노동부 Ministry of Labor 중소기업직업훈련컨소시엄

Korea Software Technology Association

UML2.x 기초 다루기

훈련기간: 2010.01.25 ~ 02.05

강사명: 손재현 *-넥스트리소프트*
-jhsohn@nextree.co.kr

Copyright © 2009 Nextree Soft, All rights reserved

과정 개요

❑ 교육 목표 & 특징

- UML2.x의 이해
- 유스케이스 작성
- 객체모델링 이해
- UML2.x의 다양한 다이어그램 이해 및 활용
- 모델링 도구 사용법 습득

Copyright © 2009 Nextree Soft, All rights reserved

강의 요구 기술

- **본 강의는 아래 기술에 대한 이해를 필요로 합니다.**
  - 객체지향 언어(Java) 기초
  - 개발프로세스 이해

Copyright © 2009 Nextree Soft, All rights reserved

교육 일정표

❑ 교육은 매 회 4 시간씩 총 5회에 걸쳐 진행합니다.

1 일차	2 일차	3 일차	4 일차	5 일차
- UML 개요 UML 소개 UML 역사 UML 다이어그램분류	- 구조 다이어그램 클래스 객체 컴포넌트 배치	- 행위 다이어그램 유스케이스 액티비티 상태기계	- 상호작용 다이어그램 상호작용 Overview 시퀀스 커뮤니케이션 타이밍	- 유스케이스 I 유스케이스 개요 유스케이스 내용 유스케이스 다이어그램
6 일차	**7 일차**	**8 일차**	**9 일차**	**10 일차**
- 유스케이스 II 유스케이스 목표수준 유스케이스 명세 유스케이스 패턴	- 유스케이스 III 유스케이스 분석기법 분석클래스 제어클래스 실체클래스	- 요구사항 모델실습 I 유스케이스 사용자 시나리오 핵심개념 모델	- 요구사항 모델실습 II 인터페이스 추출 유스케이스 분석 컴포넌트 식별	- 설계모델 실습 컴포넌트 설계 유스케이스 설계 도메인 모델

Copyright © 2009 Nextree Soft, All rights reserved

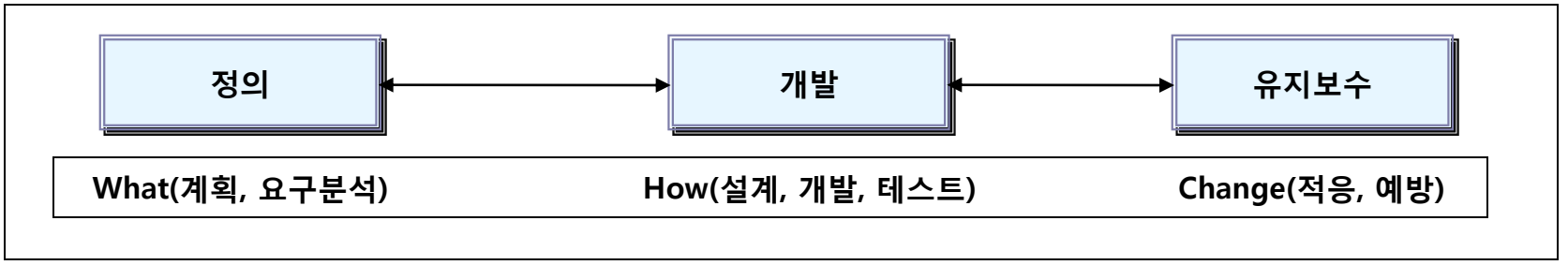
1일차 – 보충자료

Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 프로세스

❑ 소프트웨어 공학 프로세스

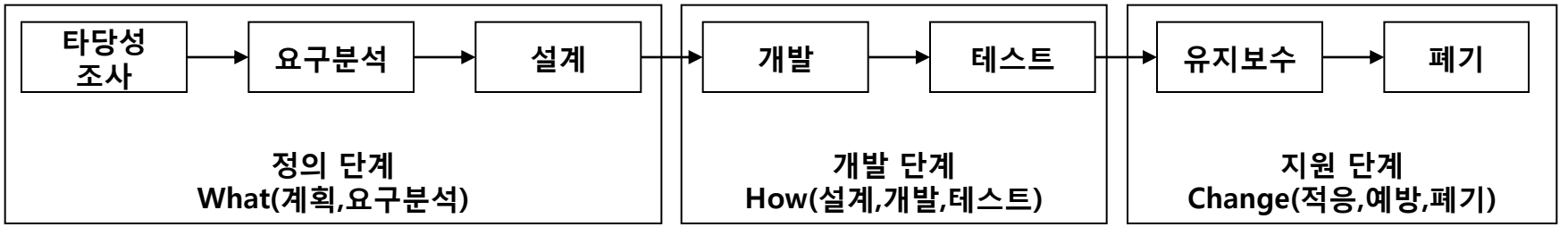
- 정의단계*Definition Phase* : 무엇*What*에 초점
  - 처리되는 정보, 성능과 기능, 인터페이스, 설계제약 조건, 검증기준 등의 기술
  - 시스템과 소프트웨어의 주요 요구사항 결정단계
  - 소프트웨어 프로젝트 계획, 요구분석 단계
- 개발단계*Development Phase* : 어떻게*How*에 초점
  - 데이터 구조화, 소프트웨어 기능 및 설계, 개발, 테스트에 대한 기술 단계
  - 소프트웨어 설계, 코드생성, 소프트웨어 테스트 단계
- 지원단계*Support Phase* : 변화*Change*에 초점
  - 오류수정, 소프트웨어 사용환경변화에 따른 변화, 사용자 요구에 따른 변경
  - 기존 소프트웨어의 성질은 변화시키지 않는 범위에서 적용시킴
  - 종류:수정(결함수정), 적응(환경변화), 강화(기능추가), 예방(품질향상)



Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 프로세스

❑ 소프트웨어 개발 생명주기 모델의 단계



구분	내용
정의단계 Definition Phase	타당성, 요구 명세화 타당성 조사, 소프트웨어의 기능과 제약조건을 정의하는 명세서 작성하는 단계
개발단계 Development Phase	개발 단계 명세서를 만족하는 소프트웨어를 실제 개발, 구현하는 단계 (설계, 구현) 확인, 검증 단계 요구 명세화 내용을 기준으로 사용자가 원하는 소프트웨어인지를 확인하는 검증 단계 (시험, 설치)
지원단계 Support Phase	유지보수, 폐기 단계 운영환경의 변화 및 사용자 요구사항의 변화를 수용하기 위한 진화/적응 및 프로그램 폐기

Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

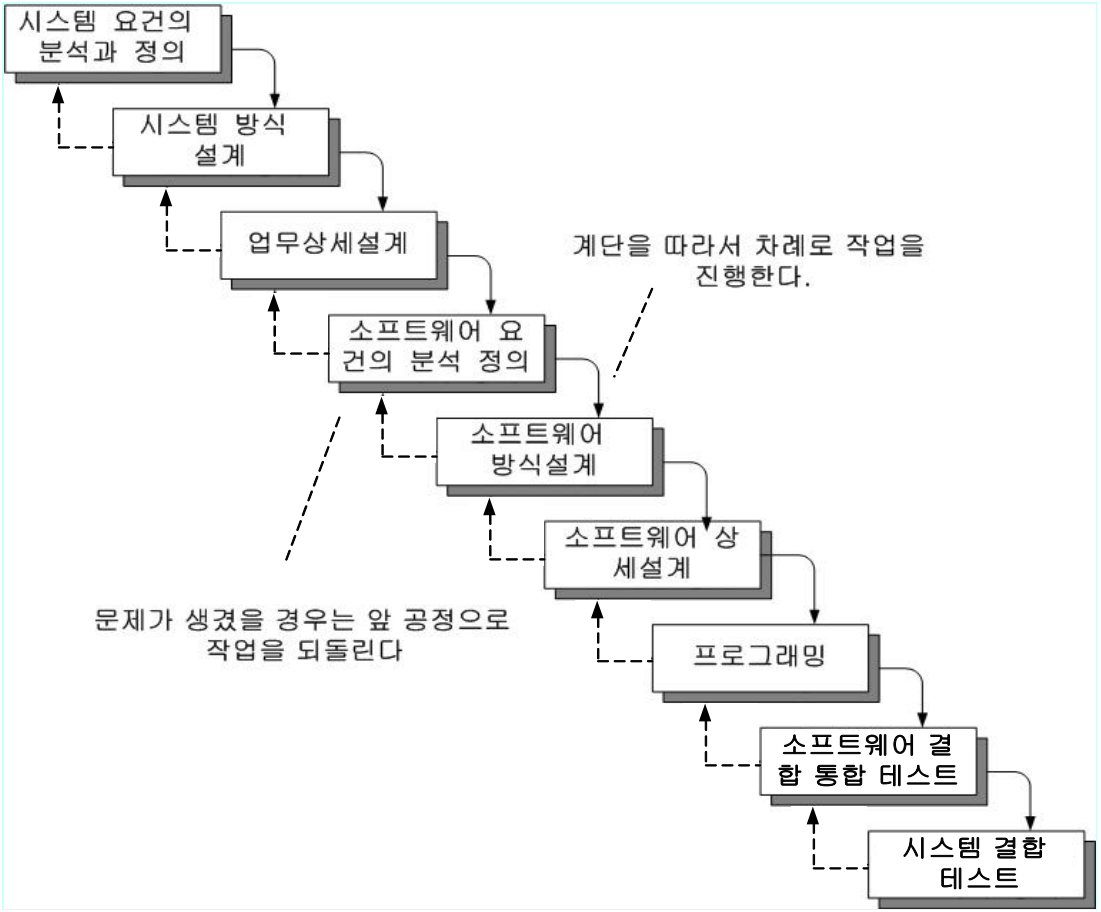
- **생명주기 개념에 따라서 세운 것**
  - 시스템이 어떠한 순서를 밟아서 개발되며, 운용, 보수되어 생애를 마칠 때까지의 작업 프로세스를 모델화(modeling)한 것
- **소프트웨어 프로세스**
  - 고객과 개발자, 고객과 도구(technology), 개발자와 도구들간의 상호작용
  - 좋은 품질의 소프트웨어를 얻기 위한 업무의 프레임워크(framework)
  - 소프트웨어가 설계되는 접근방법.
- **프로세스 모델은**
  - 프로젝트와 어플리케이션의 성격
  - 사용되어질 메소드와 도구
  - 요구되는 관리와 산출물 등에 따라 선택
- **제안모델**
  - 선형순차적 모델
  - 나선형 모델
  - 프로토타이핑 모델
  - 컴포넌트 기반 모델

Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 선형 순차적 모델

- 종종 고전적(classic) 또는 단계적(phased)생명주기라 불리기도 하고, 가장 널리 사용하는 것은 폭포수(waterfall)모델



Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 특징

- 공업제품의 개발을 근본으로 한 모델
- 공정이 이미 정해져 있으며, 어떤 공정에서 문제가 생겨도 앞의 공정 또는 앞에 앞의 공정 등에 되돌아가는 것이 곤란

❑ 장점

- 소프트웨어의 개발과정을 개념적으로 자연스럽게 표현
- 단계별 작업진행으로 해당 단계의 진척 관리가 쉬움
- 반드시 각 공정마다 눈에 보이는 성과물 출력
- 다음 공정의 개시 예정이 세워져 있어 일정관리가 쉬움
- 사양이 분명해서 유사한 시스템의 개발 경험이 있는 경우 효율, 품질 면에서 우수

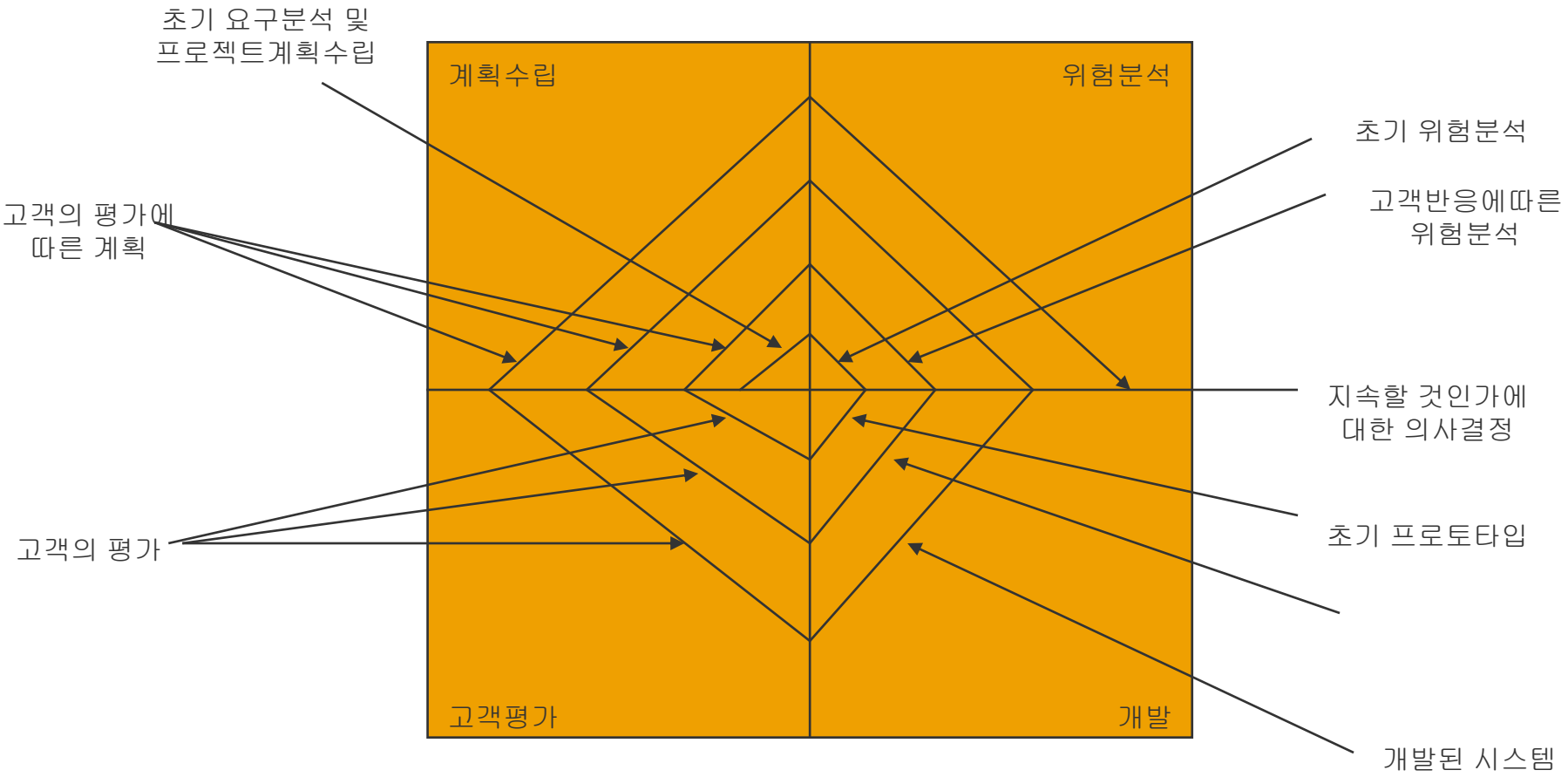
❑ 단점

- 앞 공정으로 되돌아가서 다시 반복하는 등 많은 시스템 개발(사양이 애매모호한 시스템, 신규개발 시스템)에서는 작업공정의 변경이 곤란
- 처음 단계에서 지나치게 강조하다 보면 코딩이나 테스트 작업 지연
- 사용자의 요구를 만족하는지는 최종적인 성과물이 완성되어야만 판명

Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 나선형 모델



Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 정의

- 기업 내에서 경영환경의 변화에 따른 업무개선을 하는 데에 사용되는 PDCA(Plan-Do-Check-Action)주기의 생각을 시스템 개발에 적용하고, 명세의 변경도 적극적으로 받아들이자는 생각으로 고안된 것
- PDCA : 계획을 세우고, 실행하고, 체크하고, 행동하는 활동

❑ 장점

- 시스템 개발 중에 일어나는 변경에 대하여 유연하게 대응.
- 프로토타이핑(prototyping)에 의해 요구를 조기에 확인할 수 있고, 사용자 요구에 일치된 시스템이 개발

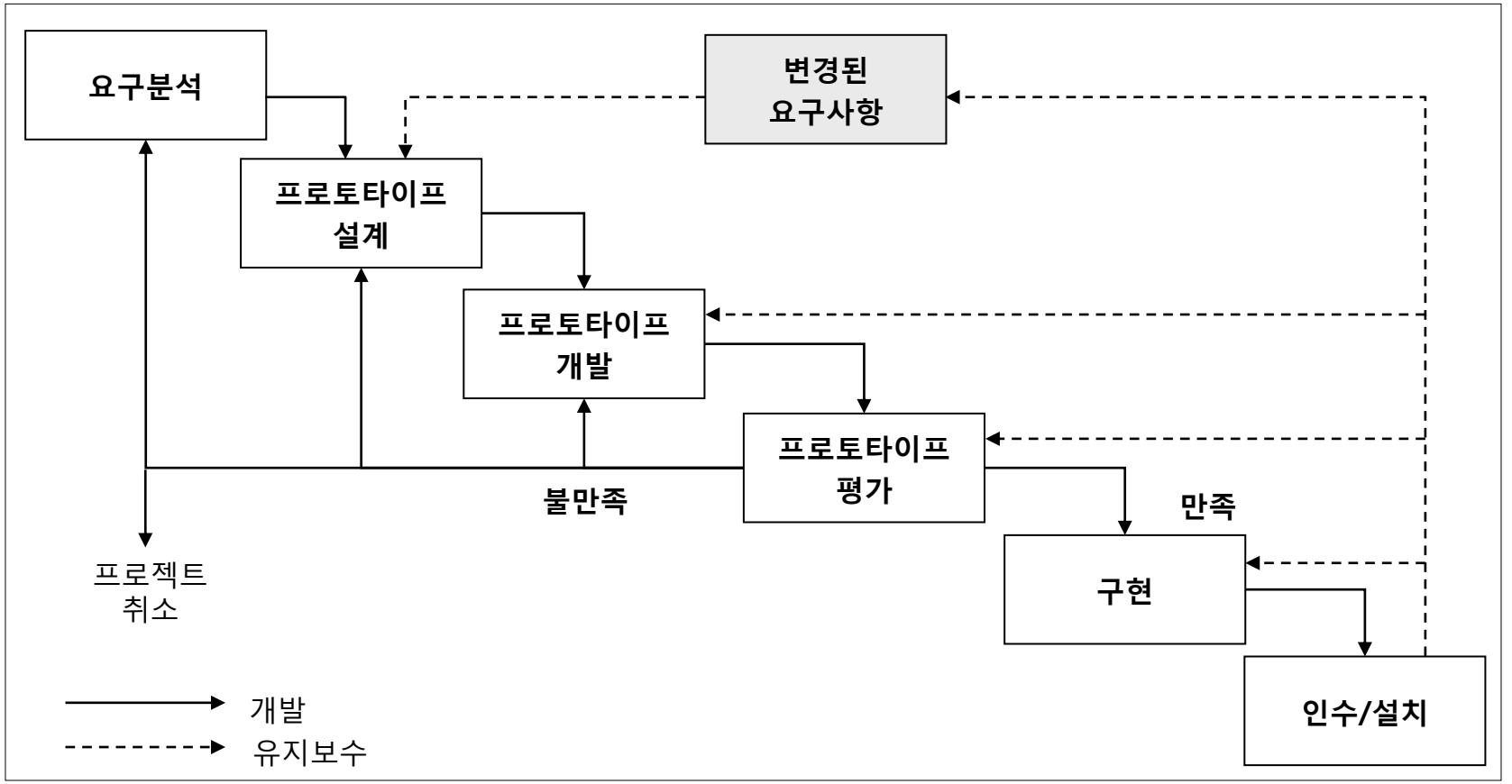
❑ 단점

- 필요에 따라 똑같은 단계를 반복함으로 공정관리와 개발요원의 배치가 복잡
- 이 모델은 다른 모델에 비해 프로젝트의 관리가 어려워서 프로젝트 성공을 위해서는 리스크(Risk)를 다룰 수 있는 전문가 필요

Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 프로토타이핑 모델



Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 정의

- 프로토타이팅(시스템 일부 또는 시스템 모델을 만드는 과정)을 통하여 개발하려는 시스템주요기능을 사용자의 요구사항을 최대한 반영하여 초기에 개발하는 방법
- 폭포수 모델(Waterfall Model)의 단계적 개발방법의 단점 보완하여 실제의 소규모운영 모델을 단기간에 구체화 시키는 방법
- 일회용, 진화용 시제품으로 구체적인 요구분석에 임하는 전략 또는 방법

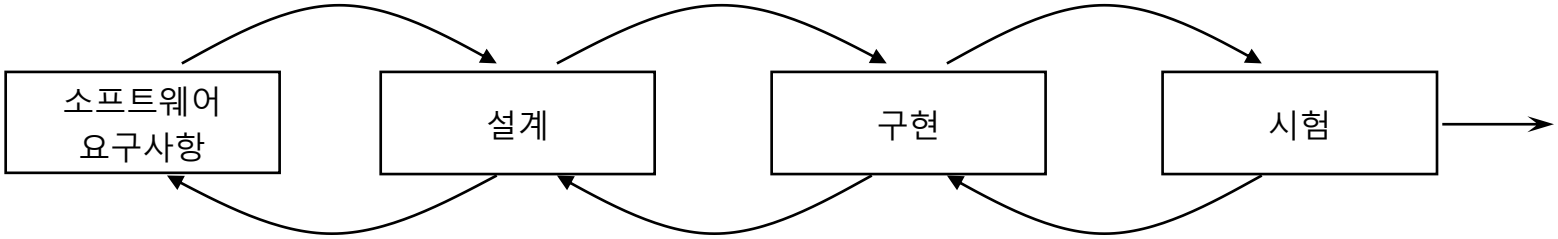
❑ 목적

- 사용자의 요구사항 추출과 확인
- 시스템 외부에서 본 동작이나 사용자 인터페이스의 모델화
- 시스템의 내부구조, 기능의 모델화

Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 반복적 개발 모델



각각의 사이클은 정해진 Time-Box 내에 수행될 수 있도록 적절한 부하를 할당
(Time-Box는 2주~2달 정도가 적절)

Copyright © 2009 Nextree Soft, All rights reserved

소프트웨어 개발 모델

❑ 정의

- 소프트웨어의 구조적 관점에서 하향식 계층구조의 수준별 증분을 개발하여 이들을 통합하는 개발 모델
- 사용자의 요구사항의 일부분을 제품의 일부분으로 반복적으로 개발하여 최종제품을 완성하는 방법으로 전통적인 폭포수 모델과 프로토타이핑 모델을 결합한 것

❑ 특징

- 폭포수, 프로토타이핑 개발모델의 개념을 포괄하는 보다 진보된 개발 모델
- 사용자 요구사항의 일부분을 제품의 일부분으로 반복적으로 개발하여 최종제품을 완성하는 중/대규모 시스템 구축에 적합한 모델
- 모든 프로세스 단계마다 프로덕트가 생산되는 가시화 가능한 진행
- 진화 또는 증분 개발모델로 시스템의 복잡도가 급격하게 늘지 않음
- 손쉽게 사용자의 평가가 이루어져서, 시장 변화에 효율적인 대응이 가능

Copyright © 2009 Nextree Soft, All rights reserved

애자일 개발 프로세스

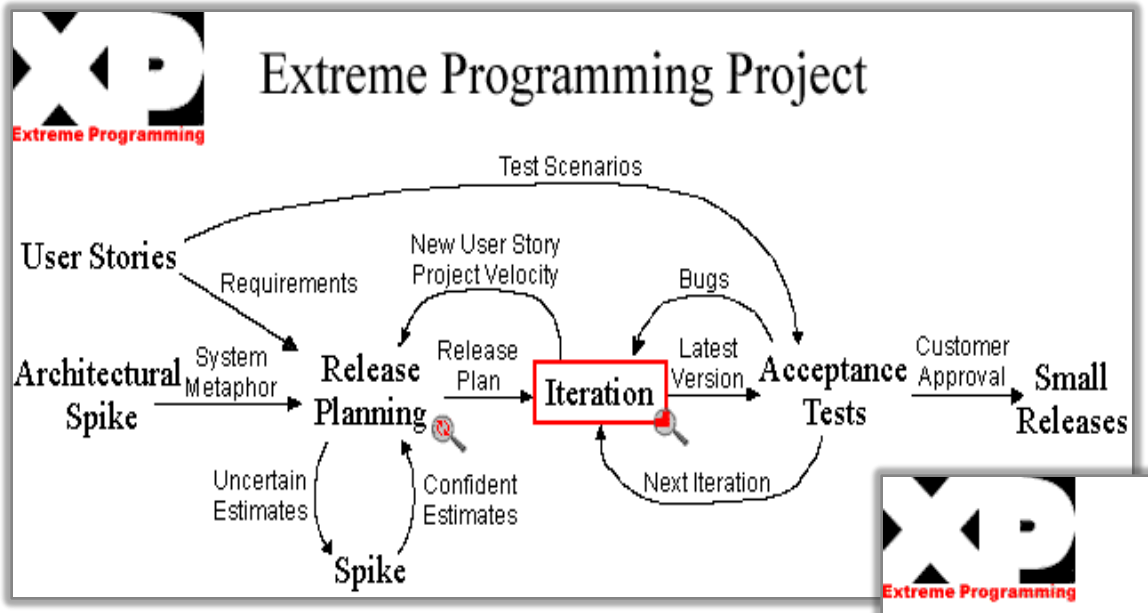
❑ Agile 소프트웨어 개발 실천사항의 중요단계

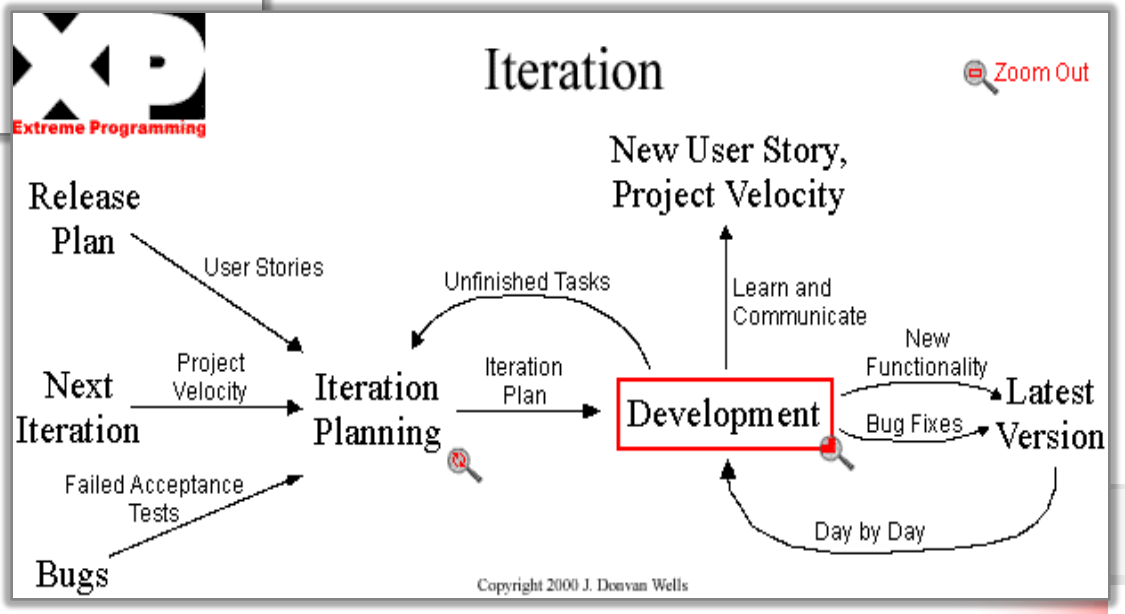
- 계획단계 *Planning game*
- 작은규모 릴리즈 *Small Release*
- 단순한 설계 *Simple Design*
- 테스팅 *Testing*
- 지속적인 통합 *Continuous integration*
- 리팩토링 *Refactoring*
- 짝 프로그래밍 *Pair Programming*
- 코드 공동 소유 *Collective Ownership*
- 40시간 내 해결 *40-Hour week*
- 현장 고객 상주 *On-site customer*
- 메타포 *Metaphor*
- 코딩 표준 *Coding standard*

Copyright © 2009 Nextree Soft, All rights reserved

애자일 개발 프로세스

❑ eXtreme Promgramming



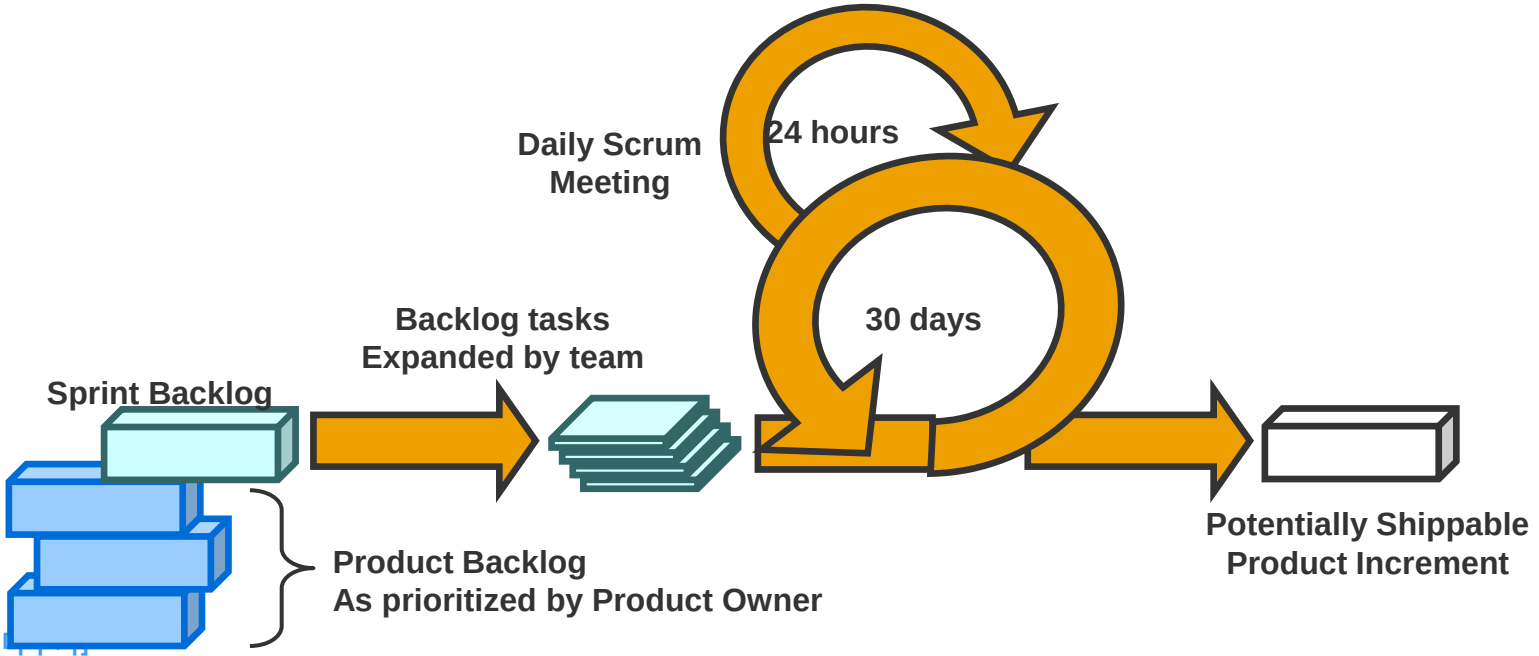


Copyright © 2009 Nextree Soft, All rights reserved

애자일 개발 프로세스

❑ Scrum

- 어떻게 코드를 작성(또는 관리)하는지 언급하지 않음
- 민첩하고 경량화된 프로세스로서 소프트웨어나 제품 개발을 관리하고 제어
- 점진적이고 반복적인 과정을 사용
- 적응성과 경험적 시스템 개발 중심적이다.
- 간결한 프로세스를 추구하여 생산성을 높이고 개발시간 단축
- 모든 프로젝트나 모든 조직에 적합한 것은 아님



Copyright © 2009 Nextree Soft, All rights reserved

개발 프로세스와 UML

❑ UML은 모델링 언어일 뿐 개발 프로세스는 아님

- 개발 프로세스는 업무 처리 절차에 대한 정의와 각각 업무에 대한 지침
- 모델링 언어는 표기법(또는 다이어그램) 만 제시
- UML이 여러 다이어그램을 제시함으로써 소프트웨어 개발과정의 산출물을 비주얼하게 제공
- 또한 개발자와 고객 또는 다른 개발자와 의사소통을 원활하게 할 수 있음
- 어떤 개발 프로세스에서도 모델링 언어로 UML을 사용할 수 있음

Copyright © 2009 Nextree Soft, All rights reserved

객체지향 이외의 언어와 UML

❑ UML은 OO를 위한 도식적 설계 언어

- SSAD와 OOAD 프로세스 이해 필요

UML로 구조적 프로그래밍 설계를 하고 있습니다.정말 안맞습니다.
Submitted by ctcquatre on 목, 2005/05/12 - 4:56pm.

UML로 구조적 프로그래밍 설계를 하고 있습니다.

정말 안맞습니다. 그래서 표식을 전부다 다른 의미로 쓰고 있습니다.

뭐 예를 들면 --> 는 핵심 호출부, 연계관계..

이렇게 다시 정의해서 그냥 저 편한대로 쓰고 잇습니다.

왠만하건 워크그룹단위로 나눠버리고요.

정말 UML은 정말이지 구조적과는 전혀 맞지 않는것 같습니다.
구조적 접근을 염두해두지 않고 만들었으니 당연한거라 생각합니다.
혹시 책이나 문서르 본다면.. 꼭 알려주세요.

»
Login or register to post comments

설계요..? 그런거 하십니까?
Submitted by magingax on 월, 2007/08/13 - 9:14am.

학교 대닐땐 UML, 객체지향설계 니 뭐니 참 많이 배웠는데..
실제로 프로젝트 해보면..
그냥 생각나는대로..
화이트보드에 모듈그려가며 계속 고치고, 수정하고 단순화하고
뭐 그렇게 합니다.
설계 그거 어차피 하다보면 다 바뀝니다.
끝없는 리펙토링이 있을뿐..

»
Login or register to post comments

Copyright © 2009 Nextree Soft, All rights reserved