



제출일	2023.06.02	학과	컴퓨터공학전공
과목	S/W품질관리및테스팅	학번	2018112007
담당교수	이호정 교수님	이름	이승현



1. Use-Case Testing은 무엇이며 이것의 한계점에 대해 기술하시오.ㄸ

Use-Case Testing은 소프트웨어 시스템의 기능과 사용자 시나리오 간의 상호작용을 테스트하기 위한 테스트 기법이다. 이 테스트 기법은 사용자 요구사항을 충족하는지 확인하고, 시스템의 동작을 검증하는 데 중점을 둔다. Use-Case Testing은 시스템이 예상된 사용자 시나리오에 따라 올바르게 작동하는지 확인하기 위해 사용자 상호작용을 시뮬레이션하고 검증하는 것을 목표로 한다.

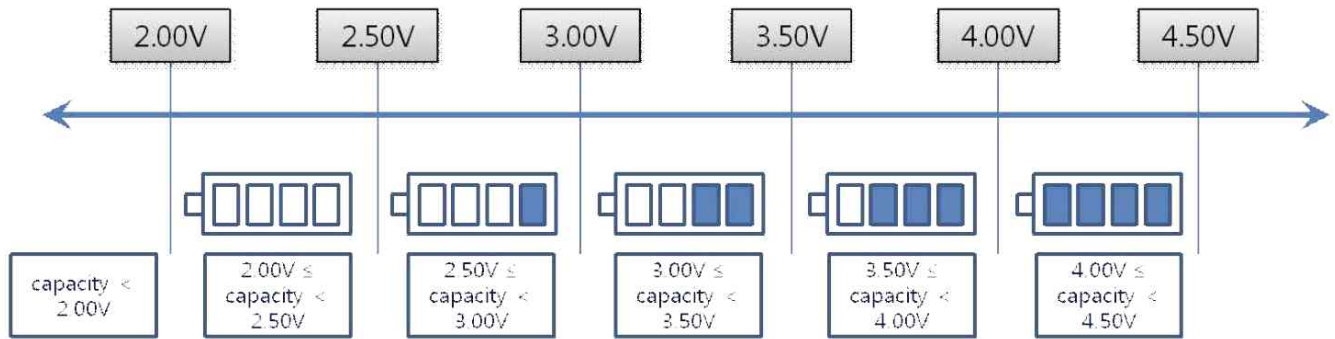
<한계점>

1. 불완전한 시나리오: 모든 가능한 시나리오를 커버하기 어렵다. 사용자의 다양한 동작과 상호작용을 고려하기 어렵기 때문에 일부 시나리오를 놓칠 수 있다.
2. 상호작용 복잡성: 시스템의 기능이 복잡하거나 다양한 시스템 간 상호작용이 있는 경우 Use-Case Testing은 복잡성을 다루기 어렵다.
3. 변경 관리: Use-Case Testing은 사용자 요구사항이 변경될 때마다 테스트 케이스를 업데이트해야 한다. 이는 유지보수 비용을 증가시킬 수 있다.
4. 비즈니스 로직: Use-Case Testing은 시스템의 비즈니스 로직을 다루기 어렵다. 복잡한 비즈니스 규칙과 조건을 테스트하는 데 한계가 있을 수 있다.
5. 선행 조건과 의존성: Use-Case Testing은 시나리오를 수행하기 위해 필요한 선행 조건과 의존성을 관리하기 어렵다. 이로 인해 테스트 환경 설정과 관리가 복잡해질 수 있다.

2. 동등 분할(동치 클래스 분할)과 경계값 분석의 한계점은 무엇인가?

1. 불완전한 테스트 커버리지: 동등 분할과 경계값 분석은 테스트 케이스의 수를 줄여 효율적인 테스트를 가능하게 한다. 그러나 이러한 기법은 모든 가능한 시나리오를 커버하지 않을 수 있다. 특정 동등 클래스나 경계값에 대한 테스트만 수행되므로 다른 중요한 시나리오나 예외 상황을 놓칠 수 있다.
2. 사용자의 실수 간과: 동등 분할과 경계값 분석은 시스템의 외부에서 입력 값을 고려한다. 그러나 사용자의 실수나 잘못된 입력에 대한 테스트는 충분히 수행되지 않을 수 있다. 이러한 실수는 경계값이나 동등 클래스를 벗어나는 입력으로 인해 발생할 수 있으며, 이러한 경우에 대한 테스트 케이스를 추가로 고려해야 한다.
3. 복잡한 시스템에 대한 한계: 동등 분할과 경계값 분석은 단순한 입력과 출력 관계에 적용하기 적합하다. 하지만 복잡한 시스템의 경우 다양한 요소와 상호작용이 복잡하게 얽혀있을 수 있으며, 이러한 경우에는 추가적인 테스트 기법이 필요할 수 있다.
4. 버그 미발견 가능성: 동등 분할과 경계값 분석은 테스트 케이스를 선택하는 데 있어서 효과적이지만, 이들 기법만으로는 모든 버그를 찾아내지 못할 수 있다. 다른 테스트 기법과 함께 사용하여 테스트 커버리지를 향상시키는 것이 중요하다.
5. 유지보수 비용: 동등 분할과 경계값 분석은 초기 단계에서 효과적인 테스트 케이스를 선택하는 데 도움을 준다. 그러나 시스템이 변경될 때마다 이러한 기법에 따라 테스트 케이스를 업데이트해야 한다. 따라서 유지보수 비용이 증가할 수 있다.

3. 다음 조건을 갖는 충전 상황을 동등 분할하여 테스트하려고 한다. 이 조건에 맞는 테스트 케이스를 작성시오.



<동치 클래스 분해>

입력 조건	정상적 EC	비정상적 EC
capacity	V1: 4.00V ≤ capacity < 4.50V V2: 3.50V ≤ capacity < 4.00V V3: 3.00V ≤ capacity < 3.50V V4: 2.50V ≤ capacity < 3.00V V5: 2.00V ≤ capacity < 2.50V	I1: 4.50V ≤ capacity I2: capacity < 2.00V

<테스트 케이스>

테스트케이스	입력	예상 결과
1	capacity = 5.00V	invalid
2	capacity = 4.50V	invalid
3	capacity = 4.25V	4칸
4	capacity = 4.00V	4칸
5	capacity = 3.75V	3칸
6	capacity = 3.50V	3칸
7	capacity = 3.25V	2칸
8	capacity = 3.00V	2칸
9	capacity = 2.75V	1칸
10	capacity = 2.50V	1칸
11	capacity = 2.25V	0칸
12	capacity = 2.00V	0칸
13	capacity = 1.75V	invalid
14	capacity = 0.00V	invalid
15	capacity = -1.00V	invalid

4. 소프트웨어 테스팅을 가로막는 장애는 어떤 것들이 있는가?

1. 제한된 시간과 자원: 테스트를 수행하는 데 필요한 시간과 자원이 제한되어 있는 경우 테스트 커버리지를 충분히 확보하기 어려울 수 있다. 이는 모든 시나리오를 충분히 테스트하지 못하고 중요한 결함을 놓칠 수 있는 원인이 될 수 있다.
2. 부적절한 테스트 환경: 테스트를 위한 적절한 환경을 구축하는 데 어려움이 있는 경우 테스트 프로세스가 방해받을 수 있다. 테스트 환경의 제약사항, 액세스 권한 부족, 테스트 데이터의 부재 등은 효과적인 테스트를 방해할 수 있다.
3. 부정확한 또는 불완전한 요구사항: 테스트는 요구사항을 충족시키는지 확인하기 위해 수행되므로, 부정확하거나 불완전한 요구사항은 테스트의 품질을 저하시킬 수 있다. 명확하고 완전한 요구사항이 제공되지 않으면 테스트 케이스 작성과 검증이 어려워질 수 있다.
4. 복잡성과 상호의존성: 복잡한 소프트웨어 시스템이나 다양한 모듈 간의 상호의존성은 테스트를 어렵게 만들 수 있다. 시스템의 복잡성이 증가하면 테스트 커버리지를 확보하고 테스트 케이스를 관리하는 것이 더 어려워질 수 있다.
5. 결함 추적 및 관리의 어려움: 테스트 중 발견된 결함을 추적하고 관리하는 프로세스의 부재는 테스트 효과성을 방해할 수 있다. 결함 관리 도구의 부재, 테스트 결과와 결함 추적의 연계 부재 등은 테스트 결과의 품질과 추적성을 저하시킬 수 있다.
6. 테스트 자동화의 한계: 테스트 자동화는 효율적인 테스트를 위해 필수적이다. 그러나 모든 종류의 테스트를 자동화하는 것은 어려울 수 있다. 특히 사용자 인터페이스(UI) 테스트, 비기능적인 테스트 등은 자동화하기 어렵거나 제한적일 수 있다.
7. 인적 요인: 인적 요소도 테스트에 영향을 줄 수 있다. 테스터의 기술 수준, 경험 부족, 의사소통의 문제, 팀 간 협업의 어려움 등은 테스트 품질과 생산성에 영향을 미칠 수 있다.

5. 다음은 페어와이즈 조합 테스팅에 관련한 질문이다.

(가). 페어와이즈 조합 테스팅과 직교배열 테스팅의 공통점과 차이점은 무엇인가?

<공통점>

1. 효율적인 테스트 커버리지: 페어와이즈 조합 테스팅과 직교배열 테스팅은 입력 변수 간의 상호작용을 고려하여 효율적인 테스트 케이스 조합을 선택한다. 이는 적은 수의 테스트 케이스로 다양한 조합을 커버하며, 결함을 발견할 확률을 높인다.
2. 결함 탐지 용이성: 둘 다 입력 변수 간의 결함 및 상호작용에 대한 테스트를 진행하므로 결함을 탐지하고 식별하는 데 도움이 된다. 이는 테스트의 품질을 향상시키고 시스템의 신뢰성을 높인다.

<차이점>

1. 테스트 케이스 생성 방법: 페어와이즈 조합 테스팅은 각 변수 간의 가능한 모든 조합을 찾아내고 해당 조합에 대한 테스트 케이스를 생성하는 방법이다. 직교배열 테스팅은 입력 변수의 레벨을 기반으로 테스트 배열을 생성하여 효율적인 조합을 선택하는 방법이다.
2. 변수 조합 범위: 페어와이즈 조합 테스팅은 변수 간의 모든 가능한 조합을 고려하므로, 변수의 수와 레벨이 증가하면 테스트 케이스의 수가 증가한다. 직교배열 테스팅은 변수의 레벨을 기준으로 테스트 배열을 생성하므로 변수의 수와 레벨에 상관없이 테스트 케이스의 수가 일정하다.

(나). 다음은 특정 시스템을 설계했을 때의 작동에 관련한 매개변수를 표로 작성하였다. 이를 기반으로 하여 페어와이즈 조합을 만드는 과정을 보이시오.

동작모드	설정	이퀄라이저
순차	Hold	Off
순차	Hold	Live
순차	착신	Off
순차	착신	Live
순차반복	Hold	Off
순차반복	Hold	Live
순차반복	착신	Off
순차반복	착신	Live

<표-1>

1.

동작모드	설정		동작모드	이퀄라이저		설정	이퀄라이저	동작모드	설정	이퀄라이저
순차	Hold		순차	Off		Hold	Off	순차	Hold	Off
순차	착신		순차	Live		Hold	Live	순차	착신	
순차반복	Hold		순차반복	Off		착신	Off	순차반복	Hold	
순차반복	착신		순차반복	Live		착신	Live	순차반복	착신	

Off: 2개의 상호작용 커버 -> 선택

Live: 2개의 상호작용 커버

2.

동작모드	설정		동작모드	이퀄라이저		설정	이퀄라이저	동작모드	설정	이퀄라이저
순차	Hold		순차	Off		Hold	Off	순차	Hold	Off
순차	착신		순차	Live		Hold	Live	순차	착신	Live
순차반복	Hold		순차반복	Off		착신	Off	순차반복	Hold	
순차반복	착신		순차반복	Live		착신	Live	순차반복	착신	

Off: 1개의 상호작용 커버

Live: 2개의 상호작용 커버 -> 선택

3.

동작모드	설정		동작모드	이퀄라이저		설정	이퀄라이저	동작모드	설정	이퀄라이저
순차	Hold		순차	Off		Hold	Off	순차	Hold	Off
순차	착신		순차	Live		Hold	Live	순차	착신	Live
순차반복	Hold		순차반복	Off		착신	Off	순차반복	Hold	Live
순차반복	착신		순차반복	Live		착신	Live	순차반복	착신	Live

Off: 1개의 상호작용 커버

Live: 2개의 상호작용 커버 -> 선택

4.

동작모드	설정		동작모드	이퀄라이저		설정	이퀄라이저	동작모드	설정	이퀄라이저
순차	Hold		순차	Off		Hold	Off	순차	Hold	Off
순차	착신		순차	Live		Hold	Live	순차	착신	Live
순차반복	Hold		순차반복	Off		착신	Off	순차반복	Hold	Live
순차반복	착신		순차반복	Live		착신	Live	순차반복	착신	Off

Off: 2개의 상호작용 커버 -> 선택

Live: 0개의 상호작용 커버

<최종 페어와이즈 조합>

동작모드	설정	이퀄라이저
순차	Hold	Off
순차	착신	Live
순차반복	Hold	Live
순차반복	착신	Off

(다). 다음 <표-2>은 <표-1>을 기반으로 하여 직교 배열 테스트 기법으로 얻은 결과라고 한다. 이것이 직교 배열 조합에 맞는 것인지 확인하십시오.

동작모드	설정	이퀄라이저
순차	Hold	Off
순차	착신	Live
순차반복	Hold	Live
순차반복	착신	Off

(라). <표-3>을 기반으로 Allpairs.exe를 통해 얻은 것이 <표-4>이다. 검증팀(V&V Team: validation and verification team)에서 확인한 결과에 의하면 <표-3>의 내용을 분실하였다고 한다. 왼쪽 <표-3>에 들어갈 내용을 옳게 기입하십시오.

Mode	Volume	EQ
순차	크게	Classic
순차반복	보통	Rock
1곡반복	작게	Ballad
무한반복	~크게	Dance

표-3 테스트 매개변수

case	Mode	Volume	EQ	pairings
1	순차	크게	Classic	3
2	순차	보통	Rock	3
3	순차	작게	Ballad	3
4	순차반복	보통	Classic	3
5	순차반복	크게	Rock	3
6	순차반복	크게	Ballad	2
7	순차반복	작게	Dance	3
8	1곡반복	작게	Classic	3
9	1곡반복	크게	Rock	2
10	1곡반복	보통	Ballad	3
11	무한반복	작게	Rock	3
12	무한반복	크게	Dance	3
13	무한반복	보통	Classic	2
14	순차	보통	Dance	2
15	1곡반복	~크게	Dance	1
16	무한반복	~크게	Ballad	1

표-4 Allpairs.exe로 만들어진 결과