Agile

2020년 6월 7일 일요일 오후 11:15

Agile
2.애자일방법,기술,규모

3.애자일필요성
소프트웨어 요구사항이 계속 변하기 때문에 불가피하게
빠르게 대응,적응 할 필요가있다.
계획기반 plan-driven development가 변화에 대응하기 힘듬.

4 애자일 개발
Specification ,design ,Impremention 동시에 일어남.
증분형태로 버전형태로 발전하고 stakeholder가 참여
새로운 버전이 빈번히 발생
Testing tools이 쓰임
문서작업 최소화 및 코드에 포커스
Plandrived는 Specification이 문서작업으로 이루어짐

6 plan-driven VS Agile development
Plan : 각각의 단계가 분리되어있음.
단계에따른 output이 주어짐
Waterfall model, incremental development 등등
Agile : Specification ,design ,Impremention 동시에 일어남.

8 Agile Methods
1) Code>design
2) 반복적 접근
3) 빠르게 소프트웨어 개발 사용자에게 전달

9 성명서
프로세스와 tools 보다 개인의 상호작용
문서보다 software만드는것
Customer과 협동하여
계획을짜는것보다 변화에 대응하는형식으로

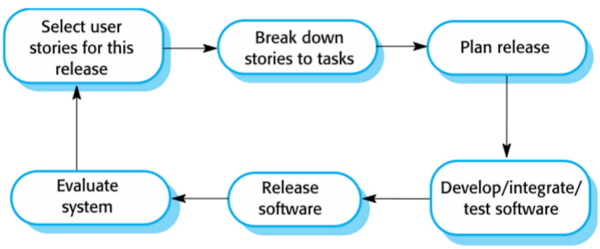
10 agile method의 원리
1)customer involvement :customer의 개입및 평가
2)증분형태로 개발되어 customer에게 전달
3)process중심이 아님
4)변화의 수용및 빠르게대응
5)단순성 유지 : 코드를 읽기쉽게 구현해야한다,

11 Agile method applicability
Customer에 관해 명확히 나와있고 개발에 참여할수 있을때
사용함.
만드는 소프트웨어가 법규와 같은 외부의 제한요소가 별로 없는 환경

13Agile method technique

Extreme programming
새로운버전이 하루에 여러번만들어짐
이주에 한번씩 custommer에게 증분개발하여 보냄
개발을 할때마다 test해야함.

14 extreme programming relase cycle



15 Extreme progrmming practices

Incremental planning
Small releases
Simple design
Test-first devlopment-자동화된 테스트 프레임워크
Refactorint계속 개선

16 Extreme progrmming practices(B)
Pair programming: 두명이 하나의 프로그램 함수를작성

Collective ownership: 시스템 전체적으로 두명이서 짬. 전체팀이 코드를 책임지는형태
Continuous integration: 계속적인 통합 테스트
Sustainable pace: 알을 너무 시키면안된다.
On-site customer: 최종사용자가 full-time으로 개발팀과 일할수있고 available 해야한다.

17 XP(extreme programming) and agile principles
-Incremental development(증분 개발): 작고 빈번히 업데이트 되는 시스템 releasese
-Customer involvement : full-time customer의 관여
-pair programming,collective onwership등을 통해서 진행됨
People not process
-변화에 잘대응(releases 를 통해서)
-변화가 단순하게 이루어져야함 코드의 단순화

18 영향력있는 XP practices
1)UserStories
-Customer들도 user의 일원, 결정하는데 있어 책임을 져야함
-Requirements는 시나리오,Userstoies로 작성됨
-개발팀들이 이것기반으로 다시 구현가능한 task로 분해, 스케줄링하고 비용산출하게됨
-다음 releases에 어떤것들이 포함되어야할지 우선순위와 스케줄을 고려하여 스토리를 선택
2)Refactoring 코드개선 쉽게변화에 대응하기위해서 코드잘짜자자
문제가있을때마다 바꾸는게아님.(따라서 문서작성필요x)
구조화하여 짤짜는것->변화잘줌
아키택쳐자체를 바꾸는것은 힘듬
예 이름바꾸거나 메서드
3)test-first development
모든 기능들을 test 시나리오에 따라서
자동화된 test tool 존재(Promgram형태로) 효율적임
모든 component들을 test
4)customer의 관여:팀의 일원

29 test automation
먼저 testcode부터 작성!!
Task에대한 이해를 하는데 큰도움이됨

30 단점:
1)사람들이 예외상태를 제대로 고려하지않은 상황이 발생할 수 있음
2)복잡한 user interface의 경우 짜기 힘들 수 있음.
3)테스트가 잘됐는지 모든 coverage에대해 test 하기 힘든경우가 발생할 수 있음

31 Pair programming: 두명이 하나의 컴퓨터를 가지고 코드를 개발
1)지식이 전파하게됨
2)모든 코드에 대해 review를 하게됨
비효율적이지않고 생산적일수 있음
숙련된 개발자들의 경우 효율이 떨어짐.

33 **Agile project management**
제한된 예산과 시간안에 프로젝트짜는게 매니저역할
애자일에서 문서가없기에 해당 단계의 manage하기가 쉽지않음

35 Scrum (manage방법)
Iterative development 사용
Initial 단계 : 목적, 아키텍처 디자인
sprint cycle:
Closure:
36 scrum 용어

Development team	개발팀
Potentialy shippable product increment	Sprint부터 만들어지는 소프트웨어 increment
Product backlog	scrum팀이 구현해야 할 사항을 정의한 리스트이

반복 개발이란?
1. 복잡한 문제를 다루기 쉬운 작은 단위로 분해한다.
2. 분해된 작은 단위 문제를 해결하는 코드를 개발하고 테스트한다.
3. 모든 분해된 문제를 해결할 때까지 2번 단계를 반복한다

Potentially shippable product increment	Sprint후의 건물어지는 소프트웨어 increment
Product backlog	scrum팀이 구현해야 할 사항을 정의한 리스트이다. 다양한 요구명세가 있고 우선순위로 나뉘어져 있다
Owner	우선순위정하기,Backlog점검,확인하는작업
Scrum	매일 이루어지는 회의(해야할일들에대해 우선순위)
Scrum master	Project에 대한 상황같은것과 잘 이루어질수있도록 이끄는사람 Scrum Master는 개발의 측면에서 PO가 말한 비즈니스 기능들을 개발 가능한 단위로 쪼개고 조율하는 역할을 합니다. 외부의 방해를 예방 개발한것들을 리뷰하고 이해관계자들에게보여줌
Sprint	개발 iteration 2주에서 4주
Velocity	개발역량

38 Scrum sprint cycle : 고정된 길이 2주에서 4주
Backlog로부터 개발하는시점
팀은 customer로 부터 분리되어서 개발
대화가 필요할때는 마스터를 통해서 개발

41 Teamwork in scrum
모든팀원 매일회의에 참여하고 진행 과정에 대해서 이야기나눔
문제가 발생 하였을때 빠르게 해결하기 위한 방법찾음

42 이점

1.product가 관리가능하고 이해가능한 조각으로 쪼개져있다. 개발관리 쉬어짐
2.불안정한 요구사항 일 방해하지않음
3.팀의 상황이 어떻게 돌아가고 있는지 볼 수 있기에 효율적인 communication
4.customer가 적절한 시기에 product에 대한 feedback을 줄 수 있다.
5.개발자들과 customer사이의 신뢰관계가 쌓이게됨

43 분산 scrum(참고)

44 Scaling agile method(에자일 기법의 규모정하기)

애자일 기법을 조금더 크고 긴 프로젝트로 적용하기 위한 방법

46 Scailing out vs Scailing up

Scailing up: 거대한 소프트웨어 시스템에 agile method를 적용시키기 위한방법

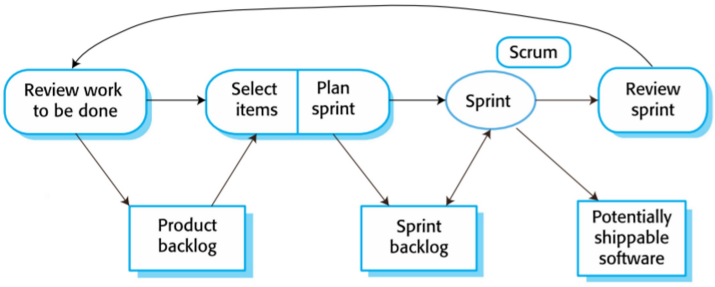
Scailing out: 에자일 방법을 오래된 소프트웨어 개발 경험이있는 거대한조직에 적용시킬 수 있는방법

47 애자일 기법의 문제점

1.기존 큰 규모의 회사에서 애자일 방법에서의 간소한 계약을 적용시키기에 적합하지가 않다.
2.새로운 소프트웨어 만드는데 적합하지만 기존에 있는 소프트웨어를 유지보수하는데는 적합한방법이아님
3.애자일방법은 작은팀단위의 개발에 적합.
분산된팀의경우 적합하지가않음

48 Specification이 명확하게 없기에 계약형태에 적합하지가않음.(End user 참여)
49 에자일 방법의 유지보수
유지보수를 지원할 수 있어야함
이슈1) 문서를 최소화하면서 계속해서 유지보수 가능한시스템인가?
이슈2) 애자일기법이 customer의 요구가 바뀌는것에 대해 효율적으로

2. 분해된 작은 단위 문제를 해결하는 코드를 개발하고 테스트한다.
3. 모든 분해된 문제를 해결할 때까지 2번 단계를 반복한다



사용할 수 있을것인가?

50 유지보수 어려운점:
1.문서부족
2.customer관여
3.개발팀이 계속해서 유지되기 힘듦

51 애자일 vs plan-driven balnce

-매우 자세한 Specification과 설계가 구현되어있다면 Plan-driven이 나을 수 있음
-현실적으로 점진적 delivery strategy가 가능하고 customer에게 feedback을 받을 수 있다면
-개발할려는 시스템이 얼마나큰지에 따라서 나누어짐

52 Agile 원칙, Organization practice(실무적)
고객관여:
요구사항변화:이해관계당사자들의 우선순위가 다름
점진적 전달(incremental delivery)
단순성유지: 개발중간의 압박으로 인해 어려울 수 있음
사람이 프로세스보다 중요: 사람의 인성이 중요''

시스템이슈
분석이필요할때는 plan-driven이유리
시스템라이프타임이길때 plan-driven이중요

People and teams 애자일기법은 plan-based보다 더 큰역량이필요

LargeSystem에 애자일기법 어려움.
대형 시스템과 그 개발 과정은 종종 개발될 수 있는 방법을 제한하는 외부 규칙과 규제에 의해 제약을 받는다.
∘대규모 시스템은 조달 및 개발 시간이 길다. 그 기간 동안 시스템에 대해 알고 있는 일관성 있는 팀을 유지하는 것은, 필연적으로 사람들이 다른 직업과 프로젝트로 옮겨가기 때문에 어렵다.
∘대규모 시스템에는 일반적으로 다양한 이해관계자 집합이 있다. 이러한 서로 다른 이해관계자 모두를 개발 과정에 참여시키는 것은 사실상 불가능하다.

62 **Scailng Up to large systems**(하이브리드형태)
완벽한 incremental aproach는 힘듬

1)계약에 필요한 정도 의, 다른팀과의 협업을 위한	**수준의 requirement는 초기에 만들어 놓음.**

상세한 requirement는 paln-driven으로 실행함
2)지속적으로 다양한 사람과 의사소통
3).code focus도 어느정도함
4)주기적으로 integration, 개발하면서 적용
63참고