2013. 6. 17. [제56호]

테스트 진행단계를 어떻게 측정할 것인가: 차트와 그래프 기반의 대시보드 활용방법

소프트웨어공학센터 경영지원TF팀

Contents

▶ 서론
▶ 1. 테스팅은 언제 완료될 것인가
▶ 2. 테스터는 얼마나 진행하고 있는가
▶ 3. 개발자는 얼마나 진행하고 있는가
▶ 4. 시간 경과에 따른 결함 동향
▶ 5. 결론

정보통신산업진흥원 부설 nipa 소프트웨어공학센터

Key Message

모든 테스트 관리자는 중요한 두 가지 질문에 맞닥뜨리게 되는데 첫째는 "현재 테스트의 진행단계는 어디인가" 이며, 둘째는 "얼마나 더 테스트를 진행해야 하는가" 임. 이러한 질문들에 대한 답을 구하기 위해 차트와 그래프 기반의 테스트 프로젝트 대시보드 활용 사례를 제시함.

▶ 서론

- 모든 테스트 관리자가 답변을 해야 하는 두 가지 중요한 질문이 있음
  - 테스트 프로젝트의 현재 상태는 어떠한지 그리고 얼마나 더 테스트를 해야 하는지에 대한 사항임
  - 테스트 진행 상황을 측정하기 위해 차트와 그래프를 사용하여 대시보드를 작성한다면 위의 두 질문에 정확히 대답할 수 있음
  - 관리자가 일이 어떻게 진행되고 있는지를 질문하기 위해 왔을 때 대시보드는 빠른 응답을 제공함
- 테스트 관리자들은 테스트 프로젝트의 진행 상황을 추적함에 있어 불량률에 너무 많이 의존하고 있음. 결함 숫자는 중요하지만, 그것들이 전체를 말하는 것은 아니기 때문에 테스트 프로젝트가 진행하는 방법에 대한 큰 그림을 만들려면 테스트 관리자는 측정하고 보고해야 함
  - 소프트웨어의 특정기능이 테스트 과정을 통해 진행하는 방법
  - 테스트 실행 및 테스트 통과의 수와 연계된 계획된 테스트의 수
  - 제한된 결함과의 관계 내에서 결함회귀비율(The fault feedback ratio, FFR)[1)]
  - 발견된 새로운 결함 및 제한된 결함과의 관계 내에서 누적 결함의 동향
- 여기서는 테스트 대시보드에 추가할 수 있는 네 개의 샘플 차트와 그래프를 제시하고, 이러한 측정들이 중요한 이유와 테스트 프로젝트 진행 방법의 큰 그림을 그려볼 수 있는 방법을 설명할 것임

1) 모든 버그 수정 중에서 잘못 수정된 비율

▶ 1. 테스팅은 언제 완료될 것인가

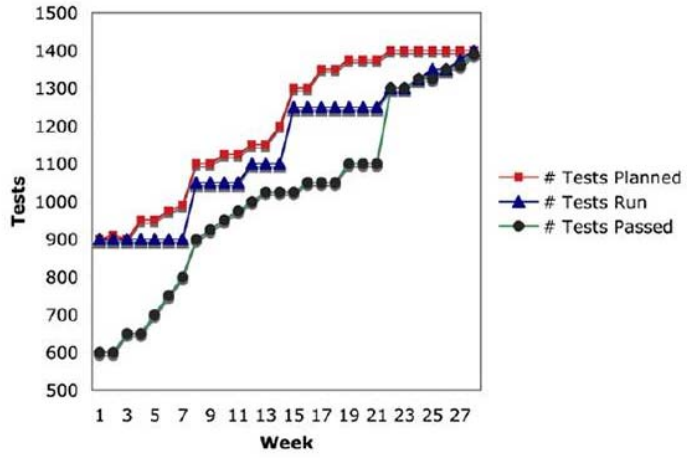
그림 1_ 테스팅은 언제 완료될 것인가

FEATURE OR AREA OR MODULE	LAST TEST DATE	STATE	NEXT PLANNED TEST
Feature Set 1	March 3 Build 145	Pass	Build 150
Feature Set 2	March 3 Build 146	Pass	Build 150
Performance Scenario 3	March 1 Build 140	Fail. Waiting on Jeff and Andy to fix.	Build 150
Feature 14	March 2 Build 142	Passed all regression tests	Build 147 for customer acceptance
Performace Scenario 4	March 2 Build 144	Pass	Build 150
OVERALL STATUS	AS OF MARCH 3 10:08 AM	PARTWAY DONE WITH THIS ITERATION'S DEVELOPMENT SEEMS TO BE ON TRACK	PROJECTED LAST BUILD FOR THIS ITERATION: BUILD 155

- 테스트가 언제 완료될 것인가에 대한 질문에 대답하기 위해 〈그림 1〉의 대시 보드를 제공함
  - 이것은 제품의 기능 및 성능 시나리오를 보여주는 것으로 제품의 현재 상태를 제공하고 상태가 다시 변경될 때를 나타냄
- 이 대시 보드는 프로젝트의 여러 기능을 중심으로 구성되어 있음
  - 모듈 또는 영역으로 작업할 때, 이들의 주위에 차트를 구성할 수 있음
  - 모듈 또는 영역들은 고객이 실제로 사용하는 것이나 구매하는 것을 더 정확하게 기술하기 때문에 이러한 기능들을 추천함
- 성능 시나리오와 같은 배포 기준을 사용하는 경우 가장 왼쪽 열에 표시된 대로 그것들을 추적할 수 있고, 고객들이 사용하는 것과 배포를 위해 고려해야 하는 것들을 추적하는 테스트 대시보드를 가질 수 있음

▶ 2. 테스터는 얼마나 진행하고 있는가

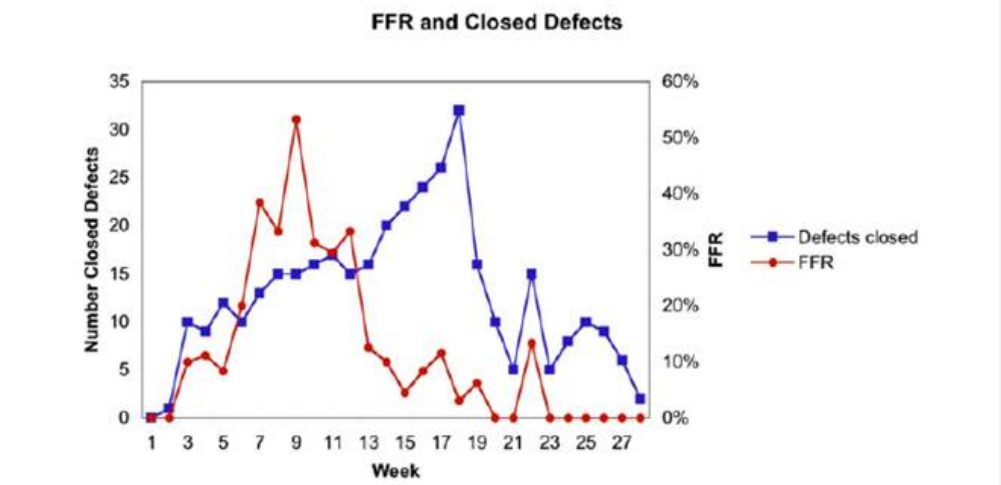
그림 2_ 테스터는 얼마나 진행하고 있는가



- 또 다른 주요 측정기준은 시간이 지남에 따라 테스터들이 얼마나 진행하고 있는가임
- 단일 차원 측정은 너무 쉬운 방법이기 때문에 완료된 테스트 케이스의 수만을 측정할 수는 없음
- 계획된 테스트 케이스의 수, 실행된 테스트 케이스의 수, 통과된 테스트 케이스 수라는 다차원 측정을 하고자 한다면, 〈그림 2〉처럼 표현하기 위한 더 완성된 스토리가 있어야 함
- 이 그래프에서 보면 요구사항에 대한 것들이 진행의 후반에 증가하기 때문에 배포할 때가 다가오면 계획된 테스트의 수가 늘어나는 것을 알 수 있음
  - 통과된 테스트 수가 측정기간의 시작 부분에 적은 이유는 개발자들이 보고된 회기 부분을 중요하게 생각하지 않기 때문임
  - 또한 개발자가 이전의 결함을 발견하고 수정하자마자 테스터가 새로운 결함을 지적하기 때문에 테스트가 측정을 실행하는 것을 알려주는 지표에 몇 번의 정체기가 있음
  - 다른 말로 하면 테스터가 새로운 결함을 발견하는 것과 같은 비율로 결함들을 종결함
- 이러한 측정들 중 단지 하나만 나타내게 된다면, 그래프는 진행되고 있는 전체 스토리를 의미하는 것이 아님

▶ 3. 개발자는 얼마나 진행하고 있는가

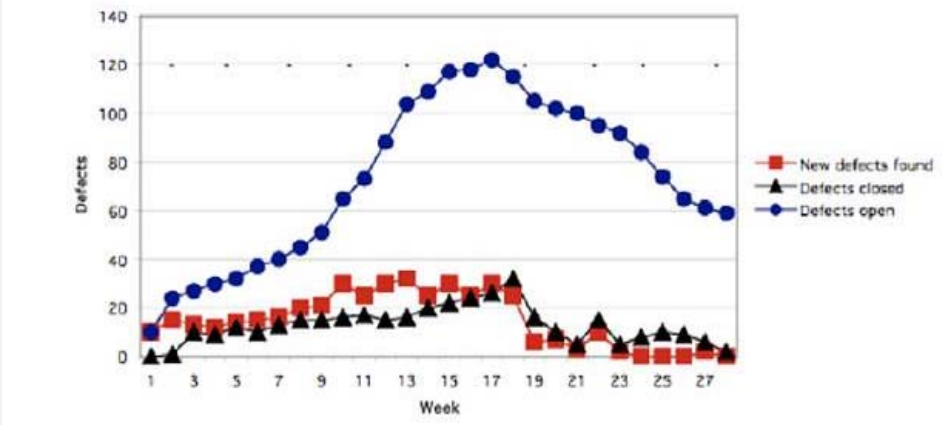
그림 3_ 개발자는 얼마나 진행하고 있는가



- 테스터의 진행 상황을 추적하는 것뿐만 아니라, 개발자들이 수행하는 방법을 측정해야함
  - 대시보드에서 정보를 얻는 프로젝트 관리자와 함께 일을 할 수 있음
  - 개발자가 어떤 진행 중에 있는지 확인하기 위해, 요구 사항의 변화와 함께 결함 회귀비율(FFR)를 측정함
  - 개발자들은 단지 유지할 수 있는 만큼만 빨리 진행하는 경향이 있음
- 결함회기비율은 모든 버그 수정 중에서 잘못된 수정의 비율임
  - 8 % 이하의 비율을 유지해야 하며, 만약 결함회기비율이 그 수준 이상으로 올라가기 시작하면 개발자들은 공황에 빠져 더 이상 진전을 보이지 않음
  - 하지만 결함회기비율이 낮아질 때, 개발자는 진행하게 되고 〈그림 3〉과 같이 더 많은 결함을 제거함
- 테스트 프로젝트가 충분히 빠르게 진행되지 않을 때 개발자를 비난하기 위해 이와 같은 그래프를 오용하는 상황을 볼 수 있음
  - 만약 조직에서 이러한 상황이 발생할 경우, 공개 대시보드에 이 그래프를 포함하지 않아야 함

▶ 4. 시간 경과에 따른 결함 동향

그림 4_ 시간 경과에 따른 결함 동향



- 〈그림 4〉는 시간이 지남에 따라 결함의 완전한 정보를 나타냄
- 결함에 대한 하나의 범주만을 중요하고 긴급한 것처럼 보고하지 말아야함
  - 모든 고객은 자신이 직면한 결함이 가장 중요하다 생각하기 때문에, 만약 중요하거나 긴급한 것을 추적하기 시작한다면, 결함의 우선순위를 매기는 회의를 계속하게 될 것임
  - 일단 중요한 것은 결함의 전체 개수이기 때문에, 단지 한 주에 공개하고 제한하는 개수만을 추적하면 모든 결함의 총체적 영향력을 발견하기 어려움
- 〈그림 4〉에서 추적한 제품에 대해 주목해 보면, 결코 일주일에 20개 이상의 결함이 발견되지는 않음
  - 대부분 20개 미만으로 결함이 발견되었고 발견된 만큼 수정함
  - 이 그림을 보고받은 최고경영자는 100개 이상 공개 결함이 있다는 것을 알고 충격을 받았으며, 모든 잔여 공개 결함의 누적 개수는 놀라운 것이었음
  - 매주 결함을 발견한 만큼 결함을 수정하지 않았다면 그 결과가 매우 충격적일 것임

▶ 5. 결론

- 테스트가 완료될 때를 설명하기 위해, 요구사항의 변화에 관련된 다른 데이터를 추적할 필요가 있는지를 결정해야 할 경우가 발생함
  - 요구사항이 계속 변경되는 경우, 테스트는 결코 완료되지 않을 것임
- 전체 상황 파악을 위해 수집해야 할 데이터를 결정해야 하는 것이 우선임

참고 자료

1. https://wiki.jenkins-ci.org/display/JENKINS/Test+In+Progress+Plugin
2. http://www.softwaretestingtimes.com/2010/08/test-progress-monitoring-reporting.html
3. http://www.youtube.com/watch?v=y1au6NUMNe8