|  |  |
| --- | --- |
| **API** | 데이터를 송신 시스템의 DB에서 읽어와 제공하는 애플리케이션 프로그래밍 인터페이스 프로그램 |
| **APT (지능형 지속 위협)** | 특정 타겟을 목표로 다양한 수단을 통한 **지속적이고 지능적인** 맞춤형 공격기법 |
| **ARIA** | 국가정보원과 산학연협회가 개발한 학계, 연구기관, 정부의 영문 앞 글자를 따 명명한 양방향 알고리즘의 한 종류 |
| **BGP** | 라우팅 프로토콜은 RIP, OSPF 등이 있는데 라우팅 프로토콜 중에서 EGP의 단점을 보완한 것 6" |
| **cat** | UNIX 명령어 중 파일 내용을 화면에 표시하는 명령어 |
| **chmod** | 특정 파일 또는 디렉토리 퍼미션 수정 명령어 |
| **chown** | 파일이나 디렉토리의 소유자, 소유 그룹 수정 명령어 |
| **CMS (Contents Management System)** | 다양한 미디어 포맷에 따라 각종 콘텐츠를 작성.수집.관리.배급(생산~활용~폐기) 전 공급과정을 관리하는 기술 |
| **CSRF (Cross-Site Request Forgery)** | 보안기능 보안 약점 중 하나 악성 스크립트 구문을 삽입하여 **자신의 의도와 무관하게** 공격자가 의도한 행위를 하게 되는 공격 |
| **DBMS** | 데이터베이스를 관리해주는 소프트웨어 |
| **DFD 데이터 흐름도** | 데이터가 각 프로세스를 따라 흐르면서 변환되는 모습을 나타낸 그림 |
| **DLP 데이터 유출 방지** | 내부 정보 유출 방지를 의미하며 기업 내에서 이용하는 다양한 주요 정보들을 보호하고 외부 유출 방지를 위해서 사용하는 보안기술 |
| **DNS  (Domain Name System)** | 도메인 네임을 IP 주소로 변환하는 시스템 |
| **DOI (Digital Object Identifer)** | 디지털저작물의 저작권보호 및 정확한 위치추적이 가능한 시스템 |
| **DRM 컨트롤러** | DRM 구성요소 중 하나로 배포된 콘텐츠에 대한 **이용권한**을 통제 |
| **EAI** | 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간의 정보연결, 연계, 통합을 가능하게 해주는 솔루션 |
| **ERD (E-R다이어그램)** | 업무분석 결과로 도출된 엔티티와 엔티티 간의 관계를 도식화한 다이어그램 |
| **ESB** | 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간 하나의 시스템으로 관리 운영할 수 있도록 **서비스 중심**의 통합을 지향하는 아키텍처 또는 기술 |
| **fsck** | UNIX 명령어 중 파일 시스템을 검사 및 보수하여 무결성을 검사하는 명령어 |
| **FTP** | TCP/IP 프로토콜을 가지고 서버와 클라이언트 사이의 파일을 전송하기 위한 프로토콜 |
| **GRANT OPTION FOR** | 다른 사람에게 주는 권한을 뺏자 |
| **Hipo** | 하향식 소프트웨어 개발 문서화 도구. 기능과 자료의 의존 관계를 동시에 표현할 수 있어 이해하기 쉬움. 차트 종류로 가시적 도표, 총체적 도표, 세부적 도표가 존재 |
| **HRN** | 대기중인 프로세스 중 현재 응답률이 가장 높은 것을 선택 우선순위 = (대기시간 서비스 시간) / 서비스 시간 |
| **MPEG-21** | 멀티미디어 관련요소 기술들이 통일된 형태로 상호운용성을 보장하는 멀티미디어 표준규격 |
| **MVC패턴** | 사용자 인터페이스로부터 비즈니스 로직을 분리하려 애플리케이션의 시각적 요소나 그 이면에서 실행되는 비즈니스 로직을 서로 영향없이 쉽게 고칠 수 있는 패턴 |
| **OLAP** | 사용자가 정보에 대해 직접 접근해 대화식으로 정보를 분석하고 의사결정에 활용하는 과정 다차원으로 이루어진 데이터로부터 통계적으로 요약한 정보를 분석하여 의사결정에 활용하는 방식 |
| **OPT** | 페이지 교체 알고리즘 중 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 방식 |
| **OSI 7계층** | ISO에서 개발한 컴퓨터 네트워크 프로토콜. 디자인과 통신을 계층으로 나누어 설명한 개방형 시스템 상호 연결 모델 |
| **PFF (페이지 부재 빈도)** | 페이지 부재율의 상한과 하한을 정해 직접적으로 페이지 부재율을 예측하고 조절해서 페이지 교체 현상을 줄이는 기법 |
| **PHP** | 동적 웹페이지 제작에 사용되는 언어. 인터페이스 방식으로 자체 인터프리터를 제공하며, 대중적인 언어의 문법과 유사하기 때문에 배우기 쉬움 |
| **Ping of Death (죽음의핑)** | 인터넷 프로토콜 범위 이상의 패킷을 전송하여 네트워크를 마비시키는 공격 |
| **PnP (Plug and Play)** | 운영체제에 하드웨어를 새로 설치했을 때 해당 하드웨어를 사용하는데 필요한 시스템 환경을 운영체제가 자동으로 구성해주는 기능. Window의 특성 중 하나 |
| **RAID** | 여러 개의 하드디스크에 일부 중복된 데이터를 나눠 저장하는 기술 |
| **REST** | HTTP URL을 통해 자원을 명시하고 HTTP 메소드를 통해 해당 자원에 대한 생성.조회.갱신.삭제 등의 명령을 적용할 수 있는 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처 |
| **rollback** | 트랜잭션 수행 시 오류가 발생하면, 오류 이전의 특정 지점으로 되돌려주는 명령어 |
| **RPO (Recovery Point Object)** | 재해복구 프로세스 중 하나.  재해로 인해 서비스가 중단되었을 때 유실을 감내할 수 있는 손실 허용 시점. 복구시점 |
| **RTCP** | RTP 패킷 전송 품질 제어, 세션에 참여한 참여자에게 제어 정보 전송, 하위 프로토콜은 데이터 패킷과 제어 패킷의 다중화 제공의 역할을 하는 프로토콜 |
| **RTO (Recovery Time Object)** | 재해복구 프로세스 중 하나. 재해로 인해 서비스가 중단되었을 때 서비스를 복구하는데 걸리는 시간 |
| **Ruby on Rails** | Ruby 언어를 기반으로 한 서버 개발 프레임워크 |
| **RUDY** | 요청 헤더의 Content-Length를 비정상적으로 크게 설정하여 메시지 바디 부분을 매우 소량으로 보내 계속 연결 상태를 유지. 자원을 소진시키는 기법 |
| **SDLC** | 소프트웨어 생명 주기. 소프트웨어 개발의 전 과정을 각 단계별로 나눈 것 |
| **Slowloris** | HTTP 프로토콜의 취약점을 이용한 공격. 헤더와 바디를 구분하기 위해 헤더 끝에 /r/n/r/n을 전송해야 하는데 /r/n만 보내 나머지 /r/n이 올 때까지 기다리게 하는 공격 |
| **SMTP** | TCP/IP 응용 계층의 프로토콜 중 전자 우편 전송을 위한 프로토콜 |
| **SNY flooding** | 공격자가 가상의 클라이언트로 위장하여 의도적으로 3-way-handshake를 중단하여 서버가 대기 상태에 놓여 정상적인 서비스를 하지 못하게 하는 서비스 공격 유형 |
| **SRT** | 선점 스케줄링 기법 중 하나로 비선점 스케줄링 기법인 SJF를 선점방식으로 변형한 것 |
| **SSL** | 데이터를 송,수신하는 두 컴퓨터 사이에 위치하여 인증, 암호화, 무결성을 보장하는 업계 표준 프로토콜. 웹 브라우저와 서버 간의 통신에서 정보를 암호화하여 도중에 정보가 유출되더라도 정보의 내용을 보호할 수 있는 보안 솔루션 |
| **SSO (Single Sing On)** | 한 번의 시스템인증을 통해 여러 정보시스템에서 재인증절차 없이 접근할 수 있는 통합 로그인 기술 |
| **TCP** | 전송계층의 통신 프로토콜. 근거리 통신망이나 인트라넷 인터넷에 연결된 컴퓨터에서 실행되는 프로그램 간에 일련의 옥텟을 안정적으로 순서대로 에러없이 교환하게 해주는 프로토콜 |
| **TRUNCATE** | 데이터가 하나도 없이 테이블 구조만 남은 최초의 테이블이 만들어진 상태로 돌아가도록 하는 명령어 |
| **UDP** | 전송계층의 통신 프로토콜. 비연결성이고 신뢰성이 없음. 순서화되지 않은 데이터그램 서비스를 제공. |
| **UI** | 사용자와 시스템 간 상호작용을 원활하게 하기 위한 인터페이스 |
| **URL** | 인터넷이 있는 자원을 나타내는 유일한 주소 |
| **UX** | 제품과 시스템, 서비스 등을 사용자가 직/간접적으로 경험하면서 느끼고 생각하는 총체적 경험 |
| **V-모델** | 소프트웨어 생명주기 단계 별 개발자 관점에서 명세서에 맞게 소프트웨어가 개발되었는지 검증, 사용자 입장에서 요구사항이 잘 반영됐는지 확인을 지원하기 위한 테스트 모델 |
| **VPN (Virtual Private Network)** | 공중망에서 터널링, 암호화기법 등을 사용해 **사설 통신망과 같은** 서비스를 제공하는 **가상의 네트워크**. 기업에서 내용을 바깥 사람에게 드러내지 않고 통신할 목적으로 쓰임 |
| **WAN** | 광대역 통신망. 멀리 떨어져 있는 기기들 간의 통신을 지원하는 네트워크 구조 |
| **WAS (웹 애플리케이션 서버)** | 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어 |
| **WITH GRANT OPTION** | 다른 사람에게 권한 나눠주기도 가능 |
| **XSS (크로스 사이트 스크립팅)** | 특정 게시물 또는 이메일 링크 클릭 시 악성 스크립트가 실행되는 공격 기법  (CSRF는 시스템을 공격. XSS는 사용자를 공격) |
| **xUnit** | 다양한 언어를 지원하는 단위테스트를 하기 위한 프레임워크 |
| **가상화** | 물리적인 리소스들을 사용자에게 하나로 보이게 하거나 하나의 물리적인 리소스를 여러 개 로 보이게 하는 기술 |
| **가용성** | 사용하고자 할 때 언제라도 사용할 수 있는 정도 |
| **강도 테스트** | 시스템에 과도한 정보량이나 빈도를 부과하여 과부하 시에도 소프트웨어가 정상적으로 실행되는지를 확인하는 테스트 |
| **개념모델링** | 요구사항 분석 기법 중 하나로 쉬운 이해를 위해 현실세계 상황을 단순화하여 개념적으로 표현하는 것 |
| **개체 무결성** | 기본키를 구성하는 어떤 속성도 NULL 값이나 중복 값을 가질 수 없다 |
| **결함 분포** | 결함 추이 분석 중 모듈 또는 컴포넌트에 포함되는 특정 속성에 해당하는 결함 수 |
| **결함 에이징** | 결함 추이 분석 중 특정한 결함 상태의 지속시간을 측정하여 분석 |
| **결함추세** | 결함 추이 분석 중 테스트 진행시간에 따른 결함 상태의 지속시간 |
| **경쟁상태** | 메모리가 공유되는 하나의 객체에 두개 이상의 스레드가 동시에 변경하려고 할 때 발생하는 성능저하 문제점 |
| **관계대수** | 관계형 데이터베이스에서 원하는 정보와 그 정보를 어떻게 유도하는가를 기술한 절차적 정형 언어 |
| **관계해석** | 관계데이터의 연산을 표현하는 방법 비절차적이며 질의어로 표현함 |
| **교착상태** | 둘 이상의 프로세스가 서로가 가진 한정된 자원을 요청하는 경우, 프로세스가 진전되지 못하고 모든 프로세스가 대기 상태로 돌아가는 현상 |
| **데이터 마이닝** | 대규모로 저장된 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 찾아내는 기술 |
| **데이터 모델** | 현실세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형 |
| **데이터 웨어하우스** | 의사결정용 데이터베이스 급증하는 다량의 데이터를 분석하여 정보화하고 이를 여러 계층의 사람들이 효율적으로 사용할 수 있도록 한 데이터베이스 |
| **델파이 기법** | 전문가 감정 기법을 보완한 방법으로 다수의 전문가와 한 명의 중재자를 통해 비용을 산정하는 하향식 비용 산정 기법 |
| **동치 분할 테스트** | 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 방법 프로그램의 입력 조건에 타당한 입력자료와 타당하지 않은 입력자료의 개수를 균등하게 하여 테스트 케이스를 정하고, 해당 입력 자료에 맞는 결과가 출력되는지 확인하는 테스트 기법 |
| **디스패치 (Dispatch)** | 준비상태에 있는 여러 프로세스를 선정하여 CPU에 할당하는 것. 문맥교환이 발생하고 프로세스를 준비상태에서 실행상태로 전이하는 프로세스 상태 전이 |
| **로킹(Locking)기법** | 병형제어 기법 중 하나. 하나의 트랜잭션이 데이터를 액세스하는 동안, **다른 트랜잭션이 그 데이터 항목을 엑세스할 수 없도록 제어**하는 기법 |
| **메소드** | 프로그램의 구성요소인 클래스의 기능으로 작업의 최소단위. 반드시 호출되어야만 기능을 사용할 수 있는 요소 |
| **명령형 언어** | 문제를 해결하기 위한 방법을 기술하는 언어. 알고리즘을 명시함 |
| **모듈화** | 모듈을 사용해 성능을 향상시키고 복잡성을 줄이고 시스템 디버깅 시험 수정이 용이하도록 설계하는 기법 |
| **문맥교환** | CPU가 현재 실행하고 있는 프로세스의 문맥 상태를 프로세스 제어블록(PCB)에 저장하고 다음 프로세스의 PCB로부터 문맥을 복원하는 작업 |
| **미들웨어** | 분산 컴퓨팅 환경에서 응용 프로그램과 프로그램이 운영되는 환경 간에 원만한 통신이 이루어질 수 있도록 제어해주는 소프트웨어 |
| **반정규화** | 시스템의 성능 강화. 개선을 위해 정규화한 데이터들을 통합. 분리하는 모델링 기법 |
| **백도어** | 시스템 설계자가 기술자의 액세스 편의를 위해 시스템 보안을 제거하여 만들어 놓은 비밀통로 |
| **백본망** | 네트워크를 상호 연결하는 컴퓨터 네트워크의 일부. 각기 다른 LAN선이나 부분 망 간에 정보를 교환하도록 제공하는 망 |
| **보안 컨테이너** | DRM의 구성요소 중 안전 유통을 위한 보안장치 |
| **브룩스의 법칙** | 소프트웨어 개발 일정이 지연된다고 해서 새로운 개발 인력을 진행 중인 프로젝트에 투입할 경우 작업 적응 기간과 부작용으로 인해 일정이 더욱 지연된다는 법칙 |
| **빅데이터** | 시스템, 서비스, 조직 등에서 주어진 비용, 시간 내에 처리 가능한 데이터의 범위를 넘어서는 방대한 양의 비정형 데이터 |
| **사용성 테스트** | 사용자가 제품을 사용하면서 시나리오에 맞춰 과제를 수행한 후 질문에 답하도록 하는 테스트 |
| **사이트 맵** | 화면의 정보를 한 눈에 파악하기 위한 시각적인 콘텐츠 모형 |
| **살충제 패러독스** | 동일한 테스트 케이스로 반복적인 테스트를 감행할 경우 더 이상 오류를 찾을 수 없는 소프트웨어 테스트 원리 |
| **서브넷 마스크** | IP주소 중 네트워크 주소와 호스트 주소를 구분하기 위한 비트 |
| **선언형 언어** | 무엇을 수행할 것인가 문제를 기술하는 언어. 목표를 명시함 |
| **선점형 스케쥴링** | 현재 프로세스가 CPU를 점유하고 있는데 다른 프로세스가 더 우선시 된다면 현 프로세스가 중단되고 새로이 점유되는 스케쥴링 |
| **세그먼테이션 기법** | 가상 기억장치에 보관되어 있는 프로그램을 다양한 크기의 단위로 나눈 후 주기억장치에 적재시켜 실행하는 기법 내부 단편화는 발생하지 않으나 외부 단편화는 발생할 수 있음 |
| **스니핑** | 네트워크 상 떠돌아다니는 패킷을 몰래 들여다 보는 수동적 공격기법 |
| **스래싱(Thrashing)** | 가상 기억 장치를 구현 시 시스템 성능에 영향을 미치는 요소 중 하나 프로세스 수행에 소요되는 시간보다 페이지 교체에 소요되는 시간이 더 커지는 현상 |
| **스레드(Thread)** | 프로세스 내 작업단위 |
| **스머핑(SMURFING)** | IP나 ICMP 특성을 이용해 한 사이트에 대량의 데이터를 집중 전송하여 네트워크 또는 시스템의 상태를 불능하게 만드는 공격 기법 출발지 주소를 공격 대상의 IP로 설정하여 네트워크 전체에게 ICMP echo 패킷을 직접 브로드캐스팅하여 마비시키는 공격 |
| **스토리보드** | UI 화면 설계를 위해서 구축하는 서비스를 위한 모든 정보가 담겨져 있는 문서. 설계 산출물 |
| **스푸핑** | 공격 대상 컴퓨터와 서버(스위치) 사이의 MAC 테이블을 속여 권한을 획득하고 정보를 가로채는 공격기법 |
| **스피어 피싱** | 특정 대상을 선정한 후 메일의 링크나 파일의 클릭을 유도한 뒤 개인정보를 탈취하는 공격기법 |
| **시퀀스 다이어그램** | 상호 작용하는 시스템이나 객체들이 주고받은 메세지를 표현하는 다이어그램 |
| **시큐어 코딩** | SW를 개발하는 과정에서 보안상의 취약점을 제거하고자 하는 코딩기법 |
| **신뢰도** | 시스템이 주어진 문제를 정확하게 해결하는 정도 |
| **에이징** | 프로세스가 자원을 기다리고 있는 시간에 비례하여 우선순위를 부여함으로써 무기한 대기하는 문제를 방지하는 기법 |
| **오류-부재의 궤변** | 테스트의 원칙 중 하나로, 모든 결함을 제거하여도 사용자의 요구사항을 만족하지 못한다면 품질이 높다고 말할 수 없음 |
| **온톨로지 (Ontology)** | 실세계의 존재하는 모든 개념들과 개념들의 속성, 그리고 개념들 간의 관계 정보를 컴퓨터가 이해하기 쉽게 서술해 놓은 개념화 명세서 |
| **옵티마이저** | SQL을 가장 빠르고 효율적으로 수행할 최적의 처리 경호를 생성해주는 DBMS 핵심엔진. 모듈 |
| **와이어프레임** | 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하기 위해 제작하는 UI 설계 도구 이해관계자들과의 화면구성을 협의하거나 서비스의 간략한 흐름을 공유하기 위해 화면 단위의 레이아웃을 설계하는 작업 |
| **요구공학** | 시스템의 요구사항을 개발(도출 분석 명세 검증 관리)하기 위해 수행되는 구조화된 활동의 집합 소프트웨어를 설계하기에 앞서 무엇을 개발해야 하는지 요구사항을 정의하고 분석 및 관리하는 프로세스를 연구하는 것. 소프트웨어 프로젝트 실패 최소화를 목표로 함 |
| **운영체제 (OS)** | 사용자와 하드웨어 간의 인터페이스로서 동작하는 시스템 소프트웨어 |
| **워킹셋 (Working Set)** | 프로세스가 일정 시간 동안 자주 참조하는 페이지 집합을 메모리 공간에 계속 상주시켜 빈번한 페이지 교체 현상을 줄이는 기법 |
| **이상(Anomaly)** | 테이블에서 일부 속성들의 종속으로 인해 데이터의 중복이 발생하고, 이 중족으로 인해 테이블 조작 시 문제가 발생하는 현상 |
| **인스턴스** | 객체 지향 프로그래밍에서 해당 클래스의 구조로 컴퓨터 저장공간에 할당된 실체 |
| **인터프리터** | 프로그램 문장을 하나씩 변역하고 실행할 수 있도록 하는 프로그램 컴파일 과정이 없기 때문에 개발하는 과정에서 사용하면 유용하다는 장점이 있지만, 실행 속도가 느리고 메모리 사용이 효율적이라는 단점이 있음 |
| **정규화 (Normalization)** | 테이블의 속성들이 상호 종속적인 관계를 갖는 특성을 이용해 테이블을 무손실 분해하는 **데이터 모델링 기법**. 중복을 최소화하고 이상현상이 생기지 않도록 함 |
| **정황 시나리오** | 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사하는 것이며 이를 토대로 UI 요구사항을 작성한다 |
| **제로데이** | 보안 취약점이 발견된 후 널리 공표되기 전에 취약점을 공략해 공격하는 기술 |
| **제어의 역흐름** | 프레임워크의 특성(재확모역) 중 하나. 개발자가 관리하고 통제해야 하는 객체들의 제어 권환을 프레임워크에 넘겨 생산성을 향상시킴 |
| **참조 무결성의 CASCADE 법칙** | 연관 데이터에서 오류나는 일을 방지. 참조하는 테이블이 삭제될 경우 참조받는 테이블도 연쇄로 함께 삭제된다. |
| **컴포넌트** | 특정한 기능을 수행하기 위해 독립적으로 개발되어 보급되는 잘 정의된 인터페이스룰 가지며 다른 부품과 조립되어 응용시스템을 구축하기 위해 사용하는 소프트웨어 프로그램 |
| **코드 난독화** | 역공학을 통한 공격을 막기 위해 프로그램 소스 코드를 알아보기 힘든 형태로 바꾸는 기술 |
| **클러스터** | 데이터 저장시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장방법 |
| **클리어링 하우스** | DRM(디지털 저작권 관리)의 구성요소 중 하나로 저작권에 대한 사용 권한과 라이선스를 발급 |
| **키로거 공격** | 컴퓨터 사용자의 키보드 움직임을 탐지하여 ID, PW등의 개인정보를 빼내는 공격 |
| **텍스트 마이닝** | 대규모의 텍스트 데이터를 가지고 체계적인 규칙 패턴분석을 통해 원하는 정보를 찾아내는 기술 |
| **트랜잭션** | DB시스템에서 **하나의 논리적인 기능을 수행**하기 위한 작업의 **기본 단위**이자 **한꺼번에 모두 수행**되어야 할 **일련의 연산** |
| **트로이 목마** | 정상 프로그램으로 위장하고 있다가 해당 프로그램이 실행되었을때 활성화 되는 공격 (자기 복제 능력은 없다) |
| **트리거** | DB시스템에서 이벤트가 발생할 때 마다 관련 작업이 자동으로 수행되는 절차형 SQL |
| **파레토 법칙** | 테스트의 원리 중 결함집중과 관련된 단어로 20%의 모듈에서 80%의 결함이 발견된다는 의미로 사용 |
| **파티션** | 데이터베이스에서 대용량의 테이블이나 인덱스를 작은 논리적 단위인 ( )으로 나누는 것 |
| **패키저 (Packager)** | DRM 구성요소 중 콘텐츠를 메타 데이터와 함께 배포 가능한 단위로 묶는 기능 (+암호화) |
| **페이징 기법** | 가상 기억장치에 보관되어 있는 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나눠진 페이지를 동일하게 나눠진 주기억장치의 영역에 적재시켜 실행하는 기법 외부 단편화는 발생하지 않으나 내부 단편화는 발생할 수 있음 |
| **포렌식** | 범죄 사실을 사법기관에 제출하기 위해 디지털 증거자료를 수집/복사/분석/제출하는 일련의 과정을 지칭하는 용어 |
| **프로시저** | 복잡한 트랜잭션을 처리하기 위해 SQL을 이용해 생성된 데이터를 조작하는 프로그램. 일정 조건을 달성하면 자동으로 수행함 |
| **프로토콜** | 시스템끼리 원활하게 정보를 주고받기 위해 표준화된 통신규약 |
| **함수적 종속** | 어떤 테이블 R에서 X와 Y를 각각 R의 속성 집합의 부분 집합이라고 할 때 속성 X의 값 각각에 대해 시간에 관계없이 항상 속성 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 ( )이라고 하고, X->Y로 표기한다 |
| **해시(HASH) 암호화** | 임의의 길이 입력데이터나 메세지를 고정된 길이의 값이나 키로 변환하는 단방향 암호 알고리즘 |
| **형상 관리 (SCM)** | 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동 |
| **회선 교환 방식** | 라우팅 방식 중에서 교환기를 이용하여 물리적으로 접속하는 방식으로 기존의 음성 전화망이 이에 해당한다. 공간 분할 교환 방식과 시분할 교환 방식이 있다 |
| **NoSQL (Not only SQL)** | 데이터 저장에서 고정된 테이블 스키마가 필요하지 않고 조인 연산을 사용할 수 없으며, 수평적으로 확장이 가능한 DBMS |
| **TearDrop Attack** | **IP 패킷의 재조합 과정**에서 **잘못된 Fragment Offset** 정보로 인해 수신시스템이 문제를 발생하도록 만드는 DoS 공격 |
| **동등 분할 테스트** | 입력 데이터의 영역을 유사한 도메인별로 유효 값/무효 값을 그룹핑하여 대표 값을 테스트 케이스로 도출하여 테스트 하는 기법 |
| **시멘틱 웹** | 인터넷과 같은 분산환경에서 리소스에 대한 정보와 자원 사이의 관계-의미 정보를 컴퓨터가 처리할 수 있는 **온톨로지 형태로 표현**하고, 이를 자동화된 컴퓨터가 처리하도록 하는 **지능형 웹** |
| **테일러링** | **프로젝트의 특성과 필요에 따라** 소프트웨어 개발 프로세스, 기법, 산출물 등을 비즈니스적으로 또는 기술적인 **요구에 맞도록 최적화**하는 과정 및 방법론 |

|  |  |
| --- | --- |
| 요구사항 개발 프로세스 (4단계) | (타당성조사) - 도출 – 분석 – 명세 - 확인 |
| 논리 데이터 모델링의 특성 (4가지) | [**정포완독**] 정규화 - 포용성 - 완전성 - 독립성 |
| 이상의 종류 (3가지) | 삽입 이상 / 삭제 이상 / 갱신 이상 |
| 분산 데이터베이스 저장 방식의 목표 (4가지) | 병행 투명성 / 장애 투명성 / 중복 투명성 / 위치 투명성 |
| 검증 기준 (3가지) | 기능 기반 커버리지 / 라인 커버리지 / 코드 커버리지 |
| 와이어프레임 도구 (4종류) | 파워포인트, 키노트, 스케치, 일러스트 |
| 인터페이스 설계 지침 (9가지) | 사용자 중심 / 가시성 / 접근성 / 일관성 / 단순성 / 결과 예측 가능 / 표준화 / 명확성 / 오류 발생 해결 |
| 결함 추적 순서 (6단계) | 결함 등록 - 검토 - 할당 - 수정 - 종료 - 해제 |
| 릴리즈 노트 작성 프로세스 (6단계) | [**모정개영정추**] 모듈 식별 - 릴리즈 정보 확인 - 릴리즈 노트 개요 작성  - 영향도 체크 - 정식 릴리즈 노트 작성 - 추가 개선 항목 식별 |
| 애플리케이션 성능 측정 지표 (4가지) | [**처응경자**] 처리량 / 응답시간 / 경과시간 / 자원사용률 |
| 운영체제의 성능 측정 지표 (4가지) | [**처반사신**] 처리능력 / 반환시간 / 사용가능도 / 신뢰도" |
| 공통모듈 구현 시 준수 할 명세기법 (5가지) | [**정명완일추**] 정확성 / 명확성 / 완전성 / 일관성 / 추적성 |
| 제품 소프트웨어 패키징 작업 순서 (7단계) | [**기모빌환시변배**] 기능식별 - 모듈화 - 빌드 진행 - 사용자 환경 분석  - 패키징 및 적용 시험 - 패키징 변경 개선 – 배포 |
| TCP/IP 4계층 | 네트워크 액세스 계층(물데) - 인터넷 계층(네) - 전송 계층(전) - 응용 계층(세표응) |
| 데이터베이스 내 데이터 (4종류) | 운영 데이터 / 통합된 데이터 / 저장된 데이터 / 공용 데이터 |
| 결함 관리 프로세스 (6단계) | [**발등분확할조**] 에러 발견 - 에러 등록 - 에러 분석  - 결함 확정 - 결함 할당 - 결함 조치 - (결함 조치 검토 및 승인) |
| 보안 점검 항목 (8가지) | [**세입보시에코캡에**] 세션통제 / 입력 데이터 검증 및 표현 / 보안 기능  / 시간 및 상태 / 에러처리 / 코드 오류 / 캡슐화 / API오용 |
| Windows의 특징 | 선점형 멀티태스킹 / Single-User / OLE / PnP |
| 서버 프로그램 구현 단계 (5단계) | [**디스다써클**] DTO/VO - SQL - DAO - Service - Controller 구현 |
| 배치 스케줄러 종류 (3종류) | 스프링배치 / Quartz / Cron |

소프트웨어 아키텍처

|  |  |
| --- | --- |
| **소프트웨어 아키텍처** | 소프트웨어 구성요소의 특성. 관계를 표현하는 시스템 구조 |
| **소프트웨어 아키텍처 프레임워크** | 소프트웨어 집약적인 시스템에서 아키텍처가 표현해야 하는 내용 및 이들 간의 관계를 제공하는 아키텍처 기술 표준 |
| 소프트웨어 아키텍처 프레임워크의 구성요소 (6가지) | 명세서 / 이해관계자 / 뷰 / 근거 / 관점 / 관심사 |
| **소프트웨어 아키텍처 4+1뷰** | 고객의 요구사항을 정리해 놓은 시나리오를 4개 관점으로 접근하는 방법 [**논구배프유**] |
| 논리 뷰 | 설계 모델의 추상화. 기능적 요구사항 지원 |
| 프로세스 뷰 | 런타임 시 태스크.스레드.프로세스의 관계 표현. 비기능적 요구사항 고려 |
| 구현 뷰 | 소프트웨어 구현.관리 측면 표현. 개발자 관점 |
| 배포 뷰 | 물리적인 구성. 상호 연결 관계 표현 |
| 유스케이스 뷰 | 아키텍처를 도출.설계하는 작업 주도. 다른 뷰 검증에 이용 |

JAVA 웅앵웅

|  |  |
| --- | --- |
| **Ajax** | 자바스크립트를 사용한 **비동기 통신기술**. 클라이언트와 서버 간에 XML 데이터를 주고받는 기술 |
| **JSON** | 속성-값 쌍으로 이루어진 데이터객체를 전달하기위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷. 비동기처리를 하는 ajax에서는 XML을 대신하여 사용됨 |
| **JDBC** | DBMS 접속 기술 중 하나. 직접 연계 방식 중 하나. **JAVA에서 데이터베이스에 접속할 수 있도록 하는 자바 API**. 수신 시스템의 프로그램에서 드라이버를 이용하여 송신 시스템 DB와 연결하는 연계 기술 |
| **MyBatis** | DBMS 접속 기술 중 하나. **JDBC 코드를 단순화한 SQL Mapping 기반 오픈 소스 접속 프레임워크**. SQL을 거의 그대로 사용 가능하여 국내 환경에 적합 |
| **Spring** | JAVA 플랫폼을 위한 오픈소스 애플리케이션 프레임워크. **전자정부 프레임워크** 기반 기술로 사용되는 서버 개발 프레임워크 |

DBMS

|  |  |
| --- | --- |
| **DBMS** | 데이터베이스를 관리해주는 소프트웨어 |
| DBMS의 종류 | Oracle / IBM DB2 / MySQL / Microsoft SQL Server / SQ Lite |
| DBMS 접속기술 (API 또는 프레임워크) (3가지) | JDBC. ODBC. MyBatis |
| **JDBC** | 직접 연계 방식 중 하나. JAVA에서 데이터베이스에 접속할 수 있도록 하는 자바 API. 수신 시스템의 프로그램에서 드라이버를 이용하여 송신 시스템 DB와 연결하는 연계 기술 |
| **ODBC** | 데이터베이스에 접근하기 위한 표준 개방형 API |
| **MyBatis** | JDBC 코드를 단순화한 SQL Mapping 기반 오픈 소스 접속 프레임워크 SQL을 거의 그대로 사용 가능하여 국내 환경에 적합 |
| DBMS 필수 기능 (3가지) | 정의기능 / 조작기능 / 제어기능 |

테스트

|  |  |
| --- | --- |
| **테스트 케이스** | **사용자의 요구사항을 준수**하는지 확인하는 테스트 항목에 대한 **명세서**  입력값, 실행 조건, 예상된 결과의 집합 |
| **테스트 하네스** | 테스트를 지원하기 위한 코드, 데이터 등 환경의 일부분 |
| 테스트 하네스의 구성요소 | [**스드슈스케목**] 테스트 스텁 / 드라이버 / 슈트 / 스크립트 / 케이스 / 목 오브젝트 |
| 테스트 스텁 | 상향식 통합 테스트에서 사용. 타 모듈의 기능을 단순히 수행하는 도구 |
| 테스트 드라이버 | 하향식 통합 테스트에서 사용. 하위 모듈을 호출하는 상위 더미 모듈 |
| **테스트 슈트** | **단순히** 테스트 케이스를 묶은 집합. |
| **테스트 시나리오** | 테스트 케이스를 적용하는 **순서에 따라** 여러 개의 테스트 케이스를 묶은 집합 |
| **테스트 오라클** | 테스트 케이스에 테스트의 예상 결과(참 값)를 대입해 비교하는 기법 |
| 테스트 오라클의 종류 | 참 오라클 / 샘플링 오라클 / 추정 오라클 / 일관성 검사 오라클 |
| 목 오브젝트 | 사전에 조건부를 입력해두고 조건이 만족할 시에 입력한 조건부가 실행 |
| 테스트의 분류 | 프로그램 실행 여부 : **정적 테스트**(X) vs **동적 테스트**(O)  테스트의 기반 : 명세 기반 테스트 vs 구조 기반 테스트 vs 경험 기반 테스트  시점 : **검증 테스트**(개발자) vs **확인 테스트**(사용자)  목적 : 회복 / 안전 / 성능 / 구조 / 회귀 / 병행 / 강도 테스트 |
| **정적 테스트** 종류 (3가지) | 동료검토 / 워크스루 / 인스펙션 |
| **동료검토** | **2~3명**이서 진행하는 리뷰. **요구사항 명세서 작성자**가 요구사항 명세서를 설명하고, **이해관계자**들이 설명을 들으면서 코드에 대한 결함을 발견하는 기법 |
| **워크스루** | 검토 자료를 **회의 전에 배포**해서 사전에 검토를 진행한 후 **짧은 시간 회의**를 통해 코드의 오류를 검출하고 문서화하는 기법 |
| **인스펙션** | 소프트웨어 요구, 설계, **원시 코드** 등의 **저작자 외의 다른 전문가** 또는 팀이 검사하여 오류를 찾아내는 **공식적 검토 기법** |
| **동적 테스트** 종류 (2가지) | 화이트 박스 테스트 / 블랙 박스 테스트 |
| **화이트박스 테스트** | 모듈의 원시 코드를 오픈 시킨 상태에서 원시 코드의 논리적인 모든 경로를 확인하는 테스트 |
| **블랙박스 테스트** | 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완벽히 작동되는 것을 입증하는 테스트 |
| 테스트 커버리지 | 주어진 테스트 케이스에 의해 수행되는 소프트웨어의 테스트 범위를 측정하는 테스트 품질 측정 기준. 테스트의 정확성과 신뢰성을 향상시키는 역할 |
| 테스트 커버리지 유형(3가지) | [**기라코**] 기능 기반 커버리지 / 라인 커버리지 / 코드 커버리지 |
| **코드 커버리지** | 소스 코드의 구문, 조건, 결정 등의 구조 코드 자체가 얼마나 테스트되었는지를 측정하는 방법 |
| 코드 커버리지 유형 (6가지) | 구문 커버리지 / 결정 커버리지 / 조건 커버리지 외 3가지. (구조 테스트 유형과 동일) |
| **구문 커버리지** | 프로그램 내의 모든 명령문을 적어도 한 번 수행하는 커버리지 |
| **결정 커버리지** | 프로그램 내의 전체 결정문이 적어도 한 번은 참과 거짓을 결과를 수행하는 커버리지 |
| **조건 커버리지** | 결정 명령문 내의 각 조건이 적어도 한 번은 참과 거짓을 결과가 되도록 수행하는 커버리지 |

프레임워크 / 라이브러리 / API

|  |  |
| --- | --- |
| **프레임워크** | 소프트웨어의 구체적 설계 구현을 클래스로 제공해 도움을 줌 (어떻게 개발할지 결정 틀) |
| 프레임워크의 특성 (4가지) | [**재확모역**] 재사용 / 확장성 / 모듈화 / 제어의 역흐름 |
| **라이브러리** | 미리 함수들을 만들어 두고 편리하게 사용하게 제공함 (필요한 것을 빌려 쓰는 대상) |
| **API** | 프로그램 또는 애플리케이션이 처리를 위해 호출하는 함수의 집합 (라이브러리 + 명세)  데이터를 송신 시스템의 DB에서 읽어와 제공하는 애플리케이션 프로그래밍 인터페이스 프로그램 |

웹 서비스 구성요소

|  |  |
| --- | --- |
| **SOAP** | HTTP, HTTPS, SMTP 등을 활용해 XML 기반의 메세지를 네트워크 상에서 교환하는 프로토콜 |
| **WSDL** | 웹 서비스 명, 서비스 제공 위치, 프로토콜 등 웹 서비스에 대한 상세 정보를 XML 형식으로 구현 |
| **UDDI** | WSDL을 등록하여 서비스와 서비스 제공자를 검색하고 접근하는데 사용하는 레지스트리 |

\_\_ML

|  |  |
| --- | --- |
| **HTML** | 하이퍼텍스트 문서를 작성하기 위해 사용하는 마크업 언어 |
| **UML** | 객체지향 모델링 언어 |
| **XML** | 다른 특수한 목적의 마크업 언어를 만들기 위해서 사용하는 다목적 마크업 언어 |
| **XrML** | 디지털 콘텐츠. 웹서비스 권리 조건을 표현한 XML 기반의 마크업 언어 |

\_aaS

|  |  |
| --- | --- |
| **IaaS (인프라형 서비스)** | 서버, 스토리지 같은 시스템 자원을 클라우드로 제공하는 서비스 |
| **PaaS (플랫폼형 서비스)** | 인프라를 생성, 관리하는 복잡함 없이 애플리케이션을 개발, 실행, 관리 할 수 있게 하는 플랫폼을 제공하는 서비스 |
| **SaaS (소프트웨어형 서비스)** | 소프트웨어 및 관련 데이터는 중앙에 호스팅돠고 사용자는 웹 브라우저 등의 클라이언트를 통해 접속하여 소프트웨어 서비스 형태로 이용하는 서비스 |

UNIX

|  |  |
| --- | --- |
| 유닉스 구성요소 (5단계) | 하드웨어 - 커널 - 쉘 - 유틸리티 - 사용자 |
| **커널(Kernel)** | 유닉스의 구성요소 중 가장 핵심이며 프로그램-하드웨어 사이 인터페이스 역할을 담당하는 하드웨어 보호기 |
| **쉘(Shell)** | 유닉스의 구성요소 중 시스템-사용자 간 인터페이스를 담당하는 사용자의 명령어 해석기 |

무결성

|  |  |
| --- | --- |
| **개체 무결성** | 기본키를 구성하는 어떤 속성도 NULL값이나 중복된 값을 가질 수 없다 |
| **도메인 무결성** | 주어진 속성값이 정의된 도메인에 속한 값이여야 한다 |
| **참조 무결성** | 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다 |

옵티마이저

|  |  |
| --- | --- |
| **옵티마이저** | SQL을 가장 빠르고 효율적으로 수행할 최적의 처리 경호를 생성해주는 DBMS 핵심엔진. 모듈 |
| **실행 계획** | 옵티마이저가 SQL을 가장 빠르고 효율적으로 수행할 수 있도록 생성한 최적의 처리 경로 |
| **힌트(Hint)** | 옵티마이저가 비정상적으로 수립한 계획에 엑세스 경로 및 조인순서를 제어할 수 있도록 하기 위한 기법. 항상 최적으로 실행계획을 수립하기 위해 명시적인 이것을 사용. |
| 옵티마이저 종류 (2가지) | RBO(Rule-Based Optimization) / CBO(Cost-Based Optimiation) |
| **RBO** | 통계정보가 없는 상태에서 사전에 등록된 규칙에 따라 질의 실행 계획을 선택하는 옵티마이저 |
| **CBO** | 통계정보로부터 모든 접근 경로를 고려한 질의 실행 계획을 선택하는 옵티마이저 |

네트워크 연결 방식

|  |  |
| --- | --- |
| **DAS (Direct Area Network)** | 서버와 저장장치를 전용케이블로 직접 연결하는 방식 |
| **NAS (Network Attached Storage)** | 서버와 저장장치를 네트워크로 연결하는 방식 |
| **SAN (Storage Attached Network)** | 서버와 스토리지를 광케이블 및 광채널스위치를 통해 근거리 네트워크 환경을 구성. 빠른 속도로 데이터를 처리할 수 있는 특수 목적용 고속 네트워크 |

DTO / VO / DAO

|  |  |
| --- | --- |
| **DTO** | 프로세스 사이에서 데이터를 전송하는 객체. 데이터 저장 회수 외 다른 기능 없음 |
| **VO** | 간단한 엔티티를 의미하는 작은 객체가 클래스인 DTO와 달리 고정 클래스를 가짐 |
| **DAO** | 특정 타입의 DB에 추상 인터페이스를 제공하는 객체. 세부내용 노출없이 데이터를 조작 |