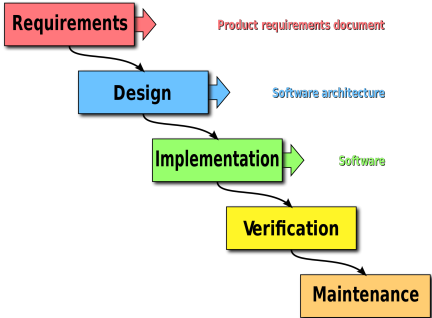


질문	답변
폭포수 방법론이란 뭔가요?	<p>폭포수 모델(waterfall model)은 순차적인 소프트웨어 개발 프로세스로, 개발 흐름이 마치 폭포수처럼 지속적으로 아래로 향하는 것처럼 보이는 데서 이름이 붙여졌습니다.</p> <p>이 폭포수 모델의 흐름은</p> <ul style="list-style-type: none"> - 소프트웨어 요구사항 분석 - 소프트웨어 설계 - 소프트웨어 구현 - 소프트웨어 시험 - 소프트웨어 통합 - 소프트웨어 유지보수 <p>단계에 이릅니다.</p> <p>폭포수 모델을 따르기 위해서는, 완전히 순차적으로 한 단계, 한 단계를 진행해 나가야 합니다.</p> <p>예를 들어, 가장 먼저 요구사항 기술을 진행하여 이를 확정하여야 하며, 그런 이후에 설계를 진행할 수 있습니다.</p> <p>소프트웨어가 설계된 후, 그 설계도가 구현자(개발자)에게 따라서 구현해야할 계획으로 전달된다.</p> <p>따라서 설계가 완전히 완료된 후에 설계에 대한 구현이 코더에 의해 진행될 수 있는 것이다.</p> <p>이 구현의 마지막 단계에 이르면, 각각의 생성된 컴포넌트를 결합하여, 새로운 기능을 실현시키고 그때까지 발생한 버그를 해결하게 된다.</p> <p>폭포수 모델은 전 단계가 수행되어 완료되기 전에는 다음 단계를 진행할 수 없도록 제한하는 것이 특징이다.</p>
	 <pre> graph TD R[Requirements] --> D[Design] D --> I[Implementation] I --> V[Verification] V --> M[Maintenance] R --- PRD[Product requirements document] D --- SA[Software architecture] I --- S[Software] V --- V[Verification] </pre>
애자일 방법론이란 뭔가요?	<p>애자일은 신속한 반복 작업을 통해 실제 작동 가능한 소프트웨어를 개발하여 지속적으로 제공하기 위한 소프트웨어 개발 방식입니다.</p> <p>작동하는 소프트웨어의 작은 구성 요소를 신속하게 제공하여 고객의 만족도를 개선하는 것이 애자일 방법론의 핵심입니다.</p> <p>애자일 방법론은 다음과 같은 부분에 가치를 둡니다.</p> <ul style="list-style-type: none"> - 개인과 개인 간의 상호작용이 프로세스 및 툴보다 우선 - 작동하는 소프트웨어가 포괄적인 문서보다 우선 - 고객과의 협업이 계약 협상보다 우선 - 변화에 대응하는 것이 계획을 따르는 것보다 우선

CI/CD란 뭔가요?	<p>CI/CD는 애플리케이션 개발 단계를 자동화하여 애플리케이션을 보다 짧은 주기로 고객에게 제공하는 방법입니다.</p> <p>CI/CD의 기본 개념은 지속적인 통합, 지속적인 서비스 제공, 지속적인 배포입니다. CI/CD는 새로운 코드 통합으로 인해 개발 및 운영팀에 발생하는 문제(일명 "인테그레이션 헬(integration hell)")을 해결하기 위한 솔루션입니다.</p> <p>CI는 개발자를 위한 자동화 프로세스인 지속적인 통합 Continuous Integration을 의미합니다. CI를 성공적으로 구현할 경우 애플리케이션에 대한 새로운 코드 변경 사항이 정기적으로 빌드 및 테스트되어 공유 리포지토리에 통합되므로 여러 명의 개발자가 동시에 애플리케이션 개발과 관련된 코드 작업을 할 경우 서로 충돌할 수 있는 문제를 해결할 수 있습니다.</p> <p>CD는 지속적인 서비스 제공 Continuous Delivery 및 지속적인 배포 Continuous Deployment를 의미합니다. 지속적인 제공이란 개발자들이 애플리케이션에 적용한 변경 사항이 버그 테스트를 거쳐 리포지토리에 자동으로 업로드되는 것을 뜻합니다. 지속적인 배포란 개발자의 변경 사항을 리포지토리에서 고객이 사용 가능한 프로덕션 환경까지 자동으로 릴리스하는 것을 의미합니다</p>
CDN이 무엇인가요?	<p>CDN(콘텐츠 전송 네트워크)은 지리적으로 분산된 여러 개의 서버입니다. 웹 콘텐츠를 사용자와 가까운 곳에서 전송함으로써 전송 속도를 높입니다. 전 세계 데이터센터는 파일 복사본을 임시로 저장하는 프로세스인 캐싱을 사용합니다. 따라서 사용자는 가까운 서버를 통해 웹 활성화 디바이스 또는 브라우저에서 인터넷 콘텐츠에 빠르게 접속할 수 있습니다. CDN은 웹 페이지, 이미지, 비디오 등의 콘텐츠를 사용자의 물리적 위치와 가까운 프록시 서버에 캐싱합니다. 이렇게 하면 콘텐츠가 로딩될 때까지 기다릴 필요 없이 영화 감상, 소프트웨어 다운로드, 은행 잔고 확인, 소셜 미디어 포스팅, 구매 등의 작업을 할 수 있습니다.</p> <p>인터넷으로 그래픽과 비디오 등 용량이 큰 웹 콘텐츠를 전송하면 트래픽이 폭주할 때처럼 네트워크 혼잡 문제가 발생할 수 있습니다. CDN 서비스는 이러한 문제를 해결하기 위해 개발되었습니다. 중앙 서버에서 개별 사용자에게 콘텐츠를 전송하려면 시간이 너무 오래 걸렸습니다. 이제 CDN은 텍스트, 그래픽, 스크립트, 미디어 파일부터 소프트웨어 다운로드, 문서, 포털, 이커머스, 라이브 스트리밍 미디어, 온디맨드 비디오 스트리밍 미디어, 소셜미디어 사이트에 이르기까지 모든 것을 처리합니다.</p> <p>CDN의 미션은 지연 시간을 줄이는 것입니다. 지연 시간은 웹 페이지 또는 비디오 스트리밍 콘텐츠가 디바이스에 완전히 로딩되기 전에 발생하는 불편한 지연을 의미합니다. 지연 시간은 밀리초 단위로 측정됩니다. 하지만 사용자가 체감하는 시간은 매우 길며, 시간 초과 또는 로딩 오류가 발생할 수 있습니다. 콘텐츠가 사용자에게 도달하기 위해 이동해야 하는 물리적 거리를 줄여 지연 시간을 줄이는 콘텐츠 전송 네트워크도 있습니다. 따라서 CDN이 보다 광범위하고 넓게 분산되어 있으면 사용자와 최대한 가까운 곳에 콘텐츠를 배치함으로써 웹 콘텐츠를 보다 빠르고 안정적으로 전송할 수 있습니다.</p> <p>어떤 사람들은 CDN을 ‘엣지’라고 부릅니다. 엣지는 물리적 환경과 디지털 세계가 네트워크 경계에서 만나 상호작용하는 곳입니다. 엣지는 전 세계에 분산된 수천 개의 PoP (Points of Presence)로 구성되며, 이러한 PoP는 CDN 제공 업체가 관리하며 콘텐츠 캐싱, 보관, 및 전송에 사용됩니다.</p>
CDN을 사용하는 대표적인 기업과 CDN 서비스를 제공하는 대표적인 기업에 대해 아는대로 말해주세요	<p>CDN 을 사용하는 대표적인 기업 - 넷플릭스, 슬랙</p> <p>넷플릭스: 넷플릭스는 온라인 동영상 스트리밍 서비스를 제공하는 회사로, 수많은 사용자에게 고화질의 비디오 콘텐츠를 제공하기 위해 CDN을 사용합니다. 이를 통해 지연 시간을 줄이고, 끊김 없는 스트리밍을 제공합니다. 슬랙: 슬랙은 업무 협업 및 메시징 플랫폼으로, 사용자 간의 실시간 메시징 및 협업을 지원합니다. 슬랙 또한 CDN을 사용하여 채팅 및 파일 공유와 같은 기능을 안정적으로 제공하며, 사용자 경험을 향상시킵니다.</p> <p>CDN 서비스를 제공하는 대표적인 기업 - Amazon CloudFront</p> <p>Amazon CloudFront: Amazon Web Services(AWS)의 일부로서, Amazon CloudFront는 CDN 서비스를 제공합니다. 이 서비스는 전 세계 여러 지역에 서버 및 콘텐츠 배포 지점을 설정하여 사용자에게 빠른 콘텐츠 전송을 제공합니다. Amazon CloudFront는 동영상 스트리밍, 웹 페이지, 애플리케이션 데이터 등 다양한 콘텐츠를 가속화하고 보안을 제공하는 데 사용됩니다.</p>
테스트는 무엇인가요?	<p>애플리케이션이 요구사항에 맞게 동작하는지를 검증하는 행위입니다</p>

테스트의 예는 어떤 것들이 있나요?	<p>DB에 데이터를 입력하는 API를 개발 > API 호출 > DB값 검증</p> <p>디자인 시안에 맞게 HTML/CSS를 작성 > 브라우저에서 실제 렌더링된 결과를 확인</p> <p>새로운 기능을 추가하기 위해 기존 모듈을 리팩토링 > 영향을 받는 다른 모듈의 실행 결과를 확인</p> <p>버그를 수정하기 위해 기존 함수를 수정 > 버그 수정 확인 & 영향을 받는 다른 모듈의 실행 결과를 확인</p> <p>개발 환경에서 테스트된 어플리케이션을 리얼 환경에 배포 > 배포 과정에서 발생한 문제가 없는지 확인</p>
테스트는 왜 해야 하나요?	<p>개발자는 작성한 프로그램의 퀄리티에 대한 책임이 있습니다.</p> <p>자동화된 테스트를 작성해두지 않으면, 어플리케이션이 복잡해질 수록 테스트 비용이 증가합니다.</p> <p>개발 기간이나 인력 등이 한정되어 있기 때문에, 테스트를 소홀히 하게 되는 경우가 있는데 이럴 경우 QA와의 커뮤니케이션 비용이 늘어나, 업무의 효율이 떨어질 수 있습니다.</p>
유닛 테스트란 무엇인가요?	<p>단위(Unit) 테스트</p> <p>모듈(함수/클래스) 단위의 테스트</p> <p>작성 비용이 적게 들고 실행 속도가 빠름</p> <p>실패했을 때 문제가 생긴 부분을 비교적 정확하게 파악할 수 있음</p>
통합 테스트란 무엇인가요?	<p>통합(Integration) 테스트</p> <p>주로 단위 테스트보다 큰 범위의 테스트를 의미</p> <p>개발 모듈(함수/클래스)들이 연결되어 제대로 상호작용하는지를 테스트</p> <p>단위 테스트에 비해 실패 시 문제가 생긴 부분을 정확히 파악하기가 어려움</p>
E2E 테스트란 무엇인가요?	<p>E2E(End to End) 테스트</p> <p>실제 사용자가 사용하는 것과 같은 조건에서 전체 시스템을 테스트</p> <p>단위/통합 테스트에 비해 작성이 어렵고 실행 속도가 비교적 느림</p> <p>API 서버, DB 등의 외부 서비스들을 모두 사용하여 통합된 시스템을 테스트</p>