

정보처리기사 필기

소프트웨어 개발 7 애플리케이션 테스트 관리①

양문자 선생님

출처 : ncs 학습모듈(NCS능력단위 애플리케이션테스트관리)

소프트웨어 개발

차례

1 데이터 입출력 구현

2 통합 구현

3 제품소프트웨어 패키징

4 애플리케이션 테스트 관리

1) 애플리케이션 테스트케이스 설계

2) 애플리케이션 통합 테스트

3) 애플리케이션 성능 개선

5 인터페이스 구현

테스트 케이스

- 특정한 프로그램의 일부분 또는 경로에 따라 수행하거나, 특정한 요구사항을 준수하는지 확인하기 위해 설계된 입력 값, 실행 조건, 기대 결과로 구성된 테스트 항목의 명세서.

테스트 케이스 작성절차

- (1) 테스트 계획 검토 및 자료 확보
- (2) 위험 평가 및 우선순위 결정
- (3) 테스트 요구사항 정의
- (4) 테스트 구조 설계 및 테스트 방법 결정
- (5) 테스트 케이스 정의
- (6) 테스트 케이스 타당성 확인 및 유지 보수

소프트웨어 개발

테스트 오라클

(1) 테스트 오라클 정의

테스트의 결과가 참인지 거짓인지를 판단하기 위해서 사전에 정의된 참 값을 입력하여 비교하는 기법 및 활동.

(2) 테스트 오라클 유형

(가) 참(True) 오라클

모든 입력 값에 대하여 기대하는 결과를 생성함으로써 발생한 오류를 모두 검출할 수 있는 오라클이다.

(나) 샘플링(Sampling) 오라클

특정한 몇 개의 입력 값에 대해서만 기대하는 결과를 제공해 주는 오라클이다.

(다) 휴리스틱(Heuristic) 오라클

샘플링 오라클을 개선한 오라클로, 특정 입력 값에 대해 올바른 결과를 제공하고, 나머지 값들에 대해서는 휴리스틱(추정)으로 처리하는 오라클이다.

(라) 일관성 검사(Consistent) 오라클

애플리케이션 변경이 있을 때, 수행 전과 후의 결과 값이 동일한지 확인하는 오라클이다.

문제풀이

- 다음이 설명하는 테스트 용어는?

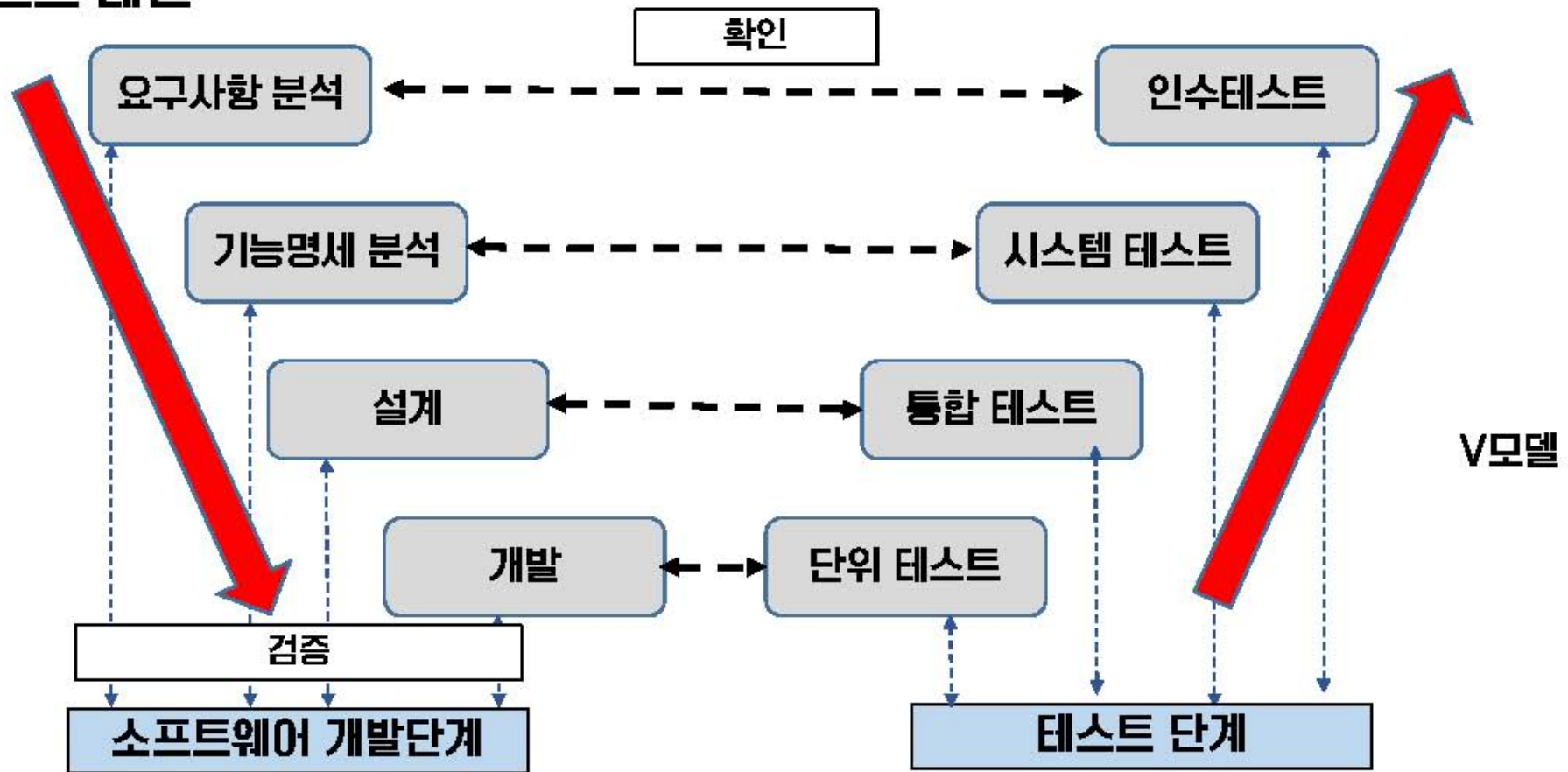
- 테스트의 결과가 참인지 거짓인지를 판단하기 위해서 사전에 정의된 참값을 입력하여 비교하는 기법 및 활동을 말한다.
- 종류에는 참, 샘플링, 휴리스틱, 일관성 검사가 존재한다.

- ① 테스트 케이스 ② 테스트 시나리오
- ③ 테스트 오라클 ④ 테스트 데이터

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 개발)

소프트웨어 개발

테스트 레벨



소프트웨어 개발

테스트 레벨 종류

종류	설명
단위 테스트	<ul style="list-style-type: none">구현된 모듈(함수, 서브루틴, 컴포넌트 등)의 기능 수행 여부를 판정내부에 존재하는 논리적 오류를 검출할 수 있는 방안을 파악
통합 테스트	<ul style="list-style-type: none">모듈 간의 인터페이스 연계를 검증하고, 모듈 간의 인터페이스 오류를 확인모듈 간의 상호 작용 및 연계 동작 여부를 판정하는 방안을 파악상향식, 하향식 테스트
시스템 테스트	<ul style="list-style-type: none">단위, 통합 테스트 후 전체 시스템이 정상적으로 작동하는지 판정하는 기능 명세를 확인하는 방안을 파악기능, 비기능 요구사항 테스트
인수 테스트	<ul style="list-style-type: none">사용자가 요구분석 명세서에 명시된 사항을 모두 충족하는지 판정시스템이 예상대로 동작하고 있는지를 판정하는 방안을 파악알파, 베타 테스트

문제풀이

- 알파, 베타 테스트와 가장 밀접한 연관이 있는 테스트 단계는?

- ① 단위 테스트
- ② 인수 테스트
- ③ 통합 테스트
- ④ 시스템 테스트

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 개발)

테스트 시나리오

- 테스트 수행을 위한 여러 테스트 케이스의 집합으로서, 테스트 케이스의 동작 순서를 기술한 문서이며 테스트를 위한 절차를 명시한 문서이다.
- 테스트 수행 절차를 미리 정함으로써 설계 단계에서 중요시되던 요구사항이나 대안흐름과 같은 테스트 항목을 빠짐없이 테스트하기 위함이다.

테스트 시나리오 작성 시 유의점

- 테스트 시나리오 분리 작성
테스트 항목을 시스템별, 모듈별, 항목별 테스트 시나리오로 분리하여 작성한다.
- 고객의 요구사항과 설계 문서 등을 토대로 테스트 시나리오를 작성한다.
- 각 테스트 항목은 식별자 번호, 순서 번호, 테스트 데이터, 테스트 케이스, 예상 결과, 확인 등의 항목을 포함하여 작성한다.

테스트 지식 체계

소프트웨어 테스트의 원리

- 테스트는 결함이 존재함을 밝히는 활동이다.
- 완벽한 테스트는 불가능하다.
- 테스트는 개발 초기에 시작해야 한다.
- 결함 집중(Defect Clustering)
- 살충제 패러독스(Pesticide Paradox)
- 테스트는 정황(Context)에 의존한다.
- 오류-부재의 궤변(Absence of Errors Fallacy)

소프트웨어 개발

소프트웨어 테스트 원칙 7가지

1. 테스트는 반드시 해당 S/W 개발과 상관없는 자들에 의해서 수행되도록 할 것.
2. 가장 능력이 좋은 사람이 테스트 할 것.
3. 오류가 발견되지 않을 것이란 가정하에서 테스트 계획을 수립하지 말 것.
4. 예상하지 못한 경우들에 대해서도 테스트를 수행할 것.
5. 오류가 남아 있을 확률은 이미 발견된 오류의 수에 직접적으로 비례한다.

<파레토(Pareto) 법칙>

- 시스템의 20% 모듈이 전 자원의 80%를 소비
 - 시스템의 20% 모듈이 실행시간의 80%를 소비
 - 오류의 20%가 오류 수정비용의 80%를 차지
 - 사용도구의 20% 기능만이 도구 사용의 80%를 차지
6. 테스트 케이스를 체계적으로 관리할 것.
 7. 각각의 테스트 결과를 철저하게 점검할 것.

문제풀이

- 소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙은?

- ① Brooks의 법칙
- ② Boehm의 법칙
- ③ Pareto의 법칙
- ④ Jackson의 법칙

(2020년 1, 2회 정보처리기사 필기 기출문제 소프트웨어 개발)

소프트웨어 개발

소프트웨어 테스트의 유형

1. 프로그램 실행 여부

(1) 정적 테스트

프로그램 실행 없이 소스 코드의 구조를 분석하여 논리적으로 검증하는 테스트
(인스펙션, 코드 검사, 워크스루)

(2) 동적 테스트

프로그램의 실행을 요구하는 테스트
(화이트박스 테스트, 블랙박스 테스트)

소프트웨어 개발

2. 테스트 기법

(1) 화이트박스 테스트

- 프로그램의 내부 로직(수행 경로 구조, 루프 등)을 보면서 테스트를 수행한다.
- 기초 경로 검사
- 제어 구조 검사(조건검사, 루프검사, 데이터 흐름 검사)

(2) 블랙박스 테스트

- 프로그램의 외부 사용자 요구사항 명세를 보면서 테스트, 주로 구현된 기능을 테스트 한다.
- 동치 분할 검사 : 타당한 입력자료와 타당하지 않은 입력자료의 개수를 같게 테스트케이스 작성
- 경계값 분석 : 입력조건이 중간값보다 경계값에서 오류가 발생할 확률이 높다는 점을 이용
- 원인-효과 그래프 검사
- 오류 예측 검사
- 비교 검사

문제풀이

- 평가 점수에 따른 성적부여는 다음 표와 같다. 이를 구현한 소프트웨어를 경계값 분석 기법으로 테스트 하고자 할 때 다음 중 테스트 케이스의 입력 값으로 옳지 않은 것은?

평가 점수	성적
80 ~ 100	A
60 ~ 79	B
0 ~ 59	C

① 59 ② 80 ③ 90 ④ 101

(2020년 1, 2회 정보처리기사 필기 기출문제 소프트웨어 개발)

문제풀이

- 블랙박스 테스트의 유형으로 틀린 것은?

- ① 경계값 분석
- ② 오류 예측
- ③ 동등 분할 기법
- ④ 조건, 루프검사

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 개발)

소프트웨어 개발

3. 테스트에 대한 시각

(1) 검증(Verification)

제품의 생산 과정을 테스트한다(Are we building the product right?).
올바른 제품을 생산하고 있는지 검증하는 것을 의미한다.

(2) 확인(Validation)

생산된 제품의 결과를 테스트한다.(Are we building the right product?)
생산된 제품이 정상적으로 동작하는지 확인하는 것을 의미한다.

종류	설명
형상 검사	• 구성 요소, 목록, 유지보수를 위한 모든 사항이 표현되었는지 검사
알파 검사	• 개발자의 장소에서 사용자가 개발자 앞에서 행하는 검사 기법 • 통제된 환경에서 오류, 문제점등을 사용자와 개발자가 함께 확인
베타 검사	• 선정된 최종 사용자가 여러명의 사용자 앞에서 행하는 검사 기법 • 실제 업무를 가지고 사용자가 직접 테스트하며 제어되지 않은 상태에서 검사함

문제풀이

- 검증 검사 기법 중 개발자의 장소에서 사용자가 개발자 앞에서 행하는 기법이며, 일반적으로 통제된 환경에서 사용자와 개발자가 함께 확인하면서 수행되는 검사는?

- ① 동치 분할 검사
- ② 형상 검사
- ③ 알파 검사
- ④ 베타 검사

(2020년 1, 2회 정보처리기사 필기 기출문제 소프트웨어 개발)

문제풀이

- 검증(Validation) 검사 기법 중 개발자의 장소에서 사용자가 개발자 앞에서 행해지며, 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인하면서 검사하는 기법은?

① 디버깅 검사

② 형상 검사

③ 자료구조 검사

④ 알파 검사

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 개발)

코드 최적화

- 읽기 쉽고 변경 및 추가가 쉬운 클린 코드를 작성하는 것으로, 소스코드 품질을 위해 기본적으로 지킬 원칙과 기준을 정의하고 있다.

1. 나쁜 코드(Bad Code)

(1) 개요

- 다른 개발자가 로직(Logic)을 이해하기 어렵게 작성된 코드이다. 대표적인 사례로 처리 로직의 제어가 정제되지 않고 서로 얹혀 있는 스파게티 코드, 변수나 메소드에 대한 이름 정의를 알 수 없는 코드, 동일한 처리 로직이 중복되게 작성된 코드 등이 있다.

(2) 문제점

- 스파게티 코드의 경우 잦은 오류가 발생할 가능성이 있다.
- 소스 코드 이해가 안 되어 계속 덧붙이기할 경우 코드 복잡도가 증가한다.

소프트웨어 개발

2. 클린 코드(Clean Code)

(1) 개요

- 잘 작성되어 가독성이 높고, 단순하며, 의존성을 줄이고, 중복을 최소화하여 깔끔하게 잘 정리된 코드

(2) 효과성

- 중복 코드 제거로 애플리케이션의 설계가 개선된다.
- 가독성이 높으므로 애플리케이션의 기능에 대해 쉽게 이해할 수 있다.
- 버그를 찾기 쉬워지며, 프로그래밍 속도가 빨라진다.

소프트웨어 개발

클린 코드 작성 원칙

원칙	원칙 설명
가독성	<ul style="list-style-type: none">- 이해하기 쉬운 용어를 사용한다.- 코드 작성 시 들여쓰기 기능을 사용한다.
단순성	<ul style="list-style-type: none">- 한 번에 한 가지 처리만 수행한다.- 클래스/메소드/함수를 최소 단위로 분리한다.
의존성	<ul style="list-style-type: none">- 영향도를 최소화한다.- 코드의 변경이 다른 부분에 영향이 없게 작성한다.
중복성	<ul style="list-style-type: none">- 중복된 코드를 제거한다.- 공통된 코드를 사용한다.
추상화	<ul style="list-style-type: none">- 클래스/메소드/함수에 대해 동일한 수준의 추상화를 한다.- 상세 내용은 하위 클래스/메소드/함수에서 구현한다.

소스 코드 최적화 기법의 유형

(1) 클래스 분할 배치 기법을 파악한다.

클래스는 하나의 역할, 책임만 수행할 수 있도록 응집도를 높이고, 크기를 작게 작성한다.

(2) 느슨한 결합(Loosely Coupled) 기법을 파악한다.

클래스의 자료 구조, 메소드를 추상화할 수 있는 인터페이스 클래스를 이용하여, 클래스 간의 의존성을 최소화해야 한다.

(3) 코딩 형식 기법을 파악한다.

줄바꿈으로 개념을 구분, 종속(개념적 유사성 높음) 함수를 사용, 호출하는 함수를 먼저 배치하고 호출되는 함수는 나중에 배치, 지역 변수는 각 함수 맨 처음에 선언할 때 사용하는 등의 형식을 취한다.

(4) 좋은 이름 사용 방법을 파악한다.

기억하기 좋은 이름, 발음이 쉬운 용어, 접두어 사용 등 기본적인 명명 규칙(Naming Rule)을 정의하고 정의된 이름을 사용한다.

(5) 적절한 주석문 사용 방법을 파악한다.

소스 코드 작업 시 앞으로 해야 할 일을 기록하거나, 소스상의 중요한 부분을 강조할 때 사용한다.

문제풀이

- 외계인코드(Alien Code)에 대한 설명으로 옳은 것은?
 - ① 프로그램의 로직이 복잡하여 이해하기 어려운 프로그램을 의미한다.
 - ② 아주 오래되거나 참고문서 또는 개발자가 없어 유지보수 작업이 어려운 프로그램을 의미한다.
 - ③ 오류가 없어 디버깅 과정이 필요 없는 프로그램을 의미한다.
 - ④ 사용자가 직접 작성한 프로그램을 의미한다.

(2020년 1, 2회 정보처리기사 필기 기출문제 소프트웨어 개발)

문제풀이

- 다음에서 설명하는 클린 코드 작성 원칙은?
 - 한 번에 한 가지 처리만 수행한다.
 - 클래스/메소드/함수를 최소 단위로 분리한다.

① 다형성

② 단순성

③ 추상화

④ 의존성

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 개발)

- 다음 중 클린 코드 작성원칙으로 거리가 먼 것은?

① 누구든지 쉽게 이해하는 코드 작성

② 중복이 최대화된 코드 작성

③ 다른 모듈에 미치는 영향 최소화

④ 단순, 명료한 코드 작성

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 개발)

소프트웨어 개발

- 알고리즘 기법

기법	설명
분할정복 (Divide and Conquer)	주어진 문제의 입력을 분할하여 문제를 해결하는 방식
그리디 (Greedy)	가능한 해들 중에서 가장 좋은 해를 찾는 방식(욕심쟁이, 탐욕)
동적계획 (Dynamic Programming)	작은부분 문제들을 먼저 해결한 후에 그 해들을 이용하여 보다 큰 부분 문제들을 해결하고, 최종적으로 원래 주어진 문제를 해결하는 방법
백트래킹 (Backtracking)	해를 찾는 도중 해가 나올 가망이 없으면 되돌아가서 다시 해를 찾아가는 기법

문제풀이

- 알고리즘 설계 기법으로 거리가 먼 것은?

- ① Divide and Conquer
- ② Greedy
- ③ Static Block
- ④ Backtracking

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 개발)