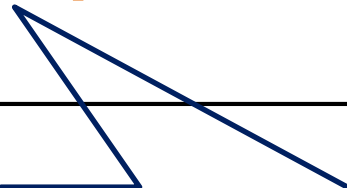

애플리케이션 테스트 수행

2001020227_16v4



요구사항대로 응용 소프트웨어가 구현되었는지를 검증하기 위해서 분석된 테스트 케이스에 따라 테스트를 수행하고 결함을 조치할 수 있다



1

애플리케이션 테스트 수행하기

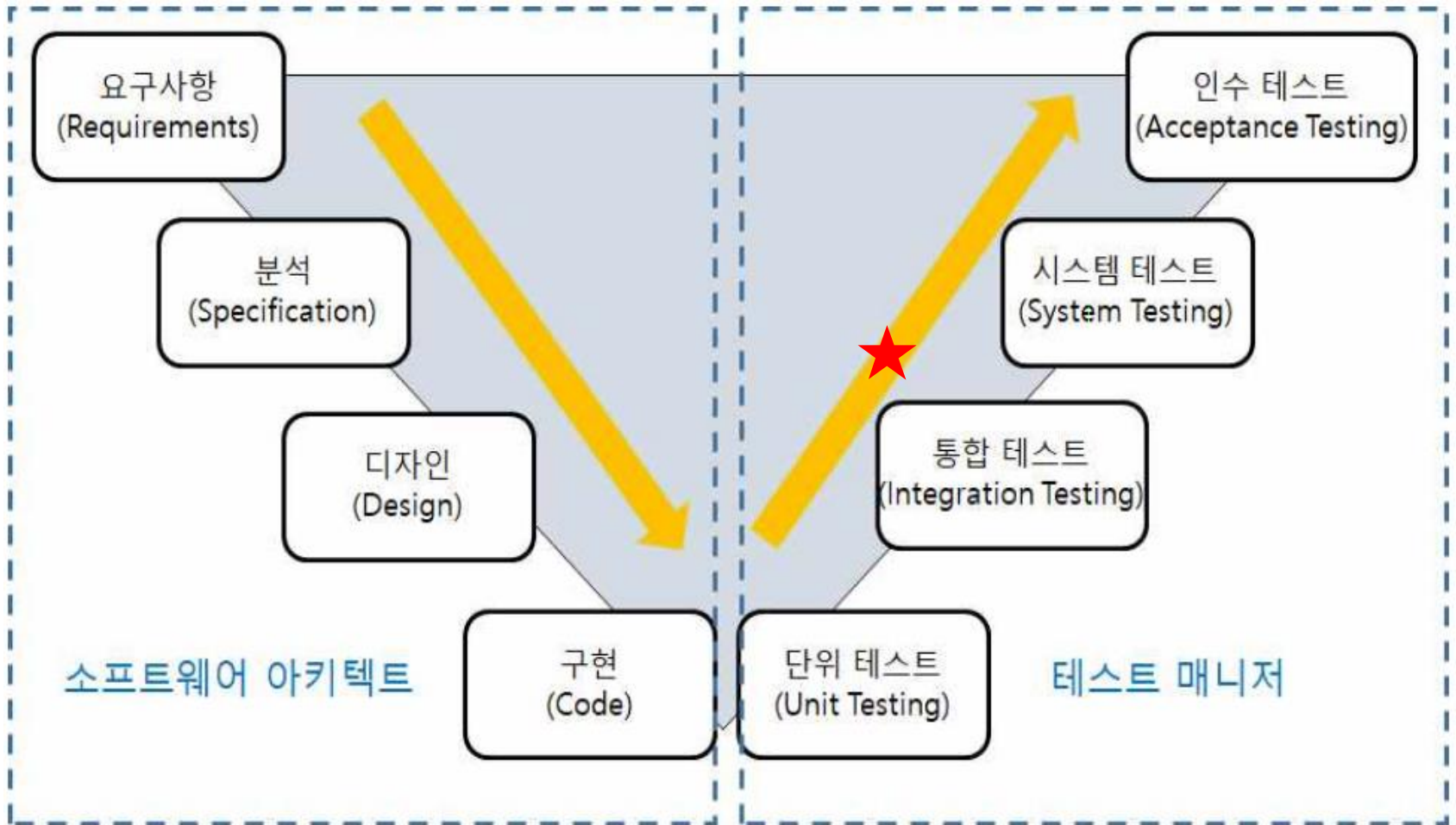
2

어플리케이션 결함 조치하기

1

애플리케이션 테스트 수행하기

1-1. 테스트 수행



[그림 1-1] 소프트웨어 생명 주기의 V-모델

1-1. 테스트 수행

프로젝트 수행 단계에 따른 테스트의 분류

1. 단위 테스트 : 테스트 가능한 단위로 작게 분리된 SW내에서 결함을 찾고 기능을 검증하는 테스트 활동

특징

개발자에 의해 행해지며, 과거에는 시간부족을 이유로 생략되었으나, 최근 개발도구의 발전으로 개발과정 중에 자동으로 진행된다. 개발 도구에서 지원하지 않아도 반드시 수행.

구조적 테스트, 기능성 테스트, 리소스 관련테스트 등 특정 비기능성 테스트 등이 포함되어 수행.

컴포넌트 명세, SW 상세설계, 데이터 모델 명세 등을 이용하여 테스트한다

방법과 목적

테스트 방법	테스트 내용	테스트 목적
구조 기반	- 프로그램 내부 구조 및 복잡도를 검증하는 <u>화이트박스(White Box) 테스트</u>	제어 흐름, 조건 결정
명세 기반	- 목적 및 실행 코드 기반의 실행을 통한 <u>블랙박스(Black Box) 테스트</u> ★	동등 분할, 경계 값 분석

1-1. 테스트 수행

프로젝트 수행 단계에 따른 테스트의 분류

2. 통합 테스트 : 단위테스트가 끝난 모듈을 통합하는 과정에서 발생할 수 있는 오류를 찾는 테스트

특징 ★

컴포넌트 간 인터페이스 테스트를 하고 운영체제, 파일 시스템, 하드웨어 또는 시스템 간 인터페이스와 같은 각각 다른 부분과 상호 연동이 정상적으로 작동하는지 여부를 테스트

빅뱅 방식(모든 모듈을 한꺼번에 테스트)보다는 순차적 형태(상향, 하향)의 테스트 방식으로 진행

1-1. 테스트 수행

프로젝트 수행 단계에 따른 테스트의 분류

3. 시스템 테스트 : 컴퓨터 시스템을 완벽하게 검사하기 위한 목적 또는 성능 목표를 가지고 테스트

특징

실제의 최종 사용자 환경과 유사하게 시스템 성능, 관련된 고객의 기능, 비기능적인 요구사항 등이 완벽하게 수행되는지 테스트하며, 이 때 요구사항 명세서, 유스케이스 결과 등을 이용한다.

방법과 목적

테스트 방법	테스트 내용
기능적 요구사항	요구사항 명세서, 비즈니스 절차, 유스케이스 등 명세서 기반의 <u>블랙박스(Black Box) 테스트</u>
비기능적 요구사항	성능 테스트, 회복 테스트, 보안 테스트, 내부 시스템의 메뉴 구조, 웹 페이지의 네비게이션 등의 구조적 요소에 대한 <u>화이트박스(White Box) 테스트</u>

1-1. 테스트 수행

프로젝트 수행 단계에 따른 테스트의 분류

4. 인수 테스트 : 최종사용자와 업무에 따른 이해관계자 등이 테스트를 수행. 실제 업무 적용 전 수행.

인수테스트 종류

테스트 종류	테스트 내용
사용자 인수 테스트	비즈니스 사용자가 시스템 사용의 적절성 여부를 확인
운영상의 인수 테스트	시스템 관리자가 시스템 인수 시 수행하는 테스트 활동으로 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등을 확인
계약 인수 테스트	계약상의 인수/검수 조건을 준수하는지 여부를 확인
규정 인수 테스트	정부 지침, 법규, 규정 등 규정에 맞게 개발하였는지 확인
알파 테스트	개발하는 조직 내 잠재 고객에 의해 테스트 수행
베타 테스트	실제 환경에서 고객에 의해 테스트 수행

1-1. 테스트 수행

테스트 자동화 도구

- 테스트 자동화란 사람이 하던 반복적 테스트 절차를 자동화 도구를 활용하여 시간과 인력 투입을 최소화하면서 운영 중인 시스템의 모니터링 또는 UI가 없는 서비스의 경우 정밀한 테스트가 가능하다
- 자동화 도구는 비공개 상용 SW인 경우 인력과 교육에 대한 유지관리 비용 투자가 필요하다.
- 도구의 한계성으로 모든 수동 테스트 과정을 자동화할 수는 없다. ★
- 프로젝트 초기에 테스트 엔지니어와 도구의 계획이 수립되어야 한다.

1-1. 테스트 수행

테스트 수행 순서

1.

❖ 테스트 계획서 :



- ✓ 테스트 목적과 범위 확인
- ✓ 테스트 실행 계획 확인
- ✓ 테스트 케이스 확인
- ✓ 테스트 데이터 확인
- ✓ 테스트 환경 확인

2.

❖ 기능 단위 테스트 수행:

- ✓ 업무별 기능과 테스트 항목 정의
- ✓ 단위 테스트 수행
- ✓ 통합 테스트 수행
- ✓ 성능 테스트 수행 후 결과 보고서 작성

<표 1-7> 쇼핑몰 테스트 케이스 예시

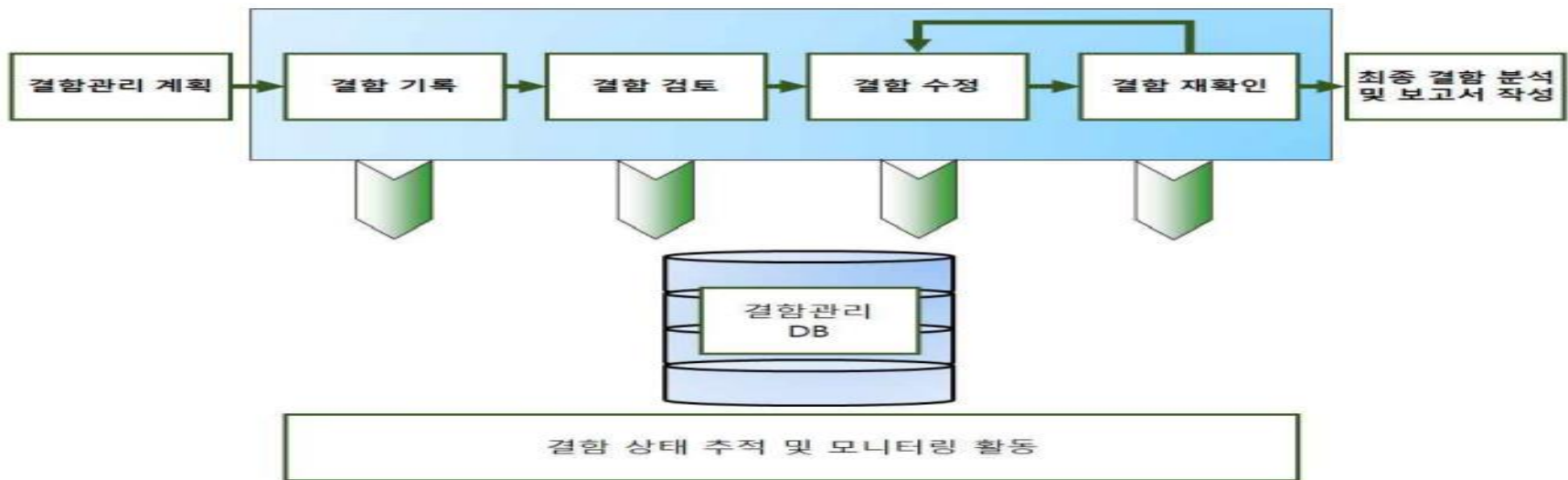
No.	테스트 케이스 명	기능 단위	주요 테스트 내용
1	로그인	서버 모듈	- 회원 로그인 SSL(Secure Socket Layer) 작동 여부
		인터페이스	- 로그인 후 회원에 따른 화면 이동 예시) 6개월 동안 암호 변경 없는 회원은 비밀번호 변경 페이지로 이동
2	상품 선택	화면 모듈	- 장바구니 등록 및 바로 구매 선택 화면 이동 - 로그인인 경우 해당 상품의 상세 페이지 이동 - 로그인아웃 상태인 경우 로그인 화면 이동
		서버 모듈	- 서버 세션 종료 시간 1분으로 설정 후, 자동 로그인아웃 점검
3	상품 구매	인터페이스	- 상품 수량 입력란에 문자 등록 시 오류 또는 안내메시지 표시 - 옵션 상품의 경우 미선택 시 구매 불가 기능 작동 여부
		데이터 입출력	- 입력한 수량에 따른 금액의 정상적인 변경 여부
4	주문서 작성	인터페이스 데이터 입출력	- 주문자 및 배송지 정보의 정상 등록 여부 테스트(이름, 주소, 휴대폰)
		화면 모듈	- 결제 구분 선택(무통장입금, 카드결제)
5	결제	서버 모듈 데이터 입출력	- 세금계산서 및 현금영수증 데이터 등록
		인터페이스 서버 모듈 데이터 입출력	- 결제 금액 데이터의 카드회사 정상 전송 점검 - 카드사별 카드결제 정상 연동 점검
6	주문 완료	서버 모듈	- 세금계산서 및 현금영수증 데이터 등록
		데이터 입출력 화면 모듈	- 주문 내역 이메일 발송 점검 - 주문 내역 조회 주문 데이터 점검 - 주문 취소 연결 모듈 점검

1-2. 결함관리

결함이란? 프로그램과 명세서간의 차이, 업무 내용과 불일치.

기대결과와 실제 테스트한 결과 간의 차이. 변경이 필요한 모든 것.

결함관리 프로세스



- ★ 결함기록 : 테스터는 발견된 결함에 대한 정보를 결함 관리 DB에 기록한다.
- ★ 결함검토: 등록된 결함에 있어서 주요 내용을 검토하고 결함을 수정할 개발자에게 전달한다.
- ★ 결함수정: 개발자는 할당된 결함의 프로그램을 수정한다.
- ★ 결함재확인: 테스터는 개발자에게 전달된 결함을 수정하였는지 확인하고 다시 테스트한다

1-2. 결함관리

결함종류★

1. 시스템 결함★

- 비정상적 종료/중단
- 응답시간지연
- 데이터베이스 에러

2. 기능 결함

- 요구사항 미반영/불일치
- 부정확한 비즈니스 프로세스
- 스크립트 에러
- 타 시스템과 연동 시 오류

3. GUI 결함

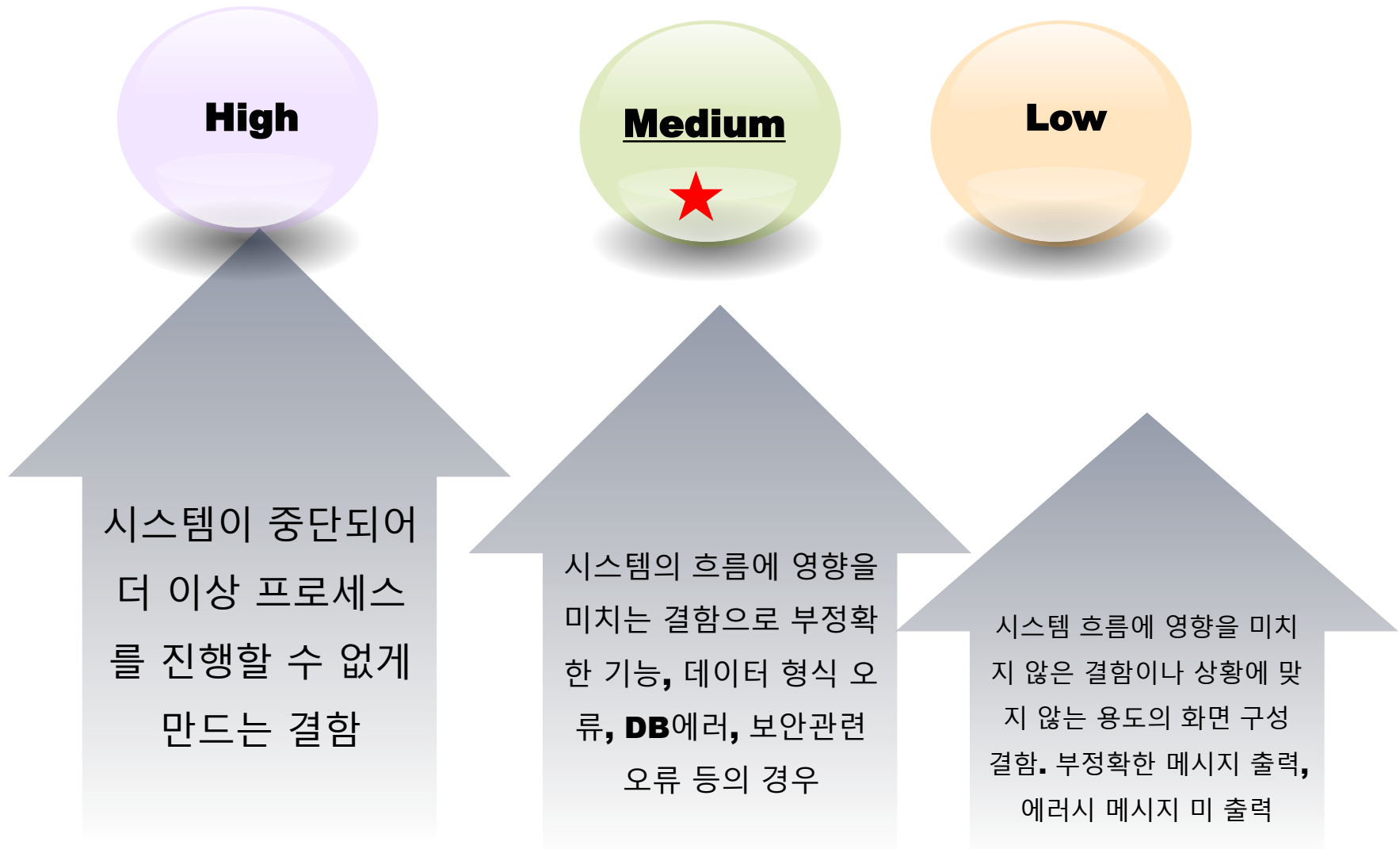
- UI의 비일관성
- 부정확한 커서/메시지
- 데이터 타입의 표시오류

4. 문서 결함

- 매뉴얼 및 요구사항 등의 각종 문서와 불일치

1-2. 결함관리

결함 심각도



2

애플리케이션 결함 조치하기



결함 관리의 이해

에러, 오류, 실패, 결함

결함판단 기준



기능명세서에 가능하다고 명시된 동작을 수행하지 않는 경우

기능 명세서에 불가능하다고 명시된 동작을 수행하는 경우

기능 명세서에 명시 되어 있지 않은 동작을 수행하는 경우

기능 명세서에 명시되어 있지 않지만 수행해야 할 동작을 수행하지 않는 경우

테스터의 시각에서 볼 때 문제가 있다고 판단되는 경우

테스터의 시각에서 결함으로 판단되는 경우 예시

이해하기 어려운 기능, 사용이 까다로운 기능, 비정상적으로 느린 기능 등