**백서** | 2016년 4월

# 초기 단계로 이전에서 테스터로서 살아남기

테스트 권위자 및 컨설턴트인 Paul Gerrard가 일련의 문서를 통해 테스트와 관련된 여러 주제를 다룹니다. 초기 단계로 이전이란 프로세스의 초기 단계에 주로 테스트를 고려한다는 의미입니다. 이 문서에서 Paul은 초기 단계로 이전을 해야하는 이유, 테스트의 역할 변화, 그리고 애자일 세상이 우리에게 미치는 영향을 대해 이야기합니다.

Paul Gerrard
Gerrard Consulting

후원:
ca technologies

# 초기 단계로 이전에서 테스터로서 살아남기

## 배경

약 5년 전 필자는 테스트 비즈니스의 변화를 설명하기 위해 "재분배된 테스트"라는 용어를 만들었습니다. 사용자, 분석가, 개발자 및 테스터는 테스트에 대한 책임을 재분배하고 보다 효율적으로 협업합니다. 이러한 변화에는 테스트 활동(및 책임)을 초기 단계로 이전하는 과정이 포함되며 이 접근 방식에 보통 사용되는 용어가 "초기 단계로 이전"입니다.

초기 단계로 이전은 개발자가 자신의 테스트에 더 높은 책임감을 가진다는 것을 의미할 수 있고, 테스터가 초기 단계에 개입하여 요구 사항을 제시하고 BBD(행동 중심 개발) 프로세스를 통해 개발자에게 예제를 공급하는 것을 의미할 수도 있습니다. 또한 사용자와 BA가 개발자와 함께 테스트에 대해 완전한 책임을 진다는 의미일 수 있고, 테스트 팀과 테스터가 없다는 의미일 수도 있습니다. 물론 이에 대한 "정답"이란 없으며 모든 구성을 눈여겨보아야 합니다.

이 문서에서는 애자일 세상이 우리에게 미칠 영향과 테스트의 역할 변화에 대해 설명합니다.

## 새로운 개념이 아닌 초기 단계로 이전

필자가 테스트 비즈니스에 발을 들였을 때(1992년), 그리고 그 이전부터 테스트 분야에서는 "초기에 테스트하고 자주 테스트하라"라는 격언이 있었습니다. 필자의 친구이자 동료인 Paul Herzlich가 1993년에 도입한 W-Model[1]에서는 단계화된 프로세스의 모든 아티팩트(문서와 소프트웨어 모두)를 테스트할 수 있으며 자주 테스트해야 한다고 제안했습니다.

당시에는 하향식(Waterfall)이 지배적인 수명 주기 접근 방식이었지만 단계의 수나 지속 기간은 중요한 것이 아닙니다. 기본 원칙은 *소프트웨어 디자인 및 개발의 방향을 제시한 지식의 출처를 검증 또는 테스트해야 한다*는 것이었습니다. 단계화된 프로젝트에서는 이를 위해 공식 검토가 포함될 수 있습니다. 애자일 프로젝트에서는 테스터(또는 개발자, BA 또는 사용자)가 요구 사항 또는 스토리의 작성자에게 코드 작성 전에 구체적 예제와 토론을 주의 깊게 생각하기 위한 질문을 하는 시나리오(또는 예제)를 제안할 수 있습니다.

*초기 단계로 이전이란 프로세스의 초기 단계에 주로 테스트를 고려한다는 의미입니다.*

그렇다면 초기 단계에 필요한 사항이 테스터가 초기에 개입하고 거북한 질문을 던지는 것으로 국한될까요? 그렇게 간단할까요? 꼭 그렇지는 않습니다.

## 초기 단계로 이전이 필요한 이유

시장에는 몇 가지 변화가 일어나고 있으며 이로 인해 우리 업계에 새로운 행동이 요구되고 있습니다. 약 5년 전에 시작된 것도 있고 보다 최근에 시작된 것도 있습니다. 여기서 시작되었다는 의미는 무엇일까요? 시작되었다는 것은 성공의 증거를 보여주었으며 다른 사람에게 도입을 권장할 만큼 신뢰성 있는 새로운 접근 방식을 장려하는 충분한 수의 사람들이 있었음을 의미합니다. Geoffrey Moore는 이러한 접근 방식을 "경계 넘기"라고 설명했으며,[2] 이는 더 다양한 기업 및 IT 커뮤니티에서 실용적 대안이 되었습니다.

초기 단계로 이전 현상에 관련된 주요 변화는 다음과 같습니다.

1. BDD(행동 중심 개발) 접근 방식을 통해 개발자, 사용자/BA 및 테스터는 비즈니스 스토리를 사용하여 협업할 수 있습니다. BDD는 애자일 팀의 협업을 향상시키기 때문에 더 널리 도입되었습니다.

2. 또한 CD(지속적 배포)의 개념[3] 역시 5~10년 전부터 있었으며 대형 온라인 비즈니스에서 개척한 고도로 자동화된 빌드 및 릴리스 자동화 접근 방식에 그 뿌리를 두고 있습니다. 지금은 온라인 활동을 하는 대부분의 조직이 이를 도입하고 있습니다.
3. CD는 자동화를 통해 릴리스 프로세스를 체계화 및 가속화했습니다. 하지만 이전에는 빌드, 테스트 및 릴리스 프로세스가 느렸기 때문에 드러나지 않았던 프로덕션 배포 및 인프라 변화에서의 지연이 더 두드러지게 되었습니다. 데브옵스는 개발자가 특히 운영 담당자와 훨씬 더 긴밀하게 협업하는 문화적, 심적 변화입니다. 최근에는 새로운 도구가 거의 매일 나타나며 많은 공급업체가 데브옵스를 "차세대 혁신 기술"로 장려하고 있습니다. 이는 매우 흥미진진하며 역동적인 상황이라 할 수 있습니다.
4. 조직이 비즈니스를 관리하는 방식의 전환과 모바일 분야에서 시스템의 변화를 대표하는 것이 바로 SMAC(소셜, 모바일, 분석, 클라우드)입니다. 프로덕션 시스템 변화로 구현되는 비즈니스의 실험은 상당히 구체적으로 모니터링됩니다. 캡처된 "빅 데이터"를 처리하여 얻은 분석 결과를 토대로 비즈니스 의사 결정이 이루어집니다.

프로덕션 시스템을 자주 실험함으로써 "마케팅 속도에 맞춰" 비즈니스 혁신이 가능합니다. 실험은 2010년대에 가장 중요한 화두인 "디지털 전환"의 핵심 사항입니다. 현재 디지털 전환 또는 "디지털"은 가장 큰 관심(및 예산)을 받고 있습니다. 마케터는 더 많은 채널(대부분 모바일)을 통해 고객에게 더 빠르고 우수한 액세스를 제공할 것을 약속하고 있습니다.

필자가 작성한 문서 "디지털 전환, 테스트 및 자동화"[4]에서 디지털 혁명을 설명하고 몇 가지 대응을 제안하고 있으므로 참조하시기 바랍니다.

## 초기 단계로 이전이 테스터에게 주는 의미

초기 단계로 이전은 팀이 목표, 요구 사항, 디자인 또는 구현에 대해 질문하고 이를 이해하며 향상시키는 데 도움이 될 피드백을 제공할 수 있을 때마다 이러한 피드백을 반드시 제공해야 함을 의미합니다. 이러한 행동은 모두는 아니지만 많은 테스터의 후천적인 본능으로 나타납니다. 사용자, BA, 개발자, 그리고 전체 팀은 이러한 방식으로 피드백을 제공하고 수용할 준비가 되어야 합니다. 어느 정도 저항이 있을 수 있지만 전체적인 목표는 더 풍부한 정보에 근거한 더 나은 프로젝트를 실행하는 것입니다.

테스트의 초기 단계로 이전 상황에서 테스터는 어떤 역할을 할까요? 이 행동을 가장 쉽게 요약하자면 가능한 이른 시기에 "초기 단계에 개입"하는 것입니다. 아이디어, 요구 사항 및 특정 단계의 결과물이 프로젝트의 최종 결과가 갖는 가치에 크게 기여하는 모든 단계에서 협업하고 토론에 함께 참여하는 것입니다. 간단히 말하자면 테스터는 지식의 출처를 묻습니다. 즉, 해당 출처가 이해 관계자, 사용자, 개발자, 비즈니스 스토리, 문서 또는 통념 중 무엇인지를 묻습니다.

가장 일반적인 접근 방식은 "예제를 통해 질문"하는 것입니다. 모든 단계에서 이러한 예제를 테스트로 간주할 수 있습니다. 이러한 질문은 사용 후 바로 폐기되거나 테스트 자동화 또는 수동 검사에 명문화될 수도 있습니다. 이러한 예제는 인간의 사고에 대한 결함을 찾아내기 위해 단순히 전략적으로 사용되거나 개발자에게 개발자 테스트를 위한 아이디어나 촉발점으로 제공될 수 있습니다. 또한 사용자나 개발자가 더 나은 테스트를 만드는 방법을 이해하는 데 도움을 줄 코칭 지원 예제로 사용될 수도 있습니다.

소프트웨어 프로젝트는 지식 습득 프로세스로 설명되어 왔습니다.[4] 이 지식은 프로젝트 전체에서 수집되며 시간이 흐름에 따라 발전하기도 합니다. 초기 단계로 이전의 목표는 출처에 근접한 테스트와 질문을 통해 이 지식을 보증하고 코드에 확정되기 전에 신뢰할 수 있는지 확인하는 것입니다.

초기 단계로 이전은 테스트 우선주의 철학을 더욱 진전시킵니다. 애자일은 항상 협업과 신속한 피드백을 장려했으며 초기 단계로 이전은 궁극적인 빠른 피드백 접근 방식으로 간주할 수 있습니다.

이를 도입할 경우 초기 단계로 이전은 테스터의 작업 방식에 깊은 영향을 미칩니다.

## 테스터는 초기 단계로 이전을 어떻게 적용해야 하는가?

초기 단계로 이전은 단순한 유행이 아니며 우연히 찾아오는 것도 아닙니다. 테스터가 시스템 테스트 팀에서 근무하고 있다면 이것이 어떤 영향을 미칠까요? 테스터가 애자일 팀의 일원이라면 어떤 영향이 있을까요? 테스터는 과연 어떻게 대처해야 할까요?

우리는 애자일 프로젝트의 테스트 전략에 있어 한동안 초기 단계로 이전을 핵심 요소로 주장했었습니다. 애자일 컨텍스트에서 테스트 전략은 일련의 "애자일 개입"으로 간주할 수 있습니다. 모든 프로젝트에는 피드백을 수집하고 제시할 기회가 생기는 중요한 순간이 있습니다. 테스터는 이 중요한 순간에 집중하고 이때 기여를 할 준비가 되어 있어야 합니다.

필자는 최근 웨비나에서 이 접근 방식의 이면에 놓인 생각을 소개했는데,[6] 여기에서는 고객 사례 연구를 사용하여 이러한 개입이 일어날 수 있는 상황을 제시합니다. 자신의 프로젝트에서 자신만의 "중요한 순간"을 식별해야 하며 자신과 자신의 팀이 할 수 있는 선택을 식별해야 합니다. 예를 들어 개발자를 위한 단위 테스트를 작성하거나, 개발자가 시작할 수 있도록 예제를 제공하거나, 개발자의 테스트 역량을 향상시키기 위한 교육을 실시해야 합니까?

어떤 경우든 여러분의 역할은 바뀌게 될 것이 거의 확실합니다. 테스터는 생각만 하고 있는 것이 아니라 초기 단계로 이전 중일 수 있으며, 여러분은 개발자의 테스트 고용인이 될 수 있습니다. 이는 여러분이나 여러분의 프로젝트에 최선의 결과는 아닐 것입니다. 따라서 중요한 순간을 식별하고 여러분의 기여를 제안하며 팀과 협상할 것을 제안합니다. 테스트 작업에 대한 책임을 자진해서 맡는 것이 아니라 더 많은 테스트 리더십과 지침을 제공해야 합니다. 이러한 접근 방식을 취할 경우 팀에 많은 테스터가 필요하지 않으므로 여러분의 가치를 입증하기가 훨씬 더 쉬워질 것입니다.

## 테스트 리드/관리자라면 무엇을 해야 할까?

여러분이 현재 테스트 관리자 또는 테스트 리드인 경우, 관리 팀의 의도가 초기 단계로 이전하여 테스트 비용을 줄이려는 것이라면 여러분의 역할을 입증하기가 더 어려울 수도 있습니다. 조직이 이 방향으로 전환하고 있다면 장기적인 경력 결정을 내려야 할 수 있습니다. 5년 후 어떤 일을 하고 싶으신가요? 6개월 후는 어떤가요? 여기에는 다섯 가지의 폭넓은 선택이 있을 수 있습니다.

1. **기업에 테스트 및 보장 능력 제공** — 이해관계자와의 관계에서 먹이 사슬의 우위를 선점함으로써 여러분의 역할은 IT 프로젝트의 제어권을 갖고자 하는 비즈니스 리더에게 조언을 제공하는 것일 수 있습니다. 독립적 에이전트로서 비즈니스 우려 사항을 파악하고 프로젝트에서 이를 제시합니다. 프로젝트 리더십에게 조언 및 설득을 하고, 그들의 성과와 달성을 검토하고, 결과를 해석하여 이해 관계자에게 조언을 합니다.
2. **요구 사항 지식 관리** — 이 역할의 경우에는 시스템을 정의 및 구축하는 데 필요한 지식을 제어합니다. 자신의 중요한 기술을 이용하여 요구 사항 및 사용 중인 특징을 나타내는 예제의 정밀도와 명확성을 보장합니다. 소프트웨어를 합당하게 빌드 및 테스트할 수 있을 정도로 요구 사항을 신뢰할 수 있다면 비즈니스 및 개발자가 결정을 내리도록 돕습니다. 비즈니스 개념 및 데이터 항목의 사용에 대한 사전 및 용어집과 요구 사항을 관리합니다. 비즈니스 영향 분석 서비스를 제공합니다.
3. **테스트 마스터 되기** — 팀, 프로젝트 및 이해 관계자에게 보증 기능을 제공합니다. 위의 1과 유사한 역할이지만 보다 애자일 지향적 환경입니다. 여러분은 애자일 프로젝트를 투명하게 유지하는 전문 테스트 및 보증 전문가입니다. 현장 고객 및 제품 소유자와 긴밀하게 작업합니다. 프로젝트가 위험을 인식하고 대처하도록 도우며, 팀을 교육하고 조언을 제공하며, 테스트 활동을 관리하고, 경우에 따라 테스트도 수행합니다.
4. **데브옵스 마스터 되기** — 데브옵스 프로세스(자동화된 빌드, 테스트 및 배포 프로세스) 내/외부의 중요한 정보 흐름을 관리합니다 이 정보 흐름은 매우 중요합니다. 변화, 테스트 및 딜리버리의 제어를 가능하게 하는 흐름을 관리하는 데 사용되는 프로세스를 정의 및 감독할 수도 있습니다.
5. **아웃소싱/외부 팀 관리** — 이 경우에는 자신의 현장 테스트 팀을 떠나서 아웃소싱 또는 외부 공급업체로의 작업 이전을 관리합니다. 여러분은 정보 흐름 전문가로, 아웃소싱 테스트 팀과의 관계를 관리하고 이들의 성과를 모니터하며 결과를 보장합니다.

아직 초기 단계로 이전하지 않았다면 팀 내/외부를 모두 살펴보고 가까운 미래에 자신의 역할이 어떻게 바뀔 것인지를 생각해 보아야 합니다. 언젠가 여러분의 역할은 바뀌겠지만 그 역할을 어떻게 발전시켜 나갈 것인지를 선택할 수 있어야 합니다. 훌륭한 선택을 하시기 바랍니다.

# 저자 소개

Paul Gerrard는 컨설턴트, 교사, 저자, 웹마스터, 개발자, 테스터, 컨퍼런스 연설자, 로잉 코치 겸 발행자입니다. 그는 테스트 보장을 전문으로 하여 소프트웨어 테스트 및 품질 보증의 모든 측면에서 다양한 컨설팅을 수행해 왔습니다. 또한 유럽, 미국, 호주, 남아프리카 등지의 테스트 관련 컨퍼런스에서 기조 연설 및 강습을 진행했으며 이에 대한 상을 수상하기도 했습니다.

Oxford 및 Imperial College London에서 공부했으며, 2010년 Eurostar European Testing excellence Award, 2013년 The European Software Testing Awards(TESTA) Lifetime Achievement Award를 수상했습니다.

2002년에는 Neil Thompson과 함께 “Risk-Based E-Business Testing”을 집필했습니다. 2009년에는 “The Tester’s Pocketbook”을 집필했습니다. 2011년에는 Susan Windsor “The Business Story Pocketbook”을 공동 집필했고, 2014년에는 “Lean Python”을 집필했습니다.

2014년에는 더블린 EuroSTAR 컨퍼런스에서 프로그램 의장을 맡았습니다.

현재 그는 Gerrard Consulting Limited의 사장 및 TestOpera Limited의 이사이며 Test Management Forum의 운영자입니다.

메일: paul@gerrardconsulting.com
Twitter: @paul_gerrard
웹: gerrardconsulting.com

## 자세한 정보는 CA 테크놀로지스의 **개발 및 테스트**를 참조하십시오.


**ca.com/kr을 통해 CA 테크놀로지스를 만나 보십시오.**

CA 테크놀로지스(NASDAQ: CA)는 회사가 변화를 통해 애플리케이션 경제의 기회를 잡을 수 있도록 하는 소프트웨어를 만듭니다. 소프트웨어는 모든 업종, 모든 기업의 핵심입니다. 계획부터 개발, 관리 및 보안에 이르기까지 CA는 전 세계의 회사를 도와 모바일, 프라이빗 및 퍼블릭 클라우드, 분산 및 메인프레임 환경에서 생활, 거래 및 소통의 방식을 바꾸고 있습니다. 자세한 내용은 **ca.com/kr**을 참조하십시오.

참조 자료

1. “The W-Model,” http://blog.gerrardconsulting.com/?q=node/531
2. “Crossing the Chasm” 및 Geoffrey A Moore의 다른 저서, http://www.chasminstitute.com/
3. Continuous Delivery definition, Martin Fowler, http://martinfowler.com/bliki/ContinuousDelivery.html
4. “Digital Transformation, Testing and Automation,” Paul Gerrard의 블로그, http://blog.gerrardconsulting.com/?q=node/660
5. “The Laws of Software Process,” Philip G Armour
6. 웨비나: “Agile Test Strategy,” Paul Gerrard, http://blog.gerrardconsulting.com/?q=node/627


CS200-181168_0416