

비즈니스 백서

최신 애플리케이션 기능 테스트를 위한 서바이벌 가이드



애플리케이션을 현대화해야 하는 이유

Forrester 조사에 따르면, 애플리케이션 현대화 작업의 주된 비즈니스 관심사는 비즈니스 민첩성, 혁신 및 비용 절감인 것으로 나타났습니다.¹ “즉각적인” 것을 선호하는 시대에는 신속하게 방향을 전환할 수 있는 민첩한 기업만이 빠르게 변화하는 세상에서 성공할 수 있음을 의미합니다. 혁신은 이러한 성장을 가속화하고 경쟁 차별화 요소를 제공하는 핵심입니다. 비즈니스 애플리케이션의 경우, 고객에게 소셜 미디어에서 경험할 수 있는 기능을 제공하는 것이 한 예가 될 수 있습니다. 콘텐츠를 자유롭게 배치할 수 있고, 애플리케이션 및 다른 사용자와의 상호 작용을 경험하며, 동영상과 같은 매체를 콘텐츠에 내장하는 등의 기능을 예로 들 수 있습니다. 이러한 새로운 요소들을 빠르게 제공하면서 동시에 운영 비용의 절감과 운영 프로세스의 개선이 이루어져야 합니다.

이러한 비즈니스 요구 사항을 충족시키기 위해, 민첩성을 선호하는 기업은 클라우드 컴퓨팅, Web 2.0, 복합 애플리케이션 및 모바일 기술과 같은 새로운 기술 및 아키텍처는 물론 Agile 및 기타 반복적인 개발 프로세스 같은 새로운 방법을 채택하고 있습니다. 그러나 이처럼 중요한 새로운 IT 이니셔티브는 때로 기업에 업무상 혼란을 가져올 수도 있고, 품질 보증(QA) 부서에 있어 기능 테스트를 더욱 어렵게 만들 수도 있습니다.

요약

본 문서에서는 기능 테스트 팀이 새로운 IT 이니셔티브 문제를 해결하는 방법에 대해 살펴 볼 것입니다.

Agile 같은 반복적인 개발 프로세스는 보다 신속하게 테스트를 수행해야 할 뿐 아니라, 테스트 시 애플리케이션에 대해 제공되는 문서(정보) 또한 적습니다. 게다가 이러한 반복 프로세스는 그 동안의 예비 테스트 수행에 대한 요구를 증가시켜 왔는데, 이러한 예비 테스트에서는 테스트 설계와 테스트 실행이 동시에 수행됩니다.

복합 애플리케이션 아키텍처, 클라우드 컴퓨팅 및 모바일 플랫폼 같은 새로운 애플리케이션 기술은 저마다 해결해야 할 문제를 가지고 있습니다. 새로운 툴킷을 배우고 사용해야 하는 문제, 다양한 플랫폼 및 장치간의 검증용 수행해야 하는 문제, 클라우드 및 On-Premises 환경에 대한 테스트를 수행해야 하는 문제, 여러 개의 그래픽 사용자 인터페이스(GUI) 및 비 GUI 계층을 가지면서 서로 상호 작용을 하는 복잡한 애플리케이션에 대한 테스트를 수행해야 하는 문제 등이 바로 해결해야 할 문제들입니다.

이러한 문제를 해결하기 위해, QA 팀은 기능 테스트 수행 방법을 변경해야 합니다. Best Practice로는 애플리케이션이 테스트 할 준비를 마치기 전에 가능한 한 빨리 테스트를 시작하는 방법, 전체적으로 테스트 항목을 확인하는 방법, 그리고 사전에 효과적이면서도 언제든지 변경 가능한 테스트 전략을 수립하는 데 시간을 투자하는 방법 등이 있습니다. 또 테스트 생성 및 유지 보수 작업을 보다 효율적으로 진행할 수 있도록, 재사용 가능한 테스트 구성 요소를 포함한 프레임워크의 사용을 고려해 볼 수 있습니다. 수동 테스트의 방식을 현대화 하여 오류가 나기 쉬운 반복적인 작업을 최소화하고 테스트 및 결함에 대한 의사소통을 원활히 할 수 있습니다. 또한 테스트 자동화의 도움을 받을 수 있습니다. 중요한 것은 현대의 복합 애플리케이션의 검증을 위해서는 GUI 계층에 대한 테스트 및 비 GUI 계층에 대한 테스트, 그리고 완전한 엔드-투-엔드(end-to-end) 프로세스에 대한 테스트를 수행해야 하고 가능한 테스트를 자동화 해야 한다는 것입니다.

테스트 팀에 수많은 과제를 제시하는 IT 이니셔티브

끊임없는 기술 개발과 프로세스의 유입은 복잡성을 가중시키고 테스트 주기의 단축을 요구합니다. 그렇기 때문에 준비가 안 된 테스트 팀은 비즈니스 민첩성 및 시장 출시 시간 단축에 걸림돌로 보일 수 있습니다. 다음은 기능 테스트가 왜 이러한 IT 이니셔티브에 있어서 특히 어려울 수 밖에 없는지에 관한 몇 가지 이유를 알려 줍니다.

민첩성: 많은 IT 부서들이 현재 시장 출시 시간을 단축하고 변경 비용을 절감하기 위해 Agile이나 하이브리드, 또는 기타 반복적인 애플리케이션 개발 프로세스를 채택하고 있습니다. 이러한 프로세스는 피드백을 통합하고 장기 계획 단계에서 낭비되는 시간을 최소화할 수 있는 메커니즘을 제공합니다. 기능 테스트 방식은 Agile 및 Agile류 방식의 장점을 소멸시키지 않도록 적절한 변경이 필요합니다. 해당 테스트는 수명 주기 초반부에 이루어져야 하며, 종종 개발과 동시에 이루어지기도 합니다. 대부분의 경우, 일정에 따라 반복이 가능하도록 테스터와 개발자가 동시에 “협업” 방식으로 테스트에 참여 합니다. 테스터는 애플리케이션에 대한 최소한의 자료 밖에 제공되지 않았더라도 자신의 작업을 끝마쳐야 하며, 빠른 문제 해결이 가능하도록 계속 종합적인 결함에 대해 설명해야 합니다. 그리고 종종 완벽한 최종 버전이 아닌 부분적인 애플리케이션 및 기능을 테스트해야 할 경우도 있습니다.

이러한 문제들은 탐색적 테스트 (Exploratory Testing) 같은 방식의 구현을 촉구합니다. 테스터는 먼저 애플리케이션의 비즈니스 목표 및 그 활용 사례에 대해 확인하기 위해 주요 이해 관계자와 논의한 다음, 독창성이나 비판적 사고와 같은, 테스터로서의 지식과 기술을 활용해 신속하게 애플리케이션을 검사하고 결함을 찾아내야 합니다. 이러한 테스트 방식의 가장 큰 이점은 테스트 수행은 물론 테스트 설계 및 문서 작성 작업에서 많은 시간을 절약할 수 있다는 것입니다.

복합 애플리케이션: QA 부서는 리치 GUI, 비 GUI 비즈니스 논리 계층, 웨어드 서비스(중중 SOA 기반) 및 패키지 애플리케이션 통합을 비롯해, 복잡한 아키텍처가 포함된 현대적인 복합 애플리케이션을 테스트할 수 있도록 기술과 능력을 개발해야 합니다. 또 GUI/비 GUI 서비스 및 구성 요소 확인 뿐 아니라, 여러 개의 복합 애플리케이션 계층을 아우르는 비즈니스 프로세스의 통합 테스트 시나리오를 시각화하고 확인 및 보고하는 일도 필요합니다. 그러나 비 GUI 구성 요소를 테스트하는 일은 말처럼 쉬운 일이 아닙니다. 애플리케이션 GUI를 통해 이를 테스트한다는 것은 애플리케이션 수명 주기의 후반부까지 기다렸다가 버그를 탐색한다는 의미로 많은 비용을 요구합니다. GUI를 통한 테스트는 논리 계층의 기능 하위 집합만 테스트하게 될 가능성이 있어, 애플리케이션 어딘가에 결함을 남겨 두거나 이를 실행할 때 다른 애플리케이션에서 결함이 발생할 수 있습니다. 그리고 비 GUI 구성 요소 및 서비스를 테스트해야 할 때, 현재 테스트 도구가 필요한 기술을 지원하지 못한다는 사실을 발견하게 될 지도 모릅니다. 더불어 비 GUI의 인터페이스 및 예상 작동 방식은 GUI 기반 구성 요소와 비교해, 이어나 정리하기가 어렵습니다.

예를 들어, 예금을 하고 데이터베이스 호출을 처리하며, 고객에게 예금 업무가 성공적으로 처리되었음을 알리는 텍스트를 발송하는 은행 업무라면 어떻게 테스트해야 할까요? 이러한 프로세스 중 너무나 많은 수가 GUI를 통해 기능을 제공하지 않을 것이며, 결국 다른 방법으로 확인할 수 밖에 없을 것입니다. 그리고 고객에게 텍스트를 전송하는 서비스가 과도한 인출액에 대한 공지를 전송하는 작업에도 사용된다면, 아직 발견하지 못한 추가 결함이 있진 않을까요?

다양한 애플리케이션에서 수많은 구성 요소들을 사용하기 때문에 테스트 팀에서 구성 요소와 이러한 구성 요소가 사용하는 애플리케이션 간의 종속성을 이해하는 것은 매우 어려운 일일 수 있습니다. 특히 일부 서비스가 작동되지 않거나 타사가 제공한 검증되지 않은 서비스일 경우엔 더더욱 그렇습니다.

많은 기업은 복잡한 복합 애플리케이션을 완벽하게 테스트하는 데 필요한 모든 테스트 사례 변경 사항을 기반으로 테스트 자동화를 시도할 것입니다. 그러나 종종 테스트 자동화에 필요한 작업을 과소 평가하는 경우가 있어, 기본적으로 몇 가지 문제를 야기 시킵니다. 자동화는 소프트웨어 개발로써, 설계, 확인 및 유지 보수에 대한 숙련된 리소스를 갖춘 적절한 인력을 필요로 합니다. 뿐만 아니라 모든 또는 대부분의 필수 테스트 시나리오에 맞는 도구 집합을 찾는 일도 어려울 수 있습니다. 각각의 다양한 기술이나 애플리케이션에 맞게 특화된 도구를 사용하면 도구 비용과 교육 비용이 늘어날 수 있고, 기업 전반을 표준화하는 능력이 제한될 수 있습니다. 그리고 테스트 스크립트의 작성자가 바뀌는 문제가 있어, 간혹 왜 그 테스트가 중요하고 충족시켜야 할 요구 사항이 무엇인지도 알지 못한 채 몇 년씩 진행되기도 합니다.

자세한 내용은 클라우드 및 Web 2.0 부분을 참조하십시오.

모바일: 모바일 인터페이스로 확장되는 엔터프라이즈 애플리케이션은 PC 애플리케이션만큼 철저하게 테스트해야 합니다. 품질 기대치가 자신의 개인 스마트폰 애플리케이션에 맞춰져 있는 사용자는 적절하게 작업을 수행하지 못하는 모바일 엔터프라이즈 애플리케이션을 견뎌내지 못합니다. 따라서, 이러한 모바일 엔터프라이즈 애플리케이션의 기능이 정확하게 작동하고 있는지를 테스트하는 것은 생산성 및 고객 충성도에 매우 큰 영향을 미칩니다. 모바일 애플리케이션은 복합 애플리케이션 테스트의 모든 복잡한 문제를 다 안고 있으면서도, 데스크톱 환경을 전혀 따라가지 못합니다. 자체적인 사용자 인터페이스 요구 사항과 비즈니스 프로세스 흐름 및 인프라 종속성을 갖고 있습니다. 그리고 지원해야 할 장치가 너무 많아서 특히 QA 부서에 많은 문제를 야기합니다. Forrester 조사에 따르면 “고객은 다양한 개발 환경을 기반으로 다양한 장치를 보유하고 있습니다. 동일한 OS의 다양한 버전, 서로 다른 화면 크기 그리고 엄청난 수의 장치는 결국 애플리케이션 포팅, 유지 보수 및 프로모션 비용이 매우 높을 것이란 사실을 의미합니다.”² 수동 테스트는 스마트폰, PDA, 태블릿 및 렌트카 회사의 스캐닝 및 인쇄 핸드헬드 장치 같은 전문적인 장비를 통해 이루어질 경우 매우 노동 집약적인 작업이 될 수 있습니다. 이러한 현상은 엄청난 데이터, 사용 시나리오, 모바일 네트워크 환경 및 캐리어의 변화를 고려할 경우 더욱 심각해집니다. 물론 구현 자체가 문제가 될 수도 있지만, 이는 테스트 자동화 솔루션에 있어 매우 발전된 분야입니다.

클라우드 애플리케이션

종종 기능 테스트 팀은 클라우드를 기반으로 하는 일부 서비스가 포함된 복합 애플리케이션을 테스트해야 할 경우가 있습니다. 클라우드 계층을 비롯해 전형적인 복합 애플리케이션을 테스트하려면 테스트 팀에서 교차 인프라 테스트 프로젝트에 액세스하여 이를 관리할 수 있어야 합니다.

뿐만 아니라 테스트를 목적으로 특정 클라우드 기반 서비스에 액세스하는 데 어려움을 겪을 수도 있습니다. 예를 들어, 온라인으로 구두를 주문할 수 있는 복합 애플리케이션은 신용 카드 처리 및 배송 서비스 같은 클라우드 기반 서비스를 포함시켜야 할 것입니다. 그러나 신용 카드 및 배송이 이미 제공되고 있는 타사 서비스이기 때문에 테스트를 할 수 없거나 테스트에 많은 비용이 들 수 있습니다. 이런 문제는 복합 애플리케이션 테스트를 지연시킬 수 있습니다.

반면 클라우드에 대한 액세스는 애플리케이션 테스트를 위해 주문형 방식의 거의 무제한에 가까운 능력을 제공하기 때문에 테스트 팀에게는 매우 좋은 기회를 제공합니다. 특히 성능 테스트에 매우 큰 도움이 됩니다.

민첩한 기업을 위한 테스트 현대화 방법

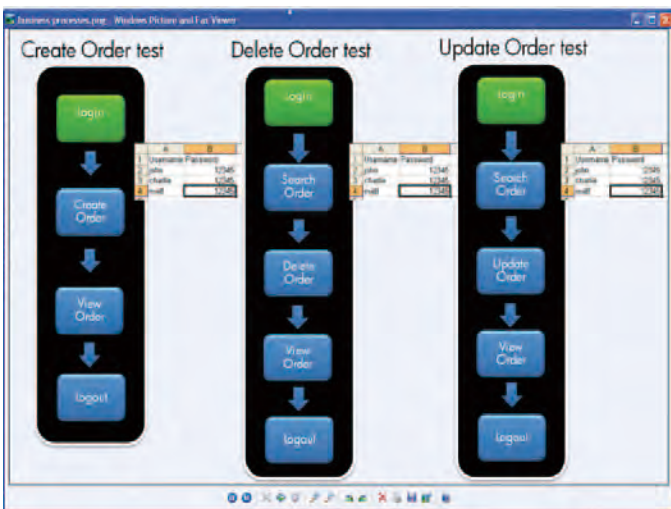
QA 부서는 이러한 문제를 해결하고 오늘날의 현대적인 애플리케이션 구현 작업에 완벽한 파트너가 될 수 있습니다. QA 부서는 기술 집합을 확장하고 테스트 의뢰 항목이라는 업무적 측면에 더 관심을 기울임으로써 스스로를 현대화할 수 있습니다. QA 부서는 각종 지표와 결과를 공유하여, 개발 및 비즈니스 분석 팀과 파트너 관계를 형성해야 합니다. 그리고 보다 예측 가능한 애플리케이션 지원으로 고객 만족도를 높이고 결함을 줄일 수 있도록 품질 관리에 힘써야 합니다. 이러한 모든 일은 수명 주기 초반에 테스트를 진행할 수 있는 프로세스 및 방법을 채택하고, 가능한 모든 분야에서 자동화 기능을 향상시키며, 상세한 테스트 설계 문서 정보 없이도 원활하게 테스트를 진행할 수 있게 될 때 실현할 수 있습니다. 대개 미래 지향적 사고를 추구하는 QA 부서는 전체론적인 측면에서 테스트 방식을 검토하고 테스트 설계에 투자하여 다음과 같은 결과를 가져옵니다. 그 결과 보다 많이, 보다 빨리, 그리고 보다 이른 시점에 테스트할 수 있게 됩니다.

다음은 기능 테스트 팀이 현대화를 위해 실천할 수 있는 몇 가지 과업입니다.

1. 설계부터 시작: 비록 이 과정이 종종 무시되기도 하지만, 시간을 들여 주의 깊게 테스트 전략을 수립하면 보다 효율적이고 쉽게 테스트를 유지 관리할 수 있습니다. 비즈니스 프로세스 모델과 플로우 형태를 정리할 수 있는, 재사용 가능한 테스트 구성 요소 라이브러리를 구축하는 프레임워크 사용을 권장합니다.

예를 들어, 주문 생성, 삭제, 업데이트라는 3개의 비즈니스 프로세스를 통해 확인해야 한다고 가정해 보겠습니다. 일단 비즈니스 프로세스를 모델화하면, 그림 1에서처럼 3개의 프로세스 모두가 동일한 로그인 기능을 사용하게 되어 중복 문제를 쉽게 확인할 수 있습니다. 또 모듈식 방식으로, 로그인 기능에 대한 테스트 구성 요소를 3번이 아닌 한 번만 생성해주면 되기 때문에 작업 중복을 피할 수 있습니다. 그리고 로그인 기능에 대한 변경 사항이 있을 때는 관련 테스트 구성 요소만 한차례 변경해 주면 해당 구성 요소를 사용하는 모든 테스트가 업데이트 됩니다.

그림 1: 재사용 가능한 테스트 구성 요소를 사용해 비즈니스 프로세스를 확인함으로써 테스트 생성 및 테스트 유지 보수 시간을 단축합니다.



이러한 “구성별 테스트” 방식은 테스트 생성 및 유지 보수 작업에 드는 시간과 노력을 줄일 수 있습니다. 그리고 이 놀라운 효율성은 테스트의 자동화나 수동 실행 여부에 상관 없이 실현할 수 있습니다.

HP BPT(HP Business Process Testing) 소프트웨어는 재사용 가능한 테스트 구성 요소 라이브러리를 생성하는 데 필요한 프레임워크를 제공합니다. 테스트 팀이 테스트 생성과 유지 보수에 필요한 시간을 줄임으로써 애플리케이션 품질을 향상시킬 수 있습니다. 이는 정기적인 변경 작업과 급격한 변화에 있어 품질 비용을 절감합니다.

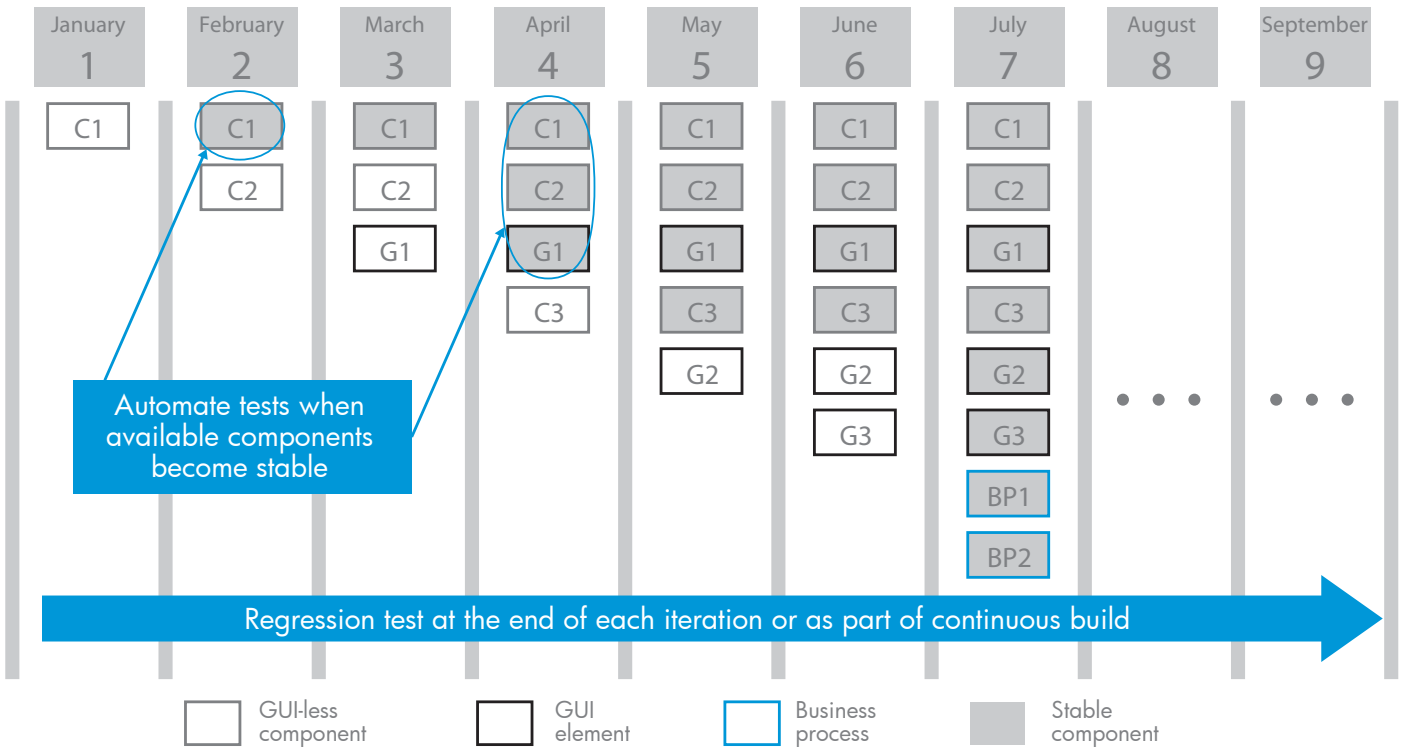
HP BPT를 통해 사전에 미리 테스트 전략을 세움으로써 얻을 수 있는 또 다른 이점은, 신속하게 수동으로 테스트할 구성 요소와 자동으로 테스트할 구성 요소를 결정할 수 있는 능력입니다. 대부분의 경우 구성 요소는 수동으로 테스트되기 시작하고, 필요에 따라 자동 테스트로 전환됩니다. HP BPT는 테스트 계획 문서 자동 작성과 HP Quality Center 소프트웨어 요구 사항에 대한 세부적인 연관 관계 형성이라는 추가 기능을 제공합니다. 잘 정리되고 관리된 테스트는 변화된 상황 및 반복적인 향후 테스트 작업에도 테스트 담당자가 계속 사용할 수 있도록 해줍니다.

Web 2.0 애플리케이션

RIA(Rich Internet Application) 및 Web 2.0 기술은 풍부한 GUI를 지원하지만 한편으로는 테스트를 수행하는데 어려움을 주기도 합니다. 기능 테스트 도구는 Microsoft® Silverlight™ 같은 다양하고 확장적인 툴킷과 Dojo Toolkit, Yahoo! User Interface(YUI), ASP.NET AJAX 및 Google® Web Toolkit(GWT)을 비롯한 광범위한 Ajax 기술을 테스트할 수 있어야 합니다. 그러나 이 도구가 대부분의 기본적인 웹 인프라를 변경하지 않기 때문에, 각 툴킷은 고유한 방식을 가진 서로 다른 웹 컨트롤을 갖게 되고 그 결과 기능 테스트 도구에서 자동으로 인식하지 못할 수도 있습니다. 게다가 애플리케이션 간의 공유 데이터는 물론 데이터의 동적 특성까지 가세하면 매우 복잡한 테스트 시나리오가 탄생할 수 있습니다.

그림 2:

수동으로 현재 반복되는 작업을 테스트하고 과거 반복된 작업의 테스트 과정을 자동화하는 것도 좋은 방법입니다.



2. 자동화 방식의 증가 및 개선: 비록 자동화가 수동 테스트를 모두 대체할 순 없지만 자동화할 수 있는 부분을 찾아 내면, QA가 많은 테스트 작업을 보다 신속하게 진행할 수 있도록 도움을 줄 것입니다. QA 팀에서 제한된 리소스로 많은 구성 및 반복 작업을 테스트할 수 있게 함으로써, 자동화는 실제 환경에서 쉽게 장애 발생 위험을 줄일 수 있는 테스트 범위를 향상시킵니다.

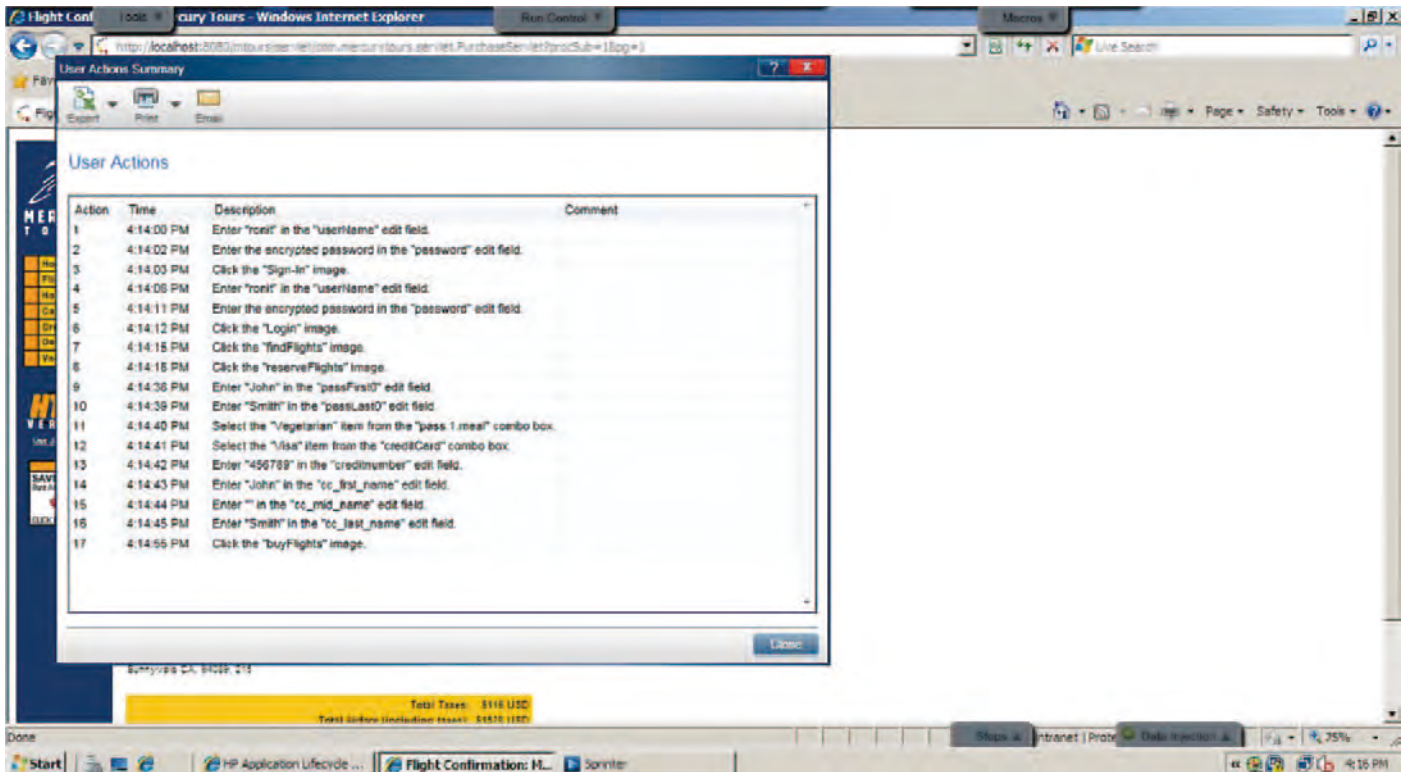
자동화 기능을 최대한 활용하려면 이를 적절하게 적용해야 합니다. 다음은 자동화할 항목과 시기에 대한 권장 사항입니다.

- 회귀 테스트 - 회귀 테스트는 수동 테스트와 함께 사용되지만 쉽게 자동화되는 “기본” 테스트입니다. 새 릴리스에서 버그를 찾아내기 보다는 최신 변경 사항이 이전 항목에 문제를 발생시키지 않도록 하기 위한 테스트입니다. 자동화 기능 사용하여 하루 24시간 계속 실행되는 빠른 반복 실행 회귀 테스트 집합을 생성할 수 있습니다. 이러한 자동 회귀 테스트와 빌드 프로세스를 통합하면 신속하게 문제를 발견하고 개발자들이 항상 잘 알려진 안정된 기반에서 작업할 수 있는 “지속적인 빌드” 방식을 활성화할 수 있습니다.
- 과거 반복 작업 - Agile과 같은 반복 방식에서는 1회 반복이나 실행으로 결과를 얻거나 자동화하는 것이 어려울 수 있습니다. 종종 시간이 충분하지 않고 코드가 유동적일 가능성이 있습니다. 회귀 테스트와 비슷하게 현재 반복되는 작업은 수동으로 테스트하고 이전 반복 작업의 테스트는 자동화하는 것이 바람직합니다(그림 2 참조). 이는 QA 팀으로 하여금 테스트를 조기에 빠르게 수행할 수 있게 하여 Agile 프로세스에 꼭 필요한 두 가지 요소를 제공합니다.

- 보다 안정적인 애플리케이션 및 구성 요소 - 자동화에 대한 투자 수익률(ROI)을 분석하는 것이 중요합니다. 자동화의 이점은 시간이 지남에 따라 자동화 스크립트 개발에 투자했던 비용과 유지 관리에 드는 비용을 상쇄합니다. HP Unified Functional Testing 소프트웨어 같은 일부 도구는 여러 가지 변화를 대상으로 테스트를 강화하는 내장 기능을 갖추고 실시간으로 스크립트를 업데이트함으로써 테스터에게 지침을 제공하지만, 유동적인 상태의 애플리케이션은 자동화를 비용 효과적인 방식으로 만들기에는 자동화 스크립트에 대한 변경 사항이 지나치게 많을 수 있습니다.
- 모바일 애플리케이션 - 비록 오늘날 대부분은 수동 방식으로 테스트되고 있지만, 장치, 운영 체제, 사용 시나리오 및 모바일 애플리케이션에 필요한 네트워크 환경의 변화에 대한 테스트는 자동화하는 것이 유용합니다.

그림 3:

HP Sprinter는 텍스트 형식으로 모든 사용자 행동을 자세한 시간과 함께 자동으로 기록합니다.



HP UFT(HP Unified Functional Testing)는 테스트 자동화에 필요한 모든 주요 소프트웨어 애플리케이션 및 환경을 지원합니다. 최첨단 혁신 기술 기업과의 파트너십을 통해, HP는 일상적으로 사용하는 운영 체제 및 플랫폼 상의 모바일 애플리케이션 및 클라우드 인프라에 대한 업계 선두의 기능 테스트 능력을 확장해 가고 있습니다. HP UFT에는 고급 서비스 테스트 및 관리 기능을 위한 HP Service Test 및 HP Service Test Management 소프트웨어와 함께, 잘 알려진 HP QuickTest Professional 자동화 소프트웨어가 포함되어 있습니다.

Web 2.0 애플리케이션을 위해 HP UFT는 Microsoft Silverlight 및 가장 일반적으로 활용되는 Ajax 툴킷(Google Web Toolkit (GWT), Yahoo! User Interface (YUI), ASP.NET AJAX 및 Dojo Toolkit)을 비롯해, Web 2.0 기술을 활성화하는 다양한 툴킷에 대한 지원 기능을 탑재하였습니다. Web Add-in Extensibility를 통해서는 Web Add-in이 바로 지원하지 못하는 타사 및 맞춤형 웹 컨트롤에 비해 테스트 지원 기능을 개발할 수 있습니다. 그리고 Extensibility Accelerator는 바로 지원되지 않는 타사 및 맞춤형 웹 컨트롤을 위한 지원 기능의 설계, 개발 및 구현을 촉진하고 가속화하는 통합 개발 환경(IDE)입니다.

3. 수동 테스트의 현대화: 갈수록 많은 기업이 자동화 테스트를 활용하기 위해 노력하고 있지만, 모든 기능 테스트의 약 75% 정도는 여전히 수동 테스트로 이루어집니다. 따라서 테스터는 이런 상황에서 가장 지루하고 오류 발생 가능성이 높은 작업을 간소화하고 최소화하기 위해 현대적인 수동 테스트 방식을 도입할 수 있습니다. 스프레드시트의 데이터를 수동으로 입력하고,

테스트가 실행되는 동안 관련 내용을 기록하며, 테스트 제품에 결과를 기록하고 다른 환경 및 조건을 대상으로 끊임없이 테스트를 실행하는 대신, 이러한 과정을 간소화하는 테스트 솔루션을 모색해야 합니다.

HP의 새로운 수동 소프트웨어 테스트 솔루션인 HP Sprinter는 수동 테스트의 간소화 및 신속한 진행과 팀의 협업 증진에 초점을 맞춥니다. HP Sprinter는 자동으로 HP Quality Center에서 직접 테스트 사례를 기록하고 이를 화면에 표시합니다. 그리고 오류 발생 프로세스를 최소화하기 위해 외부 스프레드시트의 데이터를 자동으로 테스트 대상 애플리케이션에 입력합니다.

HP Sprinter는 HP Quality Center 또는 HP ALM(HP Application Lifecycle Management) 소프트웨어에서 자동으로 테스트 사례를 기록하고 이를 "application under test(테스트 중인 애플리케이션)" 화면에 표시합니다. 이 화면에서 사용자는 테스트 과정을 살펴 보고, 결함을 확인하며, 의견을 추가하고 실제 결과를 기록할 수 있습니다.

HP Sprinter는 다음과 같은 3가지 형식으로 모든 사용자 행동을 자동 기록합니다. 텍스트 설명 방식(사용자 행동, 그림 3 설명), 화면 캡처 방식(사용자 행동), 비디오 녹화 방식(전체 테스트 실행 과정). 테스터는 버튼 클릭만으로 새로운 결함에 이처럼 상세한 설명을 덧붙일 수 있습니다. 이는 결함에 대한 개발자와의 대화를 크게 개선하여 신속하게 문제를 수정할 수 있게 합니다.

모바일 애플리케이션을 포함하여 수동으로 테스트되는 애플리케이션의 경우, HP Sprinter 소프트웨어의 “미러 테스트” 기능이 하나의 테스트를 여러 차례 다양한 환경 및 구성으로 복제하는 데 도움을 줍니다. 따라서 HP Sprinter는 동시에 최대 4개의 기기에 이를 복제할 수 있기 때문에, 테스트는 단 한번만 테스트를 실행할 수 있습니다. 이는 전체 수동 테스트의 생산성을 50-70%까지 향상시키는 것으로 나타났습니다.³

그리고 HP Sprinter, HP BPT 및 HP UFT를 사용함으로써 HP ALM 또는 HP Quality Center에 테스트 결과 및 모든 관련 문서가 자동으로 저장됩니다. 그리하여 테스터는 물론 다른 이해 관계자에게도 단일 기록 시스템의 모든 테스트 사례, 요구 사항 및 결함에 대한 완벽한 테스트 결과를 제공할 수 있습니다.

4. 서비스 및 비 GUI 구성 요소 테스트: 복합 애플리케이션에서는 기본 논리 확인 작업을 GUI 테스트에만 의존하기 보단, 직접 서비스 계층(headless layer라고도 함)을 테스트해 보는 것이 좋습니다. 이러한 서비스는 간혹 GUI가 준비되기도 전에, QA에서 애플리케이션 수명 주기 초반에 테스트 및 자동화를 실행할 수 있도록 테스트를 진행합니다. Agile 프로세스 사용 여부에 상관없이 이는 큰 이점을 제공합니다. 분리되어 있는 구성 요소를 처리할 때 결함 위치를 보다 쉽게 잡아낼 수 있고, 적용되고 얼마 되지 않아 발견된 경우 버그를 보다 쉽게 수정할 수 있습니다.

점점 더 많은 애플리케이션 비즈니스 로직이 서비스 계층에 위치하게 되고, 다양한 애플리케이션에 의해 사용되고 있습니다. 따라서 이러한 서비스에서 미처 발견하지 못한 결함은 여러 곳에 전파되고 예상치 못한 악영향을 미칠 수 있습니다. 이러한 서비스를 독립적으로 테스트하면 위험이 줄어들고, 복합 애플리케이션의 구성 요소가 강화됩니다.

그리고 제공되는 구성 요소가 갈수록 늘고 있기 때문에, 다계층적인 복합 애플리케이션은 GUI 및 비 GUI 구성 요소를 통한 전체 과정을 자동화하는 방식으로 테스트하는 것이 좋습니다. HP UFT를 활용하면 QA 팀에서 GUI 계층, 서비스 계층 및 엔드-투-엔드(end-to-end) 프로세스를 테스트할 수 있습니다. 이는 여러 애플리케이션 계층을 대상으로 더 많은 기능을 테스트함으로써 복합 애플리케이션의 품질을 향상시키는 데 도움이 되며, GUI만을 통한 테스트에서는 놓쳤을 수도 있는 결함을 발견하는데도 도움이 됩니다.

클라우드 애플리케이션에서는 클라우드나 On-Premises의 구성 요소를 테스트해야 합니다. HP SOA Systinet는 시뮬레이션된 서비스를 비롯해, 실제/복제 애플리케이션 및 서비스의 중속성을 파악하기 위해 메타데이터 리포지토리로 사용할 수 있습니다. HP SOA Systinet 소프트웨어와 HP UFT를 함께 사용하면, QA 팀에서 클라우드 애플리케이션의 엔드-투-엔드(end-to-end) 프로세스를 테스트할 수 있습니다.

또한, 복합 애플리케이션 테스트나 다단계 비즈니스 프로세스 테스트 시나리오에 대해 제공되지 않는 구성 요소가 있을 경우에는 지연 현상이 발생할 수 있습니다. 이러한 현상은 공유되는 구성 요소가 이미 실행 중에 있는 비즈니스 크리티컬 서비스, 또는 애플리케이션이거나 테스트에 많은 비용이 드는 클라우드 서비스일 경우 일어날 수 있습니다. 서비스 가상화는 이러한 테스트 지연 현상을 예방하는 데 사용할 수 있는 방법입니다. 서비스 가상화를 활용하면, 테스트의 백엔드에 있는 서비스 또는 애플리케이션 구성 요소가 에뮬레이트되고, 실제 서비스의 행동 및 반응을 모방한 가상 서비스가 생성됩니다. 가상 서비스는 클라우드 서비스 비용을 발생시키거나 실제 환경에 영향을 미치지 않으면서 테스트에 반복적으로 활용할 수 있습니다.

기능 테스트 팀을 위한 기능 요약

QA 팀에서 클라우드 컴퓨팅, Web 2.0, 모바일 및 복합 애플리케이션의 채택 등 새롭게 떠오르는 다양한 시장 요구 사항을 해결하기 위해서는 반드시 기능 테스트 수행 방식에 변화가 필요합니다. 민첩성을 선호하는 기업의 경우, 성공적인 기능 테스트 팀들은 애플리케이션 수명 주기 초반에 테스트 프로세스를 시작합니다. 심지어 애플리케이션이 테스트될 준비가 안 된 상태에서조차, 전체적인 테스트 문제를 확인하고 시간을 들여 테스트 전략을 수립합니다. 그러한 테스트 전략에는 프레임워크의 사용 및 재사용 가능한 테스트 구성 요소 생성이 포함됩니다. 이들은 적극적으로 업무에 관여하고, 라이프 사이클 초반에 테스트를 포함시키며, 자동 문서 작성 및 자동 보고 기능을 비롯한 다양한 현대적인 수동식 테스트 방식을 활용해 예비 테스트를 수행합니다. 그리고 심지어 GUI가 제공되기 전부터 서비스 계층의 테스트를 시작합니다. 미래 지향적인 QA 팀은 상황이나 대상이 적절할 경우 항상 테스트를 자동화할 수 있는 기회를 모색합니다. 그리고 수동식 테스트를 지속하는 분야에 대해서는 지루하고 반복적인 테스트를 자동화하여, 프로세스를 가속화하고 테스트 시간을 단축합니다. 끝으로 GUI 계층, 서비스 계층 및 엔드-투-엔드(end-to-end) 프로세스를 빠짐없이 테스트함으로써 복합 애플리케이션을 완벽하게 점검합니다.

HP는 이러한 모든 과정을 지원하는 통합된 포괄적인 기능 테스트 솔루션을 제공합니다.

다음과 같은 HP 솔루션의 무료 평가판을 다운로드하면 HP 기능 테스트 솔루션에 대해 보다 자세히 알 수 있습니다.

HP Unified Functional Testing

HP Sprinter

참고 사항

- ¹ 모집단: 예산이 책정된 모더니제이션 작업에 대한 통찰력을 갖춘 206명의 IT 의사 결정자.
출처: Clearing Your Path to Modern Applications and Business Agility (애플리케이션을 모더니제이션하고 비즈니스 민첩성을 확보하는 방법), Forrester Research, 2010년 4월 6일. HP의 의뢰를 받아 Forrester Consulting이 작성한 리더십 백서
- ² Forrester, 2011 Mobile Trends(2011년 모바일 동향), Thomas Husson 및 Julie A. Ask, 2011년 1월 24일
- ³ TechValidate에 다양한 고객 추천 사례는 <http://techvalidate.com/portals/hp-sprinter>를 참조하십시오.

Get connected

hp.com/go/getconnected

기술 동향, 지원 사항 및 HP 솔루션에 대한 심층 분석 받아보기



공유하기

© Copyright 2011 Hewlett-Packard Development Company, L.P. 본 안내서의 내용은 사전 통지 없이 변경될 수 있습니다. HP 제품 및 서비스에 대한 유일한 보증 사항은 제품 및 서비스와 함께 제공되는 보증서에 명시되어 있습니다. 본 안내서에는 어떠한 추가 보증 내용도 들어 있지 않습니다. HP는 본 안내서에 대한 기술상 또는 편집상의 오류나 누락에 대해 책임을 지지 않습니다.

Microsoft는 Microsoft Corporation의 미국 등록 상표입니다.

4AA3-5237KOP, 2011년 6월 작성, 2012년 4월 업데이트, 개정판 1

