-유지단계에서의 조직을 진단하고 구조화 가이드를 제시했다. TMMi는 소프트웨어 테스트 과정을 기획-설계-개발-양산-유지단계로 나누어 테스트 조직을 진단하고 구조화 가이드를 제시한다. 국방 소프트웨어는 무기체계 뿐 아니라 전력지원체계에 사용되는 모든 소프트웨어를 지칭한다. 방위사업청은 개발 가이드를 제시하고, 국제표준에 맞추어 개정한다. 하지만 빠르게 발전하는 소프트웨어 기술과 그에 따른 테스트 기법에 맞추어 대응하기가 쉽지 않다. 미국의 경우, CMM(Capability Maturity Model)을 연구하여 현재 위치를 점검하고 기술변화에 대응하고자 노력하였다. 우리나라 또한 많은 기업체에서 CMMi(Capability Maturity Model-integration)을 도입하여 조직을 정비하고 제품 품질관리를 강화하기 위한 노력을 하고 있다. 그러나 방위사업 부문에서는 이러한 노력이 더딘 것이 사실이다. 방위사업 부문은 빠른 변화보다는 안정적이고 확실한 방법이 필요하다. 작은 결함이라도 큰 손실로 이어지는 것이 확실하기 때문에 작은 무기체계 소프트웨어라도 신중을 기해야 한다. 이러한 점은 빠르게 발전하는 소프트웨어를 활용한 새로운 무기체계를 도입하고자 하는 우리 군의 미래전력 계획과 상충한다. 상충하는 부분을 완화하기 위해서 절차를 강화하거나, 생략하는 방법이 필요하다. NIST에서는 잘 만들어진 테스트 절차가 있으면 결함을 처리하는 비용을 줄일 수 있다는 연구결과를 발표하였다[1]. 국내에서는 TMMi를 통해 테스트 조직 및 프로세스 개선 가이드가 가능함을 연구한 사례가 있으며[2], CBD방법론이 중복 개발과 유지보수 비용을 줄일 수 있음을 연구하였다[3]. CBD와 TMMi 모두 품질을 추구한다는 점에서 같으나, CBD에서 측정하지 못하는 소프트웨어 테스트 절차를 강화하는 것이 필요하다. 절차를 강화하기 위해서 현재 조직의 위치가 어디인지, 무엇이 필요한지 진단이 필요하다. 여기서 TMMi를 통해 조직의 성숙도를 측정하여 조직의 위치와 필요한 것이 무엇인지 진단하여야 한다. 본 연구에서는 TMMi와 국방 소프트웨어 테스트 절차에 대해 소개한다. 그리고 국방 소프트웨어 개발의 대표 격인 국방 CBD방법론을 토대로 TMMi를 적용하여

개선할 수 있는 방안이 무엇인지 연구해 보았다.

2. 본론

2.1 TMMi(Test Maturity Model-integration)

TMMi Foundation에서 제시하는 가이드라인에 따른 소프트웨어 테스트 성숙도 단계는 Fig 1과 같다[4].

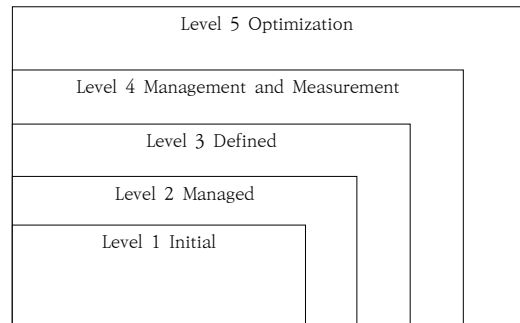


Fig. 1. TMMi Maturity Level

그리고 각 성숙도 단계별 구조와 구성요소는 Fig 2와 같다.

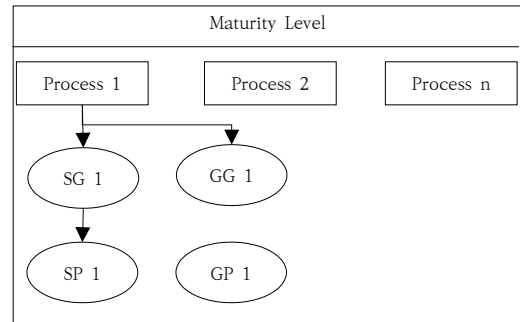


Fig. 2. Structure and component of the TMMi

만약, Level 1에서 Level 2로 올라가려면, 이전 Level 1 단계에서 SG(SpecificGoal)과 GG(GeneralGoal)를 모두 완수해야 평가에 돌입할 수 있다. 여기서 SG는 프로세스를 만족시키기 위한 구성요소로 다음 프로세스로 넘어가기 위한 결정요소라고 볼 수 있다. SG가 완수되었는지 평가하기 위해 SP(Specific Practice)가 필요하며 SP는 어떠한 행위로 SG를 달성할 수 있는지 설명해 놓은 것이다. GG는 프로세스 순서에 구애받지 않는다. 모든 프로세스에서 같은 GG가 나올 수 있다. GG는 주로

프로세스의 마지막에 나타난다. GP(Generic Practice)도 SP와 같이 GG를 달성할 수 있는지 설명해 놓은 것이다. GP도 모든 프로세스에서 같은 GP가 나올 수 있다. 성숙도 단계별 상세내용은 다음과 같다[5].

Level 1: Level 1의 테스트는 정의되지 않았거나, 테스트와 디버깅이 한 부분으로 인식되고, 조직의 안정은 조직인력의 능력과 자신감에 의존적이다. 테스트 활동과 디버깅은 버그를 제거하기 위해서만 이루어진다.

Level 2: Level 2의 테스트는 테스트가 소프트웨어 생명주기에서 하나의 독립된 단계로 정의된다. Level 2를 달성하려면 테스트 정책을 별도로 두거나 상위 정책의 일부에 정의하고 있어야 한다. 이 밖에도 문서화된 절차에 따라 테스트 환경을 확보 및 관리하고 있어야 한다.

Level 3: Level 3의 테스트는 개발 생명주기와 통합되는 단계이다. 이 레벨에서는 테스트 프로세스가 소프트웨어 개발 수명주기에 포함되어 있어야 한다. Level 2와 다른 점은, 조직차원에서 테스트 활동을 내재화 한다는 것이다.

Level 4: Level 4를 위해선 동료검토가 활성화 되어야 한다.

Table 1. TMMi check list in each maturity level[6]

Maturity Level	Check area
Level 5	5.1. Prevention defect
	5.2. Optimization test procedure
	5.3. Quality management
Level 4	4.1. Test measurement
	4.2. Product quality assesment
	4.3. Reinforced peer-review
Level 3	3.1. Test organization
	3.2. Test training program
	3.3. Test life cycle and integration
	3.4. Non-functional test
	3.5. Peer review
Level 2	2.1. Test policy and strategy
	2.2. Test plan
	2.3. Test observation and management
	2.4. Test design and execution
	2.5. Test environment
Level 1	Nothing

Table 1에서 볼 수 있듯이 TMMi의 성숙도 단계가 올라갈수록 관리영역이 넓어지고, 짧은 시간 내에 소프트웨어의 결함을 다량으로 발견하여 자원을 아낄 수 있다 [7]. 아울러 TMMi Foundation에서 제시하는 TMMi의 구성요소인 SG와 SP내 점검항목은 Level 2부터 Level 3까지 확인영역이 포함되기 때문에 Level 1은 SG와 SP가 없다. Level 4에서 Level 5의 경우 Level 2와 Level

3을 구체화 시키는 내용이므로 제외했다. 그리고 상위레벨로 진입하기 위한 필수요소가 SG와 SP이므로 해당내용을 인용하였다. 마지막으로 SG와 SP 전체 내용 중 개선을 위한 의미 있는 활동을 Table 4에 정리하였다.

2.2 국방 소프트웨어 테스트

국방 소프트웨어는 ISO/IEC 122207의 수명주기를 반영한 국방 무기체계 소프트웨어 매뉴얼에 따라 개발되고 시험된다. 특히, 국방 CBD 방법론에 무기체계 소프트웨어 테스트에 대한 절차가 상세히 나와 있다. 국방 CBD 방법론에 따른 소프트웨어 테스트 절차를 요약하면 Fig 3와 같다[8].

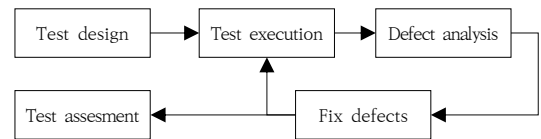


Fig. 3. Defense software test procedure

기본적으로 설계-실행-분석-수정-평가의 순서로 되어 있다. 테스트 설계에서 일정과 테스트 케이스 등을 설정하고, 테스트 활동을 통해 결함을 발견하고, 발견된 결함을 수정한 다음 총평을 하는 구조이다. 이 구조는 일반적인 소프트웨어 테스트 절차이며, TMMi가 아직 적용되지 않은 절차이다. 본 연구에서는, 국방 무기체계 소프트웨어 매뉴얼 중 가장 대표적인 국방 CBD 방법론과 TMMi를 활용하였다. 두 방법론 모두 품질향상을 추구한다는 점에서 같다, 그러나 세부적인 체크 리스트에서 차이가 난다. 국방 CBD 방법론에서는 산출물 체크 리스트이다. TMMi는 행동여부 체크 리스트이다.

2.3 TMMi를 적용한 국방 소프트웨어 테스트 발전 방안

국방 CBD 방법론에 따른 소프트웨어 통합 테스트 절차는 Fig 3과 같다. Table 2에 나온 TMMi의 성숙도별 확인영역과 결함발견 활동에 대입해 보면 국방 소프트웨어 테스트 절차는 Level 2과 같다고 할 수 있다. 테스트 계획과 설계, 실행은 만족하지만 비기능 테스트와 동료평가가 부족하다. 만약 상위 레벨에서 요구하는 동료평가와 테스트 절차 최적화 활동을 구축하고 관리한다면 TMMi에서의 Level 3 이상의 조건을 충족함과 동시에 소프트웨어 테스트에서 발생하는 다양한 문제를 줄일 수 있고

결과적으로 소프트웨어 신뢰도가 올라가며 자원의 낭비를 줄일 수 있을 것이다. 본 연구에서는 Fig 4와 같은 절차를 따라 개선방안을 고안하였다. 그리고 Level 3을 위한 테스트 절차 개선을 Fig 5와 같이 제안해 보았다.

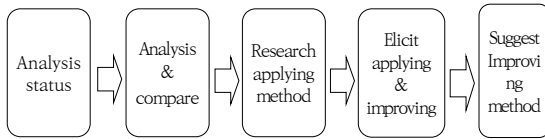


Fig 4. Research process concept

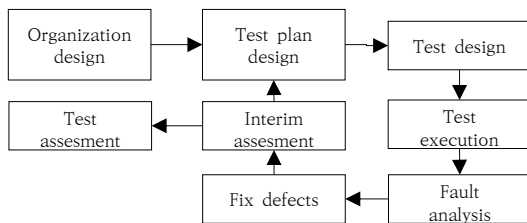


Fig. 5. Improvement measurement for defense software test procedure adjusting the TMMi

본 제안에서는 테스트수행을 설계하기 전, 테스트 조직을 설계한다. 테스트 참여자는 테스트 목표를 정하고, 테스트 기술과 대상에 대한 충분한 이해를 가지고 있다. 조직을 설계한 후, 테스트 계획을 설계한다. 기본적으로 일정과 산출물에 대한 내용, 그리고 테스트 케이스가 있다. 여기에 더해 기능적 위험과 구조적 위험에 대한 워크스루(Workthrough)도 진행한다. 테스트 설계 후, 이하 절차는 기존의 절차와 같다. 다만, 오류 수정 후 “중간평가”를 추가하였다. 오류를 수정했다 하더라도 그것이 요구조건을 위한 오류였다면, 테스트는 잘못 수행되었다고 할 수 있다. 잘못된 테스트로 인해 다른 문제점이 발생할 가능성이 있으므로 테스트 계획 단계부터 절차적으로 점검한다. 위의 반복활동 후, 중간평가를 통해 참여자가 모두 동의하면, 테스트 평가로 진입하여 최종 산출물을 검토한 뒤 마무리 한다. 이 제안은 기술적으로 정해진 테스트 도구와 케이스로는 정확한 테스트 성숙도 측정에 어려움이 있기 때문에, 비기능 테스트 항목을 넣음으로서 성숙도 측정을 병행하게끔 하였다. 제안한 절차와 기존 절차는 Table 2와 같이 정리할 수 있다.

아울러 다음과 같은 개선을 통해 소프트웨어 신뢰도 향상을 제안한다.

Table 2. Comparison with the Defense CBD

	Defense CBD method (Current)	Reflecting the TMMi (Suggestion)
Procedure	5 Steps Test Design, Test Execution, Fault Analysis, Fix Defects, Test Assesment	8 Steps Organization Design, Test Plan Design, Test Design, Fault Analysis, Fix Defects, Interim assesment, Test Assesment
Test method	Function based black-box testing	work-through, peer-review and inspection
Advantages	Strict testing by planned procedure Find fault from functional testing at user sight	Make detailed activities in each test events Always same test quality Check maturity of test organization Prevent the "Pesticide Paradox"
Disadvantages	No detailed activities in each test events Always different test quality Unknown maturity of test organization Exposure at the "Pesticide Paradox"	No control the function test result

2.3.1 형상물 통제 표준화

국방 분야에서는 다양한 SIMS(Software Inspection Management System)을 도입하여 사용하고 있지만, 표준화된 형상통제 방안이 없다. 예를 들어, 결함관리번호와 결함식별번호와 같은 객관적인 표시에 대해 전 군이 표준화된 표시방식을 갖춘다면, 결함을 식별하고 TMMi 프로세스에 따라 테스트를 함으로써 신뢰도를 향상시킬 수 있다. 물론 산출물 변경에 따른 별도의 버전을 관리하는 VMS(Version Management System)과 통합운동 또한 고려해야 할 것이다.

2.3.2 동료검토 활성화

많은 테스트 활동의 결과가 말해주듯, 테스트 수행자가 동일 테스트를 동일한 방법으로 지속한다면 살충제 패러독스(Pesticide Paradox)에 노출되기 쉽다. 그 뿐 아니라 회귀 테스트(Regression Test) 수행 시에도 본인에게 익숙한 부분은 넘어갈 가능성이 높다. 동료검토를 활성화 하면, 제 3자의 입장에서 테스트 수행을 객관적으로 바라볼 수 있는 기회가 생기고, 살충제 패러독스를 피할 수 있는 여지가 남아있다. 하지만, 동료검토 시에 별도의 테스트 케이스를 작성해서 테스트를 할 것인지, 또는

요구사항에 대한 이해가 충분한 지에 대해 이해관계자의 논의가 반드시 필요하다.

2.3.3 관리체계 활성화

국방 무기체계 소프트웨어의 품질관리는 상당히 체계적으로 이루어져 있지만, 수준이 높은지 가늠할 수는 없다. 이러한 이유로 필요한 것이 국방 소프트웨어 품질관리체계이다. 현재 사용되고 있는 정보체계가 있지만, 사용빈도와 축적된 데이터의 수가 매우 적으며 관리 또한 제대로 이루어지고 있지 않은 실정이다. 결국, 향후 필요에 따라 품질관리를 위한 새로운 정보체계 개발이 필요하다는 이야기이다. 이렇게 되면 자원과 시간 낭비는 당연한 것이고, 구축에 필요한 시간만큼 품질확보가 늦어질

가능성도 크다. 상용 관리체계도 있지만, 기술적으로 민감한 부분을 다루는 국방 무기체계 특성상 별도의 관리체계를 구축하는 것은 필요하다. TMMi에서는 기존에 사용하던 도구와 같은 자원의 제공성도 성숙도 측정 항목에 포함되어 있다. 이미 구축한 국방 소프트웨어 품질관리체계를 조직을 평가하고 테스트 활동을 관리하는 측면을 중심으로 활성화 시킨다면, TMMi의 성숙도가 올라가고, 소프트웨어 신뢰도도 향상될 것이다.

2.3.4 명확한 목표설정

TMMi의 구성요소 중 SG와 GG가 있다. 다시 말하면, 일반적인 절차를 위한 목표와 절차 속의 요구사항 만족도 테스트를 통해 현재 테스트 조직이 얼마나 요구조건

Table 4. List of the SP and the SG in each check area for improvement

Check area	SG	SP
3.1. Test Organization	SG1. Establish test organization and compose education system?	SP1. Test organization composing and test training establishment
		SP2. Test works allocation by each attendants
3.2. Test training program	SG1. Have essential education for test activities?	SP1. Test training planning
		SP1. Test training recording and evaluation
3.3. Test life cycle and integration	SG1. Is test executive plan be appropriative composed?	SP1. Integration between test life cycle and development model
		SP1. Specific test overall plan establishment and evaluation
3.4. Non-functional test	SG1. Is there verification about non-functional items around by test environment and execution?	SP1. Non-functional test design and analysis
		SP2. Non-functional test implementation and execution
3.5. Peer-review	SG1. Have a specific plan for peer-review and execution?	SP1. Peer-review items identification
		SP2. Peer-review execution and result evaluation
2.1. Test policy and strategy	SG1. Do stakeholder share the test policy?	SP1. Test policy establishment
		SP2. Test strategy and goal establishment
2.2. Test plan	SG1. Have an identification of risk? Is test plan accurately coincided with test work?	SP1. Risk identification and evaluation
		SP2. Test plan execution and test evaluation establishment
2.3. Test observation and management	SG1. Contrast of test plan, is progress appropriate?	SP1. Test progress review
		SP2. Product quality review
2.4. Test design and execution	SG1. Is test be implemented and executed according to the test design technic?	SP1. Test case execution and event reporting
		SP2. Appropriate activity establishment for test event modification
2.5. Test environment	SG1. Is test environment be appropriately implemented and be managed?	SP1. Specific test environment implementation coincidence with test process
		SP2. Test environment development and analysis

을 만족하는 프로그램 개발을 달성하였나를 스스로 측정하는 것이다. 보통의 경우, 테스트 목적은 “결함을 찾아내는 활동”이다[9]. 하지만 TMMi에서는 “특정 절차에서 결함을 찾아 어떠한 행동을 취한 것”까지 스스로 측정할 수 있도록 가이드를 제시한다. 결국 결함발견 수를 줄이고 소프트웨어 신뢰도를 높이기 위해 명확한 테스트 목표를 설정하는 것이 필요하다. 목표를 기록하고 추적하기 위해서 테스트 인력이 많이 필요하겠지만 관리 도구를 통해서 자원을 효율적으로 활용하는 방법도 있다. 현재 사용하고 있는 정보체계 도구를 통해서 목표를 기록하고 추적할 수도 있다. 그래서 국방 소프트웨어 품질관리 정보체계의 유지보수와 활성화가 필요하다.

지금까지 제안한 4가지 개선에 따른 기대효과를 정리하면 Table 3과 같다. Table 3은 4가지 개선에 따른 각 기대효과와 최종 기대효과도 정리하였다.

Table 3. 추가제안에 따른 기대효과

	Additional suggestion	Expected effect
2.3.1	Standardization a configuration control	Standardization fault identification Satisfy TMMi Level 3 maturity
2.3.2	Activation a peer-review	Prevent the "Pesticide Paradox" Satisfy TMMi Level 3 maturity
2.3.3	Activation a management system	Reduce wasted test resource and time
2.3.4	Set a cleared test purpose	Manage accomplishment of test organization
Conclusion	Match with the TMMi Level 3	Expect to highest performance of test organization Reduce wasted test resource and time

3. 결론

TMMi는 테스트 조직의 성숙도를 측정하는 방법이다. 국방 소프트웨어는 매뉴얼에 따라 개발되고 평가되어진다. 현대 무기체계에는 소프트웨어 사용 비중이 높고, 쓰인 소스코드의 수도 늘어났다. 예를 들어, F35에 쓰인 소스코드는 830만 줄에 달한다[10]. 이에 따라 소프트웨어의 복잡도와 정밀도는 증가했지만 현재 국방 소프트웨어 매뉴얼의 테스트 절차는 이러한 빠른 흐름에 대응하기 어려운 것이 현실이다.

흐름에 대응하지 못하면 개발시간이 길어지고 결함복구 및 재시험에 사용되는 자원의 낭비도 심해진다. TMMi의 측정방식을 적용하면, 테스트 조직의 성숙도 위치를 알 수 있으며, 매뉴얼을 따르면서도 더 정교한 평가를 통해 국방 소프트웨어의 신뢰도를 높일 수 있다. 신뢰도가 높은 소프트웨어는 결함복구 및 재시험에 사용되는 자원의 낭비를 줄일 수 있다. 물론 TMMi는 대상체계에 대해 기능 하나 씩 측정하지 않는다는 한계도 있다. 한계를 넘어 더 높은 레벨로 가기 위해서는 모든 이해관계자들의 참여와 최적화된 절차가 무엇인지에 대하여 끊임없는 논의가 있어야 한다. 그리고 세분화된 테스트 지침이 부족한 현실에서는 조직과 절차의 위치를 측정하는 활동을 가급적 많이 실행해야 한다. 앞으로 도입할 예정인 최신 무기체계에도 새로운 소프트웨어가 탑재될 확률이 매우 높다. 새로운 소프트웨어 운용을 극대화 하고 유지보수 비용을 줄이기 위해서는 논문에 제안된 절차 외에 다양한 아이디어가 나와 하나의 표준을 만드는 노력도 해야 할 것이라 본다.

References

- [1] Gregory Tasse, "The Economic Impacts of Inadequate Infrastructure for Software Testing", National Institute of Standards and Technology, May, 2002, pp.15.
- [2] Kidu Kim, Park Young Bom, R.Youngchul Kim, "Guideline for Test Process Improvement of Test Organization Through Correlating TMMi with TPI NEXT", Korea Information Process Society, Vol.2, No.12, pp.823-828, Dec. 2013.
- [3] Yeondae Jung, Jinsu Lim, Heebyung Yoon, "Present and Future of Advanced Defense Development Methodology", Communications of the Korean Institute of Information Scientists and Engineers, Vol.24, No.9, pp.75-80, Sep. 2006.
- [4] Erik van Veenendaal, "Test Maturity Model Integration(TMMi)", Version 2.0, TMMi Foundation, 2009, pp.10.
- [5] Jin-Wook Jang, "A Study on Implementation of Automobile Control Software Testing Process Based on TMMi", Korea IT Service Study, Vol.8, No.8, pp.161-162, Sep. 2009.
- [6] Erik van Veenendaal, "Test Maturity Model Integration(TMMi)", Version 2.0, TMMi Foundation, 2009, pp.11-12.
- [7] Erik van Veenendaal, "Test Maturity Model Integration(TMMi)", Version 2.0, TMMi Foundation,

2009, pp.7.

- [8] Jae-Min Kim, MND Component Based Development Methodology, Vol.1, Defense Acquisition Program Administration, 2010, pp.384.
 - [9] Kaner Cem, Falk Jack, Nguyen Hung Quoc, Testing Computer Software 2nd Edition, p.496, John Willy and Sons Inc, 1999, pp.25.
 - [10] Lockheed Martin, "A Digital Jet for the Modern Battlespace", Available From:
<https://www.f35.com/about/life-cycle/software>
(accessed Sep. 1, 2020).
-

박 연 동(Yeon-Dong Park)

[정회원]



- 2012년 2월 : 세종대학교 디지털 콘텐츠학과 (공학학사)
- 2012년 4월 ~ 2014년 4월 : 하만 코리아 사원
- 2014년 3월 ~ 2019년 6월 : 마그 네티마렐리 대리
- 2019년 12월 ~ 현재 : 국방기술품 질원 연구원

〈관심분야〉

소프트웨어 테스트, 소프트웨어 공학