

트랜잭션의 특성 4가지	일관성, 지속성, 원자성, 고립성
트랜잭션 실행 중 생성하는 연산의 중간 결과를 다른 트랜잭션이 접근하지 못하도록 보장하는 특성	고립성(격리성)
트랜잭션이 완료되면 시스템의 (1)이 유지된다는 특성 트랜잭션이 실행되고 나서도 데이터베이스의 상태는 무결성이 유지되고 모순되지 말아야 된다는 의미	일관성
트랜잭션의 결과는 오로지 성공 또는 철회뿐이라는 특성	원자성
완료된 트랜잭션의 결과는 영구적으로 유지된다는 특성	지속성(영구성)
데이터베이스의 회복기법 중 Rollback 시 Redo, Undo가 모두 실행되는 트랜잭션 처리법으로 트랜잭션 수행 중 갱신 결과를 바로 DB에 반영하는 기법	즉각 갱신 회복 기법
데이터베이스의 구조, 제약조건 등의 정보를 담고 있는 기본적인 구조	DB 스키마
데이터베이스 시스템에서 하나의 논리적 기능을 정상적으로 수행하기 위한 작업의 기본 단위	트랜잭션
데이터를 빠르게 찾을 수 있는 수단으로서 테이블에 대한 조회 속도를 높여주는 자료 구조. 검색 연산의 최적화를 위해 데이터베이스 내 열에 대한 정보를 구성한 데이터 구조	인덱스
일련의 쿼리들을 마치 하나의 함수처럼 실행하기 위한 쿼리의 집합	프로시저
일련의 SQL처리를 수행하고, 수행 결과를 단일 값으로 반환할 수 있는 절차형 SQL	사용자 정의 함수
데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다	트리거
SQL을 가장 빠르고 효율적으로 수행할 최적의 처리경로를 생성해주는 DBMS 내부의 핵심엔진이다.	옵티마이저
트랜잭션을 메모리에 영구적으로 저장하는 제어어	커밋(COMMIT)
오류가 발생했을 때 오류 이전의 특정 시점(SAVEPOINT, CHECHPOINT)상태로 되돌려주는 제어어	롤백(ROLLBACK)
RollBack을 위한 시점을 지정하는 제어어	체 크 포 인 트 (CHECKPOINT)
SQL성능 개선의 핵심부분으로 옵티마이저의 실행 계획을 원하는 대로 변경할 수 있게 한다	힌트(HINT)
사전 등록된 규칙에 따라 질의실행 계획을 선택하는 옵티마이저	규칙 기반 옵티마이저(RBO)
통계 정보로부터 모든 접근 경로를 고려한 질의실행 계획을 선택하는 옵티마이저	비용 기반 옵티마이저(CBO)
형상 항목의 버전 관리를 위해 변경 여부와 변경 활동을 통제하는 활동	형상통제
Linux 위에서 구동하며 휴대 전화를 비롯한 휴대용 장치를 위한 운영체제와 미들웨어, 사용자 인터페이스 그리고 표준 응용 프로그램을 포함하고 있는 자바 및 코틀린을 사용하여 개발하는 운영체제	안 드 로 이 드 (Android)
UI 화면 설계를 위해서 정책이나 프로세스 및 콘텐츠의 구성, 와이어프레임기능에 대한 정의, 데이터베이스의 연동 등 구축하는 서비스를 위한 대부분 정보가 수록된 문서	스토리보드
UI 설계 지침 중 사용자가 이해하기 쉽고 편하게 사용할 수 있는 환경	사용자중심

을 제공하며, 실사용자에 대한 이해가 바탕이 되어야함.	
UI 설계 지침 중 버튼이나 조작 방법을 사용자가 기억하기 쉽고 빠르게 습득할 수 있도록 설계해야함	일관성
UI 설계 지침 중 조작 방법은 가장 간단하게 작동되도록 하여 인지적 부담 최소화	단순성
UI설계 지침 중 작동시킬 기능만 보고도 결과 예측이 가능해야 함	결과 예측 가능
UI설계 지침 중 주요 기능을 메인 화면에 노출하여 쉬운 조작이 가능해야함	가시성
UI설계 지침 중 디자인을 표준화하여 기능구조의 선행학습 이후 쉽게 사용 가능해야 함	표준화
UI 설계 지침 중 사용자의 직무, 연령, 성별 등이 고려된 다양한 계층을 수용해야함	접근성
UI 설계 지침 중 사용자가 개념적으로 쉽게 인지해야함	명확성
UI 설계 지침 중 사용자가 오류에 대한 상황을 정확하게 인지할 수 있어야함	오류발생해결
LoC/프로그래머의 월간 생산성	Man Month
Man Month/프로젝트 인력	프로젝트기간
UI 유형 중 정적인 텍스트 기반 인터페이스로 명령어를 텍스트로 입력하여 조작하는 사용자 인터페이스	CLI(Command Line Interface)
UI 유형 중 그래픽 반응 기반 인터페이스로 그래픽 환경을 기반으로 한 마우스나 전자펜을 이용하는 인터페이스	GUI(Graphical User Interface)
UI 유형 중 직관적 사용자 반응기반 인터페이스로 키보드 마우스 없이 신체 부위를 이용하는 사용자 인터페이스. 터치, 음성 포함	NUI(Natural User Interface)
UI 유형 중 유기적 상호작용기반 인터페이스로 현실에 존재하는 모든 사물이 입출력자치로 변화할 수 있는 사용자 인터페이스	OUI(Oranic User Interface)
UML 구성요소	사물, 관계, 다이어그램
UML 다이어그램 종류 중 시스템 내 클래스의 정적 구조를 표현. 클래스는 객체들의 집합으로 속성(Attribute)과 동작(Behaior)으로 구성	클래스 다이어그램
UML 다이어그램 종류 중 클래스에 속한 사물들 즉 인스턴스를 특정 시점의 객체와 객체 사이의 관계로 표현한 다이어그램	객체 다이어그램
UML다이어그램 종류 중 시스템을 구성하는 물리적인 컴포넌트와 그들 사이의 의존 관계를 나타내는 다이어그램	컴포넌트 다이어그램
UML다이어그램 중 컴포넌트 사이의 종속성을 표현하고, 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현하는 다이어그램	배치 다이어그램(Deply ment)
UML 다이어그램 중 유스케이스나 클래스 등의 모델 요소들을	Package(UML2.0

그룹화한 패키지들의 관계를 표현한 다이어그램	에서 추가)
UML 다이어그램 중에서 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현하는 다이어그램	복합체 구조 다이어그램
UML 다이어그램 중에서 시스템이 제공하고 있는 기능 및 그와 관련된 외부 요소를 사용자의 관점에서 표현하는 다이어그램	유스케이스 다이어그램
UML 다이어그램 중에서 객체간 동적 상호작용을 시간적 개념을 중심으로 메시지 흐름으로 표현한 다이어그램	시퀀스 다이어그램
UML 다이어그램 중에서 커뮤니케이션 다이어그램은 동작에 참여하는 객체들이 주고 받는 메시지를 표현하고, 메시지뿐만 아니라 객체 간의 연관까지 표현하는 다이어그램	커뮤니케이션 다이어그램
UML 다이어그램 중에서 하나의 객체가 자신이 속한 클래스의 상태 변화 혹은 다른 객체와의 상호작용에 따라 상태가 어떻게 변화하는지 표현하는 다이어그램	상태 다이어그램
UML 다이어그램 중에서 시스템이 어떤 기능을 수행하는지를 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서대로 표현 하는 다이어그램	활동 다이어그램
활동의 시작을 의미. 하나의 다이어그램 안에는 하나의 시작점만 존재. 검은색 동그라미로 표현	시작점
실행의 흐름을 나타냄. 화살표로 표현	전이
어떠한 일들의 처리와 실행을 의미. 액션은 더 이상 분해할 수 없는 단일 작업이고, 액티비티는 몇 개의 액션으로 분리될 수 있는 작업. 모서리가 둥근 사각형으로 표현하고, 둥근 사각형 안에 액션이나 액티비티 명칭 기술	액션(Action)/액티비티(Activity)
처리의 종료를 의미. 하나의 다이어그램 안에는 여러 개의 종료 노드가 있을 수 있음. 검은색 동그라미를 포함한 원으로 표현	종료점
조건에 따른 제어 흐름의 분리를 표현 마름모로 표현하고 들어오는 제어 흐름은 한 개이고 나가는 제어 흐름은 여러 개로 표현	조건(판단)노드
여러 경로의 흐름이 하나로 합쳐진 것을 표현. 마름모로 표현하고 들어오는 제어 흐름은 여러 개이고, 나가는 제어 흐름은 한 개로 표현	병합노드
평행적으로 수행되는 흐름을 나누는 노드. 굵은 가로선으로 표현하고 들어오는 액티비티 흐름은 한 개이고, 나가는 액티비티 흐름은 여러개	포크 노드
포크 노드로 나뉜 흐름을 다시 하나로 합치는 노드. 굵은 가로선으로 표현하고 들어오는 액티비티 흐름은 여러 개이고, 나가는 액티비티 흐름은 한 개	조인 노드
액티비티 수행을 담당하는 주체를 구분하는 면. 가로 또는 세로 실선을 그어 구분	구획면
UML 다이어그램 중에서 객체 상태 변화와 시간 제약을 명시적	타이밍

으로 표현하는 다이어그램	다이어그램
작성한 코드의 빌드 및 배포를 수행하는 도구, 각각의 구성요소와 모듈에 대한 의존성 관리를 지원	빌드도구
코드의 작성과 디버깅, 수정 등과 같은 작업과 구현소프트웨어가 어떤 프로그래밍 언어로 개발되는지 선택, 프로그램을 개발할 때 가장 많이 사용되는 도구	구현 도구
코드의 기능 검증과 전체의 품질을 높이기 위해 사용하는 도구 코드의 테스트, 테스트에 대한 계획, 수행 및 분석 등의 작업가능	테스트 도구
개발자들이 작성한 코드와 리소스 등 산출물에 대한 버전 관리를 위한 도구, 프로젝트 진행 시 필수로 포함되는 도구	형상관리 도구
HTTP를 이용한 요청/응답을 처리 웹상의 정적 콘텐츠(CSS, Javascript, Image)를 처리. 주요 제품으로 Apache 웹 서버, IIS 웹 서버, Google Web Server, Nginx 등 존재	웹 서버
동적 콘텐츠(Servlet, JSP)를 처리하기 제공하기 위해 사용. 주요 제품으로 Tomcat, Weblogic, JBoss, Resin 등 존재	웹 애플리케이션 서버
데이터의 수집, 저장을 위한 용도로 사용. 연계되는 주요 DBMS로 MySQL, Oracle, Resin 등 존재	데이터 베이스 서버
물리 저장장치를 활용한 서버, 대용량 HDD, SSD 등의 장치가 존재	파일 서버
설치를 통해 사용자와 커뮤니케이션하는 프로그램	클라이언트 프로그램
웹 서비스의 형태로 서버에서 웹 애플리케이션을 응답 시 브라우저를 통해 사용자와 커뮤니케이션. 일반적인 형태의 웹 사이트가 해당	웹 브라우저
모바일 디바이스에 설치되어 활용되는 애플리케이션. App Store, Android Market 등을 통해 다운로드 가능	모바일 앱
웹 브라우저와 동일한 형태로 모바일상 웹 브라우저르 통해 서비스를 제공. 모바일에 최적화되어 제공되는 웹사이트가 해당	모바일 웹
서버의 하드웨어를 사용자 관점에서 편리하고 유용하게 사용하기 위한 소프트웨어. 프로젝트의 성격에 따른 사용	운영체제
컴퓨터와 컴퓨터 간의 연결을 쉽고 안전하게 할 수 있도록 해주고 이에 대한 관리를 도와주는 소프트웨어 웹 서비스는 WEB, WAS 서버를 설치하여 사용자 요구사항에 적합한 서비스 제공. 자바 기반 환경에서 JVM을 설치하여 컨테이너로의 기능을 주로 이용	미들웨어
데이터의 저장 및 활용을 위해 설치함 사용자의 요청에 따라 제품을 선택	DBMS
모듈 내부가 아닌 외부의 모듈과의 연관도 또는 모듈간의 상호의존성이다. 소프트웨어 구조에서 모듈 간의 관련성을 측정하는 척도이다.	결합도
모듈 내부의 각 구성요소 등이 연관이 없을 경우	우연적 응집도
유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들이 한 모듈에서 처리되는 경우	논리적 응집도
연관된 기능이라기보다는 특정 시간에 처리되어야 하는 활동들을 한 모듈에서 처리 할 경우	시간적 응집도
모듈이 다수의 관련 기능을 가질 때 모듈안의 구성요소들이 그 기능을 순차적으로 수행할 경우	절차적 응집도
동일한 입력과 출력을 사용하여 다른 기능을 수행하는 활동들이 모여	통신적 응집도

있을 경우	
모듈 내에서 한 활동으로부터 나온 출력값을 다른 활동이 사용할 경우	순차적 응집도
모듈 내부의 모든 기능이 단일한 목적을 위해 수행되는 경우	기능적 응집도
다른 모듈 내부에 있는 변수나 기능을 다른 모듈에서 사용하는 경우	내용 결합도
파라미터가 아닌 모듈 밖에 선언되어 있는 전역 변수를 참조하고 전역 변수를 갱신하는 식으로 상호작용하는 경우	공통 결합도
두 개의 모듈이 외부에서 도입된 데이터 포맷, 통신 프로토콜, 또는 디바이스 인터페이스를 공유할 경우	외부 결합도
단순 처리할 대상인 값만 전달되는 게 아니라 어떻게 처리를 해야 한 다는 제어 요소가 전달되는 경우	제어 결합도
모듈 간의 인터페이스로 배열이나 객체, 구조 등이 전달되는 경우	스탬프 결합도
모듈 간의 인터페이스로 전달되는 파라미터를 통해서만 모듈 간의 상호작용이 일어나는 경우	자료 결합도
애플리케이션이 무엇을 할 것인지를 정의. 내부 비즈니스 로직을 처리 하기 위한 역할	Model
Model이 어떻게 처리할지를 알려주는 역할. View에 명령을 보내어 화면 요청 결과를 전달	Controller
화면에 무엇인가를 보여주기 위한 역할 model, controller가 화면에 보여주려고 하는 것들을 화면에 처리	View
사용자 인터페이스로부터 비즈니스 로직을 분리하여 애플리케이션의 시각적 요소나 그 이면에서 실행되는 비즈니스 로직을 서로 영향 없이 쉽게 고칠 수 있는 패턴	MVC패턴
응용 프로그램의 내부 구조와 동작을 검사하는 소프트웨어 테스트 방식이다.	화이트박스 테스트
어떤 모듈을 제어(호출)하는 모듈의 수, 이것이 높으면 재사용 측면에서 설계가 잘 되었지만, 단일 장애점 발생가능. 이것이 높으면 관리 비용 및 테스트 비용 증가	팬인(Fan-In)
어떤 모듈에 의해 제어(호출)되는 모듈의 수. 이것이 높을 경우 불필요한 모듈 호출 여부 검토 필요. 이것이 높을 경우 단순화 여부 검토 필요	팬아웃(Fan-Out)
사용자와 만나지 않고 프론트엔드와 연동하여 핵심 로직을 처리하는 영역. DB나 인터페이스 등을 통해 시스템 구성 실체에 접근	백 엔 드 (Back-end)
사용자(user)의 화면에 나타나는 웹 화면 영역. 웹 페이지를 그리는 기술(JSP, 자바스크립트, CSS, HTML, Node.js, React.js, Angular.js 등 활용)	프론트엔드 (Front-end)
객체지향 언어인 자바와 관계형 DB프로그래밍을 좀 더 쉽게 할 수 있게 도와주는 개발 프레임워크. 개발자가 작성한 SQL 명령어와 자바 객체를 매핑해주는 기능을 제공, 기존에 사용하던 SQL명령어 재사용	Mybatis
특정 타입의 데이터베이스에 추상 인터페이스를 제공하는 객체로 세부 내용 노출 없이 데이터 조작	D A O (D a t a Access Object)
프로세스 사이에서 데이터를 전송하는 객체로 데이터 저장. 회수 외에 다른 기능 없음	D T O (D a t a Transfer Object)
간단한 엔티티를 의미하는 작은 객체 가변 클래스인 DTO와 달리 고정 클래스를 가짐	V O (V a l u e Object)

사용자와의 상호작용 없이 일련의 작업들을 작업 단위로 묶어 정기적으로 반복 수행하거나 정해진 규칙에 따라 일괄 처리하는 방법	배치(Batch)프로그램
스프링 프레임워크의 DI, AOP, 서비스 추상화 등 스프링 프레임워크의 3대 요소를 모두 사용할 수 있는 대용량 처리를 제공하는 스케줄러	스프링 배치 (Batch)
스프링 프레임워크에 플러그인되어 수행하는 작업과 실행 스케줄을 정의하는 트리거를 분리하여 유연성을 제공하는 오픈 소스 기반 스케줄러	쿼츠(Quartz) 스케줄러
스케줄러를 실행시키기 위해 작업이 실행되는 시간 및 주기 등을 설정하게 되는데 이때 이것을 통해 배치 수행시간을 설정한다.	쿼츠 크론 표현식
그 자체로 하나의 완전한 기능을 수행할 수 있는 독립된 실체	모듈
소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리를 용이하게 하여 프로그램을 효율적으로 관리할 수 있도록 시스템을 분해하고 추상화하는 기법	모듈화
다른 프로그래밍 기능을 대리하는 코드. 기존 코드를 흉내 내거나, 아직 개발되지 않은 코드를 임시로 대체하는 역할을 수행. 모듈 및 모든 하위 컴포넌트를 대신하는 더미. 모듈 제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행하는 도구로 더미 객체에의 단순 기능에 특정 상태를 가정해서 특정한 값을 리턴하거나 특정 메시지를 출력한다.	스텝
상위의 모듈에서 데이터의 입력과 출력을 확인하기 위한 더미 모듈로 상향식 통합 테스트 수행 시 사용된다.	드라이버
자바 바이트 코드를 실행할 수 있는 주체로 CPU나 운영체제 종류와 무관하게 동작하는 것을 보장하는 가상머신	JVM(Java Virtual Machine)
자바 코드를 실행하기 위한 소프트웨어	JRE(Java Runtime Environment)
자바 응용 소프트웨어 개발 키트	JDK(Java Development Kit)
조직이나 기업의 자산에 악영향을 끼칠 수 있는 사건이나 행위	위협(Threat)
위협이 발생하기 위한 사전 조건으로 시스템의 정보 보증을 낮추는 데 사용되는 약점	취약점 (Vulnerability)
위협이 취약점을 이용하여 조직의 자산 손실 피해를 가져올 가능성	위험(Risk)
보안 취약점의 근본원인이 되는 소프트웨어의 결함이나 버그를 의미한다. 이것이 존재하지 않도록 개발하는 방법을 시큐어 코딩이라고 한다 (CF) 소프트웨어 보안약점을 사전식으로 분류한 것이 CWE이고 시간에 따라 확인된 보안취약점을 정리한 목록으로 인덱스 형식에 확인 연도가 포함되는 것이 특징인 것은 CVE이다.	소프트웨어 보안 약점
특정 서버에게 수 많은 접속 시도를 만들어 다른 이용자가 정상적으로 서비스 이용을 하지 못하게 하거나, 서버의 자원을 소진시켜 원래 의도된 용도로 사용하지 못 하게 하는 공격	DOS
TCP 프로토콜의 구조적인 문제를 이용한 공격으로 서버의 동시 가용	SYN Flooding

사용자 수를 SYN 패킷만 보내 점유하여 다른 사용자가 서버를 사용 불가능하게 하는 공격	
패키지와 버킷의 합성어로 통신망을 통해 전송하기 쉽도록 자른 데이터의 전송단위	패킷
대량의 UDP 패킷을 만들어 임의의 포트 번호로 전송하여 응답 메시지(ICMP)를 생성하게 하여 지속해서 자원을 고갈시키는 공격	UDP Flooding
ICMP 패킷을(Ping)을 정상적인 크기 보다 아주 크게 만들어서 전송하면 다수의 IP단편화가 발생하고, 수신 측에서는 단편화된 패킷을 처리(재조합)하는 과정에서 많은 부하가 발생하거나, 재조합 버퍼의 오버플로우가 발생하여 정상적인 서비스를 하지 못하도록 하는 공격 기법	Ping Of Death (죽음의 핑)
출발지 주소를 공격 대상의 IP로 설정하여 네트워크 전체에게 ICMP Echo패킷을 직접 브로드 캐스팅하여 타겟 시스템을 마비시키는 공격	스머프 공격 (Smurf)
출발지 IP와 도착지 IP를 같은 패킷 주소로 만들어 보냄으로써 수신자가 자기 자신에게 응답을 보내게 하여 시스템의 가용성을 침해하는 공격기법	Land Attack
IP Fragment Offset 값을 서로 중첩되도록 조작하여 전송하고, 이를 수신한 시스템이 재조합하는 과정에서 오류가 발생, 시스템의 기능을 마비시키는 Dos 공격방식	티어 드롭(Tear Drop)
공격을 주도하는 해커의 컴퓨터	공격자
공격자에게서 직접명령을 받는 시스템. 여러 대의 에이전트를 관리하는 역할을 함	마스터
공격 대상에 직접 공격을 하는 시스템	에이전트
마스터시스템의 역할을 수행하는 프로그램	핸들러
에이전트 시스템의 역할을 수행하는 프로그램	데몬 프로그램
출발지 IP를 공격 대상 IP로 위조하여 다수의 반사 서버로 요청 정보를 전송, 공격 대상자는 반사 서버로부터 다량의 응답을 받아서 서비스 거부(DoS)가 되는 공격이다.	DRDoS(Distributed Reflection Dos)
과도한 Get 메시지를 이용하여 웹 서버의 과부하를 유발시키는 공격으로 HTTP 캐시 옵션을 조작하여 캐싱 서버가 아닌 웹 서버가 직접 처리하도록 유도, 웹 서버 자원을 소진시키는 서비스 거부 공격 기법	HTTP Get Flooding
HTTP GET 메소드를 사용하여 헤더의 최종 끝을 알리는 개행 문자열인 \r\n\r\n(Hex: 0d 0a 0d 0a)을 전송하지 않고, \r\n(Hex: 0d 0a)만 전송하여 대상 웹 서버와 연결상태를 장시간 지속시키고 연결 자원을 모두 소진시키는 서비스 거부 공격	Slowloris
Content-Length: 9999999 설정 이후 1바이트 씩 전송하여 지속적인 연결 유지를 통해 가용 자원을 소진시키는 공격	RUDY
TCP 윈도우 크기와 데이터 처리율을 감소시킨 상태에서 다수 HTTP 패킷을 지속적으로 전송하여 대상 웹 서버의 연결 상태가 장시간 지속, 연결자원을 소진시키는 서비스 거부 공격	Slow HTTP Read Dos
공격대상 웹 사이트 웹 페이지 주소(URL)를 지속적으로 변경하면서 다량으로 GET 요청을 발생시키는 서비스 거부 공격	Hulk DoS
시스템 또는 서비스의 ID와 패스워드를 크랙하기 위해 ID와 패스워드	사전(Dictionary)

가 될 가능성이 있는 단어를 파일로 만들어 놓고 이 파일의 단어를 대입하여 크랙하는 공격 기법	크래킹
패스워드로 사용될 수 있는 영문자(대소문자), 숫자, 특수문자 등을 무작위로 패스워드 자리에 대입하여 패스워드를 알아내는 공격 기법	무차별 크래킹
특정 호스트의 MAC 주소를 자신의 MAC 주소로 위조한 ARP Reply를 만들어 희생자에게 지속적으로 전송하여 희생자의 ARP Cache table에 특정 호스트의 MAC 정보를 공격자의 MAC 정보로 변경, 희생자로부터 특정 호스트로 나가는 패킷을 공격자가 스니핑하는 기법	ARP 스푸핑
메모리에 할당된 버퍼 크기를 초과하는 양의 데이터를 입력하여 이로 인해 프로세스의 흐름을 변경시켜서 악성 코드를 실행시키는 공격기법	버퍼 오버플로우
버퍼오버플로우 공격 유형에는 메모리 영역중 Local Value 나 Return Address가 저장되는 스택영역에서 발생하는(1)공격과 프로그램 실행 시 동적으로 할당되는 힙 영역에서 할당된 버퍼 크기를 초과하는 데이터를 입력하여 공격자가 원하는 임의의 코드를 실행하는 (2) 공격이 있다.	(1): 스택버퍼 오버플로우 (2): 힙 버퍼 오버플로우
함수 시작 시 복귀 주소를 Global RET라는 특수 스택에 저장해 두고, 함수 종료 시 저장된 값과 스택의 RET값을 비교해 다를 경우 오버플로우로 간주하고 프로그램 실행을 중단하는 방식으로 대응	스택섀드(stack shield)
카나리(canary)라고 불리는 무결성 체크용 값을 복귀 주소와 변수 사이에 삽입해 두고 버퍼 오버플로우 발생 시 카나리 값을 체크, 변할 경우 복귀 주소를 호출하지 않는 방식으로 대응	스택가드
버퍼 오버플로우 공격에 대한 대응 방안 중 리눅스에서 설정 가능한 기법으로 메모리 공격을 방어하기 위해 주소 공간 배치를 난수화하고, 실행 시마다 메모리 주소를 변경시켜 버퍼 오버플로우를 통한 특정 주소 호출을 차단하는 기법	ASLR(Address-Space Layout Randomization)
둘 이상의 프로세스나 스레드가 공유자원을 동시에 접근 할 때 접근 순서에 따라 비정상적인 결과가 발생하는 조건/상황에서 실행되는 프로세스가 임시파일을 만드는 경우 악의적인 프로그램을 통해 그 프로세스의 실행 중에 끼어들어 임시파일을 심볼릭 링크하여 악의적인 행위를 수행하게 하는 공격 기법	레이스 컨디션 공격
컴퓨터 사용자의 키보드 움직임을 탐지해서 저장하고, ID나 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹공격	키로거 공격
시스템 침입 후 침입 사실을 숨긴 채 차후의 침입을 위한 백도어, 트로이 목마 설치, 원격 접근, 내부사용 흔적 삭제, 관리자 권한 획득 등 주로 불법적인 해킹에 사용되는 기능을 제공하는 프로그램의 모음	루트킷(Root kit)
사회 공학의 한 기법으로 특정 대상을 선정한 후 그 대상에게 일반적인 이메일로 위장한 메일을 지속적으로 발송하여, 발송 메일의 본문 링크나 첨부된 파일을 클릭하도록 유도하여 사용자의 개인정보 탈취하는 공격 기법	스피어 피싱
특정 타깃을 목표로 하여 다양한 수단을 통한 지속적이고 지능적인 맞춤형 공격 기법	APT 공격
소프트웨어 개발사의 네트워크에 침투하여 소스 코드의 수정 등을 통	공급망

해 악의적인 코드를 삽입하거나 배포 서버에 접근하여 악의적인 파일로 변경하는 방식을 통해 사용자 PC에 소프트웨어를 설치 또는 업데이트 시에 자동적으로 감염되도록 하는 공격	공격(Supply Chain Attack)
무선 Wifi 피싱 기법으로 공격자는 합법적인 Wifi 제공자처럼 행세하며 노트북이나 휴대전화로 핫스팟에 연결한 무선 사용자들의 정보를 탈취하는 무선 네트워크 공격 기법	이블 트윈 공격
자신이 누구라고 시스템에 밝히는 행위. 객체에게 주체가 자신의 정보를 제공하는 활동	식별
주체의 신원을 검증하기 위한 활동. 주체의 신원을 객체가 인정해 주는 행위	인증
인증된 주체에게 접근을 허용하는 활동. 특정 업무를 수행할 권리를 부여하는 행위	인가
주체의 접근을 추적하고 행동을 기록하는 활동. 식별, 인증, 인가, 가사 개념을 기반으로 수립	책임 추적성
접근 통제 유형 중 시스템 정보의 허용등급을 기준으로 사용자가 갖는 접근 허가 권한에 근거하여 시스템에 대한 접근을 제한하는 방법	MAC
시스템에 대한 접근을 사용자/그룹의 신분 기반으로 제한하는 통제기법	DAC
무결성을 보장하는 최초의 모델로 No Read Down, No Write UP의 속성을 가지고 있다.	비바모델(BIba)
접근통제 보호 모델 중 기밀성을 강조하고, 보안수준이 낮은 주체는 보안 수준이 높은 객체를 읽어서는 안되며, 보안수준이 높은 주체는 보안 수준이 낮은 객체에 기록하면 안 되는 속성을 가지고 있는 모델 No read Up, No Write Down	벨-라파둘라 모델
암호화와 복호화에 같은 키를 쓰는 알고리즘으로 유형에는 블록 암호화 알고리즘과 스트림 암호화 알고리즘이 있다.	대칭키 암호 방식
키를 사용하는 메시지 인증 코드로 메시지의 무결성과 송신자의 인증 보장하는 방식	MAC(Message Authentication Code)
키를 사용하지 않는 변경 감지 코드로 메시지의 무결성 보장하는 방식	MDC(Medification Detection Code)
1999년 국내 한국인터넷진흥원이 개발한 블록 암호화 알고리즘으로 128비트 비밀키로부터 생성된 16개의 64비트 라운드 키를 사용하여 총 16회의 라운드를 거쳐 128비트의 평문 블록을 128비트 암호문 블록으로 암호화하여 출력하는 방식	SEED
2004년 국가정보원과 산학연구협회가 개발한 블록 암호화 알고리즘으로 블록 크기는 128비트이며, 키 길이에 따라 128비트, 192비트, 256비트로 분류되고, 경량 환경 및 하드웨어에서의 효율성 향상을 위해 개발된 암호화 알고리즘	ARIA(Academy, Research Institute Agency)
2001년 미국 표준 기술 연구소(NIST)에서 발표한 블록 암호화 알고리즘으로 블록 크기는 128비트이며, 키 길이에 따라 128비트 192비트 256비트로 나뉘지고, 이것의 라운드 수는 10, 12,	AES

14라운드로 분류되며, 한 라운드는 SubBytes, ShiftRows, MixColumns, AddRoundKey의 4가지 계층으로 구성되는 대칭 키 암호화 알고리즘	
1977년 3명의 MIT 수학 교수가 고안한 큰 인수의 곱을 소인수 분해하는 수학적 알고리즘을 이용하는 공개키 암호화 알고리즘	RSA(Rivest-Shamir-Adleman)
이산대수의 계산이 어려운 문제를 기본 원리로 하며 RSA와 유사하게 전자 서명과 데이터 암호.복호화에 함께 사용가능한 알고리즘	ElGamal
IP계층(3계층)에서 무결성과 인증을 보장하는 인증 헤더(AH)와 기밀성을 보장하는 암호화(ESP)를 이용한 IP보안 프로토콜	IPSec(Internet Protocol Security)
메시지 인증 코드(MAC)를 이용하여 인증과 송신처 인증을 제공해주는 프로토콜로 기밀성(암호화)은 제공하지 않는 프로토콜	인증헤더(AH) (Authentication Header)
메시지 인증 코드(MAC)와 암호화를 이용하여 인증과 송신처 인증과 기밀성을 제공하는 프로토콜. 기밀성, 무결성, 인증 제공	암호화(ESP) 프로토콜(Encapsulation Security Payload)
Key를 주고받는 알고리즘. 공개된 네트워크를 통하여 Key를 어떻게 할 것 인가를 정의, IKE 교환을 위한 메시지를 전달하는 프로토콜	키 관리(IKE)프로토콜 (Internet Key Exchange)
전송계층(4계층)과 응용계층(7계층) 사이에서 클라이언트와 서버간의 웹 데이터 암호화(기밀성), 상호 인증 및 전송 시 데이터 무결성을 보장하는 보안 프로토콜	SSL/TLS
협상된 Cipher Spec을 상대방에게 알리는 프로토콜	Change Cipher Spec protocol
서버와 클라이언트가 서로 인증하고 암호화 키를 협상하는 프로토콜	Handshake Protocol
SSL/TLS 관련 경고 메시지 전달	Alert Protocol
서버와 클라이언트가 정상적인 상태인지 확인하기 위하여 사용	Heartbeat Protocol
OpenSSL 암호화 라이브러리의 하트비트(Heartbeat)라는 확장 모듈에서 클라이언트 요청메시지를 처리할 때 데이터 길이에 대한 검증을 수행하지 않는 취약점을 이용하여 시스템 메모리에 저장된 64KB 크기의 데이터를 외부에서 아무런 제한 없이 탈취할 수 있도록 하는 취약점	하트블리드 (HeartBleed)
합의된 Cipher Suite로 암호화. 복호화, 무결성, 압축.해제 등 수행	Record Protocol
웹상에서 네트워크 트래픽을 암호화 하는 주요 방법 중 하나로 클라이언트와 서버 간에 전송되는 모든 메시지를 각각 암호화하여 전송하는 기술 HTTP를 사용한 애플리케이션에서만 사용 가능	S-HTTP
실무적으로 검증된 개발 보안 방법론 중 하나로써 SW보안의 모범 사례를 SDLC(Software Development Life Cycle)에 통합한 소프트웨어 개발 보안 생명주기 방법론	Seven TouchPoints.
미국 국토안보국의 지원을 받아 수행된 소프트웨어 보증프로젝트의 결과물 중 하나. 보안 활동의 성숙도 수준을 영역별로 측정함으로써 소	BSIMM(Building Security In

소프트웨어 개발에 필요한 보안 능력 향상을 목표로 하는 개발 프레임워크	Maturity Model)
OWASP에서 개발한 개방형 보안 프레임워크. 개방을 원칙으로 소규모, 중규모, 대규모로 점진적인 확대가 가능한 융통성 있는 프레임워크 BSIMM과는 달리 설계 리뷰와 코드 리뷰, 그리고 보안 테스트를 3개의 중요한 검증 활동으로 정의함으로써 이들 활동 간의 연계성 강조	Open SAMM(Software Assurance Maturity Model)
마이크로소프트사가 2004년이 후 자사의 소프트웨어 개발에 의무적으로 적용하도록 고안한 보안강화 프레임워크. SDL은 개발 중인 제품이 보안 위협에 대해 얼마나 강인한가를 측정하기 위해 동일한 제품에 대해 'pre-SDL'과 'post-SDL'의 두 개의 버전으로 테스트	MS-SDL
기밀성, 가용성, 무결성	정보 보안 세 가지 요소
개념 관점, 역할 기반 관점, 활동평가 관점, 활동구현 관점, 취약성 관점 등의 활동중심, 역할기반의 프로세스로 구성된 보안 프레임워크로 이미 운영 중인 시스템에 적용하기 쉬운 보안 개발방법론. 프로그램 설계나 코딩 오류를 찾아 내어 개선하기 위해 개발팀에 취약점 목록을 제공 SDLC 초기단계에 보안 강화를 목적으로 하는 정형화된 프로세스. 활동중심/역할기반의 프로세스로 구성된 집합체.	OWASP CLASP
정보시스템 운영 중 서버가 다운되거나 자연재해나 시스템 장애 등의 이유로 고객에게 서비스가 불가능 한 경우가 종종 발생한다. 이와 같은 상황에서 비상사태 또는 업무 중단 시점부터 업무가 복구되어 다시 정상 가동될 때까지의 시간을 의미하는 용어	재해복구 시간(RTO;Recovery Time Objective)
업무중단 시점부터 데이터가 복구되어 다시 정상가동 될 때 데이터의 손실 허용 시점으로 재해 시 복구 목표 지점의 선정에 활용	RPO(Recovery Point Objective)
외부입력값을 이용하여 웹페이지를 동적으로 구성하는 경우, 웹페이지 사용자의 권한으로 부적절한 스크립트 실행을 유도함으로써, 정보유출 등의 문제를 유발할 수 있다. 타 사용자의 정보를 추출하기 위해 사용되는 공격기법으로 게시판에 악성 스크립트를 작성하여 다른 사람이 그 글을 보았을 때 그 글을 본 사람의 쿠키정보를 빼내가는 해킹 기법	XSS(Cross Site Scripting)
보안 공격기법 중 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위를 특정 웹사이트에 요청하게 하는 공격기법	CSRF(cross site request forgery)
방문자들이 악성 스크립트가 포함된 페이지를 읽어 봄과 동시에 악성 스크립트가 브라우저에서 실행되면서 감염되는 기법	Stored Xss
공격용 악성 URL을 생성한 후 이메일로 사용자에게 전송하면 사용자가 URL 클릭시 즉시 공격 스크립트가 피해자로 반사되어 접속 사이트에 민감정보를 공격자에게 전송하는 기법	Reflected XSS
공격자는 DOM기반 XSS 취약점이 있는 브라우저를 대상으로 조작된 URL을 이메일을 통해 발송하고 피해자가 URL 클릭 시 공격 피해를 당하는 기법	DOM(Document Object Model)XSS
기존 SQL Injection의 확장된 개념으로 한 번의 공격으로 대량의 DB 값이 변조되어 홈페이지에 치명적인 영향을 미치는 공격방법	Mass SQL Injection
DB 쿼리 결과의 참과 거짓을 통해 의도하지 않은 SQL문을 실행함으	Blind SQL

로써 데이터베이스를 비정상적으로 공격하는 방법	Injection
HTML Form 기반 인증을 담당하는 애플리케이션의 취약점이 있는 경우 사용자 인증을 위한 쿼리 문의 조건을 임의로 조작하여 인증을 우회하는 기법	Form SQL Injection
쿼리의 UNION 연산자를 이용하여 한 쿼리의 결과를 다른 쿼리의 결과에 결합하여 공격하는 기법	Union SQL Injection
DB 쿼리에 대한 에러값을 기반으로 한 단계 씩 점진적으로 DB정보를 획득할 수 있는 공격 기법	Error-Based SQL Injection
잘못된 세션에 의한 데이터 정보 노출, 제거되지 않고 남은 디버그 코드, 민감한 데이터를 가진 내부 클래스 사용, 시스템 데이터 정보 노출 등이 있다.	캡슐화 취약점
널 포인터 역참조, 정수를 문자로 변환, 부적절한 자원 해제, 초기화되지 않은 변수 사용으로 발생할 수 있는 보안 취약점	코드 오류 취약점
코드 오류 취약점 중 일반적으로 그 객체가 널(NULL)이 될 수 없다는 가정을 위반했을 때 발생한다. 공격자가 의도적으로 이를 발생시키는 경우, 그 결과 발생하는 예외 상황을 이용하여 추후의 공격을 계획하는 데 사용될 수 있다. 이를 방지하기 위해 널이 될 수 있는 레퍼런스는 참조하기 전에 널 값인지를 검사하여 안전한 경우에만 사용해야 한다.	널 포인터 역참조
코드 오류 취약점 중 사용된 자원을 적절히 해제하지 않으면 자원 누수 등이 발생하고, 자원이 부족하여 새로운 입력을 처리할 수 없게 되는 보안약점	부적절한 자원 해제
코드 오류 취약점 중 메모리 등 해제된 자원을 참조하여 예기치 않은 오류가 발생될 수 있는 보안 약점	해제된 자원 사용
코드 오류 취약점 중 변수를 초기화 하지 않고 사용하여 예기치 않은 오류가 발생될 수 있는 보안 약점	초기화되지 않은 변수 사용
사용자 로그인/로그아웃 정보, 시스템 shutdown/reboot 정보를 확인할 수 있는 wtmp(x) 로그의 내용을 확인할 수 있는 명령어	last
현재 시스템에 로그인한 사용자 정보를 확인할 수 있는 로그로 who, w, users, finger 명령어를 통해서 확인할 수 있음	utmp(x)
로그인에 실패한 정보를 저장하고 있는 유닉스/리눅스 로그 파일명	btmp(x)
기업 내부, 외부 간 트래픽을 모니터링하여 시스템의 접근을 허용하거나 차단하는 시스템. 외부로부터 불법 침입과 내부의 불법 정보유출을 방지하고, 내/외부 네트워크 상호간 영향을 차단하기 위한 보안 시스템	방화벽(firewall)
네트워크에서 발생하는 이벤트를 모니터링하고 비인가 사용자에게 의한 자원접근과 보안정책 위반 행위를 실시간으로 탐지하는 시스템	침입탐지시스템
단말기가 내부 네트워크에 접속을 시도할 때 이를 제어하고 통제하는 기능을 제공하는 솔루션으로 바이러스나 웜 등의 보안 위협뿐만 아니라 불법 사용자에게 대한 네트워크 제어 및 통제 기능을 수행하는 장비	NAC(Network Access Control)
인터넷과 같은 공중망에서 사설망을 구축하여 마치 전용망을 사용하는 효과를 가지는 보안 솔루션	VPN

운영체제에 내재된 결함으로 인해 발생할 수 있는 각종 해킹으로부터 시스템을 보호하기 위해 보안 기능이 통합된 보안 커널을 추가한 운영체제	보안 운영체제(Secure OS)
조직 내부의 중요 자료가 외부로 빠져나가는 것을 탐지하고 차단하고, 정보 유출 방지를 위해 정보의 흐름에 대한 모니터링과 실시간 차단 기능을 제공하는 솔루션	데이터 유출 방지 (DLP; Data Loss Prevention)
비즈니스에서 재난 및 재해로 인해 운영상의 주요 손실을 볼 것을 가정하여 시간 흐름에 따른 영향도 및 손실평가를 조사하는 비즈니스 영향 분석	BIA(Business Impact Analysis)
재해복구계획의 원활한 수행을 지원하기 위해 평상시에 확보하여 두는 인적, 물적 자원 및 이들에 대한 지속적인 관리체계가 통합된 재해복구센터	DRS(Disaster Recovery System)
각종 재해, 장애, 재난으로부터 위기관리를 기반으로 재해복구, 업무복구 및 재개, 비상계획 등을 통해 비즈니스 연속성을 보장하는 체계	비즈니스 연속성 계획(BCP; Business Continuity Plan)
주 센터와 데이터복구센터 모두 운영 상태로 실시간 동시 서비스가 가능한 재해복구센터로 재해 발생시 복구까지의 소요시간(RTO)은 즉시 (이론적으로 0)	Mirror Site
주 센터와 동일한 수준의 자원을 대기 상태로 원격지에 보유하면서 동기, 비동기 방식의 미러링을 통하여 데이터의 최신 상태를 유지하고 있는 재해복구센터로 재해 발생 시 복구까지의 소요 시간(RTO)은 4시간 이내	Hot Site
Hot Site와 유사하나 재해복구센터에 주 센터와 동일한 수준의 자원을 보유하는 대신 중요성이 높은 자원만 부분적으로 재해복구센터에 보유하고 있는 센터로 데이터 백업 주기가 수 시간~1일	Warm Site
데이터만 원격지에 보관하고, 재해 시 데이터를 근간으로 필요 자원을 조달하여 복구할 수 있는 재해 복구센터로 재해 발생 시 복구까지의 소요시간은 수주~수개월	Cold Site
악의적인 해커가 불특정 웹 서버와 웹 페이지에 악성 스크립트를 설치하고, 불특정 사용자 접속 시 사용자 동의 없이 실행되어 의도된 서버 (멀웨어 서버)로 연결하여 감염시키는 공격기법	Drive By Download
독일 지멘스사의 SCADA 시스템을 공격 목표를 제작된 악성코드로 원자력, 전기, 철강, 반도체, 화학 등 주요 산업 기반 시설의 제어 시스템에 침투해서 오작동을 일으키는 악성코드	Stuxnet
미국 비 영리회사인 MITRE 사에서 공개적으로 알려진 소프트웨어의 보안 취약점을 표준화한 식별자 목록	CVE(Common Vulnerabilities and Exposures)
많은 소스에서 하나 혹은 여러 개의 목표 시스템에 대해 서비스 거부 공격을 수행할 수 있는 도구로 공격자가 공격 명령을 일련의 TFN 서버들(혹은 데몬들)에게 보냄으로써 이루어지고, UDP Flood공격 뿐만 아니라 TCP SYN flood 공격, ICMP echo요청 공격, ICMP브로드 캐스트 공격의 수행이 가능한 도구이다.	TFN(Tribes Flood Network)

3계층에서 스니핑 시스템을 네트워크에 존재하는 또 다른 라우터라고 알림으로써 패킷의 흐름을 바꾸는 공격으로 특정 목적지로 가는 패킷을 공격자가 스니핑하는 기법	ICMP Redirect
특정인에 대한 표적 공격을 목적으로 특정인이 잘 방문하는 웹 사이트에 악성코드를 심거나 악성코드를 배포하는 URL로 자동으로 유인하여 감염시키는 공격기법	위터링 홀
네트워크 경로를 알 수 없도록 암호화 기법을 사용하여 데이터를 전송하며, 익명으로 인터넷을 사용할 수 있는 가상 네트워크	토르 네트워크
웹 애플리케이션을 사용하고 있는 서버의 미흡한 설정으로 인해 인덱싱 기능이 활성화되어 있을 경우, 공격자가 강제 브라우저를 통해서 서버 내의 모든 디렉토리 및 파일 목록을 볼 수 있는 취약점	디렉토리 리스팅 취약점
암호화 알고리즘의 실행 시기의 전력 소비, 전자기파 방사 등의 물리적 특성을 측정하여 암호 키 등 내부 비밀 정보를 부 채널에서 획득하는 공격기법	부 채널 공격
미국 비영리 회사인 MTRE 사가 중심이 되어 소프트웨어에서 공통적으로 발생하는 약점을 체계적으로 분류한 목록으로, 소스 코드 취약점을 정의한 데이터베이스 7가지 보안 약점 입력데이터검증 및 표현, 보안기능, 시간 및 상태, 에러처리, 코드품질, 캡슐화, API악용	CWE(common weaknesses enumeration)
CWE보안약점중 XSS, SQL삽입, 버퍼오버플로우 등	입력데이터 검증 및 표현
CWE보안약점 중 중요정보 평문저장, 하드코딩된 패스워드	보안기능
CWE보안약점 중 Deadlock자원에 대한 경쟁조건, 세션고착	시간 및 상태
CWE보안약점 중 처리루틴 누락, 정보 노출	에러처리
CWE보안약점 중 널 포인터 역참조, 부적절한 자원해제	코드품질
CWE보안약점 중 디버거 코드, 시스템 데이터 정보	캡슐화
CWE보안약점 중 DNS Lookup에 의존한 보안 결정	API악용
동일한 테스트 케이스에 의한 반복적 테스트는 새로운 버그를 찾지 못한다는 원리로 테스트 케이스의 정기적 리뷰와 개선 및 다른 시각에서의 접근이 필요하다는 의미를 담고 있다.	살충제 패러독스
테스트 오라클 중 특정한 몇 개의 입력값에 대해서만 기대하는 결과를 제공해 주는 오라클	샘플링 오라클
테스트 오라클 중 모든 입력값에 대하여 기대하는 결과를 생성함으로써 발생된 오류를 모두 검출할 수 있는 오라클	참 오라클
테스트 오라클 중 샘플링 오라클을 개선한 오라클로, 특정 입력값에 대해 올바른 결과를 제공하고, 나머지 값들에 대해서는 휴리스틱(추정)으로 처리하는 오라클	휴리스틱 오라클
테스트 오라클 중 애플리케이션 변경이 있을 때 수행 전과 후의 결과값이 동일한지 확인하는 오라클	일관성 검사 오라클
테스트 결과가 올바른지 판단하기 위해 사전 정의된 참 값을 대입하여 비교하는 방법	테스트 오라클
요구사항을 충족시켜주지 못 한다면, 결함이 없다고 해도 품질이 높다고 볼 수 없는 소프트웨어 테스트 원리이다.	오류 부재의 궤변
애플리케이션의 테스트가 되어야 할 기능 및 특징, 테스트가 필요한 상황을 작성한 문서. 테스트 케이스와 일 대 다의 관계를 가짐	테스트 시나리오

Test Case를 실행 환경에 따라 구분해 놓은 Test Case의 집합 단 시나리오는 포함되지 않은 단순한 테스트 케이스들의 모음	테스트 스위트
테스트 케이스의 실행 순서(절차)를 작성한 문서. 테스트 스텝, 테스트 절차서라고도 함	테스트 스크립트
테스트 대상을 실행하지 않고 구조를 분석하여 논리성을 검증하는 테스트	정적 테스트
소프트웨어를 실행하는 방식으로 테스트를 수행하여 결함을 검출하는 테스트. 화이트 박스 테스트, 블랙박스 테스트, 경험 기반 테스트	동적 테스트
결정 포인트 내의 전체 조건식이 적어도 한 번은 참과 거짓의 결과를 수행하는 테스트 커버리지	결정 커버리지 or 선택커버리지 or 분기 커버리지
결정 조건 내 모든 개발 조건식의 모든 가능한 조합을 100% 보장하는 커버리지 테스트	다중 조건 커버리지
주어진 테스트 케이스에 의해 수행되는 소프트웨어의 테스트 범위를 측정하는 테스트 품질 측정 기준이며, 테스트의 정확성과 신뢰성을 향상시키는 역할을 한다	테스트 커버리지
제어 흐름 그래프에 데이터 사용현황을 추가한 그래프를 통해 테스트하는 기법	데이터 흐름 테스트
프로그램 외부 사용자의 요구사항 명세를 보면서 수행하는 기능 테스트. 소프트웨어의 특징, 요구사항, 설계 명세서 등에 초점을 맞춰 테스트가 이루어짐. 기능 및 동작 위주의 테스트를 진행하기 때문에 내부 구조나 작동 원리를 알지 못해도 가능	블랙박스 테스트
블랙박스 테스트 기법 중 테스트 데이터 값들 간에 최소한 반 번씩을 조합하는 방식이며, 이는 커버해야 할 기능적 범위를 모든 조합에 비해 상대적으로 적은 양의	
등가 분할 후 경계값 부분에서 오류 발생 확률이 높기 때문에 경계값을 포함해서 테스트 케이스를 설계하여 테스트하는 기법	경계값 분석 테스트
시스템에 고의로 실패를 유도하고, 시스템의 정상적 복귀 여부를 테스트하는 기법	회복 테스트
전체 조건식뿐만 아니라 개별 조건식도 참 한 번, 거짓 한 번 결과가 되도록 수행하는 테스트 커버리지 기법	조건/결정 커버리지
사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 사용자의 입장에서 확인하는 테스트로 알파, 베타 테스트가 있다.	인수테스트
시스템 처리 능력 이상의 부하, 즉 임계점 이상의 부하를 가하여 비정상적인 상황에서의 처리를 테스트	스트레스(강도) 테스트
짧은 시간에 사용자가 몰릴 때 시스템의 반응 측정 테스트	스파이크테스트
소프트웨어 요구, 설계, 원시 코드 등의 저작자 외의 다른 전문가 또는 팀이 검사하여 문제를 식별하고 문제에 대한 올바른 해결을 찾아내는 형식적인 검토 기법	인스펙션(Inspection)
테스트의 결과가 참인지 거짓인지를 판단하기 위해서 사전에 정의된 참값을 입력하여 비교하는 기법	테스트 오라클
사용자 요구사항에 대한 단위 모듈, 서브루틴 등을 테스트하는 단계. 인터페이스 테스트, 자료 구조 테스트, 실행 경로 테스트, 오류 처리 테스트 등의 기법이 존재	단위 테스트

통합된 단위 시스템의 기능이 시스템이세 정상적으로 수행되는지를 검증하는 테스트로 기능/비기능 요구사항 테스트 기법이 있다.	시스템 테스트
최종 사용자와 업무의 이해관계자 등이 테스트를 수행함으로써 개발된 제품에 대해 운영 여부를 결정하는 테스트로 알파 테스트, 베타 테스트 기법	인수 테스트
단위 테스트를 통과한 모듈 사이의 인터페이스, 통합된 컴포넌트 간의 상호작용을 검증하는 테스트 단계	통합 테스트
사용자 요구사항에 대한 단위 모듈, 서브루틴 등을 테스트하는 단계	단위 테스트
주로 테스트 대상 클래스와 협력하는 클래스로 가는 출력을 검증하는데 사용	테스트 스파이
실제 협력 클래스의 기능을 대체해야 할 경우에 사용. 실제 협력 클래스의 기능 중 전체나 일부를 훨씬 단순하게 구현	가짜 객체
상향식 통합 테스트와 하향식 통합 테스트 방식을 결합한 테스트 방식이다. 하위 프로젝트가 있는 큰 규모의 통합 테스트에서 사용하는 방식으로 병렬 테스트가 가능하고 시간 절약이 가능하다. 스텝과 드라이버의 필요성이 매우 높은 방식이고, 비용이 많이 소요된다.	샌드위치 통합 테스트
결함의 원인을 찾기 위해 결함을 발생시킨 입력 값, 테스트 절차, 테스트 환경을 명확히 파악하는 방법	구체화
입력값, 테스트 절차, 테스트 환경 중 어떤 요소가 결함 발생에 영향을 미치는지 분석하는 방법	고립화
결함 발생에 영향을 주는 요소를 최대한 일반화 시키는 방법	일반화
테스터가 테스트 절차를 실행하여 발견한 결함을 분석 후 구체화, 고립화, 일반화한 결함으로서 보고된 상태. 결함 보고서에 기록되어 결함 추적의 대상이 된 상태	결함 등록
개발자의 결함 처리가 합당한지, 정확한지 검증이 완료된 상태	결함확인
결함의 원인이 되는 것으로 일반적으로 사람에 의해 생성된 실수를 무엇	에러/오류
애플리케이션에 발생한 결함이 어떤 영향을 끼치며, 그 결함이 얼마나 치명적인지를 나타내는 척도	결함 심각도
결정 포인트 내의 개별 조건식이 적어도 한 번은 참과 거짓의 결과가 되도록 수행하는 테스트 커버리지	조건 커버리지
전체 조건식뿐만 아니라 개별 조건식도 참 한번, 거짓 한 번 결과가 되도록 수행하는 테스트 커버리지	조건/결정 커버리지
개별 조건식이 다른 개별 조건식에 영향을 바디 않고 전체 조건식에 독립적으로 영향을 주도록 함으로써 조건/결정 커버리지를 향상시킨 커버리지	변경조건/결정커버리지
블랙박스 테스트 유형 중에서 SW의 일부 또는 전체를 트리 구조로 분석 및 표현하여 테스트 케이스를 설계하여 테스트하는 기법	분류 트리 테스트
회귀 테스트는 오류를 제거하거나 수정한 시스템에서 오류 제거와 수정에 의해 새로이 유입되는 오류가 없는지 확인하는 일종의 반복 테스트 방법	회귀 테스트
각 응용 프로그램의 내부 구조와 동작을 검사하는 소프트웨어 테스트이다	구조 기반 테스트

검토자료를 회의 전에 배포해서 사전 검토한 후 짧은 시간 동안 회의를 진행하는 형태	워크스루																											
제어 흐름의 복잡한 정보를 정량적으로 표시하는 기법으로 해당 제어 흐름 그래프에서 선형적으로 독립적인 경로의 수를 나타낸다. edge-node+2	맥케이브																											
각 애플리케이션 모듈 또는 컴포넌트의 특정 속성에 해당하는 결함의 수를 측정하여 결함의 분포를 분석	결함분포분석																											
테스트 진행 시간의 흐름에 따른 결함의 수를 측정하여 결함 추세를 분석	결함추세 분석																											
등록된 결함에 대해 특정한 결함 상태의 지속 시간을 측정하여 분석	결함 에이징 분석																											
기능이 기대와 많이 다르게 동작하거나 그 기능이 해야 하는 것을 못하는 결함. 기능 장애 등이 속함	주요 결함																											
제품이나 프로그램이 특정 기준을 충족하지 못하거나 전체에 영향을 주지 않는 일부 기능이 부자연스러운 결함. 사소한 기능 오작동 등이 속함	보통결함																											
사용상의 불편함을 유발하는 결함. 표준위반 UI잘림 등이 속함	경미한 결함																											
클린코드 작성 원칙	가독성, 단순성, 의존성 최소, 중복성 제거, 추상화																											
컴퓨터 프로그램의 소스 코드가 복잡하게 얽힌 모습을 스파게티의 면발에 비유한 표현으로 스파게티 코드는 작동은 정상적으로 하지만 사람이 코드를 읽으면서 그 코드의 작동을 파악하기는 어려운 코드	스파게티 코드																											
(대기시간+서비스 시간)/서비스 시간	HRN 우선순위 계산식																											
<table><tr><th colspan="3">user</th><th colspan="3">group</th><th colspan="3">other</th></tr><tr><th>R</th><th>W</th><th>X</th><th>R</th><th>W</th><th>X</th><th>R</th><th>W</th><th>X</th></tr><tr><td>4</td><td>2</td><td>1</td><td>4</td><td>2</td><td>1</td><td>4</td><td>2</td><td>1</td></tr></table> ex) chmod 777 a.txt	user			group			other			R	W	X	R	W	X	R	W	X	4	2	1	4	2	1	4	2	1	
user			group			other																						
R	W	X	R	W	X	R	W	X																				
4	2	1	4	2	1	4	2	1																				
데니스 리치와 켄 톰슨 등이 함께 벨연구소를 통해 만든 운영체제이며 90% 이상 C언어로 구현되어있고 시스템 프로그램이 모듈화되어 있어서 다른 하드웨어 기종으로 쉽게 이식 가능하며 계층적 트리 구조를 가짐으로써 통합적인 파일 관리가 용이한 운영체제. 교육 및 연구 기관에서 사용되는 범용 다중 사용자 방식의 시분할 운영체제이다.	유닉스																											
생성-> 준비-> 실행-> 대기->완료	프로세스 상태 전이도																											
준비상태에서 실행 상태로 전이하는 프로세스	Dispatch																											
실행상태에서 대기하는 프로세스	Block																											
대기상태에서 준비상태로 전이하는 프로세스	Wake-up																											
프로세스 5가지 상태 중 사용자에게 의해 프로세스가 생성된 상태	Create																											
프로세스 5가지 상태 중 CPU할당을 받을 수 있는 상태	Ready																											
프로세스 5가지 상태 중 프로세스가 CPU를 할당받아 동작중인 상태	Running																											
프로세스 5가지 상태 중 프로세스 실행 중 입출력 처리 등으로 인해 CPU를 양도하고 입출력 처리가 완료까지 대기 리스트에서 기다리는	Waiting																											

상태	
프로세스 5가지 상태중 프로세스가 CPU를 할당받아 주어진 시간 내에 완전히 수행을 종료한 상태	Complete
기업 또는 조직 내부에서 보유하고 있는 컴퓨팅 자원을 이용하여 내부에 구축되어 운영되는 클라우드. 자체 컴퓨팅 자원으로 모든 하드웨어, 소프트웨어 데이터를 수용. 직접적인 통제가 가능하며 보안성을 높일 수 있음	사설 클라우드
클라우드 서비스 제공 업체에서 다중 사용자를 위한 컴퓨팅 자원 서비스를 제공하는 클라우드. 일정한 비용을 지불하고 하드웨어, 소프트웨어 등을 사용. 확장성, 유연성 등이 뛰어남	공용클라우드
기업 또는 조직 내부 자원을 이용한 사설클라우드와 공용클라우드를 모두 사용하는 클라우드	하이브리드 클라우드
클라우드 컴퓨팅 유형중 서버, 스토리지 같은 시스템 자원을 클라우드로 제공하는 서비스 가상 머신 환경제공	인프라형 서비스 IaaS(Infrastructure as a Service)
인프라 생성, 관리 하는 복잡함 없이 애플리케이션 개발, 실행, 관리할 수 있게 하는 플랫폼 제공 서비스 자원 및 개발도구 제공	플랫폼형서비스 Paas Platform as a service
소프트웨어 및 관련 데이터는 중앙에 호스팅되고 사용자는 웹브라우저 등의 클라이언트를 통해 접속하여 소프트웨어를 서비스 형태로 이용하는 서비스. 주문형 소프트웨어라고도 함	소프트웨어형 서비스(SaaS)
하나의 프로세스가 CPU를 차지하고 있을 때, 우선순위가 높은 다른 프로세스가 현재 프로세스를 중단시키고 CPU를 점유하는 스케줄링	선점형 스케줄링
프로세스는 같은 크기의 CPU 시간을 할당, 프로세스가 할당된 시간 내에 처리 완료를 못 하면 준비 큐 리스트의 가장 뒤로 보내지고, CPU는 대기 중인 다음 프로세스로 넘어가는 알고리즘	라운드 로빈(Round Robin)
가장 짧은 시간이 소요되는 프로세스를 먼저 수행하고, 남은 처리 시간이 더 짧다고 판단되는 프로세스가 준비 큐에 생기면 언제라도 프로세스가 선점되는 알고리즘	SRT(Shortest Remaining Time First)
작업들을 여러 종류 그룹으로 분할, 여러 개의 큐를 이용하여 상위단계 작업에 의한 하위단계 작업이 선점 당하는 알고리즘. 각 큐는 자신만의 독자적인 스케줄링을 가짐	다단계 큐
프로세스가 도착하는 시점에 따라 그 당시 가장 적은 서비스 시간을 갖는 프로세스가 종료 시까지 자원을 점유하는 알고리즘	SJF(Shortest job Fitst)
둘 이상의 프로세스가 서로가 가진 한정된 자원을 요청하는 경우 발생하는 상태로, 다중프로세싱 환경에서 두 개 이상의 프로세스가 특정 자원할당을 무한정 대기하는 상태	교착상태
교착상태를 해결하기 위해, 회피 기법을 사용하는 기법. 자원의 상태를 감시하고 프로세스는 사전에 자신의 작업에 필요한 자원 수를 제시하는 교착상태 회피	은행가 알고리즘
교착상태 발생 조건 중 프로세스가 자원을 배타적으로 점유하여 다른	상호배제

프로세스가 그 자원을 사용할 수 없음	
교착상태 발생 조건 중 한 프로세스가 자원을 점유하고 있으면서 또 다른 자원을 요청하여 대기하고 있는 상태	점유와 대기
교착상태 발생 조건 중 한 프로세스가 점유한 자원에 대해 다른 프로세스가 선점할 수 없고, 오직 점유한 프로세스만이 해제 가능	비선점
교착상태 발생조건 중 두 개 이상의 프로세스 간 자원의 점유와 대기가 하나의 원형을 구성한 상태	환형 대기
교착상태 해결 방안 중 상호배제를 제외한 나머지 교착상태 발생 조건을 위배(부정)하는 방안	예방
교착상태 해결 방안 중 안전한 상태를 유지할 수 있는 요구만 수락	회피
교착상태 해결 방안 중 시스템의 상태를 감시 알고리즘을 통해 교착상태 검사	발견
교착상태 해결 방안 중 Deadlock이 없어질 때까지 프로세스를 순차적으로 Kill하여 제거, 희생자 선택해야 하고 기아 상태 발생	복구
시스템 부하가 많아서 준비 큐에 있는 낮은 등급의 프로세스가 무한정 기다리는 현상	기아(Starvation)
기아 현상을 해결하기 위한 방법으로 오랫동안 기다린 프로세스의 우선순위를 높여주는 기법	에이징(Aging)
프로토콜 기본요소 3가지	구문/의미/타이밍
서로 다른 시스템에 있는 두 개체간 데이터 교환을 원활히 하기 위한 일련의 통신규약	프로토콜
OSI 계층 중 비트를 전송하는 계층	물리 계층
TCP/IP에서 오류가 발생하면 ()메시지를 보내서 오류가 발생했음을 알린다. () 안에 들어갈 용어	ICMP
원하는 정보를 원하는 수신지 또는 기기에 정확하게 전송하기 위한 기반 인프라	네트워크
비동기 전송모드라고 하는 광대역 전송에 쓰이는 스위칭 기법	ATM
전송이 가능한 작은 블록으로 나누어지는 기법	단편화
단편화 되어 온 조각들을 원래 데이터로 복원하는 기법	재조립
상위 계층의 데이터에 각종 정보를 추가하여 하위 계층으로 보내는 기법	캡슐화
데이터의 전송량이나 속도를 제어하는 기법	연결제어
전송 중 잃어버리는 데이터나 오류가 발생한 데이터를 검증하는 제어 기법	오류제어
송신과 수신 측의 시점을 맞추는 기법	동기화
하나의 통신 회선에 여러 기기들이 접속할 수 있는 기술	다중화
송신과 수신지의 주소를 부여하여 정확한 데이터 전송을 보장하는 기법	주소 지정
인접 라우터와 정보 공유하여 목적지까지의 거리와 방향을 결정하는 라우팅 프로토콜 알고리즘. 벨만-포드 알고리즘 사용	거리 벡터 알고리즘
링크 상태 정보를 모든 라우터에 전달하여 최단 경로 트리를 구성하는 라우팅 프로토콜 알고리즘. 다익스트라 알고리즘 사용	링크 상태 알고리즘
최초의 라우팅 프로토콜. 거리 벡터 알고리즘 활용. 30초 주기로 전체 라우팅 정보 갱신. 변화 업데이트 시 많은 시간 소요. 라우팅 루프 발생 가능 15홉 제한이 있다.	RIP

링크 상태 알고리즘 사용. 발생한 변경 정보에 대해 RIP보다 빠른 업데이트. 토폴로지에 대한 정보가 전체 라우터에 동일하게 유지	OSPF(Open Shortest Path First)
사용자 정보를 유지하기 dnlsk 질의 및 디렉터리 서비스의 등록, 수정, 삭제 및 검색을 위한 목적으로 개발한 인터넷 프로토콜	LDAP(Lightweight Directory Access Protocol)
ERD를 통한 자동화된 데이터 모델링, UML 자동생성	설계
개발 산출물 관리, 변경 통제, 대표적으로 Subversion, Selenium이 있음	형상관리
일정 관리를 위한 간트 차트, 네트워크 다이어그램, WBS 등 제공 대표적으로 Redmine, OpenProj가 있음	프로젝트 관리
외부 인터넷망이 차단된 상태에서 인트라넷망만을 활용하여 개발환경을 구축하는 방식	온프레미스 방식
아마존, 구글, 마이크로소프트 등 클라우드 공급 서비스를 하는 회사들의 서비스를 임대하여 개발환경을 구축하는 방식	클라우드 방식
On-Premise와 클라우드 방식을 혼용하는 방식	하이브리드 방식
애플리케이션들이 단일 운영체제 상에서 실행되도록 해주는 기술. 하이퍼바이저 없이 운영체제가 격리된 프로세스로 동작하기 때문에 오버헤드가 낮음. 대표적으로 도커 기술이 있음	컨테이너
물리적으로 떨어져 있는 다양한 장비들을 연결하기 위한 수단. 중계장치(라우터, 스위치 등)의 가상화 수행 대표적으로 SDN, NFV기술이 있음	네트워크 가상화 기술
네트워크 리소스를 특정 사용층이 독점하도록 하는 통신방식	서킷 스위칭
RIP의 문제점 개선을 위해 시스코에서 개발. 네트워크 상태를 고려하여 라우팅	IGRP
문서 이름(릴리즈 노트 이름), 제품 이름, 버전정보, 릴리즈 날짜, 참고 날짜, 노트 버전 등의 정보	헤더
제품 및 변경에 대한 간략한 전반적 개요	개요
릴리즈 버전의 새로운 기능목록과 릴리즈 노트의 목적에 대한 개요, 버그 수정 및 새로운 기능 기술	목적
버그의 간단한 설명 또는 릴리즈 추가 항목요약	이슈 요약
버그 발견에 따른 재현 단계 기술	재현 항목
수정.개선의 간단한 설명 기술	수정. 개선 내용
버전 변경에 따른 최종 사용자 기준의 기능 및 응용 프로그램상의 영향도 기술	사용자 영향도
버전 변경에 따른 소프트웨어의 지원 프로세스 및 영향도 기술	소프트웨어 지원 영향도
소프트웨어 및 하드웨어 설치 항목, 제품, 문서를 포함한 업그레이드 항목 메모	노트
회사 및 표준 제품과 관련된 메시지. 프리웨어 및 불법 복제 방지, 중복 등 참조에 대한 고지 사항	면책 조항
사용자 지원 및 문의에 관련한 연락처 정보	연락 정보
개발이 완료된 제품 소프트웨어를 고객에게 전달하기 위한 형태로 포장하는 과정을 지칭하는 용어	제품 소프트웨어 패키징

최종 사용자인 고객에게 개발 과정에서 정리된 제품의 릴리즈 정보를 제공하는 문서. 릴리즈 노트 정보들은 테스트를 진행하고, 개발 팀에서 제공하는 사양에 대해 최종 승인된 후 문서를 통해 배포된다.	릴리즈 노트
DRM구성요소 중 콘텐츠를 제공하는 저작권자	콘텐츠 제공자
DRM구성요소 중 쇼핑몰 등으로서 암호화된 콘텐츠 제공	콘텐츠 분배자
DRM구성요소 중 콘텐츠를 메타 데이터와 함께 배포 가능한 단위로 묶는 기능	패키저
DRM구성요소 중 원본을 안전하게 유통하기 위한 전자적 보안 장치	보안 컨테이너
DRM구성요소 중 배포된 콘텐츠의 이용 권한을 통제	DRM 컨트롤러
DRM구성요소 중 소비자와 유통업자 사이에 발생하는 거래에 대해 디지털 저작권 라이선싱을 중개하고 라이선스 발급을 수행하는 정산소. 모니터링 서비스를 지원하여 디지털 저작물의 이용 내역을 근거로 신뢰할 수 있는 저작권료의 정산 및 분배가 이루어짐	클리어링 하우스
디지털 저작물의 저작권 보호 및 정확한 위치 추적을 위해 특정한 번호를 부여하는 일종의 바코드 시스템	DOI(Digital Object Identifier)
멀티미디어 관련 요소 기술들이 통일된 형태로 상호 운용성을 보장하는 멀티미디어 표준 규격	MPEG-21
디지털 콘텐츠/ 웹 서비스 권리 조건을 표현한 XML 기반의 마크업 언어	XrML
역공학을 위한 공격을 막기 위해 프로그램의 소스 코드를 알아보기 힘든 형태로 바꾸는 기술	코드 난독화
커널 암호화 방식으로 데이터베이스 파일을 직접 암호화 하고, 접근 제어와 감사기록 기능이 추가된 데이터베이스 보안 강화 기술	Secure DB
안전한 소프트웨어 개발을 위해 소스코드 등에 존재할 수 있는 잠재적인 보안 취약점을 제거하고, 보안을 고려하여 기능을 설계 및 구현하는 등 소프트웨어 개발 과정에서 지켜야 할 일련의 보안활동	시큐어 코딩(Secure Coding)
사용자 측면에서 패키징 이후 설치, 제품 소프트웨어를 사용하는 데 필요한 주요 내용을 기록한 문서	제품 소프트웨어 매뉴얼
사용자가 제품을 구매한 후 설치 시 참조하는 문서	제품 소프트웨어 설치 매뉴얼
사용자가 소프트웨어 사용에 필요한 내용, 제반 절차, 환경 등의 내용을 포함하는 문서	제품 소프트웨어 사용자 매뉴얼
사용자가 사용하기 편하도록 배포 정보를 포함하여 개발된 컴포넌트 또는 패키지가 제품화된 형태	제품 소프트웨어 배포본
백업받고자 하는 데이터 전체에 대해 백업하는 방식	전체 백업
마지막 전체 백업 이후 변경된 모든 데이터를 백업하는 방식	차등 백업
정해진 시간을 기준으로 그 이후에 변경된 파일 만을 백업하는 방식	증분 백업
시스템의 요구분석부터 유지보수까지 전 공정을 체계화한 절차 요구사항분석-> 설계->구현->테스트->유지보수	소프트웨어 생명 주기(SDLC)
전체 시스템을 기능에 따라 나누어 개발하고, 이를 통합하는 분할과 정복 접근 방식의 방법론 나찌-슈나이더만 차트 사용	구조적 방법론
정보시스템 개발에 필요한 관리 절차와 작업 기법을 체계화한 방법론.	정보공학 방법론

개발주기를 이용해 대형 프로젝트를 수행하는 체계적인 방법론					
'객체'라는 기본 단위로 시스템을 분석 및 설계하는 방법론 복잡한 현실 세계를 사람이 이해하는 방식으로 시스템에 적용하는 방법론		객체지향 방법론			
소프트웨어를 구성하는 컴포넌트를 조립해서 하나의 새로운 응용 프로그램을 작성하는 방법론		컴포넌트 기반 방법론			
절차보다는 사람이 중심이 되어 변화에 유연하고 신속하게 적응하면서 효율적으로 시스템을 개발할 수 있는 신속 적응적 경량 개발방법론		애자일 방법론			
특정 제품에 적용하고 싶은 공통된 기능을 정의하여 개발하는 방법론 임베디드 소프트웨어를 작성하는데 유용한 방법론		제품 계열 방법론			
xp 12가지 가치 중 매일 여러 번 씩 소프트웨어를 통합하고 빌드해야 한다는 원리		CI(지속적인 통합)			
xp 12가지 가치 중 공통적인 이름 체계와 시스템 서술서를 통해 고객과 개발자 간의 의사소통을 원활하게 한다는 원리		메타포어			
xp 12가지 가치 중 작성해야 하는 프로그램에 대한 테스트를 먼저 수행하고, 이테스트를 통과할 수 있도록 실제 프로그램의 코드를 작성한다는 원리		테스트 기반 개발 (TDD)			
xp 12가지 가치 중 프로그램의 기능을 바꾸지 않으면서 중복제거, 단순화 등을 위해 시스템 재구성 한다는 원리		리팩토링			
소프트웨어 각 기능의 원시 코드 라인 수의 낙관치, 중간치, 비관치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 방식 (낙관치+4*중간치+비관치)/6		LOC			
보험이 제안한 모형으로 프로그램 규모에 따라 비용을 산정하는 방식		COCOMO			
소프트웨어 개발주기의 단계별로 요구할 인력의 분포를 가정하는 방식		푸트남(Putnam)			
요구 기능을 증가시키는 인자별로 가중치를 부여하고, 요인별 가중치를 합산하여 총 기능의 점수를 계산하여 비용을 산정하는 방식		기능점수(FP)			
일의 순서를 계획적으로 정리하기 위한 수렴 기법으로 비관치, 중간치, 낙관치의 3점 추정방식을 통해 일정을 관리하는 기법		PERT			
여러 작업들의 수행 순서가 얹혀 있는 프로젝트의 일정을 계산하는 기법		주 공정법(CPM)			
주 공정 연소법으로 자원제약사항을 고려하여 일정을 작성하는 방법		중요 연쇄 프로젝트 관리 (CCPM)			
소프트웨어 공학의 소프트웨어 설계에서 공통으로 발생하는 문제에 대해 자주 쓰이는 설계 방법을 정리한 패턴		디자인 패턴			
목적		생성패턴	구조패턴	행위패턴	디자인 패턴 분류
의미		객체의 생성방식을 결정하는 패턴	객체를 조직화 하는데 유용한 패턴	객체의 행위를 조직, 관리, 연합하는데 사용	
범위	클래스	Factory method	Adapter	Template method, Interpreter	
	객체	Singleton,	Bridge,	Iterator,	

		Abstract factory, Builder, Prototype	Composite , Decorator, Facade, Flyweight, Proxy	Command, Chain of Responsibility, State, Strategy, Mediator, Memento, Visitor, Observer	
복잡한 인스턴스를 조립하여 만드는 구조로 복합 객체를 생성할 때 객체를 생성하는 방법과 객체를 구현하는 방법을 분리함으로써 동일한 생성 절차에서 서로 다른 표현 결과를 만들 수 있는 디자인 패턴					Builder
처음부터 일반적인 원형을 만들어 놓고 그것을 복사한 후 필요한 부분만 수정하여 사용하는 패턴, 생성할 객체의 원형을 제공하는 인스턴스에서 생성할 객체들의 타입이 결정되도록 설정하며 객체를 생성할 때 갖추어야 할 기본 형태가 있을 때 사용되는 디자인 패턴					Prototype

1. 현행 시스템 파악 3단계 : 구성/기능/인터페이스 파악 -> 소프트웨어 구성 및 아키텍처[시스템 만들기 위한 청사진] 파악-> 하드웨어 및 네트워크 구성 파악
 소프트웨어 아키텍처: 시스템 개발에 중요한 영향을 미치는 결정들, 소프트웨어 시스템 구축 및 지속적 개선 용이하게 함

2. 요구공학 단계: 타당성 조사-> 도출[인터뷰, 관찰, 벤치마킹 문헌 조사 등]-> 분석-> 명세-> 확인/검증

요구사항들 간 상충되는 것을 해결하고, 소프트웨어 범위를 파악하며 소프트웨어가 환경과 어떻게 상호작용하는지를 이해하고, 요구사항을 정제하여 소프트웨어 요구사항을 정리하는 업무

요구사항 검토는 요구 사항 검토 담당자들이 요구 사항 명세서를 수작업으로 분석하는 방법 종류 : 동료검토, 워크 스루, 인스펙션

요구사항 관리 절차

요구사항 협상-> 요구사항 기준선[합의된 요구사항 명세서]-> 요구사항 변경관리[요구사항 기준선을 기준으로 모든 변경을 공식적으로 통제하는 기법]->요구사항 확인

요구사항 명세서 기능적 요구사항[반드시 구현해야 할 기능], 비 기능적 요구사항[품질, 성능, 보안 등의 제약사항]을 구분할 수 있다.

3. 자료흐름도 구성요소

자료의 처리/반환 과정을 표현한다(Process)

자료의 흐름을 표현한다.(Data Flow)

파일, 데이터 베이스 등 저장소의 위치를 표현한다.(Data store)

자료의 출처와 도착지를 표현한다.(Terminator)

4. 객체지향 분석 방법론 종류

럼바우 방법 (객체모델링[객체, 클래스 추출]->동적모델링[상태다이어그램 이용]->기능모델링[자료흐름도DFD이용])

부치방법 절차지향 프로그램: 동사 객체지향프로그램: 명사,
coda와 yourdon방법 E-R 다이어그램을 사용하여 객체행위 모델링,
(E-R다이어그램: 현실세계에 존재하는 데이터와 그들 간의 관계를 사람이 이해할 수 있는 형태로 명확하게 표현하기 위해서 가장 널리 이용되고 있는 모델 개체, 속성, 관계로 기술함)

Wirfs-brock[위프스-브록]방법: 분석과 설계 간의 구분이 없고 연속적으로 수행하는 기법

객체지향 설계원칙(SOLID)

단일 책임 원칙: 객체는 단 하나의 책임만을 가져야함,

개방 폐쇄 원칙 : 소프트웨어 개체는 확장에는 열려있고 수정 시에는 닫혀 있어야함

리스코프 치환의 원칙: 자식클래스는 최소한 부모 클래스에서 가능한 행위 수행가능

인터페이스 분리의 원칙: 인터페이스를 클라이언트에 특화되도록 분리

의존성 역전의 법칙: 의존 관계를 맺을 때 변화하기 쉬운 것 또는 자주 변화하는 것보다 변화하기 어렵거나 거의 변화가 없는 것에 의존해야함

객체지향의 구성

클래스, 객체, 속성(데이터), 메소드(객체의 함수), 메시지(객체간 통신)

객체지향 기법

추상화[공통 성질을 추출하여 상위클래스를 구성하는 것], 다형성[오버로딩, 오버라이딩], 정보은닉, 상속성

객체지향 소프트웨어 개발 과정에서 산출물을 명세화, 시각화, 문서화 할 때 사용되는

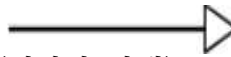
모델링 기술과 방법론을 통합하여 만든 표준화된 범용 모델링 언어(UML)

UML 다이어그램 관계유형

----->(의존관계)



(연관단계)



(일반화 단계)



(실체화 단계)



(집합관계)



(합성관계)

UML 다이어그램 종류

구조적 다이어그램, 행위 다이어그램, 상호작용다이어그램

클래스 다이어그램과 서로 밀접한 관련이 있으며 거의 유사한 표기법을 사용하는 UML다이어그램(객체 다이어그램)

문제해결을 위한 객체를 정의하고 객체 간 상호작용 메시지 흐름을 시간 또는 순서로 표현하는 다이어그램 이름(시퀀스 다이어그램)

시스템 내부에 있는 객체의 활동 간의 처리흐름을 모델링하는 범용적인 다이어그램 (액티비티 다이어그램)

UML 스테레오타입 예시

<<include>>:포함 <<extend>>:확장 <<boundary>>:경계[상호작용]

<<entity>>:엔티티[정보관리]

<<control>>:제어

5. 절차보다는 사람이 중심이 되는 방법론으로 변화에 유연하고 신속하게 적응하면서 일정한 주기를 반복함(애자일 방법론)

개인과의 상호작용이 프로세스와 도구보다 우선,

변화에 대한 대응이 계획 준수보다 우선,

동작하는 소프트웨어가 포괄적 문서화보다 우선

고객과의 협력이 계약 협상보다 우선

workflow를 가시화하고 작업 중인 항목의 제한 및 작업 소요시간을 측정할 수 있는 작업 지시서를 SW 개발에 적용한 애자일 방법론(Kanban)

스크럼 구성요소:Product Backlog, Sprint Backlog, Sprint, Daily-meeting, Burndown Chart

6. 실세계의 물리현상을 특정한 목적에 맞추어 이용하기 쉬운 형식으로 표현(모델링) 구조 모델링[컴포넌트들의 유형, 인터페이스, 내부 설계 구조 및 이들의 상호연결 구조], 행위 모델링[구성요소들의 기능들과 이들이 언제, 어떠한 순서로 기능을 수행하고 상호작용하는지]

9. 고객 요구사항을 중심으로 4가지 관점으로 소프트웨어 아키텍처를 설계하는 기법 (SW아키텍처 4+1 view) use case view, 논리관점, 구현관점, 프로세스관점, 배치관점

10. 기능적으로 클라이언트와 서버 사이의 통신을 담당하는 시스템 소프트웨어를 의미함(미들웨어)

미들웨어 솔루션 유형별 활용

데이터베이스, RPC, MOM, TP모니터, ORB, WAS

13. 하나의 릴레이션을 구성하는 속성들의 부분집합을 x와 y라 할 때, 어느 시점에서든 릴레이션 내의 모든 튜플(행)을 대상으로 한 X값에 대한 Y값이 항상 하나면 “X가 Y를 함수적으로 결정한다”(함수 종속성)

완전함수종속, 부분 함수 종속, 이행 함수종속, 결정자 함수 종속

4. 프로세스와 엔티티의 상관관계를 이용하여 구축된 엔티티 관계도 검증도구 (CRUD Matrix) C(생성)R(조회)U(변경)D(삭제)

모든 엔티티에 CRUD가 한번 이상 표기되어야함, 모든 엔티티에 C가 한번 이상 존재해야함, 모든 엔티티에 R이 한번이상 존재해야함, 모든 단위 프로세스가 하나 이상의 엔티티에 표기 되어야 함

5. 어떤 정해진 속성(컬럼) 값을 기준으로, 동일한 값을 가진 하나 이상의 테이블의 튜플을 같은 장소에 저장하는 물리적 기법(클러스터)

6. 대용량의 테이블이나 인덱스를 유지보수가 쉬운 단위로 분리하는 것

9. 슈퍼타입은 상위 공통 데이터의 내용을 담는 타입이며, 서브타입은 하위 분할된 정보를 담는 타입

10. 하나의 데이터베이스 관리 시스템이 여러 CPU에 연결 된 저장장치들을 제어하는

형태의 데이터베이스(분산 데이터베이스)

분할 투명성: 하나의 논리적 릴레이션이 여러 단편으로 분할되어 각 단편의 사본이 여러 사이트에 저장

위치 투명성: 사용하려는 데이터의 저장 장소 명시 불필요. 위치정보가 카탈로그에 유지되어야 함

지역사상 투명성: 지역 DBMS와 물리적 DB사이의 매핑 보장. 각 지역 시스템 이름과 무관한 이름 사용 가능

중복투명성: DB 객체가 여러 site에 중복되어 있는지 알 필요가 없는 성질

장애 투명성: 구성요소의 장애에 무관한 트랜잭션의 원자성 유지

병행 투명성: 다수 트랜잭션 동시 수행 시 결과의 일관성 유지, Time Stamp, 분산 2단계 Locking을 이용 구현

11. 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 성능개선을 위하여 의도적으로 정규화 원칙을 위배하는 행위(비정규화[반정규화/역정규화])

특정 속성들만 접근이 잦을 경우 컬럼을 쪼개서 테이블을 만들어 사용하는 기법(수직 분할)

스키마는 동일하지만, 그 데이터 값을 이용하는 방법이 튜플별로 구분 지어지는 경우(수평분할)

데이터베이스 키의 특징

유일성, 최소성

데이터베이스 키의 종류

후보키(candidate key)	릴레이션을 구성하는 속성들 중에 튜플을 유일하게 식별할 수 있는 하나 또는 몇 개의 속성의 집합
기본키(primary key)	릴레이션의 유일한 식별자(유일성, 최소성 모두 만족) 기본키로 지정된 속성은 같은 값을 갖지 못함 후보키 중에서 선정된 키로 중복값을 가질 수 없음 Not null, Unique, 외래키(Foreign key)로 참조될 수 있음
대체키(alternate key)	후보키가 둘 이상 되는 경우에 기본키로 선택되지 못한 후보키들 후보키 = 기본키 + 대체키
슈퍼키(super key)	유일성만 있고 최소성이 없는 속성의 집합
외래키(foreign key)	한 테이블의 키 중 다른 테이블의 튜플을 식별할 수 있는 키
관계대수	원하는 정보가 무엇인가를 정의하고 원하는 정보를 유도하기 위한 과정을 정의한 절차적 언어 종류: 일반 집합 연산자, 순수관계 연산자

관계 해석	원하는 정보가 무엇인지만을 정의하는 비절차적인 언어이다.
-------	---------------------------------

순수 관계 연산자 중 릴레이션 R에서 조건을 만족하는 튜플 반환 수평연산	σ 선택
순수 관계 연산자 중 릴레이션 R에서 주어진 속성들의 값으로만 구성된 튜플반환. 수직연산	π 프로젝트
순수 관계 연산자 중 공통 속성을 이용해 R과 S의 튜플들을 연결해 만들어진 튜플 반환	\bowtie 조인
순수 관계 연산자 중 릴레이션 S의 모든 튜플과 관련 있는 R의 튜플 반환 $R \div S$ 의 경우	\div 디비전
논리 데이터 모델링 속성	개체, 속성, 관계
객체지향 분석 방법 중 E-R다이어그램을 사용하여 객체의 행위를 모델링하며, 객체식별, 구조식별, 주체 정의, 속성 및 관계정의, 서비스 정의 등의 과정으로 구성되는 방법	Coad와 yourdon방법
\square E-R 다이어그램 기호	개체
\diamond E-R 다이어그램 기호	관계
\bigcirc E-R 다이어그램 기호	타원
\odot E-R 다이어그램 기호	다중 값 속성
- E-R 다이어그램 기호	관계-속성 연결
소프트웨어 가동을 위해 하드웨어, 소프트웨어, 네트워크 등 다양한 주변기기 등이 결합 되어 있으며 제작된 소프트웨어에 대해 언제, 어디서나 실행시키더라도 쉽게 구동가능	플랫폼
자바(Junit), C++, Net 등 다양한 언어를 지원하는 단위 테스트 프레임워크. 소프트웨어 함수나 클래스 같은 서로 다른 구성 원소를 테스트할 수 있게 해주는 도구	xUnit
DTE(데이터 터미널 장치)와 DCE(데이터 회선 종단장치)간 인터페이스를 제공하는 프로토콜	X.25
애플리케이션 소프트웨어를 구축하고 통합을 위해 정의된 프로토콜 세트	(Open API)
속성-값 쌍 또는 키-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷	Json
데이터를 연계하려는 DB시스템 간 직접 연결을 통해 데이터를 연계	DB Link
구조화된 요구 사항 명세서를 작성하기 위해 소프트웨어 생명 주기의 전체 단계를 연결해 자동화된 일관성 분석기능을 제공해 주는 통합 도구	CASE 종류 Upper CASE, Middle CASE, Lower CASE, I-CASE
인터뷰와 같이 수행하면 더 많은 정보 추출이 가능하며 훈련된 요원의 주재로 과정 정돈하는 것이 키포인트 JAD:Joint Application Development	브레인스토밍

하향식 개발을 위한 시스템 분석 및 설계 문서화 도구 가시적 도표, 총체적 도표, 세부적 도표로 구성된다.	HIPO
정규화를 거치지 않으면 데이터베이스의 데이터들이 불필요하게 중 복되어 릴레이션 조작 시 문제가 발생하는 현상	이상 현상
이상현상 종류	삽입이상, 수정(갱신)이상, 삭제이상
데이터베이스 이상현상의 원인이 되는 데이터 중복성을 제거 하여 데이터의 무결성을 보존하는 기법	정규화
타당성조사 -> 도출 -> 분석 -> 명세-> 확인/검증	요구공학 개발 절차
자료흐름도(DFD)에서 자료의 처리/변환 과정을 표현	프로세스(process)
자료흐름도(DFD)에서 자료의 흐름을 표현	자료흐름(Data Flow)
자료흐름도(DFD)에서 파일, 데이터 베이스 등 저장소의 위치를 표 현한다.	자료 저장소(Data Store)
자료흐름도(DFD)에서 자료의 출처와 도착지를 표현	단말(terminator)
자료사전 (DD: Data Dictionary)에서 자료의 정의(A는 ~로 구성되 어 있다)	=
자료사전 (DD: Data Dictionary)에서 자료의 연결(A와 B를 연결)	+
자료사전 (DD: Data Dictionary)에서 자료의 생략(없어도 되는 자 료)	()
자료사전 (DD: Data Dictionary)에서 자료의 선택(A이거나 B)	[]
자료사전 (DD: Data Dictionary)에서 자료의 반복	{ }
자료사전 (DD: Data Dictionary)에서 자료의 설명	**
객체지향 분석 방법론 중 가장 일반적으로 사용되는 방법으로 분석 활동을 객체모델, 동적모델, 기능모델로 나누어 수행하는 방법	럼바우 방법
럼바우 방법론 활동 단계 중 정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계 를 규정. 실세계 문제영역으로부터 객체와 클래스를 추출해 그들 간의 관계를 연관화, 집단화, 일반화 중심으로 규명. 클래스의 속성 과 연산을 함께 표현함으로써 시스템의 정적 구조를 생성	객체 모델링
럼바우 방법론 활동 단계 중 상태 다이어그램을 사용하여 시스템의 행위를 기술하는 모델링	동적 모델링
럼바우 방법론 활동 단계 중 자료 흐름도(DFD)를 이용, 다수의 프 로세서들 간의 자료 흐름을 중심으로 처리 과정을 표현. 어떤 데이 터를 입력하면 어떤 결과를 구할 것인지 표현	기능 모델링
UML다이어그램 중 클래스 다이어그램과 밀접한 관련이 있으며 거 의 유사한 표기법(Notation)을 사용한다. 이 두 다이어그램 모두 시 스템의 정적인 구조를 시각화하기 위해 사용된다.	객체 다이어그램
UML다이어그램 중 문제해결을 위한 객체를 정의하고 객체 간 상호 작용 메시지 흐름을 시간 또는 순서로 표현하는 다이어그램	시퀀스 다이어그램
URM다이어그램 중 시스템 내부에 있는 활동 간의 처리흐름을 모델 링하는 범용적인 다이어그램이다. 사건 발생에 따른 객체들 간의 행	액티비티 다이어그 램

위에 대한 상호관계를 표현한다.	
UML 접근 제한자 중 “#”기호 이름	Protected
UML 접근 제한자 중 “-”기호 이름	private
UML 접근 제한자 중 “+”기호 이름	public
스크럼 구성요소 중 구현할 기능과 특성, 요구사항의 우선순위 나열	Product Backlog
스크럼 구성요소 중 해당 Sprint기간 동안 수행되어야 할 Task	Sprint Backlog
스크럼 구성요소 중 통상 4~6주의 Timebox 성격을 가진 반복 개발 주기	Sprint
애자일 방법론 중 하나인 스크럼 기법에서 사용되는 스프린트가 끝난 시점이나 일정 주기별로 스프린트 주기를 되돌아 보며 정해 놓은 규칙 준수 여부, 개선점 등을 확인하고 기록하는 과정	스프린트 회고(Sprint Retrospective)
스크럼 구성요소 중 매일 약 15분 정도의 짧은 회의, 진척사항 검토 및 위험 확인	Daily-미팅
스크럼 구성요소 중 개발을 완료하기 까지 남은 작업량을 보여주는 그래프	Burndown Chart
실세계의 물리현상을 특정한 목적에 맞추어 이용하기 쉬운 형식으로 표현하는 일을 의미	모델링(modeling)
시스템에서 제공되어야 할 특정 기능을 정의함. 사용자 요구 사항 중에 수행될 기능과 관련되어 있는 입력과 출력 및 그들 사이의 처리과정이나 목표로 하는 제품의 구현을 위해 SW가 가져야 하는 기능적 속성을 의미함	기능적 요구사항
시스템의 전체적 품질이나 기능적 요구사항의 구현 시 고려해야하는 제약사항을 정의함.	비기능적 요구사항
요구사항 관리 프로세스 중 가용한 자원과 수용 가능한 위험 수준에서 구현 가능한 기능을 협상하기 위한 기법	요구사항 협상
요구사항 관리 프로세스 중 공식적으로 검토되고 합의된 요구사항 명세서	요구사항 기준선
요구사항 관리 프로세스 중 요구사항 기준선을 기반으로 모든 변경을 공식적으로 통제하기 위한 기법	요구사항 변경 관리
요구사항 관리 프로세스 중 구축된 시스템이 이해관계자가 기대한 요구사항에 부합되는지 확인하기 위한 방법	요구사항 확인
객체지향 설계원칙(SOLID) 중 객체는 단 하나의 책임만을 가져야 함. 어떤 클래스를 변경해야 하는 이유는 오직 하나뿐이어야 하며 같은 이유로 변화하는 것끼리 묶고 다른 이유로 변화하는 것끼리는 분리함	단일 책임 원칙
객체지향 설계원칙(SOLID) 중 기존의 코드를 변경하지 않으면서 기능을 추가할 수 있도록 설계가 되어야 함. 소프트웨어 개체는 확장에는 열려있고 수정 시에는 닫혀있어야 함	개방 폐쇄 원칙
객체지향 설계원칙(SOLID) 중 일반화 관계에 대한 것으로 자식 클래스는 최소한 자신의 부모 클래스에서 가능한 행위는 수행할 수 있어야 함. 하위 클래스 및 타입들은 상위 타입들이 사용되는 곳에 대체될 수 있어야 하는 설계 원칙. 자식 클래스가 부모 클래스 기능을 재정의 하지 않으면 대체가능	리스코프 치환의 원칙

객체지향 설계원칙(SOLID) 중 인터페이스를 클라이언트에 특화되도록 분리하라는 설계 원칙. 하나의 일반적인 인터페이스보다 구체적인 여러 개의 인터페이스가 나옴	인터페이스 분리의 원칙
객체지향 설계원칙(SOLID) 중 의존 관계를 맺을 때 변화하기 쉬운 것 또는 자주 변화하는 것보다는 변화하기 어렵거나 거의 변화가 없는 것에 의존하라는 것임. 추상화된 것에 의존하게 만들고 구체 클래스에 의존하도록 만들지 않도록 함	의존성 역전의 법칙
소프트웨어 시스템의 구조를 비롯한 시스템 개발에 중요한 영향을 미치는 결정들로, 소프트웨어 시스템 개발에서 특정 시스템에 대하여 요구 되는 기능과 품질을 확보하고 또한 소프트웨어 시스템의 구축 및 지속적인 개선이 용이하도록 하는 역할을 함	소프트웨어 아키텍처
시스템 요구사항 중 아키텍처에 영향을 주는 요구사항. 아키텍처 설계에 더 집중할 수 있기 위해 이것을 먼저 잘 식별한 후 아키텍처 설계에 효과적으로 반영	아키텍처 드라이버
소프트웨어 아키텍처의 설계절차에서 반복적으로 발생하는 문제에 대해 미리 만들어진 솔루션	아키텍처 패턴
아키텍처 패턴 중 하위 모듈들의 그룹으로 나눌 수 있는 구조화된 프로그램에서 사용할 수 있다. 각 하위 모듈들은 특정한 수준의 추상화를 제공하며, 각 계층은 상위계층에 서비스를 제공한다	계층화 패턴
아키텍처 패턴 중 하나의 서버와 다수의 클라이언트, 두 부분으로 구성됨. 서버 컴포넌트는 다수의 클라이언트 컴포넌트로 서비스를 제공한다. 클라이언트가 서버에 서비스를 요청하면 서버는 클라이언트에게 적절한 서비스를 제공하며 계속 클라이언트로부터의 요청을 대기함	클라이언트-서버 패턴
아키텍처 패턴 중 마스터와 슬레이브 두 부분으로 구성됨. 마스터 컴포넌트는 동등한 구조를 지는 슬레이브 컴포넌트들로 작업을 분산하고, 슬레이브가 반환한 결과 값으로부터 최종 결과 값을 계산	마스터-슬레이브 패턴
아키텍처 패턴 중 데이터 스트림을 생성하고 처리하는 시스템에서 사용할 수 있음. 각 처리 과정은 필터컴포넌트에서 이루어지며, 처리되는 데이터는 파이프를 통해 흐른다. 이 파이프는 버퍼링 또는 동기화 목적으로 사용될 수 있다.	파이프-필터 패턴
분리된 컴포넌트들로 이루어진 분산 시스템에서 사용되고, 이 컴포넌트들은 원격 서비스 실행을 통해 상호작용이 가능한 패턴으로 컴포넌트 간의 통신을 조정하는 역할을 수행	브로커 패턴
아키텍처 패턴 중 각 컴포넌트를 피어(peers)라하고 피어는 클라이언트로서 피어에게 서비스를 요청할 수도 있고, 서버로서 각 피어에게 서비스를 제공할 수도 있다. 피어는 클라이언트 또는 서버 혹은 둘 모두 동작할 수 있으며, 시간이 지남에 따라 역할이 유동적으로 바뀔 수 있다.	피어 투 피어 패턴
아키텍처 패턴 중 주로 이벤트를 처리하며 이벤트 소스, 이벤트 리스너, 채널 그리고 이벤트 버스의 4가지 주요 컴포넌트들을 갖는다. 소스는 이벤트 버스를 통해 특정 채널로 메시지를 발행하며, 리스너	이벤트-버스 패턴

는 특정 채널에서 메시지를 구독한다. 리스너는 이전에 구독한 채널에 발행된 메시지에 대해 알림을 받는다.	
MVC패턴이라고도 하는 이 패턴은 대화형 애플리케이션이며 다음의 3부분으로 나누어진다. 모델은 핵심 기능과 컨트롤러를 포함한다. 뷰는 사용자에게 정보를 표시하며, 컨트롤러는 사용자로부터의 입력을 처리한다. 이는 정보가 사용자에게 제공되는 방식과 사용자로부터 받아들여지는 방식에서 정보의 내부적인 표현을 분리하기 위해 나뉘며, 이는 컴포넌트를 분리하며 코드의 효율적인 재사용을 가능하게함	모델 - 뷰 - 컨트롤러 패턴
고객 요구사항을 중심으로 4가지 관점으로 소프트웨어 아키텍처를 설계하는 기법. 사용사례 관점은 시스템의 외부 사용자 관점에서 사용 사례들 간의 관계 정의. 시스템의 여러 가지 측면을 고려하기 위한 다양한 관점(view)을 바탕으로 정의되며 사실상의 표준임	SW 아키텍처 4+1
SW아키텍처 4+1 View의 구성 중 시스템의 외부 사용자 관점에서 사용사례들 간의 관계를 정의	사용 사례 관점(Use Case View)
SW아키텍처 4+1 View의 구성 중 상위 수준에서 시스템의 논리적인 구조/행위를 클래스 인터페이스, 협력관계로 정의	논리 관점(Logical View)
SW아키텍처 4+1 View의 구성 중 독립적으로 실행되는 컴포넌트와 이들 간 관계를 정의	구현관점
SW아키텍처 4+1 View의 구성 중 시스템의 병렬처리 및 동기화 처리를 위한 스레드와 프로세스 정의	프로세스 관점
SW아키텍처 4+1 View의 구성 중 실행되는 시스템 하드웨어와 소프트웨어 관계를 정의	배치(Deployment) 관점
객체지향 구성에서 같은 종류의 객체들의 집합에 공통 속성과 행위를 정의함. 객체지향 프로그램의 기본적인 사용자 정의 데이터형임	클래스(Class)
객체지향 구성에서 클래스의 인스턴스. 자신 고유의 데이터를 가지며 클래스에서 정의한 행위를 수행함	객체(object)
객체지향 구성에서 객체의 데이터	속성(Attribute)
객체 지향 구성에서 객체의 행위. 클래스로부터 생성된 객체를 사용하는 방법임	메소드(Method)
객체 지향 구성에서 객체간의 통신을 말함	메시지(Message)
객체 지향 기법에서 속성과 메소드를 하나로 묶어서 객체로 구성 유지보수가 용이. 재사용성이 높은 S/W 개발이 가능. 정보은닉으로 내부자료의 일관성 유지. 객체 간 인터페이스를 이용. 종속성을 최소화	캡슐화
객체 지향 기법에서 공통 성질을 추출하여 슈퍼클래스 구성. 객체 중심의 안정된 모델을 구축. 현실 세계를 자연스럽게 표현. 분석의 초점이 명확해짐	추상화
객체 지향 기법에서 캡슐화된 항목을 다른 객체로부터 숨김. 메시지 전달에 의해 다른 클래스 내의 메소드가 호출됨	정보은닉
객체 지향 기법에서 부모 클래스의 속성과 메소드를 상속받아 사용함. 부모와 자식 클래스 간의 관계가 슈퍼 클래스와 서브클래스로 유지됨. 부모클래스는 추상적이며, 자식클래스는 구체적 성질을 가	상속성

<p>집</p> <p>한 객체의 상태가 바뀌면 그 객체에 의존하는 다른 객체들한테 연락이 가고 자동으로 내용이 갱신되는 방법으로 일대다 의존성을 가지는 디자인 패턴과 서로 상호작용을 하는 객체 사이에서는 가능하면 느슨하게 결합하는 디자인을 사용해야한다.</p>	<p>옵저버패턴(Observable pattern)</p>
<p>객체를 생성하기 위한 인터페이스를 따로 정의하며 어떤 클래스에 객체를 생성하는 일을 서브 클래스가 담당토록 하는 패턴으로 객체 생성의 책임을 서브클래스에게 위임시키고 서브 클래스에 대한 정보를 은닉하고자 할 경우 사용한다.</p>	<p>팩토리 메소드 패턴</p>
<p>클래스의 재사용성을 높이기 위해 클래스 간의 기능을 변환제공하여 호환성을 확보하는 패턴으로 호환성 없는 인터페이스 때문에 함께 사용할 수 없는 클래스를 변경하여 함께 작동하도록 해준다.</p>	<p>어댑터 패턴</p>
<p>간단한 언어의 문법을 정의하는 방법과 그 언어로 문장을 구성하는 방법, 문장을 해석하는 방법을 제시하는 패턴으로 각 문법 규칙을 클래스로 표현하기 때문에 언어를 쉽게 구현할 수 있는 장점이 있다.</p>	<p>인터프리터 패턴 (Interpreter pattern)</p>
<p>서브 시스템이 복잡할 경우 간단한 인터페이스를 통해 서브 시스템의 주요 기능을 사용할 수 있는 패턴</p>	<p>파사드 패턴(Facade pattern)</p>
<p>전역 변수를 사용하지 않고 객체를 하나만 생성하도록 하여 생성된 객체를 어디에서든지 참조할 수 있도록 하는 디자인 패턴. 지정한 클래스의 인스턴스가 반드시 한 개만 존재하도록 하는 패턴. 클래스에서 만들 수 있는 인스턴스가 오직 하나 일 경우에 이에 대한 접근은 어디에서든지 하나로만 통일해 제공한다.</p>	<p>싱글톤패턴(singleton pattern)</p>
<p>조직 내.외부에 존재하는 시스템이 연동을 통해 상호 작용하기 위한 접속방법이나 규칙</p>	<p>내.외부 인터페이스</p>
<p>인터페이스 설계를 위해 필요한 데이터베이스 산출물로 개념모델링 과정에서 도출한 개체타입과 관련속성, 식별자 등에 대한 개괄적인 정보를 포함하는 문서</p>	<p>개체(Entity)정의서</p>
<p>데이터 베이스 제품 제작업체에서 제공하는 클라이언트와 데이터베이스를 연결하기 위한 미들웨어</p>	<p>데이터베이스</p>
<p>응용 프로그램의 프로시저를 사용하여 원격 프로시저를 로컬 프로시저처럼 호출하는 방식의 미들웨어</p>	<p>RPC(Remote Procedure Call)</p>
<p>메시지 기반의 비동기형 메시지 전달 방식 미들웨어. 서로 다른 이기종 분산 데이터 시스템의 데이터 동기를 위하여 주로 사용됨</p>	<p>MOM(Message Oriented Middleware)</p>
<p>온라인 업무에서 트랜잭션을 처리, 감시하는 미들웨어. 사용자 수가 증가하여도 빠른 응답 속도를 유지해야 하는 업무에 적합함</p>	<p>TP-모니터</p>
<p>코바(CORBA) 표준 스펙을 구현한 객체지향 미들웨어다. 최근에는 TP-모니터가 가지고 있는 장점(트랜잭션 처리, 모니터링 등)을 추가로 구현</p>	<p>ORB(Object Request Broker)</p>
<p>웹환경을 구현하기 위한 미들웨어 HTTP 세션 처리를 위한 웹서버 기능 뿐만 아니라 민감한 기업 업무까지 자바, EJB 컴포넌트 기반으로 구현 가능</p>	<p>WAS(Web Application Server)</p>

어떤 정해진 속성(컬럼)값을 기준으로, 동일한 값을 가진 하나 이상의 테이블의 튜플(행, Row)를 같은 장소에 저장하는 물리적인 기법. 디스크로부터 데이터를 읽어오는 시간을 줄이기 위해 조인이나 자주 사용하는 테이블의 데이터를 디스크의 같은 위치에 저장시키는 방법	클러스터
여러 개의 하드디스크로 디스크 배열을 구성하여 파일을 구성하고 있는 데이터 블록들을 서로 다른 디스크들에 분산 저장하는 기술. 데이터를 나누는 다양한 방법이 존재하며 이 방법들을 레벨이라 하는데, 레벨에 따라 저장장치의 신뢰성을 높이거나 전체적인 성능을 향상시키는 등의 다양한 목적을 만족시킬 수 있음	RAID(Redundant Array of Independent)
대용량의 테이블이나 인덱스의 유지보수 관리가 쉬운 단위로 분리하는 방법	파티셔닝
사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블	뷰
하나의 데이터베이스 관리시스템(DBMS)이 여러 CPU에 연결된 저장장치들을 제어하는 형태의 데이터 베이스. 논리적으로 같은 시스템에 속하지만, 컴퓨터 네트워크를 통해 물리적으로 분산되어 있는 데이터들의 모임. 데이터베이스를 연결하는 빠른 네트워크 환경을 이용하여 데이터베이스를 여러 지역 노드로 위치시켜 사용성/ 성능 등을 극대화 시킨 데이터베이스	분산 데이터베이스
TCP Sequence number의 보안성 문제점 때문에 발생하는 네트워크 공격	TCP 세션 하이재킹(Session Hijacking)
인터넷과 같은 분산 환경에서 리소스에 대한 정보와 자원 사이의 관계- 의미 정보를 기계(컴퓨터)가 처리할 수 있는 형태. 실세계에 존재하는 모든 개념들과 개념들의 속성, 그리고 개념들 간의 관계 정보를 컴퓨터가 이해할 수 있도록 서술해 놓은 개념 명세서	온톨로지(Ontology)
온톨로지 형태를 자동화된 기계가 처리하도록 하는 지능형 웹	시맨틱웹(semantic web)
지체되는 소프트웨어 개발 프로젝트에 인력을 추가하는 것은 개발을 늦출 뿐이다 라는 법칙으로 인력이 추가돼서 개발 생산성이 향상되지 않고 오히려 그 인력 때문에 방해된다는 의미의 법칙	브룩스의 법칙(Brook's Law)
전송계층에 위치하면서 근거리 통신망이나 인트라넷, 인터넷에 연결된 컴퓨터에서 실행되는 프로그램 간에 일련의 옥텟을 안정적으로 순서대로, 에러없이 교환할 수 있게 해주는 프로토콜	TCP
비연결성이고 신뢰성이 없으며, 순서화되지 않은 데이터그램 서비스를 제공하는 전송(4계층)계층의 통신 프로토콜이다. 1980년 데이빗 리드가 설계하였고, 현재 IERF RFC 768 표준으로 정의되어 있으며 TCP와 함께 데이터그램으로 알려진 단문 메시지를 교환하기 위해 사용되는 프로토콜	UDP

송수신 간의 패킷 단위로 데이터를 교환하는 네트워크에서 정보를 주고받는 데 사용하는 통신 프로토콜	IP
IP네트워크상에서 IP주소를 MAC주소(물리 주소)로 변환하는 프로토콜	ARP
IP호스트가 자신의 물리 네트워크 주소(MAC)는 알지만 IP주소는 모르는 경우, 서버로부터 IP주소를 요청하기 위해 사용하는 프로토콜	RARP
IP 패킷을 처리할 때 발생하는 문제를 알려주는 프로토콜로 메시지 형식은 8바이트의 헤더와 가변 길이의 데이터 영역으로 분리한다.	ICMP
호스트 컴퓨터와 인접 라우터가 멀티캐스트 그룹 멤버십을 구성하는데 사용하는 통신 프로토콜. 화상회의, IPTV에서 사용된다.	IGMP(Internet Group Management Protocol)
IPv4는 (1)Bit 주소길이를 가지며 전송방식은 유니캐스트, 멀티캐스트, (2)이다.	(1): 32 (2):브로드캐스트
IPv6는 (1)Bit 주소길이를 가지며 전송방식은 유니캐스트, 멀티캐스트, (2)이다.	(1):128 (2)애니캐스트
IPv4를 IPv6로 변환하는 방법에는 (1),(2),(3)이 있다.	터널링, 주소변환, 듀얼스택
오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼으로, 일반 PC급 컴퓨터들로 가상화된 대형 스토리지를 형성하고 그 안에 보관된 거대한 데이터 세트를 병렬로 처리할 수 있도록 개발된 자바 소프트웨어 프레임워크로 구글, 야후 등에 적용한 기술은 무엇인가?	하둡(Hadoop)
식별자 표기 시 두어에 자료형을 붙이는 표기법. 식별자 표기 시에 int형일 경우 n, char 형일 경우 c, 문자열일 경우 sz를 붙임	헝가리안 표기법
웹과 같은 분산 하이퍼미디어 환경에서 자원의 존재/상태 정보를 표준화된 HTTP 메서드로 주고받는 웹 아키텍처. RoyFielding의 2000년 논문에 의해 소개되었다. 리소스, 메서드, 메시지 3개의 요소로 구성된다	REST(Representational State Transfer)
정보보호 및 개인정보보호를 위한 일련의 조치와 활동이 인증기준에 적합함을 인터넷 진흥원 또는 인증기관이 증명하는 제도. ISMS와 PIMS를 통합하였다.	ISMS-p
해시 알고리즘 중 160비트 암호화 해시함수이다. 1993년 미국 NIST에서 개발하였다. 512비트를 입력받아서 160비트를 출력한다. 보안강도가 약해서 SHA-256이나 SHA-512로 대체되고 있다. 전자서명에 활용	SHA-1
해시 알고리즘 중 128비트 기반 암호화 해시함수이다. 결과값이 16개 문자열이다. 설계상 결함으로 사용이 줄어드는 추세이다. 메시지 무결성 검사에 쓰임	MD5
해시 알고리즘 중MD5를 변형하여 만든 해시함수. 128비트에서 256비트까지 다양한 크기가 가능하다.	HAVAL
해시 알고리즘 중 64비트 CPU에 최적화 되어있다. 32비트 CPU에	Tiger

서도 빠르게 동작가능하다.	
프로그래밍 언어의 한 종류로 하나의 응용 소프트웨어를 제어하기 위한 용도로 쓰이는 언어. 소스코드를 컴파일 하지 않고도 실행 할 수 있는 프로그래밍 언어. 인터프리터를 사용해서 코드 한 줄 한줄 즉시 해석하고 실행. 대표적으로 JavaScript, ActionScript, AutoHotKey, JSP, Perl, PHP, Python, Lua, Ruby, VBS	스크립트 언어
번역가가 문서 전체를 번역해서 넘겨주듯이 프로그램 전체 라인을 처음부터 끝까지 해독해 목적코드로 만든 후에 실행함. 이것을 사용하는 언어로는 C, C++, COBOL, Ada, FORTRAN, PASCAL 등이 있음	컴파일러 언어
통역사와 같은 작업을 함. 프로그램 라인(line) 단위로 한 줄씩 해석 하고 바로 실행. 이것을 사용하는 언어는 JavaScript, BASIC, JCL, Basic	스크립트 언어
다음에 인출할 명령어의 주소를 가지고 있는 레지스터	P C (P r o g r a m Counter)
메모리 주소를 일시적으로 저장하는 레지스터	M A R (M e m o r y Address Register)
기억장치에 쓰여질 데이터 혹은 기억장치로부터 읽혀진 데이터를 일시적으로 저장하는 버퍼레지스터	M B R (M e m o r y Buffer Register)
가장 최근에 인출된 명령어 코드가 저장되는 레지스터	I R (I n s t r u c t i o n Register)
데이터나 연산결과를 일시적으로 저장하는 레지스터	A C (A c c u m u l a t o r)
명령어 수행과정	인출-해독-실행-저장
입출력 장치 제어기(I/O Device Controller)가 CPU에 의한 프로그램의 실행없이 자료의 이동을 하는 방식	D M A (D i r e c t Memory Access)
시스템 소프트웨어 중 하드웨어와 소프트웨어 자원을 관리하고 컴퓨터 프로그램을 위한 공통 서비스를 제공하는 프로그램	운영체제(operating System)
시스템 소프트웨어 중 어셈블리어를 기계어로 변환해 주는 프로그램	어셈블러(Assembler)
시스템 소프트웨어 중 고급 언어로 작성한 원시 프로그램을 기계어인 목적 프로그램으로 바꾸어 주는 프로그램	컴파일러 또는 언어번역기
시스템 소프트웨어 중 고급언어나 코드화된 중간 언어를 입력받아 목적 프로그램 생성 없이 직접 기계어를 생성하여 실행해주는 프로그램	인터프리터 (Interpreter)
시스템 소프트웨어 중 원시 프로그램을 번역하기 전에 미리 언어의 기능을 확장한 원시 프로그램을 생성하는 시스템 프로그램	전처리기(Preprocessor)
시스템 소프트웨어 중 서로 독립적으로 작성되고 번역된 목적 프로그램을 호출 및 연계시키는 시스템 프로그램	링커
시스템 소프트웨어 중 실행 Code를 주기억장치에 적재하여 실행 가능하도록 해주는 시스템 프로그램	로더(loader)
로더의 종류 중 별도의 로더 없이 언어번역 프로그램이 로더의 기능까지 수행하는 방식	컴파일 즉시 로더 (Compile and go Loader)

UNIX에서 컴퓨터 내부를 관리하는 Kernel과 사용자 간의 인터페이스를 담당하며, 세션 별 변수 설정 등 사용자 지정 상태로 사용자가 요청한 명령어를 해석하는 기능을 하는 것	셸(Shell)
네트워크에 대한 공격이나 침입을 실시간적으로 차단하고, 유해트래픽에 대한 조치를 능동적으로 처리하는 시스템이다.	IPS(Intrusion Prevention System)
JavaScript를 사용한 비동기 통신기술로 클라이언트와 서버 간에 XML 데이터를 주고받는 기술 브라우저가 가지고 있는 XMLHttpRequest 객체를 이용해서 전체 페이지를 새로 고치지 않고도 페이지의 일부만을 위한 데이터를 로드하는 기법	AJAX
단위 시간 내에 하나 이상의 명령어를 중첩 수행하여 Performance를 향상시키는 멀티프로세스 환경에서의 명령어 처리 매커니즘	파이프라인(Pipeline)
모든 하드웨어가 가상화되어 가상자원의 풀을 구성하고 데이터센터 전체를 운영하는 소프트웨어가 필요한 기능 및 규모에 따라 동적자원을 할당, 관리 하는 역할을수행하는 데이터센터이다.	소프트웨어 정의 데이터센터 DDC : Software Defined Data Center)
서버와 저장 장치를 네트워크로 연결하는 방식으로, 구성 설정이 간편하여 별도의 운영체제를 가진 서버 한 곳에서 파일을 관리하기 때문에 서버 간 스토리지 및 파일 공유가 용이한 스토리지 장치 구성 방식	NAS(Network Attached Storage)
튜플의 개수를 지칭하는 용어	카디널리티
애트리뷰트의 개수를 지칭하는 용어	차수
하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자값들의 집합을 지칭하는 용어	도메인
웹 서비스에 대한 정보인 WSDL을 등록하고 검색하기 위한 저장소로 공개적을 접근, 검색이 가능한 레지스트리이자 표준	UDDI(Universal Description, Discovery and Integration)
프로젝트의 특성과 필요에 따라 소프트웨어 개발 프로세스, 기법, 산출물 등을 비즈니스 적으로 또는 기술적인 요구에 맞도록 최적화하는 과정 및 방법론이다.	테일러링
객체지향 분석 방법론 중 분석과 설계 간의 구분이 없고 고객 명세서를 평가해서 설계 작업까지 연속적으로 수행하는 방법	Wirfs-Brock방법
서버는 클라이언트에서 요청한 소프트웨어의 실행코드를 스트림 형태로 제공하고, 클라이언트는 서버로부터 스트리밍되는 소프트웨어 코드를 클라이언트 PC의 자원을 이용하여 실행하는 방법으로 소프트웨어의 실행을 클라이언트에서 책임지는 기술을 이르는 용어	리치클라이언트
서버와 네트워크, 프로그램 등의 정보시스템이 시스템의 장애에 대응하여 상당히 오랜 기간 동안 지속적으로 정상 운영이 가능한 성질을 이르는 용어	고가용성
메모리 관리 기법의 하나로, 프로그램이 동적으로 할당했던 메모리 영역 중에서 필요 없게 된 영역을 해제하는 기능	가비지 콜렉션(GC)
SW의 규모를 측정 및 예측하는 기법으로써 1979년 미국 IBM의	기능점수(Function

Allen J. Albercht에 의해 제안되었다.	Point)