

소프트웨어 개발생명주기

개발방법론을 이야기 하기 이전에 먼저 소프트웨어 생명주기의 개념을 알고 넘어가야 한다. 소프트웨어 생명주기와 개발방법론을 혼동하여 쓰는 경우가 종종있다.

소프트웨어 생명주기(SDLC : system Development Life Cycle)

-소프트웨어의 생성에서 소멸까지 변환되는 과정

-일반적으로 분석->설계->구현->테스트->유지보수의 단계를 거침

일단 소프트웨어 생명주기의 기본적인 개념을 간략히 요약하자면 위에 내용과 같으며, 위의 단계를 운영하는 방법에 따라 아래와 같은 유형으로 나눌 수 있습니다.

1. 폭포수 모델

-순차적으로 소프트웨어를 개발하는 전형적인 모델

-개발의 흐름이 마치 폭포수처럼 지속적으로 아래로 향한다고하여 이름이 붙여짐.

-초기 분석 단계에서 요구사항의 완벽한 정의가 어려워 요구사항 변경이 빈번하게 발생할 확률이 높고, 단계가 진행될 수록 요구사항 변경에 대한 비용이 늘어날 수 있음

2. 원형모델

-폭포수 모델의 단점을 보완하여 점진적으로 시스템을 개발해 나가는 모델

-소프트웨어 기본적인 기능을 알아보기 위해서 프로토타입을 만들고, 이후의 소프트웨어 완제품을 예측해보는 방법.

-프로토타입으로 완제품을 미리 예측할 수 있기 때문에 요구사항 도출이 용이함.

3. 나선형 모델

-폭포수 모델과 원형 모델의 장점에 위험분석을 추가한 점증적 개발 모델

-계획수립->위험분석->개발/구축->평가의 단계를 거침.

-점진적으로 프로토타입을 발전시켜 나가면서 완전한 시스템으로 개발하는 방식이기 때문에 고비용 대규모 시스템 개발에 적합함.

4. 그밖에 증분개발모델, 진화적 개발 모델등이있음.

트렌드되고있는 개발방법론

MSA아키텍처- 2014년 중반부터 떠오르기 시작하더니 국내에도 많은 시스템들이 MSA 사상으로 구현되가고 있는 것들이 보입니다. 이제 시작 단계들로 보이는데, MSA를 적용을 하고 있는 조직들은 MSA가 가지고 있는 전통적인 문제들, 분산 트랜잭션에 대한 처리, 여러개의 API를 모아서 새로운 기능을 만들어내는 개념들에서 많은 고민들을 하고 있는 것이 보입니다. 그리고 MSA를 개발하기 위한 개발환경을 셋팅하는데 많은 고민들을 하는데, MSA의 특성상 서버 컴포넌트가 많이 분산이 되고 다양한 언어로 개발현상이 조금씩 가속화 됨에 따라서, 이러한 복잡한 개발환경을 어떻게 개발자에게 전달할것인가가 새로운 키워드가 될 듯 합니다. 이에 대한 대안으로는 Docker등이 빠르게 떠오르고 있고, 사내/사외 개발용 클라우드를 구축 하는 움직임이 생기지 않을까 조심스럽게 점쳐 봅니다. MSA를 적용함에 있어서 앞단에 API gateway역할을 하는 것들이 중요해지고 있는데, 현재는 대부분 직접 개발해서 사용하는 경우가 많습니다. 그 만큼 거기에 사용할 제대로된 제품이나 오픈소스가 없다는 것인데, 아마 2015년에는 이러한 필요때문에 다양한 오픈소스가 많이 나올것이다.

데이터 스트리밍 프로세스- 빅 데이터 영역은 하둡을 중심으로 어느정도 정리가 되었으나, 근래에 들어서 실시간 데이터 분석에 대한 필요가 올라오면서 실시간 스트리밍 처리가 작년말부터 다시 주목 받는것 같습니다. 람다 아키텍처나 데이터레이크 아키텍처가 다시 언급되는 것도 같은 선상이라고 보는데, 2015년 Storm, Spark 중심의 실시간 데이터 처리기술이 부각될것이다.

머신 러닝의 보편화- 머신 러닝은 수학 통계적인 지식이 있어야 접근할 수 있는 분야였지만, 근래에는 Apache Mahout등의 프레임워크로, 주로 사용되는 머신 러닝 알고리즘 들은 대부분 프레임워크 되어 있어서 접근이 매우 쉽습니다. 약간의 지식만으로도 머신러닝을 사용할 수 있다는 겁니다.

여기에, Microsoft Azure의 ML 서비스와, IBM의 왓슨 서비스들은 클라우드 기반으로 머신 러닝 알고리즘을 서비스하는데, 사용이 매우 쉬워서, 일반 개발자들도 쉽게 머신 러닝 알고리즘을 구현 및 운영 환경에 적용이 가능합니다. 다른 빅데이터 분석들도 이런 흐름을 따라가지 않을까 싶은데 제가 보는 관점에서는 ML쪽이 선두가 되서 서비스화되는 현상이 말 부터 시작되고, 금년에는 초기 활성화 단계에 들지 않을까 합니다.

폴리 그랏- 작년도도 그랬지만, 금년도도 여러가지 프로그래밍 언어를 사용하는 폴리그랏 현상은 더욱 더 가속되지 않을까 합니다. Node.js등은 계속해서 약진할거 같고, Ruby, Groovy와 같은 기존의 스크립트 언어 뿐만 아니라 Google의 Go와, MS가 이번에 Linux까지 자사의 프로그래밍 언어를 지원하겠다고 한 이마당에, 금년에 프로그래밍 언어의 흐름은 지켜볼만 합니다.

기타- 자바스크립트의 약진, 자바스크립트 기반의 Pure 웹 클라이언트, 클라우드의 적용 가속화

이런것들은 워낙 빠른 이야기이니 별도로 언급하지 않겠다. 다만 마지막으로 지켜볼것은 중국 IT 기술의 약진으로, 금년에 중국발 오픈소스나 기술들이 인터넷으로 조금씩 공개되지 않을까 기대해봅니다.

형상관리

-형상 항목을 식별하여 그 기능적, 물리적 특성을 문서화하고, 그러한 특성에 대한 변경을 제어하며, 변경 처리 상태를 기록 및 보고하고 명시된 요구사항에 부합하는지 확인하는 지속적이고 관리적인 감독, 감시활동

-프로젝트는 진행되어 가면서 두가지 형태로 변화하는데, 첫번째는 새로운 산출물이 축적되고, 두번째는 이러한 산출물들이 계속해서 변전업이됨. 이렇게 변화되는 산출물들을 관리하는 것

변경관리, 버전관리, 빌드관리, 릴리즈관리, 배포관리 등을 모두포함

개발 프로세스 어느 단계에서 업무 환경, 기술 환경, 사용자 요구사항 변경등의 다양한 변경이 발생합니다. 소프트웨어 형상관리는 여러 요인에 의해 변경되는 각 단계별 산출물을 관리하는 일련의 활동을 말합니다. 이러한 활동을 통해 소프트웨어 개발 생명주기 기간동안 개발되는 제품의 무결성을 유지할 수 있으며, 실수를 최소화함으로써 생산성을 최대화 할 수 있습니다. 물론 관리하는 산출물에는 소스코드가 반드시 포함됩니다. 형상관리를 하기 위해서는 먼저 형상 식별의 과정이 필요한데 이는 관리할 대상을 식별하여 선정하고, 베이스라인의 기준을 정하는 것을 말합니다. 형상항목은 개발 프로세스에서 생산되거나 사용되는 작업 산출물 또는 자업 산출물들의 집합체를 말하며, 대표적인 소프트웨어 형상항목은 아래와 같습니다.

-품질관리계획서, 표준및 절차 매뉴얼 등의 계획서

-요구사항 명세서(분석)

-설계/인터페이스 명세서(설계)

-테스트설계서(시험)

-코드모듈(구현)

-보편적임=산출물다를수도있음

형상항목은 단순히 소스코드나 문서형식의 산출물만 범위에 해당하는 것이 아니라 각종 스크립트, 소프트웨어 개발 이력 등을 비롯하여 소프트웨어를 개발하는 사람들과 조직, 소프트웨어 개발프로세스, 소프트웨어가 탑재된 하드웨어 및 네트워크도 해당됩니다.

형상항목을 선정한 후에는 베이스라인 기준을 설정해야 합니다. 베이스라인이란, 생명주기 내에서 공학적, 관리적, 획득적 측면을 고려하여 정한 하나의 분기점 혹은 관리점과 그때의 산출물을 의미하는 것으로 개발 주기의 각 단계에서 산출되는 산출물에 대해 사용자의 요구조건을 만족시키는지 여부를 공식적인 검증과 확인을 거쳐 한단계를 동격하고 다음의 단계를 시작하는 기준점이 됩니다. 다시말해, 시스템의 생명주기의 일정 시점마다 그간의 제품상태(산출물)를 검토하고, 그 결과를 반영하여 다음 개발단계로 이전하는 형상관리 활동에서 산출물을 검토하고 이전하는 일정 시점을 베이스라인이라고 하는 것이지요. 베이스라인은 개발활동에 따른 시간의 순차에 의해 설정되며 일반적으로 5단계로 구분됩니다. 1. 시스템 기능분류기점 2. 소프트웨어 기능분류기점 3. 디자인기점 4. 최종산출물 생성기점 5. 운영개발기점

형상식별이 완료되면 형상통제를 수행합니다

형상통제란, 형상에 대한 변경 요청이 있을 경우, 변경 여부와 변경활동을 통제하는 것을 말하는데, 변경 요청사항에 대한 타당성을 검토하여 변경을 실행하고 그에따라 변경된 산출물에 대한 버전관리를 수행하는 것이 형상통제의 주요활동이라고 할수 있습니다.

버전관리- 형상항목이 과거부터 현재에 이르기까지 변경이 진행됨에 따라 각 변경에 대해서 버전을 부여하여 히스토리를 관리하는 것

- 버전을 통해 시간적인 변경 사항과 해당 작업 담당자를 추적 가능

변경관리 - 변경 요청사항에 대해 비용 및 기간 등을 고려하여 변경의 타당성을 평가한 뒤, 변경 여부를 결정하고 실행하는 일련의 과정

형상통제는 개발이 진행되는 도중 변경이 발생될때마다 수시로 수행되는 것입니다. 형상통제가 제대로 수행되는지 여부를 확인하기 위해 일정 주기별로 형상 관리자가 형상상태보고서를 작성하여 베이스라인의 현재 상태 및 변경 항목들이 제대로 반영되었는지 여부를 보고하는데요. 이를 형상상태보고 라고 합니다.

현재 프로젝트에서는 산출물의 형상 상태를 형상관리대장 이라는 이름으로 목록화하여 관리하고 있습니다.

예를들어, 분석단계에서 작성된 요구사항 정의서가 설계, 구현, 시험 단계를 거쳐 고객에게 최종 인수될 때까지 각 단계의 일정 시점마다 산출물의 상태를 점검하여 버전을 기록하는 것이지요. 이런것도 일종의 형상상태보고로 볼 수 있을 것 같습니다. 마지막으로 형상감사를 수행하는데 이는 형상 항목이 요구사항에 맞도록 잘 변경되었는지 형상 관리 계획서대로 형상 관리가 진행 되고 있는지 등을 확인하고 베이스라인의 무결성을 검증하는 활동입니다. 검증과정으로써 -모든승인된 변경의 반영 여부 - 관련된 항목의 갱신 여부 -승인되지 않은 변경의 반영 여부 -인도될 항목과 요구사항, 디자인과의 일치 여부 -모든 알려진 소프트웨어의 문제가 변경 승인에 의해 검토되었는지 여부 - 버전 기술 문서의 준비 여부