개발 방법론에 대한 정의

- 폭포수, 나선형, 애자일, XP, 린 등에 대해 정리

- 개발론별 장단점을 파악

- 현재 팀에 적용할 방법론에 대해 제안

----------------------------------------------------------------------

1. 소프트웨어 개발 프로세스(Software Development Process)

- 소프트웨어 제품을 개발하기위해 필요한 과정 또는 구조

- 비슷한 말로 소프트웨어 생명 주기와 소프트웨어 프로세스가 있다

2. 개발 모형

1) 개발 모형의 종류

애자일 소프트웨어 개발, DSDM, 스크럼, 클린룸, 순차 점증적 개발, 반복형 개발, RAD, RUP,

나선 모형, 폭포수 모델, 익스트림 프로그래밍(XP), V 모델, TDD 등의 여러 모형이 존재한다

(Agile, CleanRoom, Incremental, Prototyping, Spiral, V model, Waterfall)

2) 개발 모형 정리

[ 애자일 소프트웨어 개발(Agile Software Development) ]

- 문서를 통한 개발방법이 아닌, 실질적인 코딩을 통한 방법론

- 일정한 주기를 가지고 끊임없이 프로토 타입을 만들어 나가며 그때 그때 필요한 요구를 더하고 수정하여 하나의 커다란 소프트웨어를 개발해 나가는 적응형 스타일

- 애자일 개발 방법론은 애자일(Agile : 기민한, 좋은 것을 빠르고 낭비없이 만드는 것) 개발을 가능하게 해 주는 다양한 방법론 전체를 일컫는 말

|  |  |
| --- | --- |
| Framework | |
| Adaptive software development(ASD) | 소프트웨어 개발을 혼란 자체로 규정하고, 혼란을 대전제로 그에 적응할 수 있는 소프트웨어 방법을 제시하기 위해 만들어진 방법론. 다른 방법론과 유사하나 합동 애플리케이션 개발(Joint Application Development : 사용자나 고객이 설계에 참여하는 개발 방법론)을 사용함 |
| Agile modeling | 모범 사례를 기반으로 소프트웨어 시스템을 모델링하고 문서화하는 방법론. 소프트웨어 개발 프로젝트에 적용될 수 있는 가치와 원칙의 모음. 기존 모델링 방법보다 더 유연하여 빠르게 변화하는 환경에 적합함  장점 : 스크럼, 익스트림 프로그래밍과 같은  다른 애자일 방법론 보완  단점 : 개인적인 의사소통과 고객 협력에 크게 의존 |
| Agile unified process(AUP) | Rational Unified Process(RUP)의 단순화된 버전. 간단하고 이해하기 쉬운 접근 방식. Disciplined agile delivery 방식으로 대체됨 |
| Disciplined agile delivery(DAD) | 점진적이고 반복적인 솔루션 제공에 대한 프로세스 결정 단순화 가능. 스크럼, 애자일 모델링, 린 소프트웨어 개발 등을 포함함. 사람 중심의 학습 지향 하이브리드 접근 방식 이용 |
| Dynamic systems  development method(DSDM) |  |
| Extreme Programming(XP) | 고객과 함께 2주 정도의 반복 개발 진행, 테스트 우선 개발(TDD)을 특징으로 하는 명시적인 기술과 방법을 가짐 |
| Feature-driven development(FDD) | 기능마다 2주정도의 반복 개발 실시. UML을 이용한 설계 기법과 밀접한 관련을 가짐 |
| Lean software development |  |
| Lean startup |  |
| Kanban | 인간 시스템 전반에서 작업을 관리하고 개선하는 린 방법. 수요와 가용 용량의 균형을 맞추고 시스템 수준 병목 현상 처리를 개선하여 작업을 관리하는 것을 목표로 함. |
| Rapid application development(RAD) | 계획에 덜 중점을 두고 적응 프로세스에 더 중점을 둠. RAD는 사용자 인터페이스 요구 사항에 따라 구동되는 소프트웨어를 개발하는데 특히 적합함.  방법  1) 요구 사항 계획 단계  2) 사용자 설계 단계  3) 구축 단계  4) 컷 오버 단계  장점 : 더 좋은 품질을 얻을 수 있음. 위험 완화.  시간과 예산 내에서 프로젝트를 완료할 수 있음  단점 : 새로운 접근 방식의 위험.  정상 작동 시 최종 사용자에게 보이지 않는  비 기능적 요구 사항에 대한 강조 부족.  부족한 자원의 시간 필요. 통제력 감소.  형편없는 디자인과 확장성 부족 |
| Scrum | 30일마다 동작 가능한 제품을 제공하는 스프린트(Sprint) 중심. 매일 정해진 시간에 정해진 장소에서 짧은 시간의 개발을 하는 팀을 위한 프로젝트 관리 중심의 방법론 |
| Scrumban | Scrum + Kanban. 스크럼에서 칸반으로 전환하는 방법으로 설계됨. |
| Scaled Agile Framework – SAFe | 대규모 스크럼(LeSS), 훈련된 민첩한 전달(DAD), 넥서스와 함께 단일 팀을 넘어 확장 시 발생하는 문제를 해결하고자 하는 프레임 워크 중 하나. 애자일 소프트웨어 개발, 린 제품 개발, 시스템 사고라는 세 가지 주요 지식을 이용하여 개발되었으며, 수많은 팀에 걸친 조정, 협업 및 전달을 촉진함. 너무 계층적이고 융통성이 없다는 비판을 받고 있음  기본 원칙  1) 경제적 관점 취하기  2) 시스템 사고 적용  3) 가변성 가정  4) 빠른 통합 학습 주기로 점진적으로 구축  5) 작업 시스템의 객관적인 평가에 대한 기본 이정표  6) 진행중인 작업을 시각화 및 제한하고,  배치 크기를 줄이고, 대기열 길이 관리  7) 케이던스(타이밍) 적용, 도메인 간 계획과 동기화  8) 지식 근로자의 본질적인 동기 부여  9) 의사 결정 분산화  10) 가치를 중심으로 구성 |

[ 클린룸 소프트웨어 개발(Cleanroom Software Development) ]

- 내용

[ (Incremental build model) ]

- 내용

[ (Software Prototyping) ]

- 내용

[ (Spiral model) ]

- 내용

[ (V-Model) ]

- 내용

[ (Waterfall model) ]

- 내용