

소프트웨어 프로젝트

스크럼 애자일 프로세스 모델

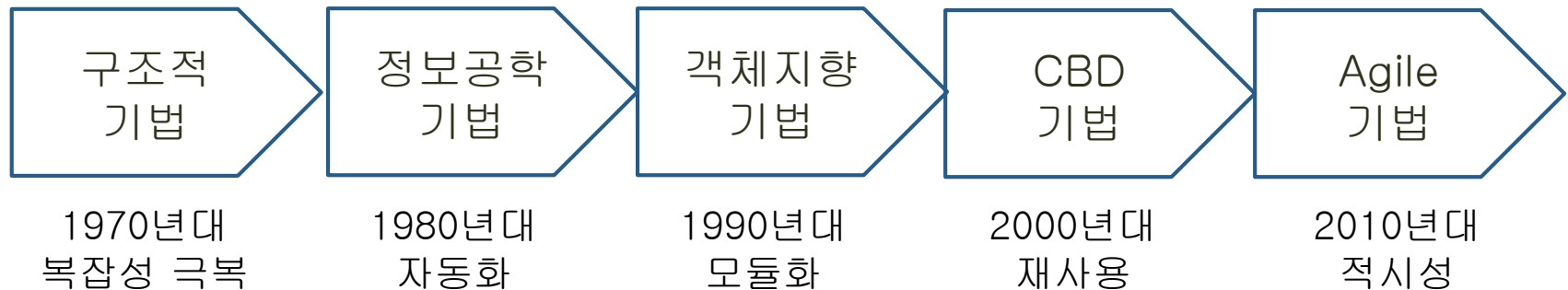
컴퓨터공학과

소프트웨어 개발 방법론

❖ 소프트웨어 개발 방법론 정의

- 소프트웨어 개발의 생명주기에 정형화된 방법 및 절차, 도구 등을 체계적으로 정리하여 표준화한 문서
- 실무적 관점에서 작업활동, 절차, 산출물, 기법 등을 정리

❖ 소프트웨어 개발 방법론의 진화



소프트웨어 개발 프로세스

- ❖ 주먹구구식 모델
- ❖ 폭포수 모델(선형 순차적 모델)
- ❖ V 모델
- ❖ 진화적 프로세스 모델 (프로토타입 모델)
- ❖ 나선형 모델
- ❖ 단계적 개발 모델
 - 점증적 개발 방법
 - 반복적 개발 방법

애자일 프로세스 모델

❖ 애자일 프로세스(agile process) 모델

- 애자일(agile)은 '날렵한', '민첩한' 등의 의미
- 애자일 프로세스 모델은 **고객의 요구에 민첩하게 대응하고 그때그때 주어지는 문제를 풀어나가는 방법론**
- 애자일 개발 방법은 반복적인 개발을 통한 잦은 출시를 목표
- 폭포수 모델처럼 무거운 방법론은 요구 사항의 변화에 유연하게 대처하기 어렵다는 큰 문제점이 있기 때문에 가볍고 비교적 변화를 수용하기 쉬운 방법론이 필요하게 됐는데 이것이 익스트림 프로그래밍(XP: eXtreme Programming), 스크럼(Scrum), 크리스털(Crystal)과 같은 방법론들
- 애자일은 가벼운 프로세스 방법론의 공통적인 특성을 정의하는 말
- 2001년 1월에 각 방법론의 전문가들이 모인 「애자일 연합(agile alliance)」 그룹에서 서로의 공통점을 찾아 “애자일 선언문”이라는 공동 선언서를 발표

애자일의 기본 가치

- ❖ 애자일 선언문은 다음과 같은 가치를 추구
 - 프로세스와 도구 중심이 아닌, 개개인과의 상호 소통을 중시한다.
 - 문서 중심이 아닌, 실행 가능한 소프트웨어를 중시한다.
 - 계약과 협상 중심이 아닌, 고객과의 협력을 중시한다.
 - 계획 중심이 아닌, 변화에 대한 민첩한 대응을 중시한다.
- ❖ 폭포수 모델이 계획, 프로세스, 프로젝트 관리, 산출물 같은 것에 중점을 둔다면, 애자일은 **고객과의 협업, 빠른 시간 안에 고객이 작동해볼 수 있는 소프트웨어, 환경과 고객의 변화에 능동적으로 대처**하는 것을 강조

애자일의 원칙

❖ 애자일 연합의 애자일 소프트웨어의 개발 원칙

- 최우선적인 목표는 고객을 만족시키기 위해 가치 있는 소프트웨어를 빨리, 지속적으로 제공하는 것이다.
- 개발 후반에 새로 추가되는 요구 사항도 기꺼이 받아들인다. 애자일 프로세스는 고객의 경쟁력을 위해 요구 사항의 변경을 받아들인다.
- 동작 가능한 소프트웨어를 짧으면 2주, 길면 2개월 간격으로 자주 고객에게 전달한다. 이때 간격은 짧을수록 좋다.
- 프로젝트가 진행되는 동안에 업무 담당자와 개발자는 매일 서로 의견을 주고받으며 함께 참여한다.
- 정보 전달을 위한 전화, 팩스, 인터넷 등 많은 매체 수단이 있으나 가장 효과적인 의사소통 방법은 역시 직접 만나 얼굴을 보면서 대화하는 것이다.
- 진척 사항의 척도를 나타내는 방법은 여러 도구로 표현할 수 있으나, 가장 좋은 방법은 실행 가능한 소프트웨어를 보여주는 것이다.
- 자율적 사고와 자유로운 분위기의 프로젝트 팀에서 최고의 아키텍처, 요구 사항, 설계가 나온다.
- 프로젝트의 효율성을 향상시키기 위해 개발 팀 스스로 정기적인 미팅을 진행하여 자신들의 행동을 되돌아보고 조율 및 수정한다.

애자일의 개발 방법

❖ 애자일 개발 방법

- 반복적인 개발을 통한 잦은 출시를 목표
- 실행 가능한 프로토타입을 만들어 사용자에게 확인받고, 좀 더 빠른 시간 안에 일부이지만 소프트웨어를 사용할 수 있게 하는 것을 중요하게 생각

애자일 개발 방법으로 문서 편집기 만드는 과정

우선 문서를 작성하는 데 가장 기본이 되는 기능만 1차 요구 사항으로 분석(문서 작성, 단순 편집, 저장, 기존 문서 불러오기 등)하고 이를 반복으로 나누어 개발한다. 그러면 사용자는 기본 기능을 빨리 확인하고 사용할 수 있다.

그 동안에 개발자는 2차 개발을 진행한다. 이때 좀 더 복잡한 편집 기능과 여러 기능을 추가하여 마찬가지로 이를 반복으로 나누어 개발한다.

애자일 방법론과 폭포수 모델의 비교

구분	애자일 방법론	폭포수 모델
추가 요구 사항의 수용	언제든지 추가 요구 사항을 수용할 수 있는 방법으로 설계	추가 요구 사항을 반영하기 어려운 구조
릴리스 시점	자주, 빨리 제품에 대한 프로토타입을 만들어 사용자에게 릴리스	최종 완성된 제품을 릴리스
시작 상태	시작 단계에서는 부족한 점이 많지만 점차 완성도가 높아짐	단계별 완성도를 최대한 높여 다음 단계로 넘어가기 위해 시작 단계에서의 완성도가 매우 높음
고객과의 의사소통	처음부터 사용자의 참여를 유도하고 많은 대화를 하면서 개발을 진행	요구 사항을 정의한 후 더 이상 추가 요구가 없다는 확답을 받고 개발에 들어감
진행 상황 점검	개발자와 사용자는 개발 초기부터 지속적으로 진행 상황을 공유	단계별 산출물에 대한 결과로 개발의 진척 상황을 점검
분석, 설계, 구현 진행 과정	분석, 설계, 구현이 하나의 단계와 그 단계 안의 반복마다 한꺼번에 진행	산출물 중심의 개발 방식이기 때문에 단계가 명확히 구별
모듈(컴포넌트) 통합	개발 초기부터 빈번한 통합을 통하여 문제점을 빨리 발견하고 수정하는 방식	구현이 완료된 후에 모듈 간의 통합 작업을 수행

애자일 프로세스 모델

❖ 애자일 개발 방법론

- 스크럼(Scrum)
- 익스트림 프로그래밍(XP: eXtreme Programming)
- 적응형 소프트웨어 개발 방법론(adaptive software development)
- 린 소프트웨어 개발 방법론(Lean software development)
- 크리스탈 패밀리(Crystal family)
- 기능 주도 개발 방법론(FDD: Feature Driven Development)
- 동적 시스템 개발 방법론(DSDM: Dynamic Systems Development Method)
- 애자일 UP(AUP: Agile Unified Process)

스크럼



❖ 스크럼(Scrum)

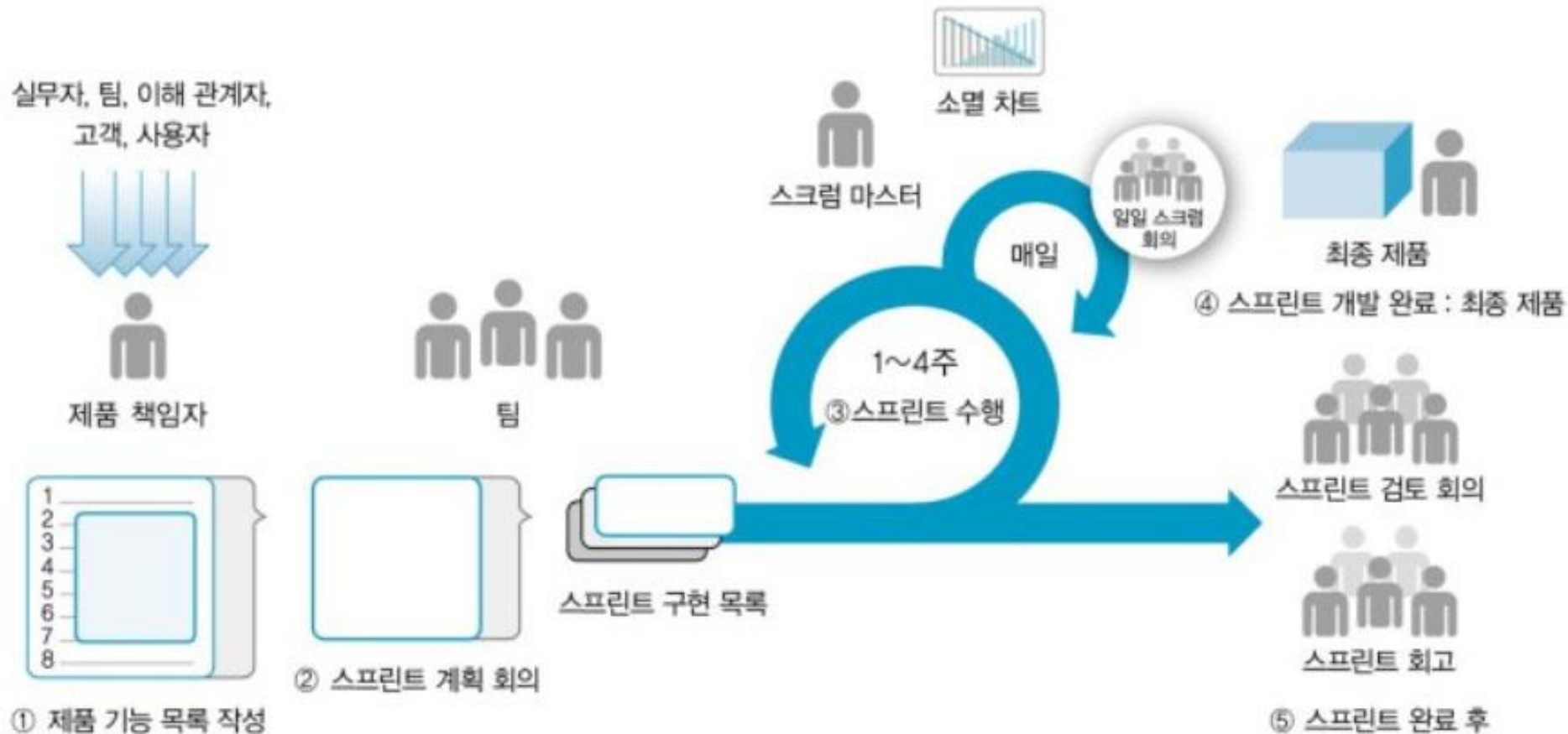
- 럭비 경기에서 쓰이던 용어인 스크럼이 소프트웨어 개발 프로세스에서 사용되는 것은 팀이라는 단어가 주는 의미를 개발 팀에 적용하여 효율적인 성과를 얻기 위해서임

스크럼

❖ 스크럼 개발 프로세스

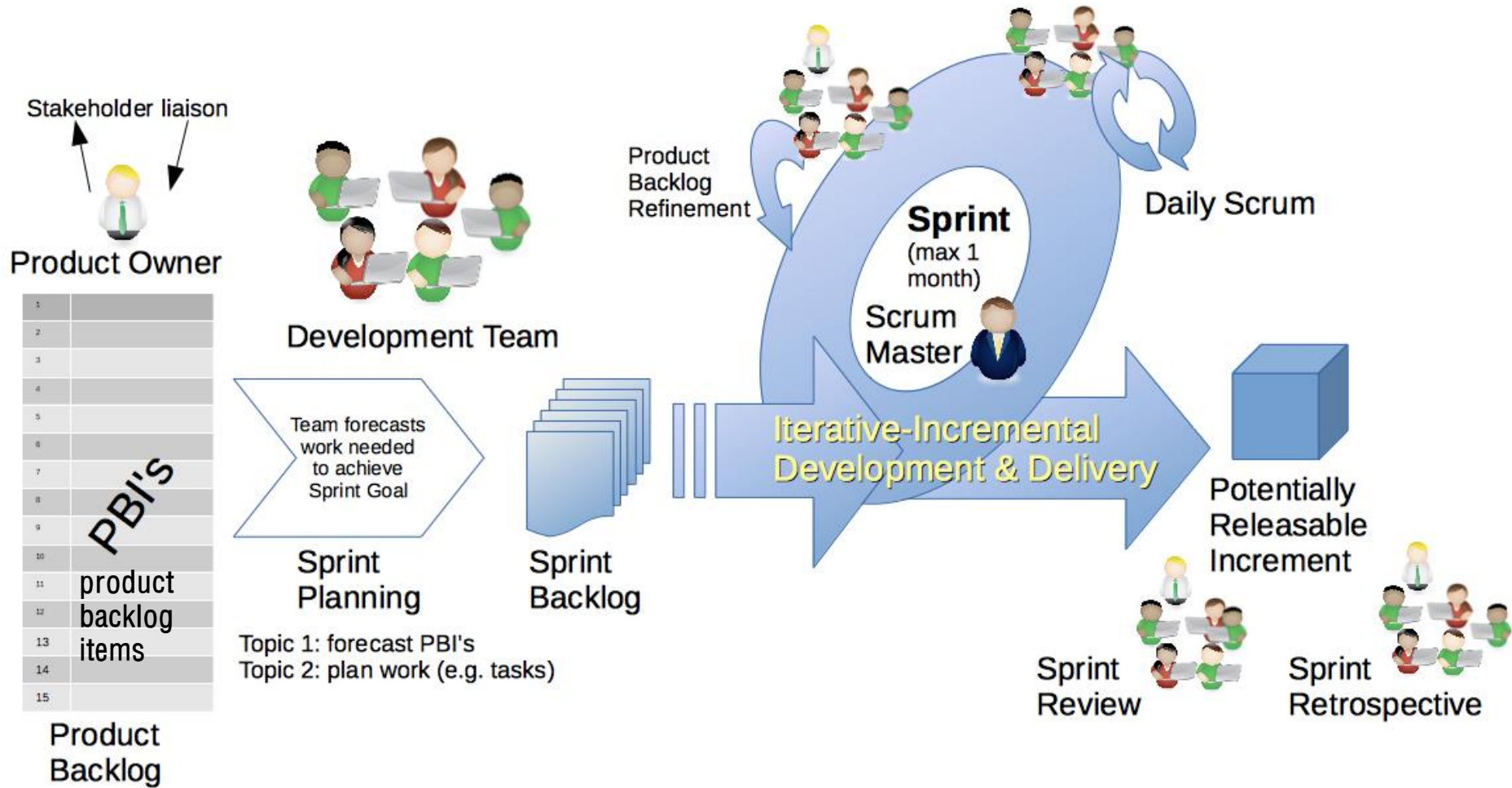
- 소프트웨어 개발보다는 팀의 개선과 프로젝트 관리를 위한 애자일 방법론
- 일종의 개발 팀(조직)을 운영하는 효율적인 운영 방식 또는 지침
- Scrum is an agile process framework for managing complex knowledge work, with an initial emphasis on software development.

스크럼 방식의 진행 과정



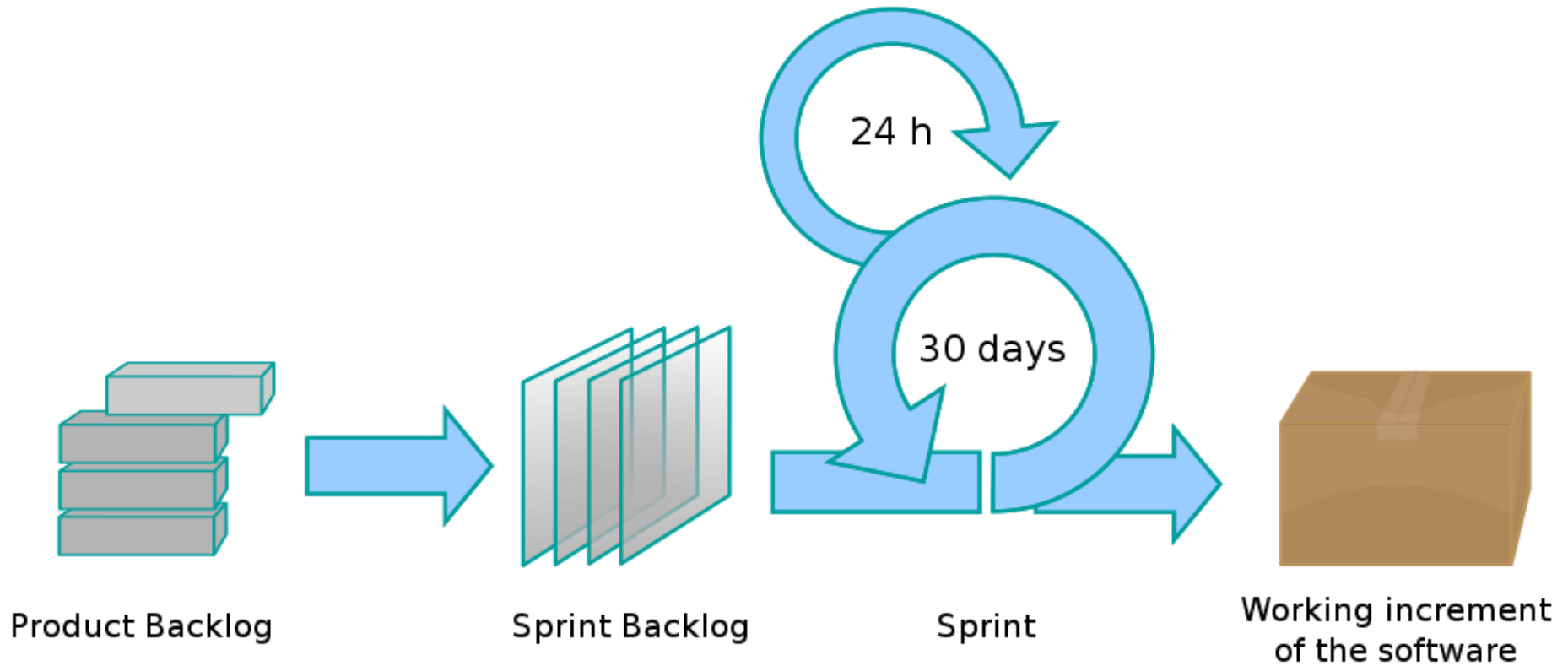
스크럼 방식의 진행 과정

스크럼 방식의 진행 과정



Scrum Framework

스크럼 방식의 진행 과정



The Scrum Process

제품 기능 목록 작성

❖ 제품 기능 목록(product backlog) 작성

- 제품 기능 목록은 우선순위가 매겨진 사용자가 요구하는 기능 목록
- 제품 책임자가 요구 사항 목록에 우선순위를 매겨 제품 기능 목록을 작성
- 한 번 결정된 제품 기능 목록은 확정된 것이 아니고 개발 중이라도 수정이 가능
- 제품 기능 목록 양식의 예

순위	요구 사항 목록	요구 사항 내역	작업 소요 기간	작업 월
1				
2				
3				
...				

사용자 스토리 작성

❖ 사용자 스토리(user story) 작성

- 사용자 스토리는 메모지 한 장에 쓰인 사용자 요구 사항
- 구현할 기능이 사용자 관점에서 사용자 언어로 작성



사용자 스토리

사용자 스토리 작성

- ❖ 사용자 스토리는 다음과 같이 설명할 수 있음
 - 제품 기능 목록에 정의된 사용자 관점에서의 기능임
 - 사용자에게 가치를 평가 받을 수 있도록 기능을 표현한 것임
 - 보통 작은 인덱스 카드를 사용해 필요한 것만 짧게 표현함
 - 고객의 요구 사항을 문서화한 것이라기보다는 표현했다고 보는 것이 적합함
 - 유스케이스보다 작은 규모임
 - 사용자 스토리는 반복을 마치면 사라지지만 유스케이스는 개발 기간 동안 지속됨
 - 사용자와 충분히 대화하여 세부 사항을 구체적으로 서술함
 - 테스트를 통해 스토리가 완료된 것을 확인함
 - 다른 스토리에 종속되지 않고 독립적이며, 협상 가능해야 함
 - 추정 및 측정 가능해야 함
 - 사용자 스토리는 스토리가 큰 것보다는 많은 것이 좋음
 - 테스트가 가능해야 좋은 사용자 스토리임

스토리 포인트 산정

❖ 스토리 포인트(story point) 산정

- 작성된 사용자 스토리는 스토리 보드에 나열
- 각각의 사용자 스토리는 우선순위가 정해져야 하고, 하나의 사용자 스토리에 대한 업무량을 파악하여 수행하는 데 걸리는 개발 기간도 산정
 - 이때 상대적인 개발 시간을 스토리 포인트라고 함
- 누가 사용자 스토리를 추출하고 스토리 포인트를 산정하는 걸까?
 - 프로젝트에 참여하는 구성원이 다 함께 진행하면 좋지만, 대체로 개발자들이 주도적인 역할
 - 또 우선순위를 매기기 위한 사용자 스토리의 중요도를 판단하는 일은 주로 업무 분석가가 함

스프린트

❖ 스프린트(sprint)

- 스프린트의 사전적 의미는 ‘전력 질주’
- 사전적 의미와 단거리 달리기(sprint race) 에서 느껴지는 것처럼 스프린트는 순발력이 좋고 출발 속도가 빠른 것
- 스프린트는 작업량으로 볼 때 그렇게 많지 않고, 개발 기간도 짧음
- 작은 단위의 개발 업무를 단기간 내에 전력 질주하여 개발한다는 뜻
- A sprint (also known as iteration or timebox) is the **basic unit of development** in Scrum. The sprint is a timeboxed effort.

스프린트

- 스크럼에서 반복 주기의 기간은 스프린트 계획 회의를 통해 결정하는데, 보통은 2~4주 정도로 수행
 - 요구 사항이 안정적이고, 개발 팀이 애자일 방법에 대해 지식과 경험이 풍부하다면 2주 정도의 짧은 기간을 스프린트 주기
 - 요구 사항의 변화가 많고, 개발 팀의 역량이 낮다면 4주 정도의 기간을 스프린트 주기
- 스프린트 주기가 결정되면, 개발 팀은 팀원의 역량에 맞게 스프린트에 배정된 작업을 중간에 멈추는 일이 없이 전력 질주해서 끝내야 함
 - 결정된 스프린트의 목표와 내용이 개발 도중에 바뀌지 않아야 하고, 그 누구도 팀원들의 동의 없이 바꿀 수 없다는 원칙을 지켜야 함
 - 이때 스크럼 마스터의 중요한 역할은 외부로부터 오는 개발 방해 요소를 차단하는 것
- 하나의 스프린트에 대한 개발이 완료되면 사용자에게 시연해 보이고 사용자는 피드백을 제공
- 계획된 일정 안에 개발을 마치지 못할 경우 하나의 스프린트는 개발 완료 여부와 관계없이 정해진 일정이 지나면 끝남

스프린트

■ 스프린트 구현 목록의 진척 관리 양식의 예

제품 기능 목록	세부 작업 항목(일)	1일	2일	3일	4일	5일	6일	7일	8일	9일	10일	...
...												
총 남은 시간												

- 스크럼의 한 주기가 끝나면 '실행 가능한 제품(shippable product)'이 만들어짐
 - 이 제품을 통해 목표가 완료되었는지 확인 가능
 - 제품의 완성도를 단순 기능 구현 정도로 할 것인지, 버그 하나 없이 완벽할 정도의 높은 수준으로 할 것인지는 회의를 통해서 결정

스프린트 구현 목록

❖ 스프린트 구현 목록(sprint backlog)

- 각각의 스프린트 주기에서 개발할 작업 목록
- 스프린트 계획 회의에서 결정하며, 이 작업 목록 하나하나가 개발 완료되면 스프린트 주기는 완성
- 작업 목록에는 세부적으로 어떤 것을 구현해야 하는지에 대한 세부 작업 항목과 작업자, 예상 작업 시간 등에 관한 정보를 작성

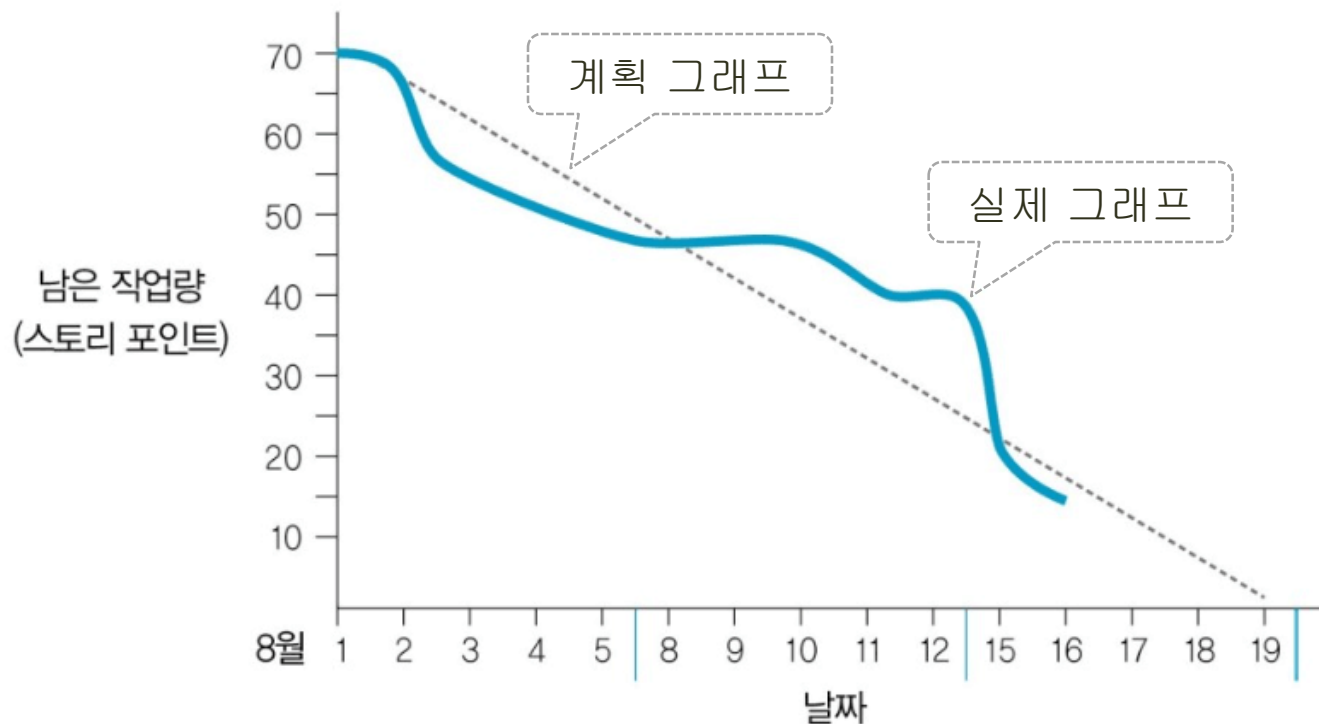
❖ 스프린트 구현 목록 양식의 예

작업 목록	세부 작업 항목	예상 작업 시간	작업자

소셜 차트

❖ 소셜 차트(burndown chart)

- 처음 계획된 작업 소진 그래프를 그려놓고, 시간이 지남에 따라 실제로 얼마나 작업량이 줄어드는지 비교할 수 있도록 개발이 완료되기까지 남은 작업량을 보여줌
- 계획 대비 작업이 어떻게 진행되고 있는지를 날짜별로 남은 작업량으로 나타냄



스프린트 계획 회의

❖ 스프린트 계획

- 각 스프린트에 대한 목표를 세우는 일과 제품 기능 목록에서 어떤 항목을 스프린트에서 진행할지 선택하는 것
- 선택된 각 항목에 대해 개발자를 배정하고 세부적인 작업 단위로 계획을 수립하는 것

❖ 스프린트 계획 회의(sprint planning meeting)

- 전체적인 스프린트 계획 회의
 - 스프린트 계획 회의 시작 단계에서는 제품 책임자를 통해 사용자가 원하는 것이 무엇인지를 파악하는 데 중점
 - 스크럼 마스터는 제품 기능 목록을 검토하면서 어떤 항목을 가장 높은 순위로 놓았는지에 관심을 갖고, 그 배경과 목표에 대해 팀원들과 토의하며 제품 책임자의 의도를 파악
- 세부적인 스프린트 계획 회의
 - 우선순위가 높은 항목을 어떻게 구현할 것인지 구체적인 작업 계획
 - 제품 기능 목록에서 개발 항목을 결정하고 스프린트 구현 목록을 작성
 - 팀원들은 정해진 작업을 수행하는 데 소요되는 시간을 추정

일일 스크럼 회의

- ❖ 일일 스크럼 회의(daily scrum meeting)는 스프린트 기간에 하는 회의로 다음과 같은 특징
 - 매일 한다.
 - 서서 한다.
 - 약속된 시간에 한다.
 - 짧게(15분 정도) 한다.
 - 진행 상황만 점검한다.
 - 스프린트 작업 목록을 잘 개발하고 있는지 확인한다.
 - 모든 팀원이 참석한다.
 - 한 사람씩 어제 한 일을 얘기한다.
 - 한 사람씩 오늘 할 일을 얘기한다.
 - 한 사람씩 문제점 및 어려운 점 정도만 얘기한다.
 - 매일 완료된 세부 작업 항목을 완료 상태로 옮겨 스프린트 현황판을 업데이트한다.
 - 개별 팀원에 대한 진척 상태를 확인한다.
 - 그날의 남은 작업량을 소멸 차트에 표시한다.

일일 스크럼 회의

- ❖ 일일 스크럼 회의에서 스크럼 마스터의 역할
 - 팀원들이 개발하는 데 방해되는 요소를 찾아 문제를 해결한다.
 - 개발자 개개인이 수행하고 있는 일의 진행 상황을 확인하다.
 - 소셜 차트에 그날의 남은 작업량을 표시한다.

스프린트 현황판

❖ 스프린트 현황판(task board)

- 개발 팀의 개발 현황(진척도, 남은 작업, 진행 속도)을 나타냄



스프린트 현황판의 예

최종 제품

❖ 최종 제품(finished work)

- 모든 스프린트 주기가 끝나면 제품 기능 목록에서 개발하려고 했던 제품이 완성

스프린트 검토 회의

❖ 스프린트 검토 회의(sprint review)

- 하나의 스프린트 반복 주기(2~4주)가 끝났을 때 생성되는 실행 가능한 제품에 대해 검토
- 스프린트 목표를 달성했는지 작업 진행과 결과물을 확인하고, 전체 흐름을 확인하여 비즈니스 가치를 점검하는 데 중점
- 스크럼 팀은 스프린트 동안 작업한 결과를 참석자(고객 포함)들에게 시연하고 요구 사항에 얼마나 부합하는지 검토
- 그리고 개선할 점 등에 관해 피드백을 받음
- 스크럼 마스터는 스프린트 동안 잘된 점, 아쉬운 점, 개선할 사항 등을 찾기 위한 회고를 진행할 수 있음
- 검토는 가능한 한 4시간 안에 마침

스프린트 회고

❖ 스프린트 회고(sprint retrospective)

- 그 동안 스프린트에서 수행한 활동과 개발한 것을 되돌아보고, 개선점은 없는지, 팀이 정한 규칙이나 표준을 잘 준수했는지 등을 검토하는 것
- 팀의 단점을 찾기보다는 강점을 찾아 더 극대화하는데 주안점을 둠
- 또한 문제점에 대한 해결 방안을 찾는 회의가 아니므로 문제점을 확인하고 기록하는 정도로만 진행
- 추정 속도와 실제 속도를 비교해보고, 차이가 크면 그 이유를 분석해봄

배포 목록

❖ 배포 목록(release backlog)

- 배포(release)는 사용자에게 시스템 일부를 제공하는 것
- 배포 목록은 제품 기능 목록의 항목 중에서 이번 배포 본에 포함하기로 결정한 항목을 말함
 - 배포 목록을 작성하면 이번 배포 본의 개발 범위와 일정을 수립할 수 있음

배포항목	내역	작업 기간	스프린트 주기
총 남은 시간			
배포 스프린트			
배포 날짜			

스크럼 방식의 진행 절차

❖ 단계별 스크럼 방식의 진행 절차

단계	수행 목록	내용
1	제품 기능 목록 작성	• 요구 사항 목록에 우선순위를 매겨 제품 기능 목록 작성
2	스프린트 계획 회의	• 스프린트 구현 목록 작성 • 스프린트 개발 시간 추정
3	스프린트 수행	• 스프린트 개발 • 일일 스크럼 회의 • 스프린트 현황판 변경 • 소멸 차트 표시
4	스프린트 개발 완료	• 실행 가능한 최종 제품 생산
5	스프린트 완료 후	• 스프린트 검토 회의 • 스프린트 회고

❖ 제품 책임자, 스크럼 마스터, 스크럼 팀의 역할

담당자	역할
제품 책임자 (product owner)	<ul style="list-style-type: none"> • 제품 기능 목록을 만들 • 비즈니스 관점에서 우선순위와 중요도를 매기고 새로운 항목을 추가함 • 스프린트 계획 수립 시까지만 역할을 수행하고, 스프린트가 시작되면 팀 운영에 관여하지 않음
스크럼 마스터 (scrum master)	<ul style="list-style-type: none"> • 제품 책임자를 돕는 조력자 • 업무를 배분만 하고, 일은 강요하지는 않음 • 스크럼 팀이 스스로 조직하고 관리하도록 지원함 • 개발 과정에서 스크럼의 원칙과 가치를 지키도록 지원함 • 개발 과정에 방해될 만한 요소를 찾아 제거함
스크럼 팀 (scrum team)	<ul style="list-style-type: none"> • 팀원은 보통 5~9명으로 구성되며, 사용자 요구 사항을 사용자 스토리로 도출하고 이를 구현함 • 기능을 작업 단위로 나누고, 일정이나 속도를 추정해서 제품 책임자에게 알려줌 • 하나의 스프린트에서 생산된 결과물을 제품 책임자에게 시연함 • 매일 스크럼 회의에 참여하여 진척 상황을 점검함

스크럼 방식의 장점과 단점

❖ 스크럼의 장점

- 반복 주기마다 생산되는 실행 가능한 제품을 통해 사용자와 충분히 의견을 나눌 수 있다.
- 일일 회의를 함으로써 팀원들 간에 신속한 협조와 조율이 가능하다.
- 일일 회의 시 직접 자신의 일정을 발표함으로써 업무에 집중할 수 있는 환경이 조성된다.
- 다른 개발 방법론들에 비해 단순하고 실천 지향적이다.
- 스크럼 마스터는 개발 팀원들이 목표 달성에 집중할 수 있도록 팀의 문제를 해결한다.
- 프로젝트의 진행 현황을 볼 수 있어, 신속하게 목표와 결과 추정이 가능하다.
- 프로젝트의 진행 현황을 볼 수 있어, 목표에 맞게 변화를 시도할 수 있다.

스크럼 방식의 장점과 단점

❖ 스크럼의 단점

- 추가 작업 시간 필요
 - 반복 주기가 끝날 때마다 실행 가능하거나 테스트할 수 있는 제품을 만들어야 한다. 이 작업이 많아지면 그만큼의 작업 시간이 더 필요하다.
- 일일 스크럼 회의를 15분 안에 마쳐야 함
 - 일일 스크럼 회의의 특성은 꼭 필요한 것만 매우 짧은 시간에 간략히 회의하는 것이다.
- 투입 공수 불측정에 따른 효율성 평가 불가
 - 투입 공수를 측정하지 않기 때문에 얼마나 효율적으로 수행되었는지는 알기 어렵다.
- 프로세스 품질 평가 불가
 - 스크럼은 프로젝트 관리에 무게 중심을 많이 둔 방법이다. 따라서 스프린트 수행 후 검토 회의를 하지만 프로세스 품질을 평가하지 않기 때문에 품질 관련 활동이 미약하고 따라서 품질의 정도를 알 수 없다.

Any Questions?

