

[소프트웨어 개발>통합 구현>모듈 구현]

[기출 예상 문제]

1. 다음 중 단위 모듈이 아닌 것은?

- ① 서비스 컴포넌트 ② 트랜잭션 컴포넌트
- ③ 화면 모듈 ④ 아키텍처 컴포넌트

[기출 예상 문제]

2. 모듈에 대한 설명으로 가장 거리가 먼 것은?

- ① 단위 모듈을 먼저 구현하고 공통 모듈 구현시 단위 모듈을 재사용 한다.
- ② 모듈 구현 시 응집도는 높이고, 결합도는 낮춘다.
- ③ 항상 예외처리 로직을 고려하여 모듈을 구현한다.
- ④ 공통 모듈은 Script로 구현하는데, 날짜 계산, 주소 정보 컴포넌트 등이 그 사례이다.

[기출 예상 문제]

3. 다음은 무엇에 대한 설명 인가?

- 작성자가 아닌 사람도 읽기 쉽고 고치기 쉽게 만든 코드이다.
- 단순하고 직접적이며 잘 쓴 문장처럼 읽힌다.
- 가독성이 뛰어나고 코드 간 의존성을 최대한 줄였다.
- 코드의 의도와 목적이 명확하다.

- ① Best Code ② Clean Code
- ③ Non-fixed Code ④ Refactoring

[기출 예상 문제]

4. 아래와 관련된 HTML 5 요소로 가장 적절한 것은?

h1 기사퍼스트
h2 기사퍼스트
h3 기사퍼스트
h4 기사퍼스트
h5 기사퍼스트
h6 기사퍼스트

- ① <header> ② <hgroup>
- ③ <article> ④ <section>

[기출 예상 문제]

5. HTML 5 문서의 구조가 아닌 것은?

- ① <footer> ② <beside>
- ③ <nav> ④ <section>

[소프트웨어 개발>통합 구현>모듈 구현]

[기출 예상 문제]

6. 다음은 무엇에 대한 설명인가?

- 하나의 웹사이트에서 PC, 스마트폰, 태블릿 PC 등 접속하는 디스플레이 종류에 따라 화면의 크기가 자동으로 변하도록 만든 웹페이지 접근 기법이다.
- 다양한 해상도의 크기 및 비율을 자유롭게 지원한다.
- 리액트(React), 뷰(Vue), 앵글러(Angular) 같은 웹 컴포넌트를 많이 활용하여 구현한다.

- ① 웹 사이트 표준 ② 반응형 웹
- ③ HTML 5 ④ 웹 3.0

[기출 예상 문제]

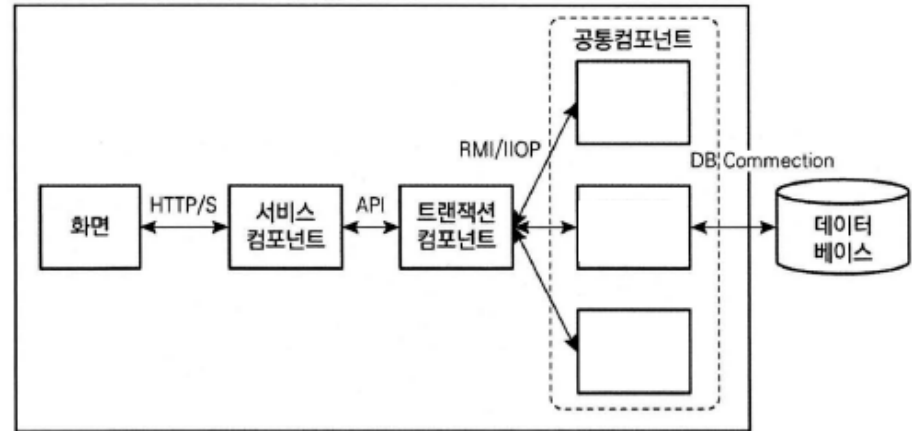
7. 다음은 무엇에 대한 설명인가?

웹 표준 기관인 월드와이드웹 컨소시엄(W3C)이 만들고 있는 차세대 웹 언어 규격이다. 문서 작성 중심으로 구성된 기존 표준에 그림, 동영상, 음악 등을 실행하는 기능까지 포함시켰다. 이것을 이용해 웹사이트를 만들면 국내 전자상거래에서 많이 쓰이는 액티브X, 동영상이나 음악재생에 필요한 어도비 플래시와 같은 플러그인 기반의 각종 프로그램을 별도로 설치할 필요가 없어진다. 이것은 모바일 환경에서 아이폰이나 안드로이드 등의 운영체제를 가리지 않고 모두 호환된다.

- ① HTML 5 ② HTTP
- ③ PHP 8.0 ④ XML

[기출 예상 문제]

8. 다음 그림의 공통 컴포넌트에 들어갈 컴포넌트로 가장 거리가 먼 것은?



- ① 인터페이스 컴포넌트
- ② 암호화 컴포넌트
- ③ 테스트 컴포넌트
- ④ DB접근 컴포넌트

[기출 예상 문제]

9. 클린코드에 대한 설명으로 가장 거리가 먼 것은?

- ① 가독성이 뛰어나다.
- ② 개체가 여러가지 작업을 수행할 수 있다.
- ③ 코드가 타인에 의해 변경이 용이하다.
- ④ 코드에 중복이 없거나 적다.

[소프트웨어 개발>통합 구현>모듈 구현]

[기출 예상 문제]

10. 검사(Test)에 대한 설명으로 가장 적절한 것은?

- ① 단위 검사는 모듈에 대한 검사로 블랙 박스 테스트 기법을 사용한다.
- ② 하향식 통합 검사는 하위 모듈을 클러스터(Cluster)로 결합하여 클러스터(Cluster)를 검사한다.
- ③ 검증 검사는 요구사항을 충족하는지를 검사하며 종류에는 알파 검사와 베타 검사가 있다.
- ④ 시스템 검사는 모듈들을 결합하여 검사한다.

[기출 예상 문제]

11. 화이트 박스 테스트에 대한 설명으로 가장 거리가 먼 것은?

- ① 구조 테스트로, 모듈 안의 작동을 자세히 관찰하는 테스트이다.
- ② 프로그램 원시 코드의 논리적인 구조를 커버하도록 테스트케이스를 설계하는 프로그램 테스트 방법이다.
- ③ 화이트 박스 테스트 종류에는 동치분할검사, 경계값 분석, 원인-효과 그래프 검사 등이 있다.
- ④ 프로그램의 제어 구조에 따라 선택, 반복 등의 부분들을 수행함으로써 논리적 경로를 제어한다.

[기출 예상 문제]

12. 블랙 박스 테스트와 화이트 박스 테스트로 검사 기법을 분류 할 때 테스트 기법이 다른 하나는?

- ① 기초 경로 검사 ② 조건 검사
- ③ 데이터 흐름 검사 ④ 오류예측검사

[이전 기출 문제]

13. 소프트웨어 테스트에서 화이트박스 기법의 설명에 해당하는 것은?

- ① 프로그램을 눈으로 보면서 확인하는 것
- ② 프로그램의 구조에 의거하여 테스트하는 것
- ③ 프로그램의 외부사양에 대하여 테스트하는 것
- ④ 프로그램의 외관상 관계를 파악하는 것

[이전 기출 문제]

14. 다음 설명에 해당하는 소프트웨어 테스트 기법은?

- 소프트웨어 인터페이스에서 실시되는 검사로 설계된 모든 기능들이 정상적으로 수행되는지 확인한다.
- 소프트웨어의 기능이 의도대로 작동하고 있는지, 입력은 적절하게 받아들였는지, 출력은 정확하게 생성되는지를 보여주는 데 사용된다.
- Equivalence Partitioning Testing, Boundary Value Analysis 등이 이 기법에 해당한다.

- ① 화이트 박스 테스트 ② 블랙 박스 테스트
- ③ 레드 박스 테스트 ④ 블루 박스 테스트

[소프트웨어 개발>통합 구현>모듈 구현]

[이전 기출 문제]

15. 다음 C 프로그램에 대하여, 조건/결정 커버리지 (condition/decision coverage)를 100% 만족하기 위한 테스트 데이터 집합으로 가장 적절한 것은? (단 short-circuit evaluation은 수행하지 않음)

```
void foo(int x, int y) {  
    if(x<50 || y>=30)  
        printf("%dWn", x+y);  
    else  
        printf("%dWn", x-y);  
}
```

- ① {(x = 0, y = 10) , (x = 5, y = 20)}
- ② {(x = 0, y = 40) , (x = 5, Y = 60)}
- ③ {(x = 60, y = 10) , (x = 5, y = 60)}
- ④ {(x = 60, y = 40) , (x = 5, y = 20)}

[기출 예상 문제]

16. 단위테스트 자동화 도구가 아닌 것은?

- ① JUnit
- ② CppUnit
- ③ unittest
- ④ iOS

[이전 기출 문제]

17. 다음의 소스 코드에 대해 테스트를 수행하려 할 때, 문장 커버리지(statement coverage) 기준으로 가장 높은 커버리지를 달성할 수 있는 테스트 케이스로 가장 적절한 것은?(단, short-circuit evaluation은 수행하지 않음)

줄번호	소스코드
1	if(x>0 && y<5)
2	x=y+5;
3	else
4	x=y-5;
5	if(x<=0)
6	z=-1*x+10;
7	else
8	z=x*x;

- ① T = {(x, y) | (2, -3), (4, 6)}
- ② T = {(x, y) | (-2, 3), (1, -4)}
- ③ T = {(x, y) | (2, -3), (4, -6)}
- ④ T = {(x, y) | (-2, 3), (-1, 4)}

[소프트웨어 개발>통합 구현>모듈 구현]

[이전 기출 문제]

18. 아래의 소스 코드에 대해 테스트를 수행하기 위하여 다음과 같이 테스트 데이터를 준비하였다. 조건 커버리지(condition coverage), 결정 커버리지(decision coverage), 문장 커버리지(statement coverage) 기준으로 평가할 때, 다음 중 가장 적절한 것은? (단, short-circuit evaluation은 수행하지 않음)

소스코드	테스트 데이터
<pre>if(x>=-2 && y <4) x=y*5; else x=y-5;</pre>	<p>T={(-3,-2), (0,6)}</p> <p>단, 테스트 데이터는 (x,y) 값에 해당함</p>

- ① 조건 커버리지: 50%, 결정 커버리지: 50%, 문장 커버리지: 50%
- ② 조건 커버리지: 50%, 결정 커버리지: 100%, 문장 커버리지: 66%
- ③ 조건 커버리지: 100%, 결정 커버리지: 50%, 문장 커버리지: 66%
- ④ 조건 커버리지: 100%, 결정 커버리지: 100%, 문장 커버리지: 100%

[소프트웨어 개발>통합 구현>통합 구현 관리]

[기출 예상 문제]

1. 다음은 무엇에 대한 설명인가?

코딩, 디버그, 컴파일 배포 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어이다. 원래 소프트웨어 개발에서는 컴파일러, 텍스트 편집기, 디버거 등을 따로 사용했지만, 이러한 프로그램들을 하나로 묶어 대화형 인터페이스를 제공한 것이다.

- ① Integrated Development Environment
- ② Software Requirement Tool
- ③ Computer Aided Software Engineering
- ④ Graphical User Interface

[기출 예상 문제]

2. IDE 도구 기능으로 가장 거리가 먼 것은?

- ① C++, Java 등의 언어를 이용하여 애플리케이션을 개발할 수 있는 환경을 제공해 준다.
- ② 외부 형상, 배포 관리 기능과 연계되어 소스 코드의 형상관리 및 자동 배포를 가능하게 한다.
- ③ 소스 코드 코딩이 완료되면 컴파일하여 문법 검사를 해 준다.
- ④ 소프트웨어 요구사항을 추적하고 관리하므로 프로젝트 관리가 수월하다.

[기출 예상 문제]

3. IDE 도구에 대한 설명으로 가장 거리가 먼 것은?

- ① 이클립스(Eclipse)는 크로스 플랫폼으로 이클립스 재단에서 개발하였다.
- ② C++빌더의 언어는 C, C++을 사용한다.
- ③ 라자루스(Lazarus)는 마이크로소프트에서 개발한 IDE 도구로 상용 라이선스를 가지고 있다.
- ④ 비주얼 스튜디오(Visual Studio)의 플랫폼은 Win32/64이고, 운영체제는 윈도우 시리즈이다.

[기출 예상 문제]

4. 다음 중 IDE 도구가 아닌 것은?

- ① Git ② Eclipse
③ J Builder ④ Lazarus

[기출 예상 문제]

5. 소프트웨어 개발 프로젝트에 사용되는 협업 도구 중 유형이 다른 하나는?

- ① 트렐로 ② 슬라이드
③ 지라(JIRA) ④ 레드마인

[소프트웨어 개발>통합 구현>통합 구현 관리]

[기출 예상 문제]

6. 협업도구의 기능이 아닌 것은?

- ① 코딩, 디버그, 컴파일러, 배포 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공한다.
- ② 개발자간 작업 공유, 오류 현상 공유, 의견 공유 등 다양한 의사소통을 제공한다.
- ③ 소스 코드 구현 도중 발생하는 오류 및 이슈에 대한 공유 혹은 프로젝트 일정에 대한 공유가 가능하다.
- ④ 구현 아이디어의 공유 혹은 새로운 아이디어를 개발자들간에 공유함으로써 집단 지성을 활용한다.

[기출 예상 문제]

7. 통합 구현에 협업 도구가 필요한 이유로 가장 적절한 것은?

- ① 소스 코드 코딩이 완료 되면 이를 컴파일하여 오류 발생 시 추적하여 수정할 수 있는 디버깅 기능이 있다.
- ② 이해관계자들 간의 실시간 커뮤니케이션과 개발자 간에 프로젝트 공유가 되어야 한다.
- ③ 프로그램의 재사용성을 높이기 위해서 협업 도구를 활용할 수 있다.
- ④ 프로젝트 도중 요구사항 변화에 대해 민감한 문제점을 해결 가능 하다.

[기출 예상 문제]

8. 팀원들과 중요한 아이디어를 공유할 수도 있고, 업무와 관련 있는 기사를 스크랩해서 공유할 수 있는 협업 도구로 가장 적절한 것은?

- ① 태스크 월드 ② 구글 캘린더
- ③ 에버노트 ④ 깃허브

[기출 예상 문제]

9. 형상관리의 항목이 아닌 것은?

- ① 형상 식별 ② 형상 기록
- ③ 형상 감사 ④ 형상 추적

[기출 예상 문제]

10. 개발자가 소스를 형상관리 도구 저장소에 업로드 후 최종적으로 업데이트가 되었을 때 형상관리 서버에 반영하도록 하는 기능은?

- ① Commit ② Checkout
- ③ Diff ④ Submit

[소프트웨어 개발>통합 구현>통합 구현 관리]

[기출 예상 문제]

11. 형상관리 도구 CVS의 특징으로 가장 거리가 먼 것은?

- ① 파일 단위로 변경 사항을 관리한다.
- ② 중앙 집중형식으로 서버 저장소를 두고 클라이언트가 접속해서 버전 관리를 실행한다.
- ③ 텍스트 기반 문서와 바이너리 문서를 지원한다.
- ④ Diff 명령어를 통해서 두 개의 파일 내용을 비교 가능하다.

[기출 예상 문제]

12. 다음은 무엇에 대한 설명인가?

- 분산형 방식의 형상관리 도구
- 각 로컬 PC에 스스로 완전한 저장소가 구성
- 필요에 따라 중앙 집중형 방식으로도 운영 가능
- 원격 저장소에 장애가 나더라도 쉽게 복구 가능

- ① Git
- ② Perforce
- ③ CVS
- ④ Subversion

[기출 예상 문제]

13. 형상관리 도구 중 CVS의 단점을 보완하여 Subversion이 가장 널리 사용되고 있다. CVS와 비교하여 Subversion의 장점으로 가장 거리가 먼 것은?

- ① 작업 단위의 변경 사항 관리
- ② 디렉토리나 파일을 자유롭게 이동해도 이력 유지
- ③ 커밋 실패 시 롤백 지원
- ④ 2개의 파일 내용을 비교하기 위한 명령어 존재

[기출 예상 문제]

14. Git의 특징으로 가장 적절한 것은?

- ① .gitbranch 를 통한 형상관리 배제 지원 지정 기능을 가지므로 형상관리가 편리하다.
- ② SVN으로 관리되던 저장소를 Git으로 전환해 주는 마이그레이션 기능이 있다.
- ③ 디렉토리가 저장되어 간편하다.
- ④ 여러 개의 commit을 원격 저장소에 한 번에 반영할 수 없으므로, 1개의 commit을 원격 저장소에 한 번씩 반영해야 하는 불편함이 있다.

[소프트웨어 개발>통합 구현>통합 구현 관리]

[이전 기출 문제]

15. 형상관리에 대한 설명 중 옳은 것은?

- ① 버전관리와는 전혀 다르다.
- ② 변경 요청 시 즉시 변경하여 융통성을 높인다.
- ③ 소프트웨어의 형상은 프로그램뿐만 아니라 데이터도 포함된다.
- ④ 소프트웨어 공학 과정의 분석 단계에만 적용된다.

[이전 기출 문제]

16. 소프트웨어 형상관리(software configuration management)에 대한 설명 중 옳지 않은 것은?

- ① 소프트웨어 형상 항목에는 시스템 명세서, 소프트웨어 프로젝트 계획서, 소프트웨어 요구사항 명세서 등이 포함된다.
- ② 소프트웨어 개발 과정에서 소프트웨어에 대한 변경 사항을 관리하기 위해 수행되는 일련의 활동들을 의미하며, 성공적인 형상관리를 위해서는 유지보수 단계에서 계획하여야 한다.
- ③ 소프트웨어 형상관리 활동에 관련된 사람들을 형상 통제 위원회라고 부르며, 구성원에는 프로젝트 관리자, 품질담당자, 기술담당자와 고객 측 담당자 등이 포함된다.
- ④ 형상관리 도구로는 Clearcase, CVS 등이 있다.

[이전 기출 문제]

17. 형상관리 기능과 관계가 없는 것은?

- ① 변경 관리 ② 형상 감사
- ③ 프로그램 실행 ④ 버전 관리

[이전 기출 문제]

18. 소프트웨어 형상 관리(configuration management)에 대한 설명으로 옳지 않은 것은?

- ① 소프트웨어 구성 항목을 관리
- ② 시스템에 행해진 변경 제어
- ③ 테스트 과정에서 소프트웨어를 통합하는 것
- ④ 진화중인 소프트웨어 제품의 여러 버전들을 관리

[정답] [소프트웨어 개발>통합 구현]

1. 모듈 구현

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		
4	1	2	2	2	2	1	3	2	3	3	4	2	2	3	4	2	3		

2. 통합 구현 관리

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		
1	4	3	1	2	1	2	3	4	1	3	1	4	2	3	2	3	3		

[정답 및 해설] [소프트웨어 개발>통합 구현>모듈 구현]

모듈 구현 1. ④

단위 모듈은 화면 모듈, 화면에서 입력받은 데이터 처리를 위한 서비스 컴포넌트, 비즈니스 트랜잭션 컴포넌트가 있다.

모듈 구현 2. ①

① 공통모듈을 먼저 구현하고, 개별 단위 모듈 구현시 이를 재사용한다.

모듈 구현 3. ②

클린 코드(Clean Code)에 대한 설명이다.

모듈 구현 4. ②

② <hgroup>은 헤딩을 다양화하기 위해 그룹으로 묶어서 <h1>~<h6>요소를 이용한다.

모듈 구현 5. ②

HTML 5 문서 구조: <header>, <hgroup>, <nav>, <section>, <article>, <aside>, <footer>

모듈 구현 6. ②

반응형 웹(Responsive Web)에 대한 설명이다.

모듈 구현 7. ①

HTML 5 에 대한 설명이다.

모듈 구현 8. ③

공통 컴포넌트에는 데이터베이스 접근 컴포넌트, 내외부 인터페이스 컴포넌트, 데이터 암호화 컴포넌트가 있다.

모듈 구현 9. ②

② 개체(Class, Method)가 한가지 작업만 수행한다.

모듈 구현 10. ③

- ① 단위 검사는 모듈에 대한 검사로 화이트 박스 테스트 기법을 사용한다.
- ② 상향식 통합 검사에 대한 설명이다.
- ④ 모듈 검사에 대한 설명이다. 시스템 검사는 해당 컴퓨터 시스템에서 수행 되는지를 검사한다.

모듈 구현 11. ③

③ 블랙 박스 테스트의 종류이다. 화이트 박스 테스트의 종류에는 기초 경로 검사, 조건 검사, 루프 검사, 데이터 흐름 검사가 있다.

[정답 및 해설] [소프트웨어 개발>통합 구현>모듈 구현]

모듈 구현 12. ④

- ①, ②, ③ 화이트 박스 테스트
- ④ 블랙 박스 테스트

모듈 구현 13. ②

화이트 박스 테스트는 구조 테스트로 프로그램 원시 코드의 논리적인 구조를 커버하도록 테스트케이스를 설계하는 프로그램 테스트 방법이다.

모듈 구현 14. ②

블랙 박스 테스트는 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하기 위한 검사이다.

모듈 구현 15. ③

조건/결정 커버리지는 전체 조건식이 참/거짓 한번씩 가지면서, 개별 조건식이 참/거짓 모두 한번씩 갖도록 조합한다. 즉, $(x < 50 \parallel y \geq 30)$: 참, 거짓, $x < 50$: 참, 거짓, $y \geq 30$: 참, 거짓 을 갖도록 테스트케이스를 설계한다.

③ $(x=60, y=10)$ 전체 조건식: 거짓, $x < 50$: 거짓, $y \geq 30$: 거짓 / $(x=5, y=60)$ 전체 조건식: 참, $x < 50$: 참, $y \geq 30$: 참

모듈 구현 16. ④

- ① JUnit: Java기반의 단위 모듈 테스트 자동화 도구
- ② CppUnit: C++언어 기반의 단위 테스트 자동화 도구
- ③ unittest: Python에서 단위테스트를 수행하기 위한 자동화 도구
- ④ iOS: 애플의 모바일 운영 체제이다.

모듈 구현 17. ②

- 각각의 x와 y값에 따라 실행문들(줄 번호:2, 4, 6, 8)이 모두 실행되어야 가장 높은 커버리지를 달성 할 수 있다.
- ② $(-2, 3)$ 의 경우 줄 번호 4와 6이 수행되고, $(1, -4)$ 의 경우 줄 번호 2와 8이 수행된다.

모듈 구현 18. ③

- 조건 커버리지: 각 개별 조건식이 참/거짓 모두 한 번씩 갖도록 개별 조건식을 조합
(-3, -2) $x \geq -2$: 거짓, $y < 4$: 참
(0, 6) $x \geq -2$: 참, $y < 4$: 거짓 = 100%
- 결정 커버리지: 전체 조건식이 최소한 참/거짓 한번의 값을 가지도록 조합
(-3, -2) $(x \geq -2 \ \&\& \ y < 4)$: 거짓
(0, 6) $(x \geq -2 \ \&\& \ y < 4)$: 거짓 = 50%
- 문장 커버리지: 조건문의 결과와 관계없이 실행된 개수로서 계산
if($x \geq -2 \ \&\& \ y < 4$): 실행
x=y*5: 실행 안함
x=y-5: 실행 = 66%

[정답 및 해설] [소프트웨어 개발>통합 구현>통합 구현 관리]

통합 구현 관리 1. ①

통합 개발 환경(IDE, Integrated Development Environment)에 대한 설명이다.

통합 구현 관리 2. ④

④ 요구사항 관리 도구에 대한 설명이다.

통합 구현 관리 3. ③

③ 라자루스(Lazarus)는 Lazarus team에서 개발한 IDE 도구로 GPL, GNU 등 라이선스를 가지고 있다. 마이크로소프트에서 개발한 IDE 도구에는 Visual Studio가 있다.

통합 구현 관리 4. ①

① Git은 형상관리 도구이다.

통합 구현 관리 5. ②

- ①, ③, ④ 프로젝트 관리 도구
- ② 문서 공유 도구

통합 구현 관리 6. ①

① 통합 개발 환경(IDE, Integrated Development Environment)에 대한 설명이다.

통합 구현 관리 7. ②

통합 구현에 협업도구 필요성

- 구현 기간 동안 다양한 개발자들과 실시간 커뮤니케이션 필요하다.
- 개발 및 작업 일정, 소스, 아키텍처가 수시로 변경되기 때문에 이에 대해서 개발자 간에 공유가 되어야 한다.
- 여러 개발자들의 아이디어를 공유해야 품질이 높아진다.

통합 구현 관리 8. ③

- ① 태스크 월드: 사용자가 자유롭게 프로젝트 일정, 팀원 설정 가능한 프로젝트 관리 도구
- ② 구글 캘린더: 구글의 일정 관리 서비스, 모바일과 PC연동이 가능한 일정 관리 도구
- ④ 깃허브: 오픈 소스 및 프로그래머들의 공동 작업이 가능한 소스 공유 도구

통합 구현 관리 9. ④

형상관리는 소프트웨어 변경 사항을 관리하기 위해서 형상 식별, 형상 통제, 형상 감사, 형상 기록을 수행한다.

[정답 및 해설] [소프트웨어 개발>통합 구현>통합 구현 관리]

통합 구현 관리 10. ①

형상관리 도구에서 Commit 명령어에 대한 설명이다.

통합 구현 관리 11. ③

③ CVS는 텍스트 기반 소스 코드만 지원한다는 단점이 있다.

통합 구현 관리 12. ①

②, ③, ④ 중앙 집중형 방식의 형상관리 도구이다.

통합 구현 관리 13. ④

④ CVS에서도 Diff 명령어를 통한 2개의 파일 내용 비교가 가능하다. 그리고 Subversion에서는 2개 이상의 파일 내용 비교가 가능하다.

통합 구현 관리 14. ②

① .gitignore를 통한 형상 관리 배제 지원 지정 기능을 가진다.

③ 디렉토리가 저장되지 않는 단점이 있다.

④ 여러 번의 commit을 실행하고, 모아진 commit을 원격 저장소에 한 번에 반영할 수 있으므로 네트워크킹 빈도는 줄고 속도는 향상되는 장점을 가진다.

통합 구현 관리 15. ③

형상관리는 소프트웨어 라이프 사이클 동안 발생하는 변경 관리 활동으로, 소프트웨어 공학 과정의 모든 단계에 적용된다. 따라서 형상관리 항목(SCI, Software Configuration Item)에는 프로젝트 계획서, 분석서, 설계서, 프로그램, 테스트 케이스, 데이터, 코드 등이 될 수 있다.

통합 구현 관리 16. ②

② 성공적인 형상관리는 유지보수 단계가 아니라 소프트웨어를 개발 전, 또는 개발 중에 계획하여야 한다.

통합 구현 관리 17. ③

형상관리 활동에는 일반적으로 형상 식별, 변경 관리 및 통제, 버전 및 배포관리, 형상 감사 및 보고 활동으로 이루어져 있다.

통합 구현 관리 18. ③

③ 통합테스트(Integration Testing)는 소프트웨어 검사(Test) 전략에 해당한다.