애플리케이션 테스트 수행
part 2

결함 관리

KOREATECH
온라인평생교육원

학습내용

- 애플리케이션 결함 관리 개요
- 애플리케이션 결함 관리 도구

학습목표

- 애플리케이션 결함의 정의와 관리 프로세스에 대해 설명할 수 있다.
- 결함의 상태 및 추적 업무 흐름도를 파악할 수 있다.
- 결함의 유형과 심각도, 결함 등록 방법에 대해 설명할 수 있다.
- 결함의 원인 분석 방법과 결함 관리 도구에 대해 설명할 수 있다.

애플리케이션 결함 관리 개요

1 결함의 정의

결함 (Defects)

- 코드 내에 포함된 실수 또는 프로그램과 명세서 간의 차이, 업무 내용 불일치
- 기대 결과와 실제 관찰 결과 간의 차이
- 시스템이 사용자가 기대하는 타당한 기대치를 만족시키지 못할 때 변경이 필요한 모든 것
- 부정확한 구문이나 데이터 정의로 인해 필요한 기능을 수행하지 못하도록 하는 애플리케이션 상의 결점

실패 (Failure)

- 정상적인 프로그램과 비정상적인 프로그램의 실행 결과의 차이
- 요구사항을 만족하지 못하는 상태

애플리케이션 결함 관리 개요

1 결함의 정의

개선

- 프로그램과 명세서 간의 차이는 아님(결함 아님)
- 결함은 아니지만, 사용자가 편의성 향상을 위해 애플리케이션의 수정을 요청하는 것

애플리케이션 테스트 케이스 측면

- 결함은 테스트 케이스 결과 실패
- 개선은 성공을 의미함

개선을 하는 경우

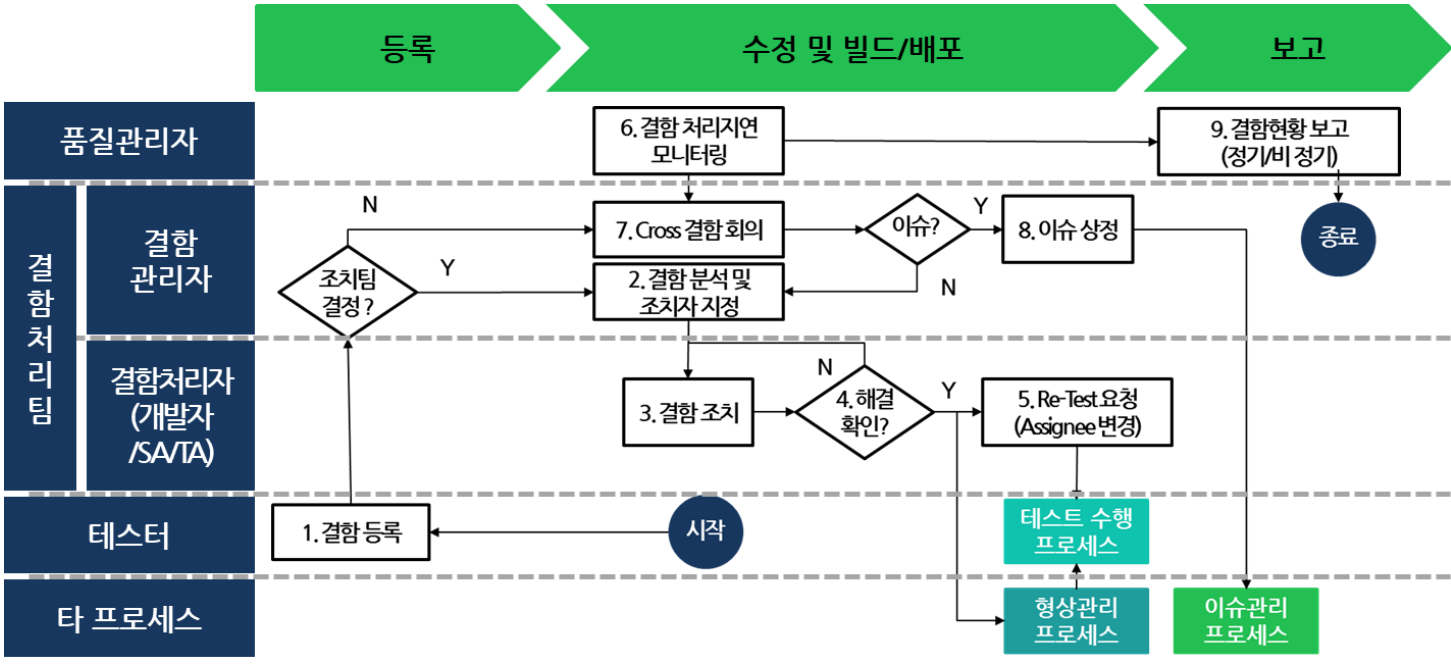
- 테스트 케이스가 성공이지만, 사용자의 편리를 증진하기 위해 개선을 하기도 함

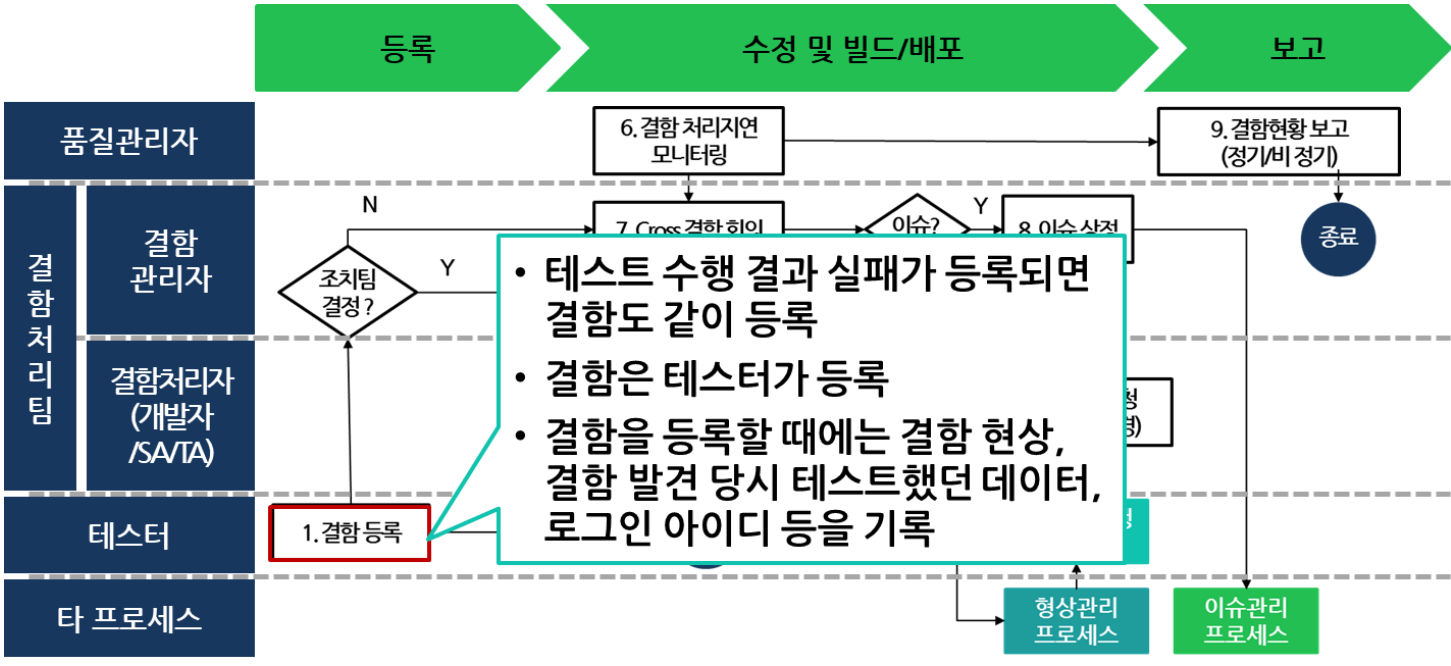
결함 관리 프로세스

- 결함을 인지하고, 기록하고, 조사하고, 분류하고, 영향력을 식별하여 행동을 취해 처리하는 일련의 절차

애플리케이션 결함 관리 개요

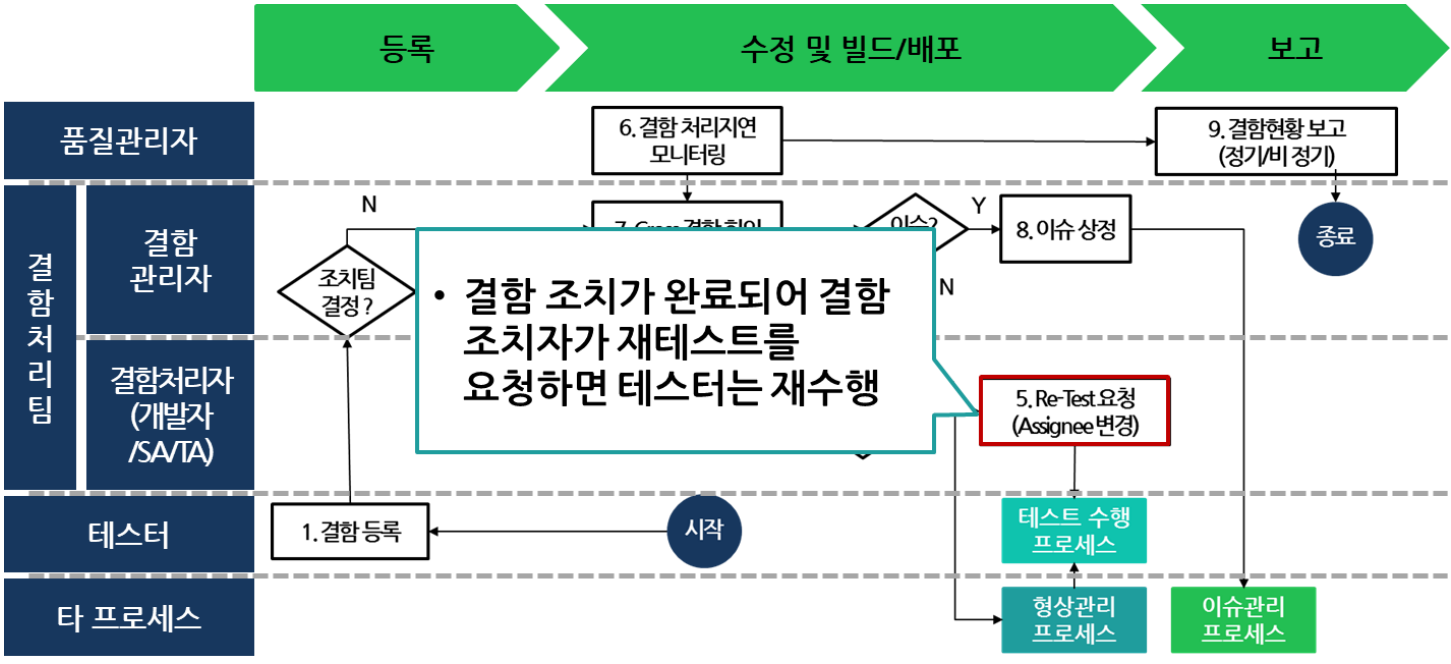
2 애플리케이션 결함 관리 프로세스

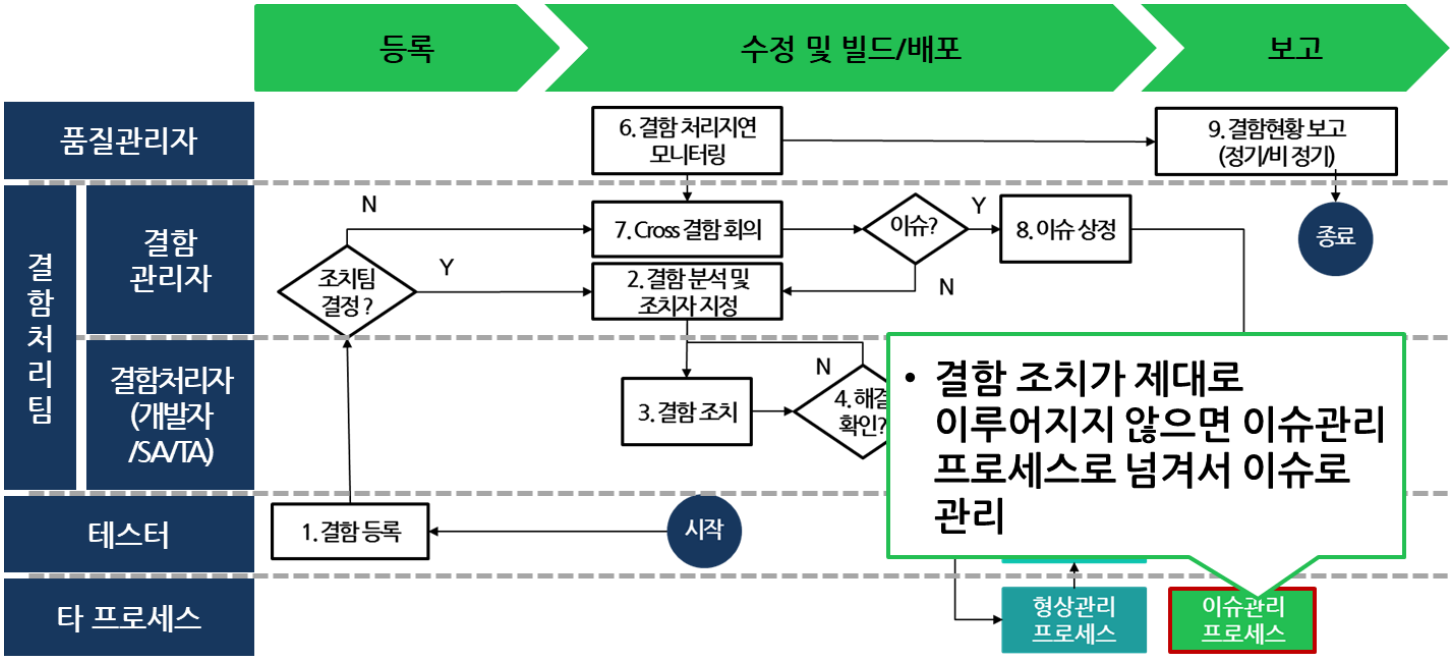




애플리케이션 결함 관리 개요

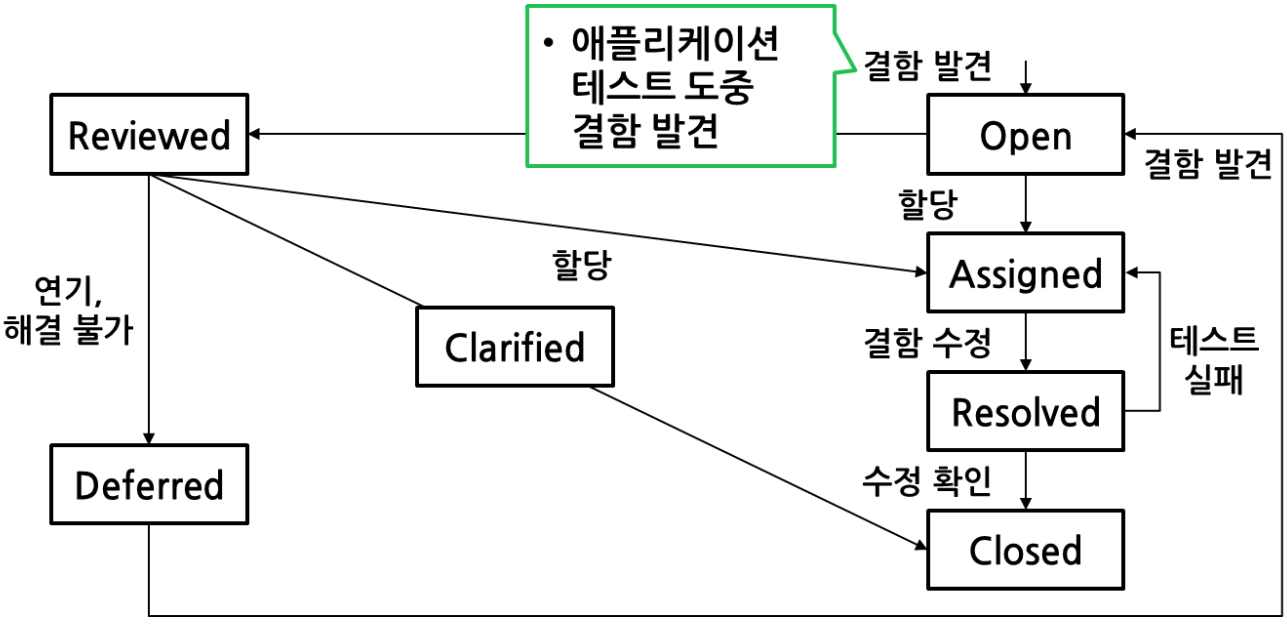
2 애플리케이션 결함 관리 프로세스

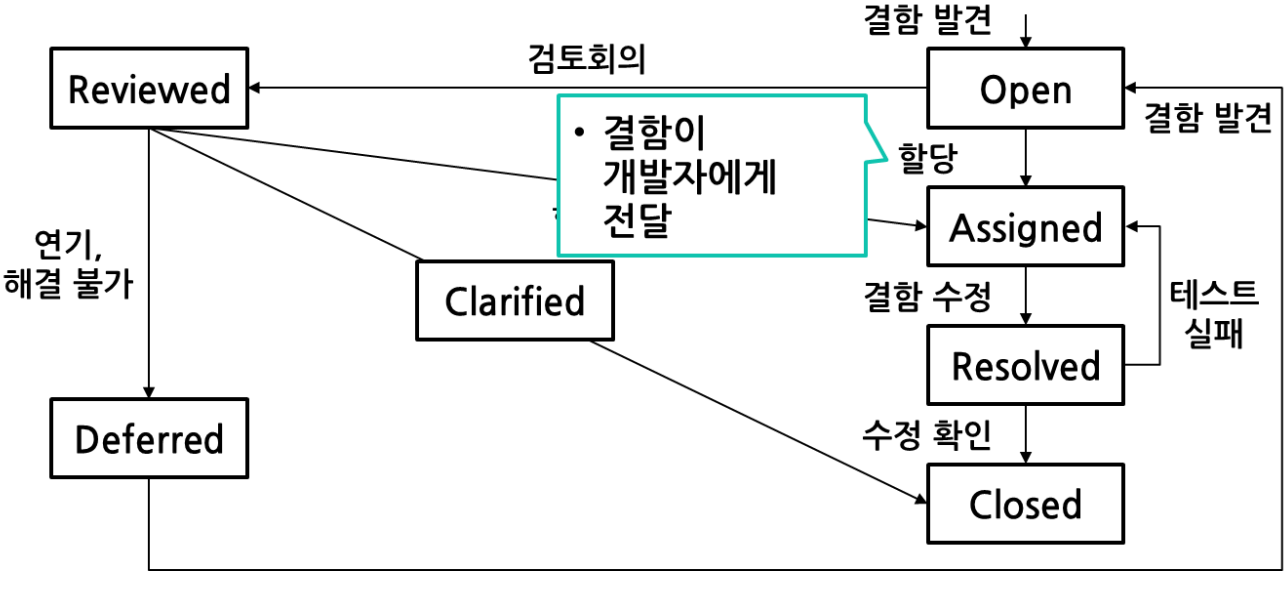




애플리케이션 결함 관리 개요

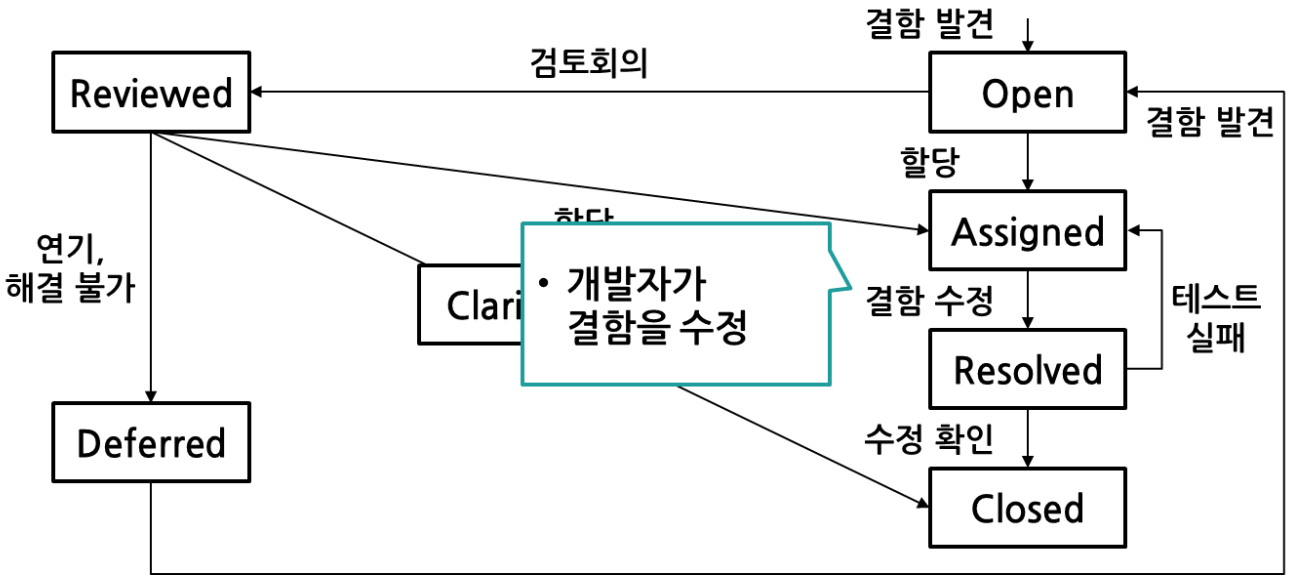
3 결함의 상태

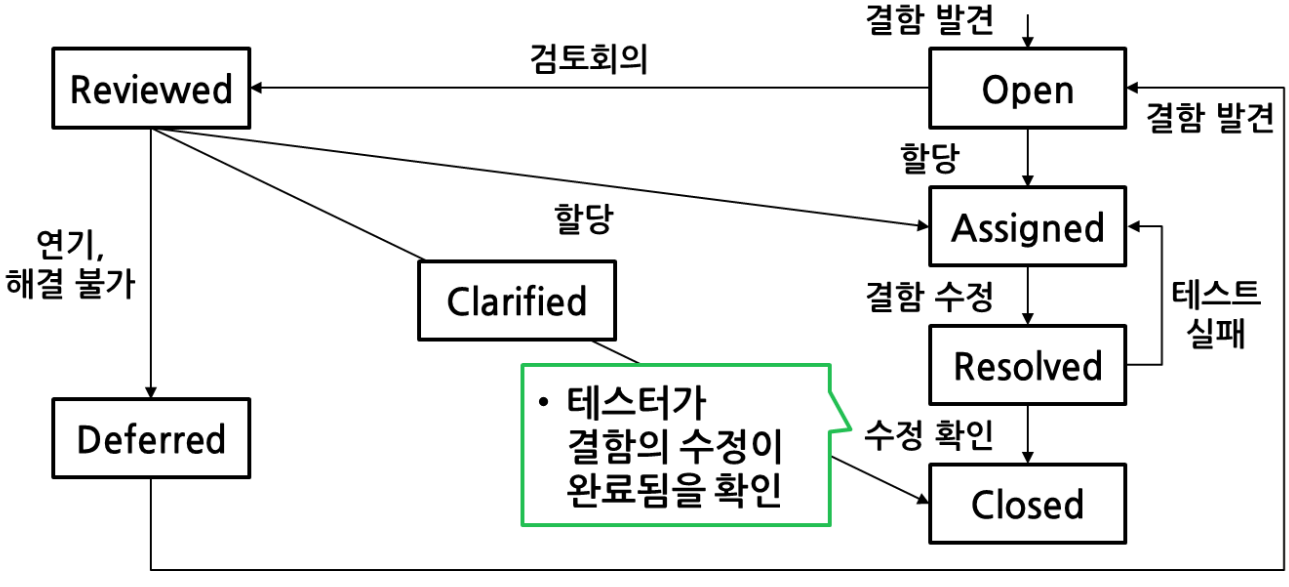




애플리케이션 결함 관리 개요

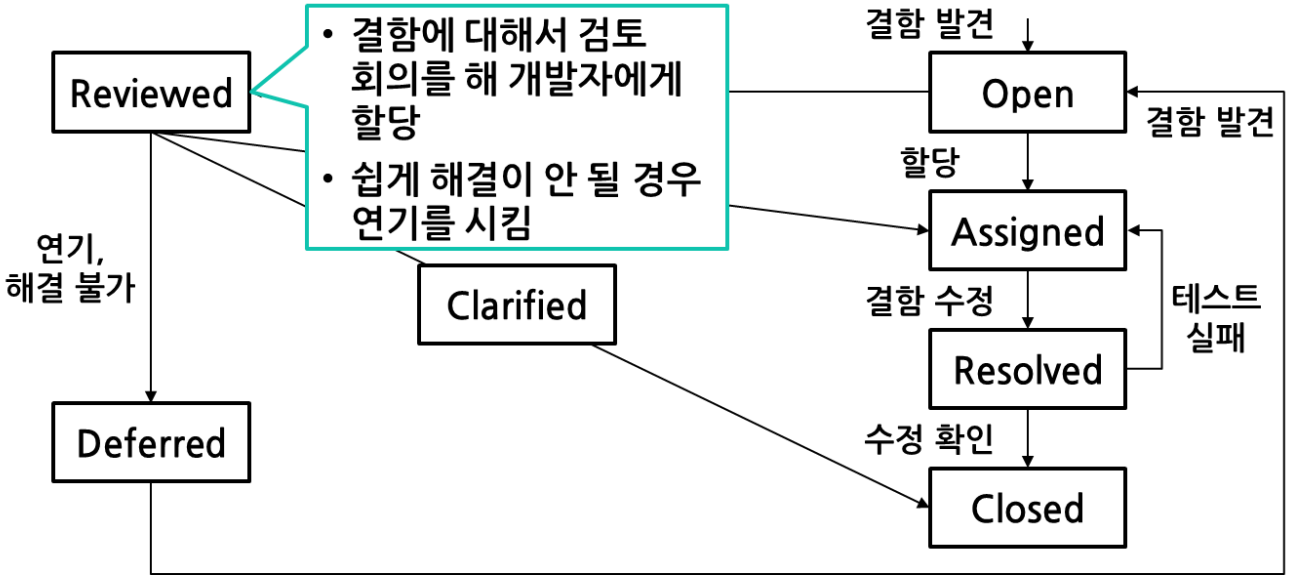
3 결함의 상태

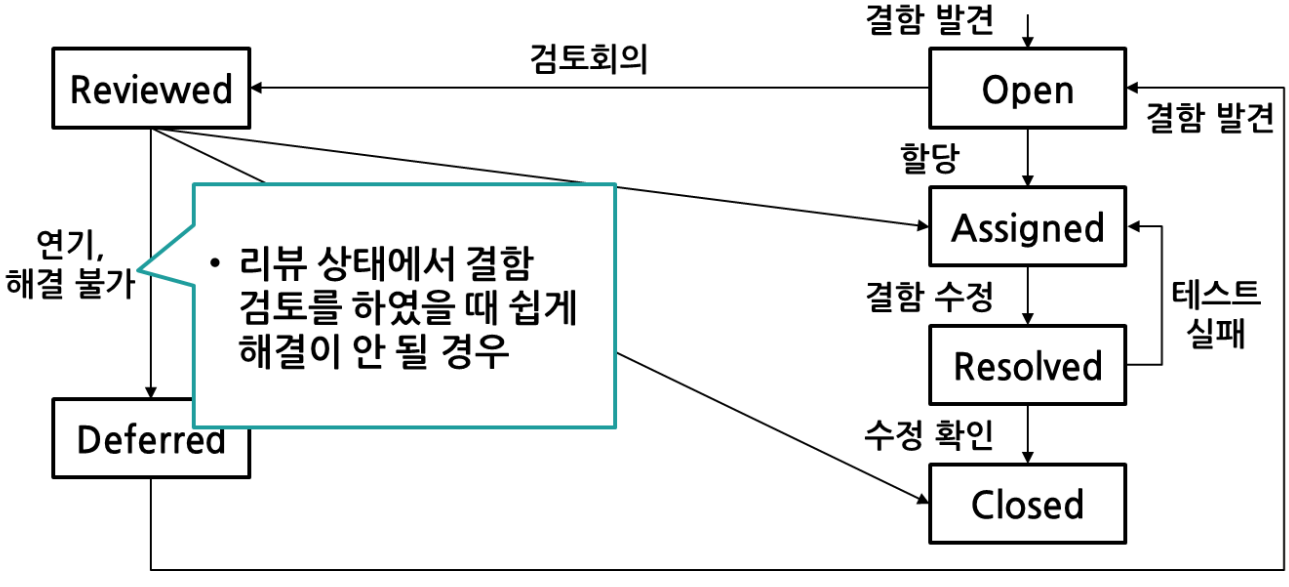




애플리케이션 결함 관리 개요

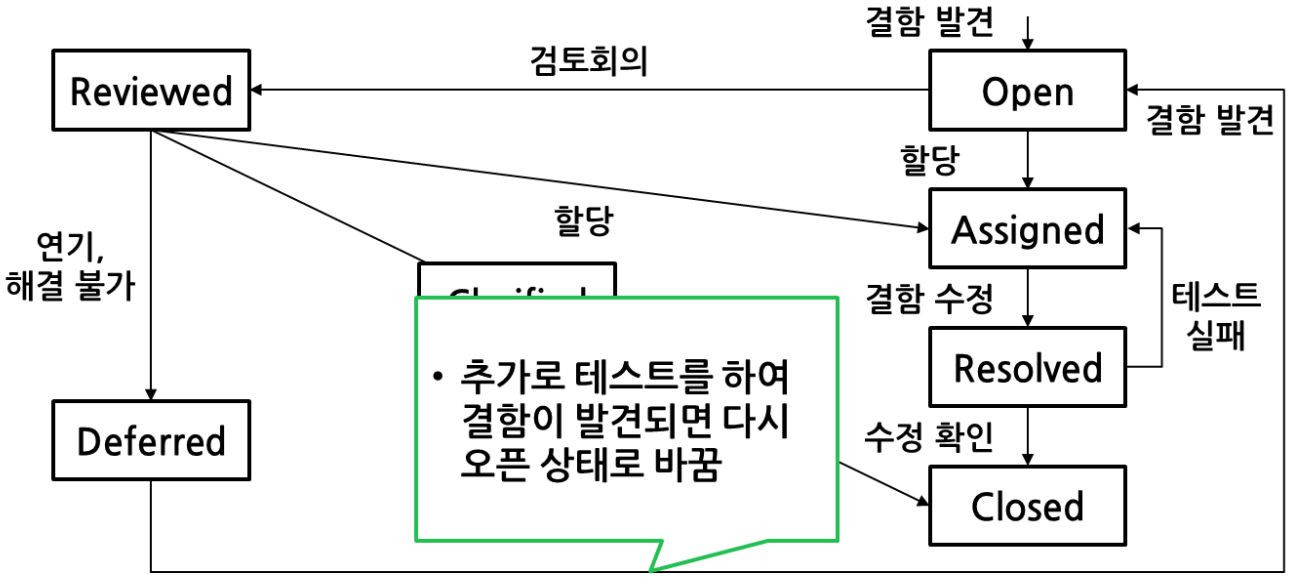
3 결함의 상태

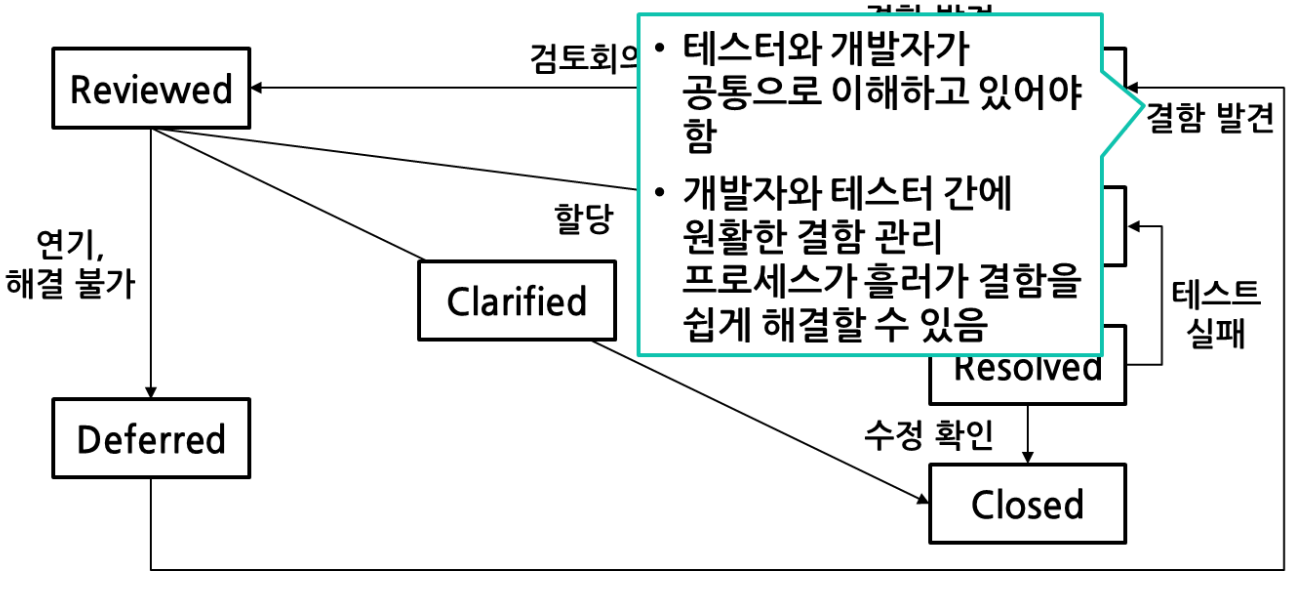




애플리케이션 결함 관리 개요

3 결함의 상태



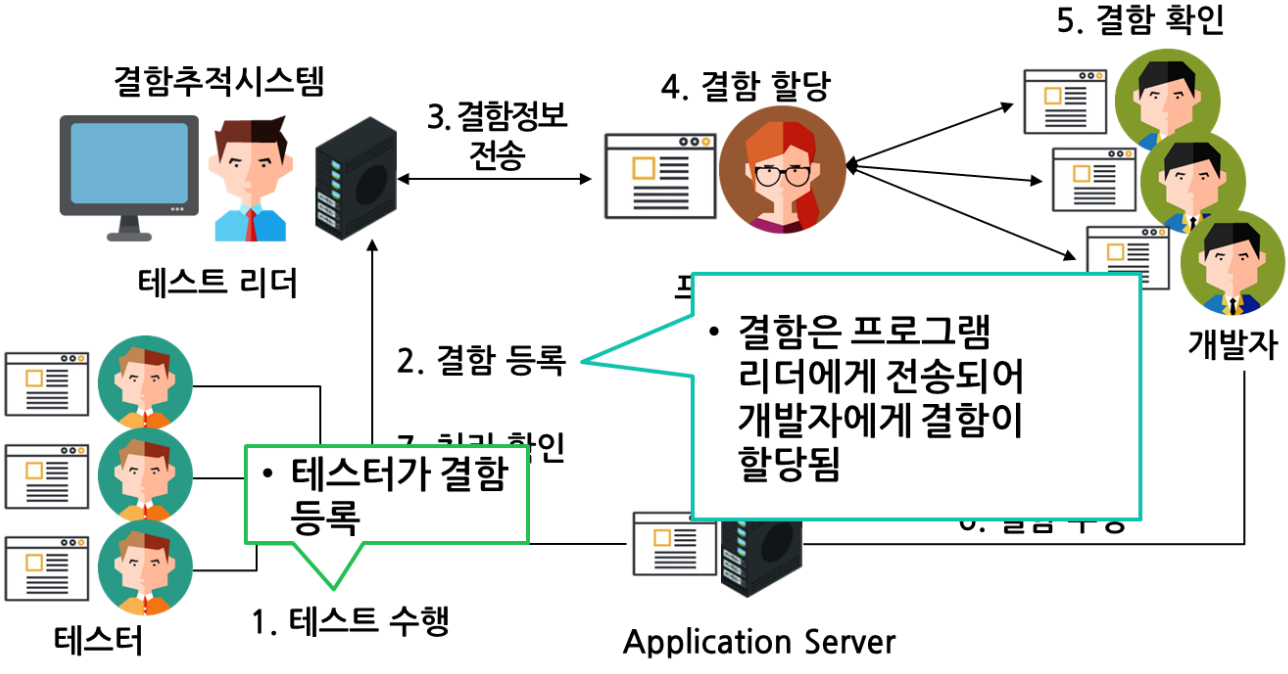


애플리케이션 결함 관리 개요

4 결함 추적 업무 흐름도

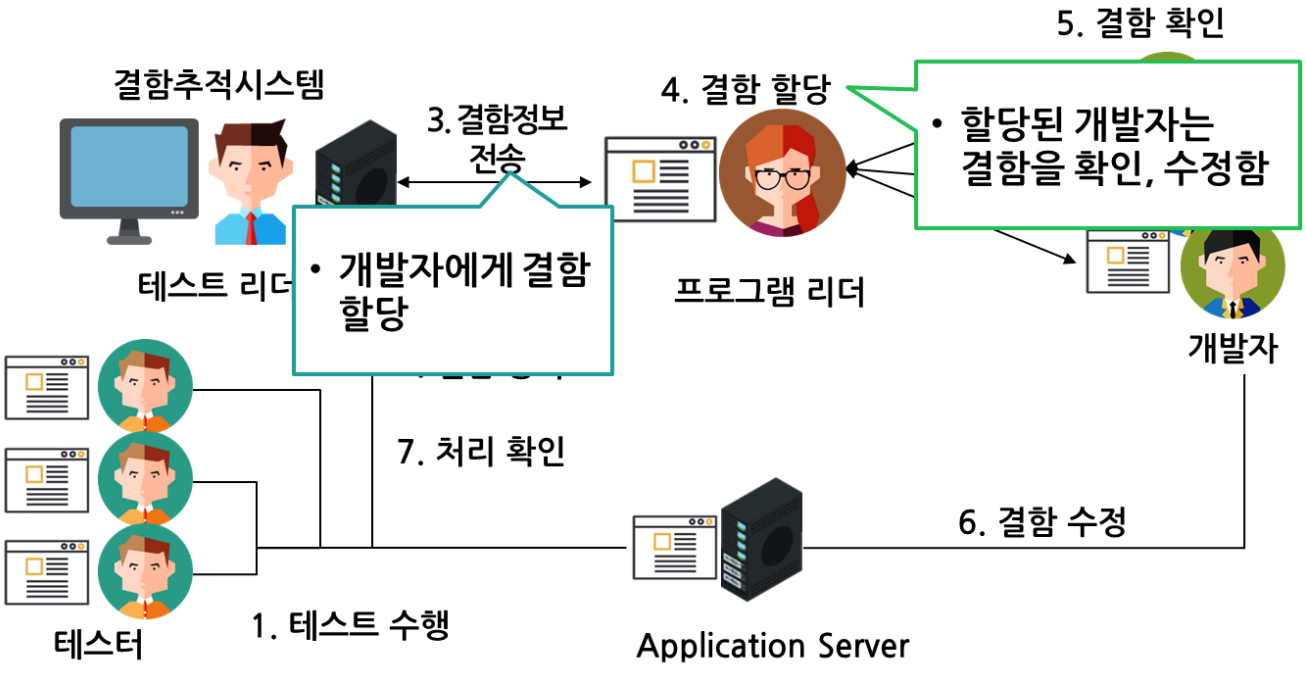
결함은 등록에서부터 최종 종료될 때까지 추적을 할 수 있어야 함

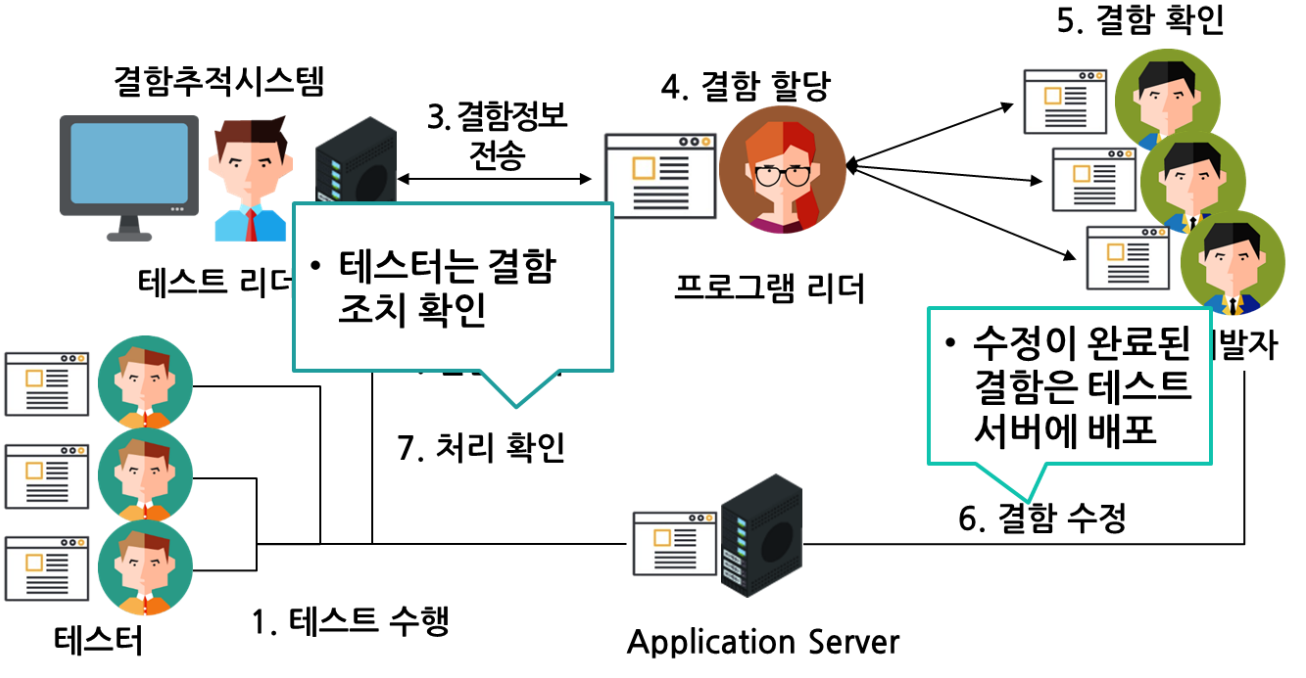
결함을 추적하는 이유는 누락된 결함이 발생하지 않고, 정해진 테스트 기간에 결함을 신속하게 처리하기 위함



애플리케이션 결함 관리 개요

4 결함 추적 업무 흐름도





애플리케이션 결함 관리 도구

1 결함의 유형

결함 유형은 결함을 분류하기 위한 기준

결함 유형	세부 내용
코드 (Code)	• 애플리케이션 코드에서 나온 결함 • UI, 서비스, 컴포넌트, Batch, Interface 등 소스 코드 결함 • 소스코드 결함은 개발자가 만들어 낸 것
데이터 (Data)	• 데이터에서 발생하는 결함 • 기준정보, 스키마 형상 불일치, Output 데이터 생성 결함 등
환경 (Environment)	• DBMS, 하드웨어, 미들웨어, 네트워크, 시스템 소프트웨어 등 환경 결함 • 개발 환경과 테스트 환경은 별도로 구분이 되어 환경문제 때문에 결함이 발생하는 경우가 많음
패키지 (Package)	• Siebel, Oracle, SAP 등 패키지 소프트웨어가 자체적으로 가지고 있는 결함으로 발생

애플리케이션 결함 관리 도구

1 결함의 유형

결함 유형은 결함을 분류하기 위한 기준

결함 유형	세부 내용
클라이언트 환경 (Client Environment)	• 클라이언트(Client)의 PC 환경 결함 • 예를 들어, 다른 테스터에서는 발생하지 않는데 유독 특정 테스터 PC에서만 발생하는 결함의 경우 클라이언트 환경이 달라서 발생하는 결함
결함 아님 (Not Defect)	• 테스터는 결함이라고 등록하였지만 사실 결함이 아닌 것들이 있음 • 테스터가 업무를 몰라서 그럴 수도 있기 때문에 결함 아님은 테스터에게 반드시 상세히 설명해야 함

애플리케이션 결함 관리 도구

2 결함 심각도(Criticality)

결함의 심각도는 해당하는 결함이 애플리케이션에 미치는 영향

결함의 심각도는 크리티컬(Critical)이 가장 심각하고, 로우(Low)가 가장 낮은 심각도임

심각도가 높다는 의미는 애플리케이션에 미치는 영향이 커서 빨리 조치가 되어야 함을 의미

심각도가 낮다는 것은 영향이 미미하기 때문에 상대적으로 여유를 가지고 결함 수정을 해도 된다는 의미

애플리케이션 결함 관리 도구

2 결함 심각도(Criticality)

심각도 (Criticality)	판단 기준
1. Critical	• 조치되지 않으면 테스트가 더 이상 진행이 불가한 경우(즉시 조치) • 로그인이 되지 않으면 어떠한 테스트도 진행될 수 없기 때문에 사용자 로그인이 되지 않는다면 크리티컬(Critical) 결함이라고 할 수 있음 • 결함의 심각도는 테스터 입장에서 측정함 • 개발자는 결함의 심각도를 낮추려는 경향이 있음 • 결함의 심각도는 테스터·사용자의 입장에서 측정하는 것이 바람직함
2. High	• 테스트 진행은 가능하나 핵심 기능 결함인 경우(1일 이내 조치)
3. Medium	• 핵심 기능은 정상이나 부가 기능 결함인 경우(2일 이내 조치)
4. Low	• 기능 개선이 필요한 경우(3일 이내 조치)

애플리케이션 결함 관리 도구

3 결함 등록

결함을 등록할 때에는 반드시 테스트 케이스 아이디와 연계해 등록

- 대부분 결함은 테스트 케이스를 실행하는 과정에서 발견되기 때문임

테스트 케이스를 실행하는 과정에서 결함이 발견되었는지 알기 위해서는 반드시 테스트 케이스 아이디와 연계를 해야 하기 때문

애플리케이션 결함 관리 도구

3 결함 등록

결함을 등록할 때 주의할 점

- 개발자가 알기 쉽도록 결함 현상을 최대한 상세하게 설명을 해야 함
- 가장 좋은 방법은 결함이 발생하였을 때의 화면을 캡처하여 결함을 등록할 때 같이 첨부하는 것
- 결함이 발생하였을 때의 화면에는 오류 메시지가 나오기 때문에 개발자가 결함의 현상과 원인을 파악하기 쉽기 때문임
- 결함 내용, 테스트케이스 번호, 결함 유형, 발견일, 심각도, 수정우선순위, 조치 예정일, 담당자, 재 테스트 결과, 종료일을 필수로 입력

애플리케이션 결함 관리 도구

3 결함 등록

테스트ID	테스트 시나리오명	테스트 케이스 ID	테스트 케이스명	결과
DOCTS-30-01	전자문서 안내와 전자문서 작성 (업무사용자)	DOCTS-30-01-01	전자안내	PASS
		DOCTS-30-01-02	보낸 주소#메일확인	FAIL
		DOCTS-30-01-03	받는 주소 등록	PASS
		DOCTS-30-01-04	첨부 문서 등록	PASS
		DOCTS-30-01-05	발송 요청	NOT EXECUTED
DOCTS-80-01	받은 문서함 관리 (업무사용자)	DOCTS-80-01-01	검색조건 등록	CLARIFIED
		DOCTS-80-01-02	문서 목록 조회	PASS
		DOCTS-80-01-03	문서 상세 조회	PASS

애플리케이션 결함 관리 도구

3 결함 등록

테스트 케이스 ID	테스트 계획일	테스트 실시일	결함내용
DOCTS-30-01-01	2018.10.01	2018. 10. 19	
DOCTS-30-01-02	2018.10.01	2018. 10. 19	보낸 주소로 메일이 전달되지 않음
DOCTS-30-01-03	2018.10.01	2018. 10. 19	
DOCTS-30-01-04	2018.10.01	2018. 10. 19	
DOCTS-30-01-05	2018.10.01	2018. 10. 19	발송 요청 문서가 전달되지 않음
DOCTS-80-01-01	2018.10.01	2018. 10. 19	문서 검색 시 특수문자 사용불가(SQL 인젝션모듈 작동)
DOCTS-80-01-02	2018.10.01	2018. 10. 19	
DOCTS-80-0103	2018.10.01	2018. 10. 19	

애플리케이션 결함 관리 도구

4 결함 원인 분석 및 도구

결함이 등록되고 개발자가 배정되면, 결함 원인을 분석하며, 결함 원인은 개발 단계마다 다를 수 있음

분석

- 불명확한 요구사항으로 인해 개발자의 업무에 대한 이해 부족으로 결함이 발생할 확률이 높음

설계

- 불완전한 설계와 설계 문서에 대한 이해가 부족하여 결함을 유발함

구현

- 개발자의 코딩 역량 부족
- 개발 도구 사용 미숙
- 테스트 데이터를 사용하여 충분한 테스트 부족

기타/공통

- 개발 표준의 미준수
- 변경 내용을 반영하지 않은 경우

애플리케이션 결함 관리 도구

4 결함 원인 분석 및 도구

1대 10대 100의 법칙

요구사항 분석 단계에서 결함의 수정 비용이 1,
설계 및 구현 단계에서 결함의 수정 비용은 10,
테스트 및 운영 단계에서 결함의 수정 비용은 100

- 애플리케이션 **요구사항 분석 단계에서 결함을 발견**하면 **수정하기 쉬움**
- **테스트 및 운영 단계에서 발생하는 결함**은 상대적으로 **비용도 많이 들고, 기간이 오래 걸려** 수정이 어려움

애플리케이션 요구사항 분석 단계에서부터 결함을 유발하는 원인을 제거해 나가야 함

애플리케이션 결함 관리 도구

4 결함 원인 분석 및 도구

① 파레토 다이어그램(Pareto Diagram)

1. 결함을 발생한 단계별로 분류한 후 그래프로 그림
2. 구현, 설계, 요구사항 분석 단계별로 결함의 개수와 비율을 측정해 다이어그램으로 그림
3. 어느 단계에서 결함이 많이 발생하였는가 분석할 수 있음

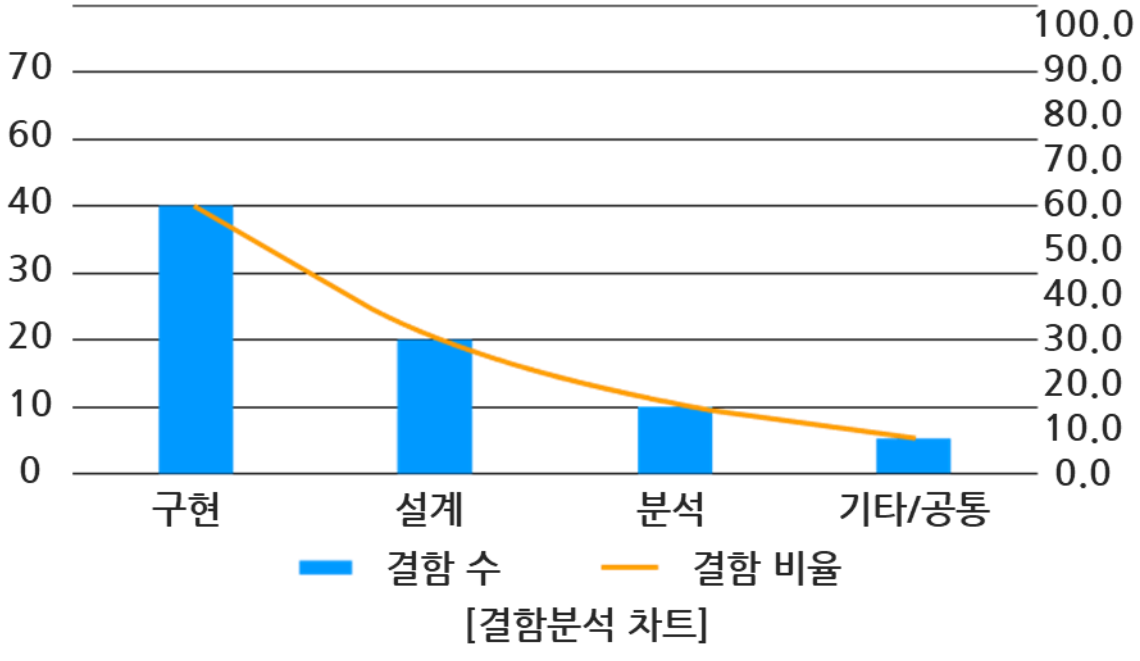
자원을 집중하여 **결함을 제거**해 나가야 함

애플리케이션 결함 관리 도구

4 결함 원인 분석 및 도구

① 파레토 다이어그램(Pareto Diagram)

문제영역	결함 수	결함비율
합계	75	100.0
구현	40	53.3
설계	20	26.7
분석	10	13.3
기타 /공통	5	6.7



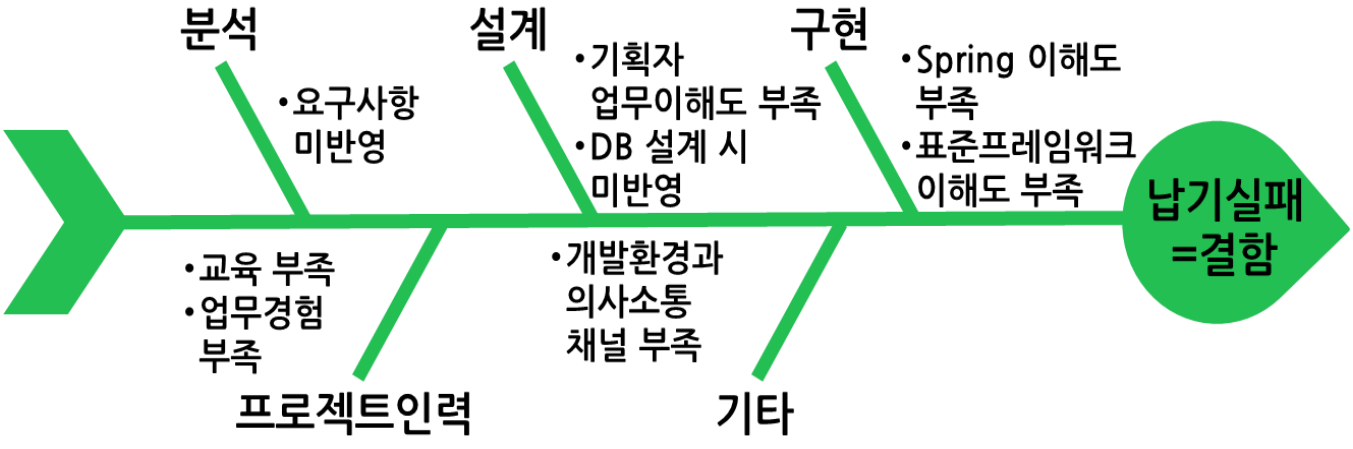
[결함분석 차트]

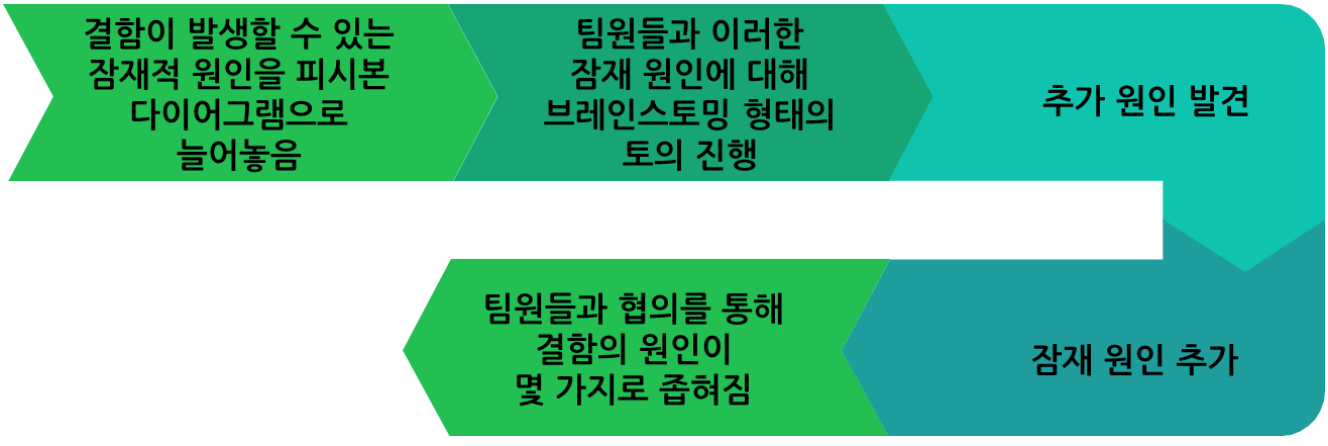
애플리케이션 결함 관리 도구

4 결함 원인 분석 및 도구

〈2〉 피시본 다이어그램(Fishbone Diagram)

물고기의 뼈와 비슷하게 생겨서 붙여진 이름





애플리케이션 결함 관리 도구

4 결함 원인 분석 및 도구

〈2〉 피시본 다이어그램(Fishbone Diagram)

결함 원인 분석에 피시본 다이어그램 사용

▶ 결함의 잠재 원인을 알아보기 쉽게 그려 줌

팀원들과 토론하기 편리함

애플리케이션 결함뿐만 아니라 다른 업무에도 많이 쓰여, 알아두면 편리함

애플리케이션 결함 관리 도구

5 결함 관리 도구

실제 프로젝트에서는 결함을 수작업으로 관리하지 않음

▶ 대규모 프로젝트에서는 결함 관리 자동화 도구를 사용해 관리함

결함 관리 자동화 도구
- 오픈 소스
- 상용 소스

대표적인 결함 관리 자동화 도구로는 JIRA, Bugzilla, Trac, Mantis가 있음

애플리케이션 결함 관리 도구

5 결함 관리 도구

지라(JIRA)

- 아틀래시안 사에서 개발한 버그 추적, 이슈 추적 관리 도구
- 상용 소프트웨어이기 때문에 구매를 한 후 사용해야 함

버그질라(Bugzilla)

- 모질라 재단이 만든 오픈 소스 도구
- 이슈관리 도구로 알려짐

맨티스(Mantis)

- 이슈관리뿐만 아니라 프로젝트 관리 도구로도 사용됨
- 오픈 소스이기 때문에 누구나 자신의 결함 관리 환경에 맞게 커스터마이징해 사용하면 됨

애플리케이션 결함 관리 도구

5 결함 관리 도구

항목	JIRA	Bugzilla	Trac	Mantis
제작사	아틀래시안	모질라 재단	Edgewall Software	Mantis
주요기능	결함 상태 관리	이슈관리	Wiki, 이슈관리	이슈관리
라이선스	상용	MPL	BSD	GPL
개발언어	PHP	Perl	Python	PHP

학습정리

01. 애플리케이션 결함 관리 개요

- 애플리케이션 테스트 수행 중 발견된 결함은 코드 내에 포함된 실수 또는 프로그램과 명세서 간의 차이, 업무 내용 불일치를 의미함
- 테스트 수행 중 결과가 실패로 등록되면 결함도 같이 등록함
- 결함 등록 시 화면 캡처를 하면 개발자가 결함을 쉽게 이해함
- 결함의 상태는 등록(Open), 할당(Assigned), 조치(Resolved), 종료(Closed)로 나누어짐

02. 애플리케이션 결함 관리 도구

- 결함 유형은 코드(Code), 데이터(Data), 환경(Environment), 패키지(Package), 클라이언트환경(Client), 결함 아님(Not Defect)으로 나누어 짐
- 결함 심각도는 Critical, High, Medium, Low로 나누어 짐
- 애플리케이션 테스트 수행 시 발견된 결함은 등록할 때에 반드시 테스트 케이스 아이디를 입력함
- 결함 원인 분석 도구에는 파레토 다이어그램, 피시본 다이어그램을 많이 사용함