[소프트웨어공학 개요]

소프트웨어공학 방법(method)과 구별하여 방법론(methodology)을 설명하시오.

보기

양식의 맨 위

양식의 맨 아래

[확인](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week01/quiz01.html)

정답 및 해설

정답 및 해설

**정답 :**

여러 방법과 기술들이 정해진 프로세스 안에서 함께 묶인 것이 방법론이다.   
즉 **프로세스(what)와 방법(how)을 함께 묶은 것**이다. 방법론의 목적은 문제 해결을 위해 사용해야 하는 방법과 기술을 지정함으로써 특정 방식의 해법을 권장하는 것이다.

**해설 :**

방법은 행위를 실현할 때 따라야 하는 일반적 가이드라인 또는 구조화된 접근 방법이다.   
도구는 방법 또는 방법론의 적용을 지원하기 위해 개발된다.

소프트웨어의 속성 중 dependability의 의미를 설명하시오.

보기

양식의 맨 위

정답 및 해설

정답 및 해설

**정답 :**

**신뢰도(reliability)**와 **보안성** 및 **안전성**을 포괄하는 속성이다.

**해설 :**

일정 기간 의도된 기능을 고장 없이 실행하는 능력(신뢰도), 시스템이 고장이 나도 생명이나 경제적 손실이 없는 정도(안전성) 및 보안 위협에도 시스템이 손상되지 않는 정도(보안성)을 포괄한다.

소프트웨어공학에서 직면하고 있는 주요 과제로 어떤 것들이 있는지 생각해 보시오.

보기

양식의 맨 위

양식의 맨 아래

[확인](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week01/quiz03.html)

정답 및 해설

정답 및 해설

**정답 :**

레가시 시스템의 유지보수, 운영 환경의 이질성, 품질과 속도의 양면성 등

**해설 :**

과거에 개발되었으나 아직도 유용하게 사용되는 레가시 시스템의 유지보수 문제, 분산된 환경에서 컴퓨터나 운영체제에 독립적으로 동작하게 하는 문제, 빠르게 변화하는 시장에서 품질을 보장하면서도 빠르게 개발하는 문제 등을 생각해 볼 수 있다.

[1강 요약]

* 소프트웨어 공학은 **소프트웨어 생산**의 모든 관점을 다루는 공학 분야이다.
* 소프트웨어는 **프로그램과 관련 문서 및 실행에 필요한 데이터**를 포함한다.
* 좋은 소프트웨어 제품이 되기 위해서는 **유지보수성, 신뢰성(dependability), 보안성, 효율성 및 수용성(acceptability)**이 보장되어야 한다.
* 소프트웨어 프로세스는 소프트웨어 개발과 관련한 모든 활동을 포함하며 기본적으로 **명세, 개발, 검증(validation)과 진화** **(evolution)**로 이루어진다.
* 시스템의 유형에 따라 개발에 사용되는 도구와 기술이 달라질 수 있다.
* 소프트웨어공학의 기본 개념은 모든 유형의 소프트웨어 시스템에 적용될 수 있다.
* 소프트웨어 엔지니어는 기술적 책무 외에도 **윤리적 책임의식**을 가져야 한다.

2강소프트웨어 프로세스

폭포수 모델의 특성을 설명하고 어떤 경우 사용하면 좋은지에 관해 설명하시오.

보기

양식의 맨 위

양식의 맨 아래

[확인](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week02/quiz01.html)

정답 및 해설

정답 및 해설

**정답 :**

폭포수 모델은 각 단계가 분리되어 독립적이기 때문에 유연하지 못하고, 결국 고객의 바뀌는 요구사항에 대응하기가 어렵다. 따라서 이 모델은 **요구사항이 이미 명확히 나온 경우** 사용하는 것이 적당하다.

**해설 :**

원칙적으로 폭포수 모델의 모든 단계는 병행되거나 거슬러 반복됨이 없이 진행된다.   
실제 작업에서는 수정을 위한 재작업을 위해 앞 단계로의 피드백을 피하기 어렵다.

Q2V&V에서 verification과 validation을 구별하여 설명하시오.

보기

양식의 맨 위

양식의 맨 아래

[확인](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week02/quiz02.html)

정답 및 해설

정답 및 해설

**정답 :**

Verification은 **결과물을 주어진 명세에 따라 만들어졌는지 확인하는 것**으로 “우리가 제품을 올바르게 만들고 있는가?”를 의미한다. Verification은 결과물이 사용자 또는 고객이 원하는 대로 만들어졌는지 검증하는 것으로 “우리가 올바른 제품을 만들고 있는가?”를 의미한다.

**해설 :**

코드 리뷰나 인스펙션과 같은 정적인 검토 방법과 동적인 프로그램 테스팅은 모두 V&V를 위한 방법이다.

Q3프로토타입을 사용하는 이유로 맞은 것에 O, 틀린 것에 X를 표시하시오.

초기 단계에 적은 비용으로 다양한 시도들을 할 수 있도록 해준다.

잘 이해가 되는 부분을 중점적으로 구현하여 완성시켜 나간다.

예외와 버그 처리를 위한 테스트를 수행하여 결함 수를 줄이는 것이 중요하다.

실제 개발에 도움이 되기 위한 프로토타입 상세 문서를 마련해야 한다.

[확인](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week02/quiz03.html)

정답 및 해설

정답입니다.

**정답 : O, X, X, X**

두번째는 점증적 개발에 해당하는 것이다. 프로토타입은 요구사항 추출과 같은 특정 목적을 달성하기 위한 것으로 대개 예외 처리나 상세 문서화 같은 작업을 수행하지 않는다.

정답 및 해설

오답입니다.

**정답 : O, X, X, X**

두번째는 점증적 개발에 해당하는 것이다. 프로토타입은 요구사항 추출과 같은 특정 목적을 달성하기 위한 것으로 대개 예외 처리나 상세 문서화 같은 작업을 수행하지 않는다.

[인쇄하기](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week02/sum01.html)

* **소프트웨어 프로세스**는 소프트웨어 생산을 위한 일련의 활동이며 **프로세스 모델**은 실제 프로세스를 추상화하여 표현한 것이다.
* 일반적 프로세스 모델은 프로세스의 구성을 기술하고 있으며, 예로서 폭포수 모델, 점증적 모델, 재사용 컴포넌트의 통합과 설정 등이 있다.
* **요구 공학**은 소프트웨어 명세서를 개발하는 과정으로 고객의 요구를 개발자에게 알려주는 과정이다.
* 소프트웨어의 확인과 검증은 명세서에 맞게 개발되었는지, 고객의 실제 요구를 충족하는지 검사하는 과정이다.
* **소프트웨어 진화**는 새로운 요구사항을 위해 기존 소프트웨어 시스템을 변경할 때 발생한다.
* 프로세스는 변경 요구에 대응할 수 있도록 구성해야 하며, 프로토타이핑 방법이나 점증적 개발 프로세스를 적용할 수 있다.

3강애자일 소프트웨어 개발

Q1올해 새로이 창립된 A업체는 은행에서 장기간에 걸쳐 사용할 ATM관리 시스템을 개발 중이다.  
이 시스템에서 단말기 관리 부분은 인도 소재 업체에 외주를 맡겼으며, 시스템은 금융감독위원회에 의해 감사를 받아야 한다.   
이 시스템에는 어떤 접근 방법이 적당하며, 그 이유는 무엇인가?

보기

양식의 맨 위

정답 및 해설

정답 및 해설

**정답 :**

이 시스템에는 계획 기반 접근법이 유효하다.

**해설 :**

ㆍ대형의 시스템 – 대규모의 개발조직이 투입됨  
ㆍ크리티컬 시스템 – 복잡한 요구사항을 사전에 검토할 필요 있음  
ㆍ분리된 개발 조직 – 기민한 접근법 적용하기 어려움  
ㆍ오래 사용될 시스템 – 유지보수를 위한 상세한 문서화 요구  
ㆍ새로이 만들어진 개발 조직 – 개발자들의 수준이 높지 않을 가능성  
ㆍ외부 공개 필요 – 감사를 위한 구체적 문서 필요

다음은 XP와 애자일 방법의 원칙들이다. 틀린 항목을 모두 고르시오.

설문지에 가능한 한 많은 내용을 담는 것이 좋다.

* [1](javascript:;)점증적 전달
* [**2**](javascript:;)고객의 개입
* [3](javascript:;)프로세스 중시
* [4](javascript:;)확정 산출물에 대한 변경 불가
* [**5**](javascript:;)단순함의 유지

[확인](javascript:;)

정답 및 해설

정답입니다.

**정답 : 3번, 4번**

* ㆍ프로세스 보다는 사람 :
* 짝 프로그래밍을 통해 지원하며 코드에 대한 공동의 책임을 지고 과도한 작업 시간을 피할 수 있다.
* ㆍ변화의 수용 :
* 규칙적인 릴리스를 통해 가능하며 테스트 후 통합을 실현한다.

정답 및 해설

오답입니다.

**정답 : 3번, 4번**

* ㆍ프로세스 보다는 사람 :
* 짝 프로그래밍을 통해 지원하며 코드에 대한 공동의 책임을 지고 과도한 작업 시간을 피할 수 있다.
* ㆍ변화의 수용 :
* 규칙적인 릴리스를 통해 가능하며 테스트 후 통합을 실현한다.
* 테스트 선행 개발이 요구사항을 이해하는데 도움을 주는 이유는 무엇인지 설명해 보시오.
* **정답 :**
* 테스트를 작성하기 위해서는 요구사항의 의도가 무엇인지를 세밀히 분석해야 하기 때문이다. 요구사항이 불완전하면 많은 경우에 테스트를 작성하는 것이 불가능하다.
* Q4다음 스프린트에 대한 설명이 맞으면 O, 틀리면 X하시오.
* 각각의 스프린트는 2~4주 정도로 정해지는 ‘기간’이다. 이 기간은 XP에서 이야기하는 하나의 릴리즈에 대응된다.
* 스프린트의 모든 활동은 백로그(backlog)를 통해 기록이 남게 된다.
* 선택 단계에서 구성원들이 목표에 대한 합의를 거치고 나면, 팀 구성원들은 목표를 달성하기 위해 새로이 팀을 구성하여, 이들의 대외적 의사소통은 누구나 할 수 있기 때문에 매우 효율적이고 즉각적이라는 장점이 있다.
* 스크럼 마스터는 스프린트 동안 외부로부터 전달되는 추가적인 요구사항을 받아들여서 효과적으로 개발 조직에 전파하는 역할을 맡게 된다.
* 정답 및 해설
* 정답입니다.
* **정답 : O, X, X, X**
* 백로그(backlog)는 스프린트 계획의 출발시점에 프로젝트에서 수행되어야 할 일들을 목록화한 것이다. 선택 단계에서 구성원들이 목표에 대한 합의를 거치고 나면, 팀 구성원들은 목표를 달성하기 위해 새로이 팀을 구성한다. 개발 기간 동안 이 팀은 고객이나 나머지 개발조직과 분리되어 따로 일하며, 이들의 대외적 의사소통은 '스크럼 마스터' 역할을 맡은 사람을 통해서만 이루어지게 된다. 스크럼 마스터는 개발 조직을 외부로부터 보호한다. 해당 스프린트 동안 스크럼은 추가적인 요구를 받아들이지 않는다.
* **정답 : O, X, X, X**
* 백로그(backlog)는 스프린트 계획의 출발시점에 프로젝트에서 수행되어야 할 일들을 목록화한 것이다. 선택 단계에서 구성원들이 목표에 대한 합의를 거치고 나면, 팀 구성원들은 목표를 달성하기 위해 새로이 팀을 구성한다. 개발 기간 동안 이 팀은 고객이나 나머지 개발조직과 분리되어 따로 일하며, 이들의 대외적 의사소통은 '스크럼 마스터' 역할을 맡은 사람을 통해서만 이루어지게 된다. 스크럼 마스터는 개발 조직을 외부로부터 보호한다. 해당 스프린트 동안 스크럼은 추가적인 요구를 받아들이지 않는다.
* 기민한 방법론은 빠른 개발, 잦은 릴리스 배포, 양질의 코드 생산에 그 주안점을 두고 있으며,   
  **고객을 직접 개발 과정에 참여**시키는 특징이 있다.
* 기민한 방법론을 적용할 것이냐, 계획 기반 방법론을 적용할 것이냐는 **개발 대상 소프트웨어의 유형, 개발 조직의 능력과 개발 조직의 문화 등에 의해 결정**되어야 한다.
* XP는 잘 알려진 애자일 방법의 일종으로, **빈번한 릴리스, 소프트웨어의 지속적 개선, 고객의 개발 조직에 참여, 짝 프로그래밍, 테스트 선행 개발** 등의 실천 기술을 장려하고 있다.
* XP의 강점은 프로그래밍에 앞서 **자동화된 테스트를 수행**하는 데서 나온다.   
  - 점증의 결과물을 시스템에 통합하기 위해서는 이러한 테스트들이 모두 통과되어야 한다.
* Scrum은 **프로젝트 관리를 중시**하는 애자일 방법의 일종이다.   
  - Scrum은 하나의 점증을 완성하기 위해 스프린트라는 짧은 작업 단위를 반복함으로써 수행된다.

4강 요구 공학

Q1다음 중 기능적 요구사항에 대한 설명이 아닌 것은 무엇인가?

* [**1**](javascript:;)시스템 속성이나 제약사항에 대한 기술
* [**2**](javascript:;)시스템이 제공해야 하는 서비스에 대한 기술
* [**3**](javascript:;)특정 입력에 대해 시스템이 어떻게 반응해야 하는지에 대한 기술
* [**4**](javascript:;)시스템이 하지 말아야 할 것(what the system should not do)에 대한 기술

정답 및 해설

정답입니다.

**정답 : 1번**

**해설 :** 시스템 속성이나 제약사항은 비기능적 요구사항에 대한 설명이다.

Q2자연어는 모호하고, 너무 유연할 뿐 아니라 모듈화가 어렵기 때문에 시스템 요구사항을 기술하는데 사용하기 어렵다.  
자연어를 사용하는 경우의 문제점을 해소하기 위한 대안들은 무엇이 있는가?

보기

양식의 맨 위

정답 및 해설

정답 및 해설

표준 형식이나 템플릿을 정의하여 구조화된 방식의 표기법을 사용하여 요구사항을 명세하거나, 프로그래밍 언어의 표현을 빌려 시스템의 모델을 기술할 수 있다. 또 그래픽 표기법을 사용하거나 수학적인 방법을 사용하여 자연어의 단점을 해소시킬 수도 있다.

Q3요구사항 관리가 필요한 이유는 무엇인가?

정답 및 해설

정답 및 해설

**정답 :**

요구사항은 불완전(incomplete)하며, 불일치(inconsistency)가 존재할 수밖에 없으므로 요구사항의 변화를 추적하고 제어할 수 있어야 한다.

**해답 :**

비즈니스 자체가 변화하거나 점차적으로 시스템에 대한 사용자의 이해도가 높아져서 요구사항이 변경되거나   
새로운 요구사항이 추가될 수 있다. 또 서로 다른 관점 때문에 다양한 요구사항이 발생할 수 있고 이 때문에 모순이 발생하기도 한다.

* 요구사항은 시스템이 **제공해야 서비스**와 동작 중이나 개발 중의 **제약사항**에 관한 기술이다.
* **사용자 요구사항**은 시스템이 무엇을 해야 하는지에 대한 고수준 명세이며, 자연어나 다이어그램으로 표현되고 고객을 위해 작성된다.
* **시스템 요구사항**은 사용자 요구사항의 내용을 보다 상세화한 것으로 구조화된 자연연어, 설계 언어, 시스템 모델 등으로 표현하고 설계 작업을 위해 작성되며 계약서의 일부가 된다.
* 요구 공학은 요구사항 추출, **요구사항 분석, 요구사항의 검증, 요구사항의 관리**를 포함하는 프로세스로 요구사항을 정의하기 위한 것이다.
* 요구사항 추출은 요구사항 발견, **요구사항 분류와 조직화, 요구사항의 우선순위 매기기와 타협, 요구사항의 문서화 활동**으로 이루어진다.
* 시스템의 이해당사자는 시스템에 직간접적으로 영향을 받으며, 타당한 관심을 가지고 있는 사람이나 조직을 말한다.

5강 시스템 모델링

Q1다음 시스템 모델에 대한 설명 중 잘못된 것은 무엇인가?

설문지에 가능한 한 많은 내용을 담는 것이 좋다.

* [**1**](javascript:;)아키텍쳐 모델은 개발할 시스템과 다른 시스템들 간의 관계를 보여준다.
* [2](javascript:;)DFD(Data flow diagrams)는 시스템의 정적인 데이터 구조를 모델링하는데 사용된다.
* [**3**](javascript:;)실시간 임베디드 시스템의 이벤트 처리를 보여주기 위해 상태 머신 다이어그램을 사용한다.
* [**4**](javascript:;)배경(context) 모델은 시스템 전체를 추상화하여 보여주는 모델이다.

[확인](javascript:;)

정답 및 해설

정답입니다.

**정답 : 2번**

**해설 :** DFD는 시스템의 데이터 처리를 모델링하는데 주로 사용되며, 정적인 데이터 구조를 모델링하는데 사용되는 것은   
ERA(Entity-relation-attribute) 모델과 같은 데이터 모델이다. DFD는 UML의 다이어그램이 아니다

Q2시퀀스 다이어그램과 비교하여 상태 다이어그램이 표현하는 것은 무엇인가?

양식의 맨 위

양식의 맨 아래

[확인](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week05/quiz02.html)

정답 및 해설

정답 및 해설

시퀀스 다이어그램이 유스케이스를 실현하기 위한 여러 객체 사이의 협력을 표현하는 것에 비해 상태 다이어그램은   
단일 객체의 상태 변화를 표현하여 객체의 행위를 모델링한다.

* 시스템 모델은 **시스템을 추상화하여 표현한 것**으로 시스템의 전체 배경, 상호작용, 구조 및 행위를   
  보이기 위해 사용된다.
* **배경 모델**은 동작 환경에서 시스템의 위치를 보여주거나(**아키텍처**) 전체 업무 프로세를 표현한다(**액티비티 다이어그램**).
* **행위 모델**은 시스템의 데이터 처리(**액티비티 다이어그램**)나 이벤트 발생에 대한 시스템의 행위(**상태 다이어그램**)를 표현한다.
* **유스케이스**는 시스템과 외부 액터 간의 상호작용을 기술한 것이며 **시퀀스 다이어그램**은 유스케이스를 실현하기 위한 객체들 간의 상호작용을 보여준다.
* **구조 모델**은 구성 요소 및 그들 간의 관계를 보여주며 클래스 다이어그램이 사용될 수 있다. **시스템 아키텍처**도 구조 모델에 해당한다.
* **모델-기반 공학(MDE)**은 시스템 모델로부터 자동으로 코드를 변환하려는 소프트웨어 개발 방법의 하나이다.
* Q3UML에서 스테레오 타입이란 무엇인가?
* **정답 :**
* UML 요소의 의미를 바꾸거나 명확하게 하기 위한 방법이다.   
  예를 들어 《 parallel 》 와 같이 키워드를 《 》 로 감싸 표현하며 특별한 아이콘을 사용하여 표현하기도 한다.
* **해답 :**
* 스테레오 타입은 거의 모든 UML 요소에 적용될 수 있다. 제약 조건(constraints), 태그 값(tagged value)과 함께   
  UML 확장법의 하나이다.
* Q4복합 상태(superstate)란 무엇인가?
* 정답 및 해설
* 하나의 상태를 세분화하여 복수 개의 상태 즉 하나의 상태 다이어그램으로 표현한 것이다.   
  복합 상태는 하나 이상의 상태 머신 다이어그램으로 표현될 수도 있는데 이 경우는 병행 상태를 의미한다.
* 시스템 모델은 **시스템을 추상화하여 표현한 것**으로 시스템의 전체 배경, 상호작용, 구조 및 행위를   
  보이기 위해 사용된다.
* **배경 모델**은 동작 환경에서 시스템의 위치를 보여주거나(**아키텍처**) 전체 업무 프로세를 표현한다(**액티비티 다이어그램**).
* **행위 모델**은 시스템의 데이터 처리(**액티비티 다이어그램**)나 이벤트 발생에 대한 시스템의 행위(**상태 다이어그램**)를 표현한다.
* **유스케이스**는 시스템과 외부 액터 간의 상호작용을 기술한 것이며 **시퀀스 다이어그램**은 유스케이스를 실현하기 위한 객체들 간의 상호작용을 보여준다.
* **구조 모델**은 구성 요소 및 그들 간의 관계를 보여주며 클래스 다이어그램이 사용될 수 있다. **시스템 아키텍처**도 구조 모델에 해당한다.
* **모델-기반 공학(MDE)**은 시스템 모델로부터 자동으로 코드를 변환하려는 소프트웨어 개발 방법의 하나이다.

6강 아키텍처 설계

Q1시스템 명세를 작성할 때 시스템의 아키텍처를 먼저 설계하는 이유를 설명하시오.

정답 및 해설

정답 및 해설

아키텍처를 이용하여 명세들을 구조화시키고 여러 서브시스템들의 명세를 동시에 작성하기 위해 필요하다.   
필요한 경우 하드웨어의 제작을 하청시킬 수 있으며 시스템 개발 비용을 추정하기 위한 기초 자료가 될 수 있다.

Q2다음 중 클라이언트-서버 아키텍처에 대한 설명이 아닌 것을 모두 고르시오.

설문지에 가능한 한 많은 내용을 담는 것이 좋다.

* [1](javascript:;)대규모 데이터를 공유하는데 효율적이다.
* [2](javascript:;)데이터의 분산이 용이하다.
* [**3**](javascript:;)새로운 서버를 추가하거나 기존의 서버를 업그레이드하기 쉽다.
* [4](javascript:;)데이터 진화(data evolution)가 어렵고 비용이 많이 든다.

[확인](javascript:;)

정답 및 해설

정답입니다.

**정답 : 1, 4번**

**해설 :** 클라이언트-서버 모델은 새로운 서버를 추가하거나 기존의 서버를 업그레이드하기 쉽지만, 공유된 데이터 모델이   
없기 때문에 데이터의 교환이 비효율적이다. 그러나 네트워크를 효율적으로 활용하기 위한 데이터의   
분산이 용이하다. ①과 ④는 저장소 아키텍처에 대한 설명이다

정답 및 해설

오답입니다.

**정답 : 1, 4번**

**해설 :** 클라이언트-서버 모델은 새로운 서버를 추가하거나 기존의 서버를 업그레이드하기 쉽지만, 공유된 데이터 모델이   
없기 때문에 데이터의 교환이 비효율적이다. 그러나 네트워크를 효율적으로 활용하기 위한 데이터의   
분산이 용이하다. ①과 ④는 저장소 아키텍처에 대한 설명이다

Q3철도 승객을 위한 티켓 발급 시스템에 적당한 시스템 구성은 무엇이라 생각하는가? 또 비디오, 오디오, 데이터를 동시에   
여러 참여자에게 제공하는 화상회의 시스템에 적당한 아키텍처는 무엇이라 생각하는가?

정답 및 해설

정답 및 해설

첫 번째는 정보의 변경을 즉시 반영할 수 있는 중앙집중식 저장소 아키텍처가 적당하며,   
두 번째는 각 멀티미디어 데이터를 처리하는 서버를 가지는 클라이언트-서버 아키텍처가 적당하다.

[인쇄하기](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week06/sum01.html)

* 소프트웨어 아키텍처는 **시스템 구성과 조직화에 관한 기술**로 성능, 보안, 가용성과 같은 시스템 속성에   
  영향을 준다.
* 아키텍처 설계 시 결정해야 하는 것으로는 **애플리케이션의 유형, 시스템의 분산, 사용할 아키텍처 스타일, 아키텍처의 문서화 및 평가 방법** 등이 있다.
* **아키텍처 패턴**은 아키텍처 재사용을 위한 것으로 자주 사용되는 제네릭 아키텍처를 말한다.
* 아키텍처 패턴의 예로서 **MVC, 계층 형(layered), 저장소(repository), 클라이언트-서버 , 파이프필터 아키텍처 패턴 등이** 있다.
* 제네릭 어플리케이션 아키텍처를 통해 어플리케이션의 동작 이해, 어플리케이션의 비교, 설계의 평가, 대규모 컴포넌트 재사용 검토 등에 도움을 받을 수 있다.

7강 소프트웨어 테스팅

Q1인스펙션과 테스트 작업의 차이점을 설명하시오.

정답 및 해설

정답 및 해설

인스펙션은 정적으로, 테스트 작업은 동적으로 이루어진다는 것이 가장 큰 차이점이다.   
즉, 테스트는 반드시 실행할 수 있는 대상을 필요로 한다. 반면에, 인스펙션은 어떤 산출물도 검토의 대상이 될 수 있다. 인스펙션이 오류 검출에 효과적이기는 하나 프로그램의 비기능적인 특성들을 파악하려면 테스트를 통해서만   
파악할 수 있다.

Q2동치 분할 방법과 경계 값 분석 방법을 설명하시오.

정답 및 해설

정답 및 해설

동치 분할은 블랙박스 테스트 방법의 하나로 가능한 입력 값의 범위를 동치 클래스(분할)로 나누고 분할의 대표 값을 테스트데이터로 선택하여 테스트하는 방법이다. 경계 값 분석은 동치 분할 방법의 변형으로 경계 값 주변에서   
오류 발생 확률이 높다는 점을 가정하여 경계 값을 가지고 테스트하는 방법이다.

Q3테스트 주도 개발은 회귀 테스트나 리팩토링과 어떤 관계에 있지 설명하시오.

양식의 맨 아래

테스트 주도 개발을 통해 회귀 테스트 자동화를 위한 테스트 집합을 형성할 수 있으며, 이와 같은 회귀 테스트 집합은   
코드 변경이 기존 코드를 깨뜨리지 않는다는 것을 확인하는 수단이 되므로, 기능의 변화 없이 코드의 구조만을 개선하는 리팩토링 활동의 기반이 된다.

[인쇄하기](http://ngradlms.knou.ac.kr/lmsdata/contents/20161108413/week07/sum01.html)

* 좁은 의미의 테스트는 **오류 발견의 목적**으로 수행되며, 결함이 없다는 것을 증명할 수는 없다.
* 개발 테스트는 소프트웨어 개발 팀의 몫이지만, 릴리스 테스트를 위해서는 별도의 팀이 존재해야 한다.
* 개발 테스트란 **결함을 발견하기 위한 테스트 작업**으로 개별 메서드와 클래스를 테스트하는 **단위테스트**, 클래스 집합을 테스트하는 **컴포넌트 테스트**, 시스템의 일부 또는 전체를 테스트하는 **시스템 테스트**로 구분된다.
* 소프트웨어를 테스트할 때는, 경험과 가이드라인에 근거해 테스트의 수준을 정하고 결함을 발견하기에 적합한 테스트케이스를 선정해야 한다.
* **테스트 주도 개발**은 코드를 작성하기 전에 코드가 통과해야 하는 테스트를 작성하는 방식을 말한다.
* **인수 테스트**는 사용자 환경에 설치되어 사용을 위한 수용 여부를 고객이 결정하기 위한 테스트 작업이다.