

2주 3강

통합, 애자일 프로세스 모델



이번 주차에는...

소프트웨어 개발 프로세스

- 통합 프로세스 모델
- 애자일 프로세스 모델

1. 통합 프로세스 모델

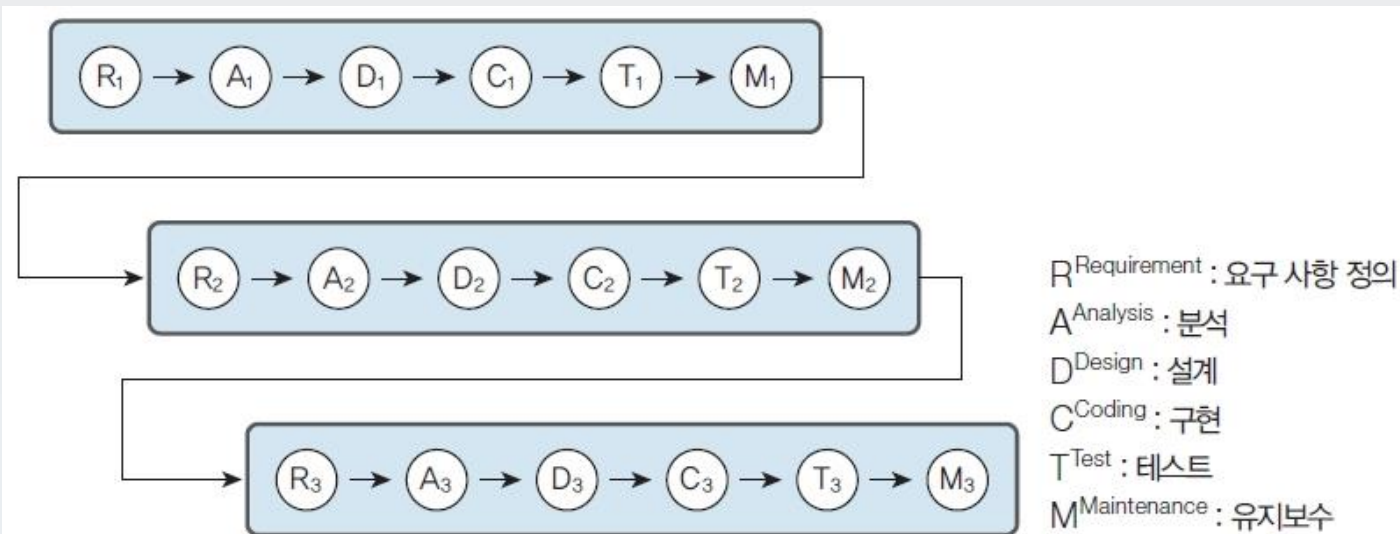


그림 2-17 반복적 개발 방법

2. 통합 프로세스(UP) 모델

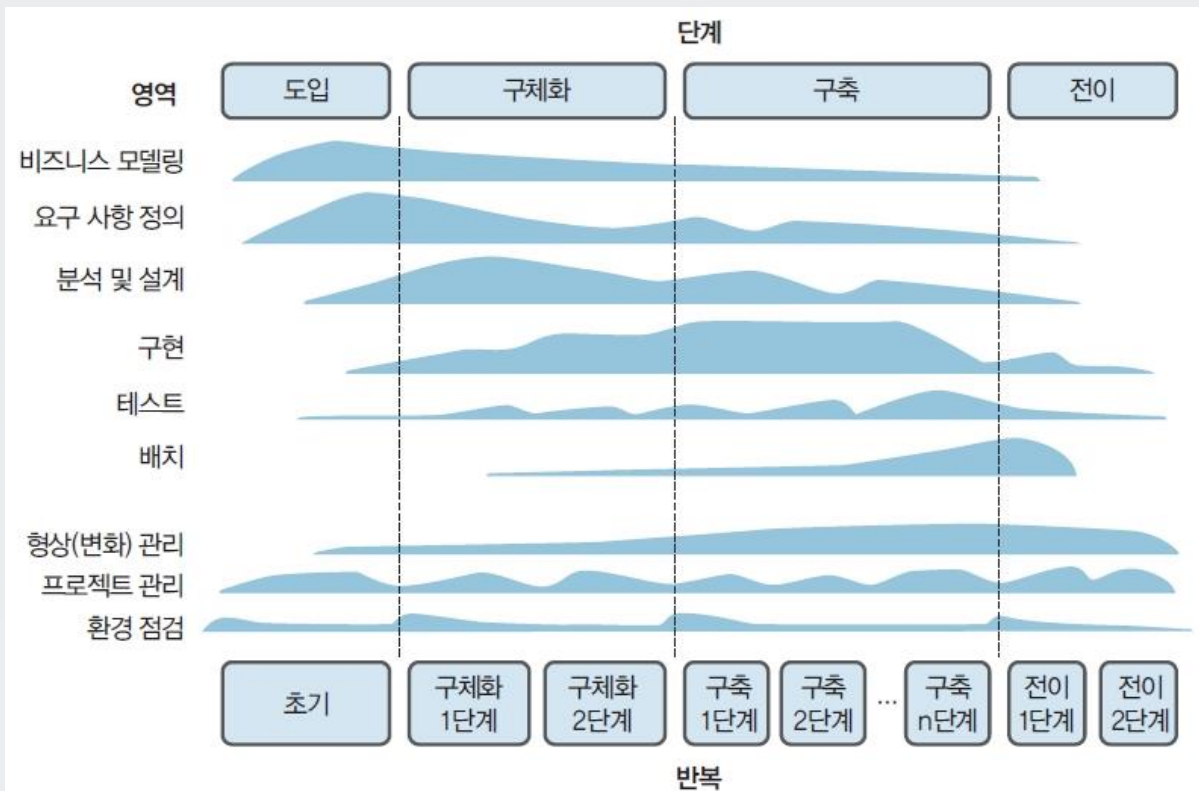
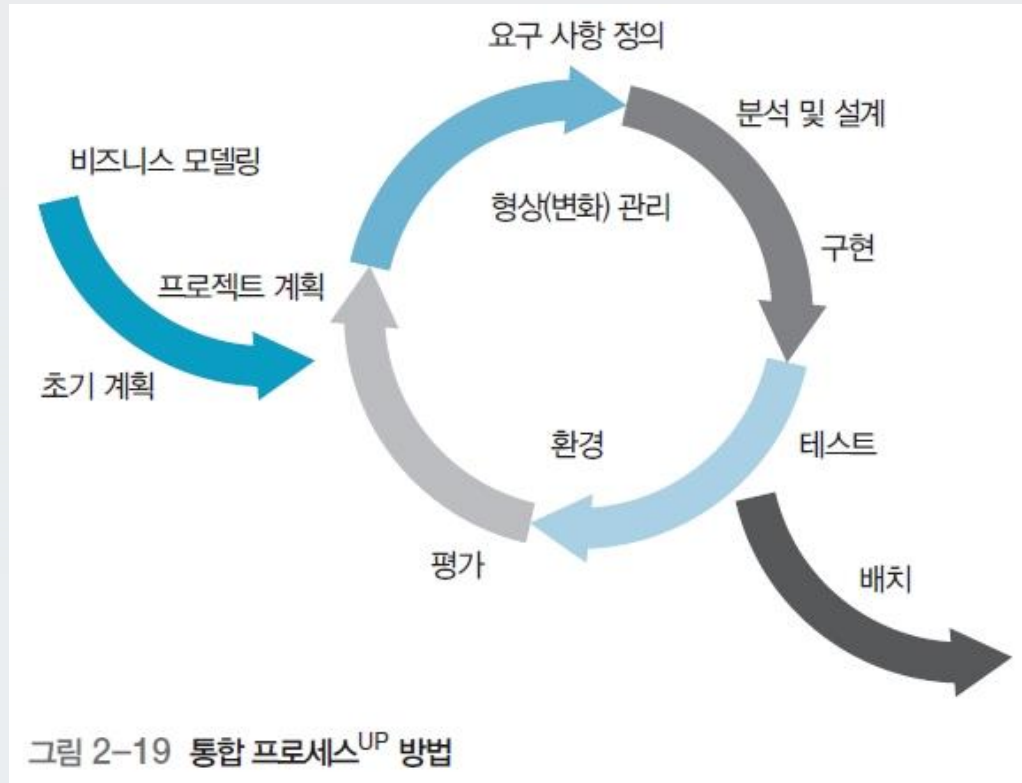


그림 2-18 통합 프로세스^{UP} 모델

3. 통합 프로세스(UP) 방법



4. 통합 프로세스(UP) 모델의 절차

- ① 도입 단계inception phase
- ② 구체화 단계elaboration phase
- ③ 구축 단계construction phase
- ④ 전이 단계transition phase
- ⑤ 도입/구체화/구축전이 단계의 공통 작업

5. 도입 단계(inception phase)

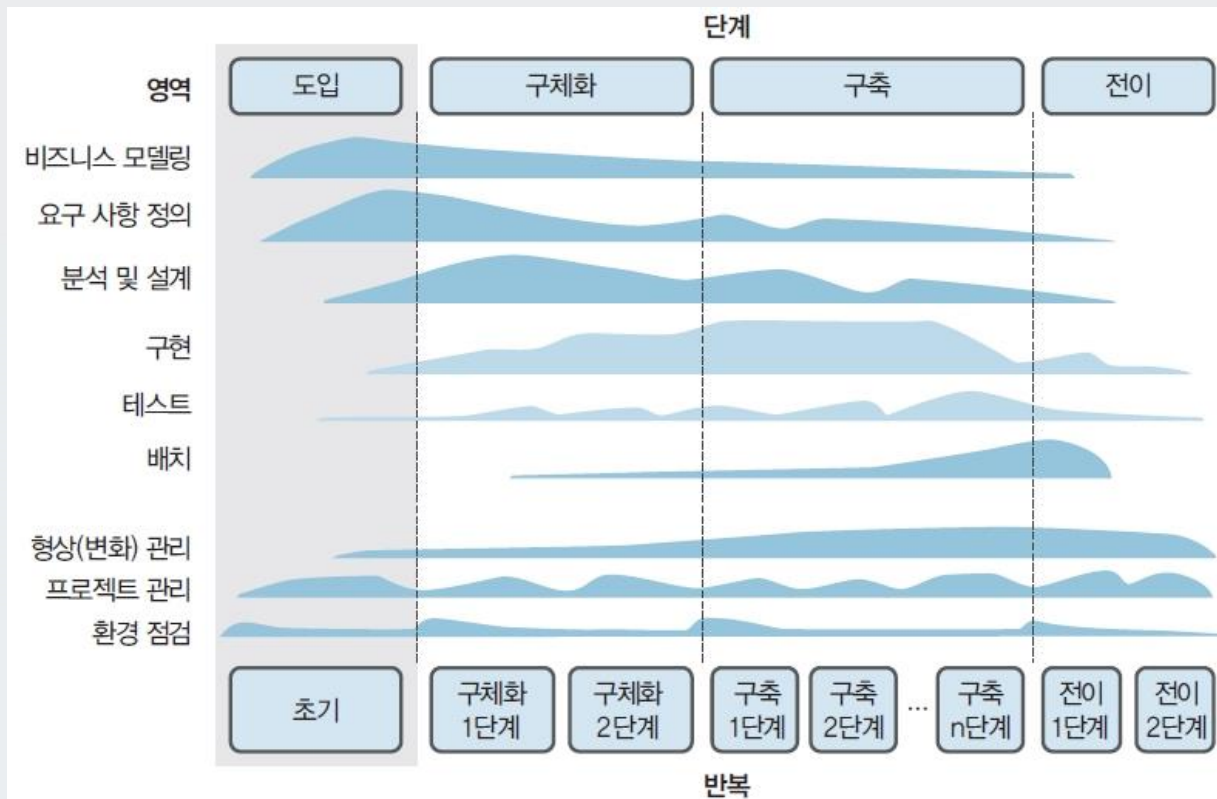


그림 2-20 통합 프로세스 모델의 도입 단계

6. 구체화 단계(elaboration phase)

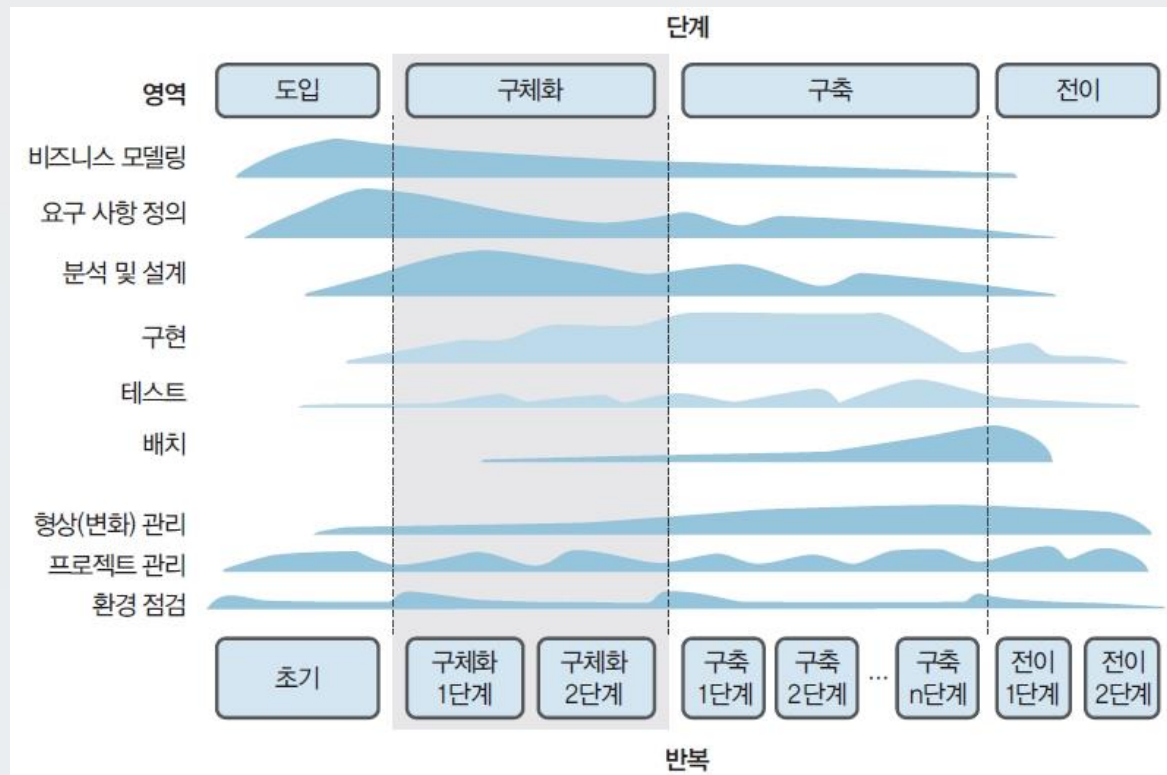


그림 2-21 통합 프로세스 모델의 구체화 단계

7. 구축 단계(construction phase)

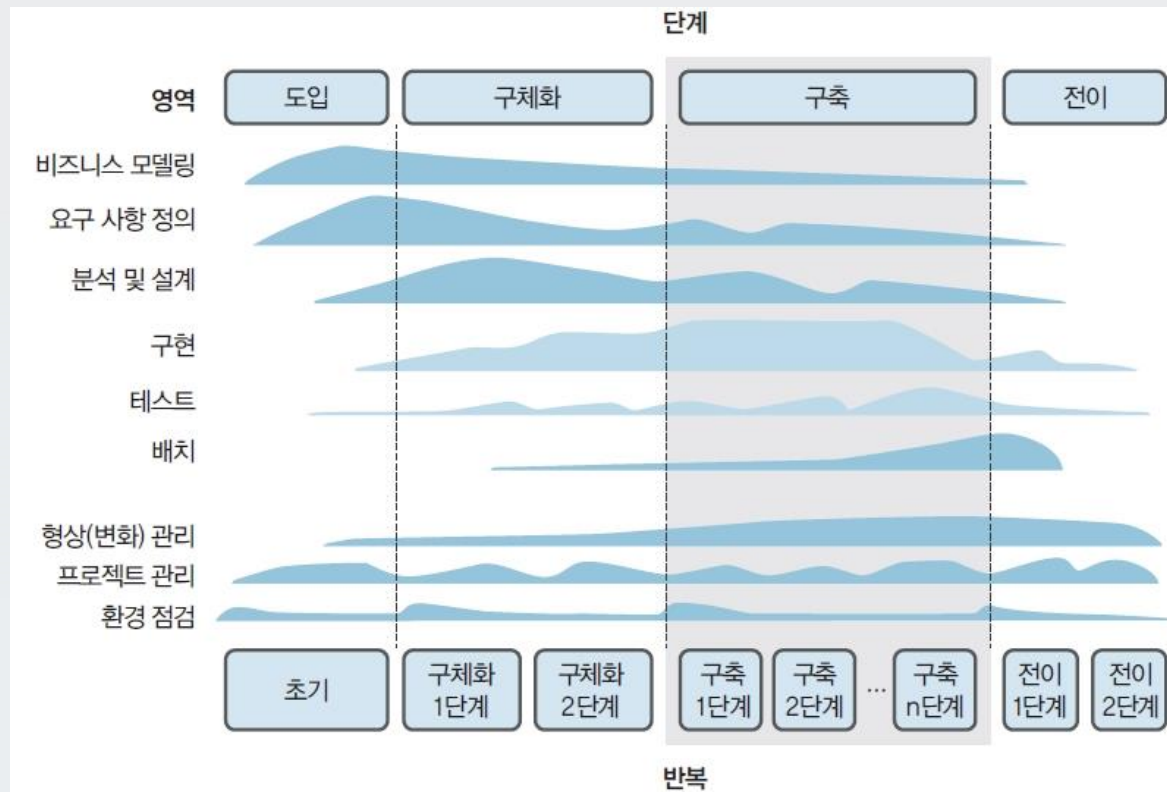


그림 2-22 통합 프로세스 모델의 구축 단계

8. 전이 단계(transition phase)

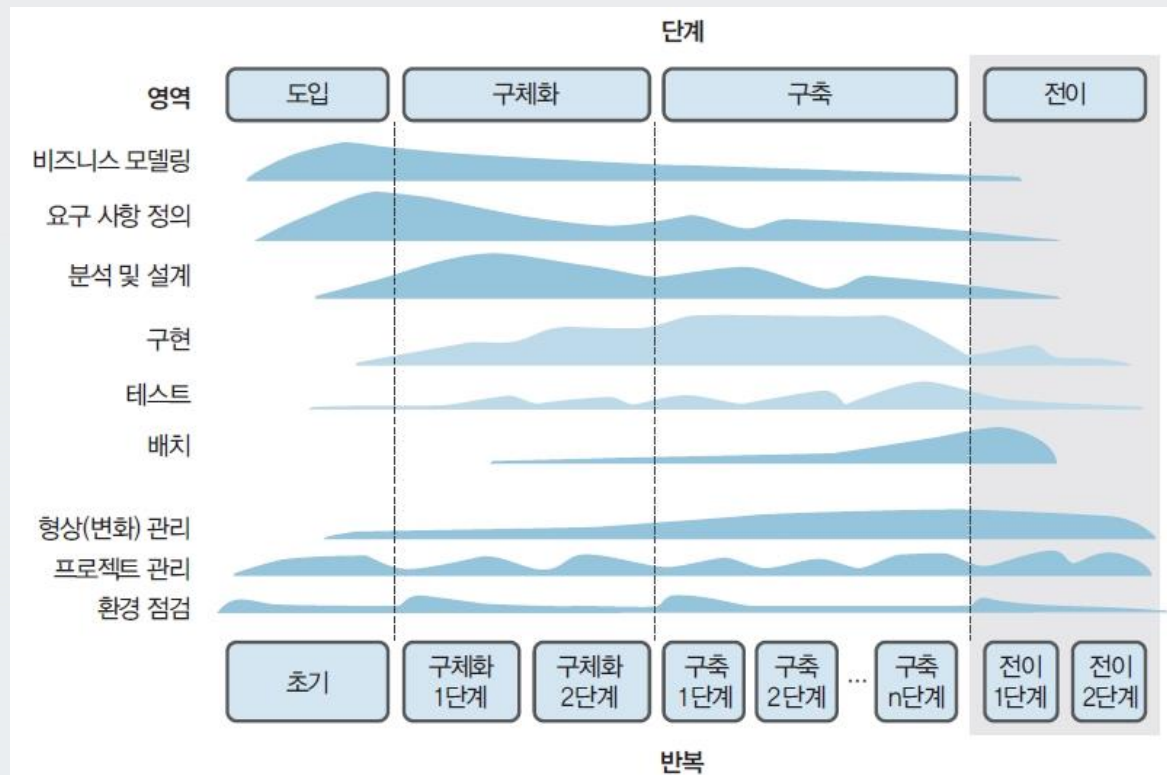


그림 2-23 통합 프로세스 모델의 전이 단계

9. 도입/구체화/구축전이 단계의 공통 작업

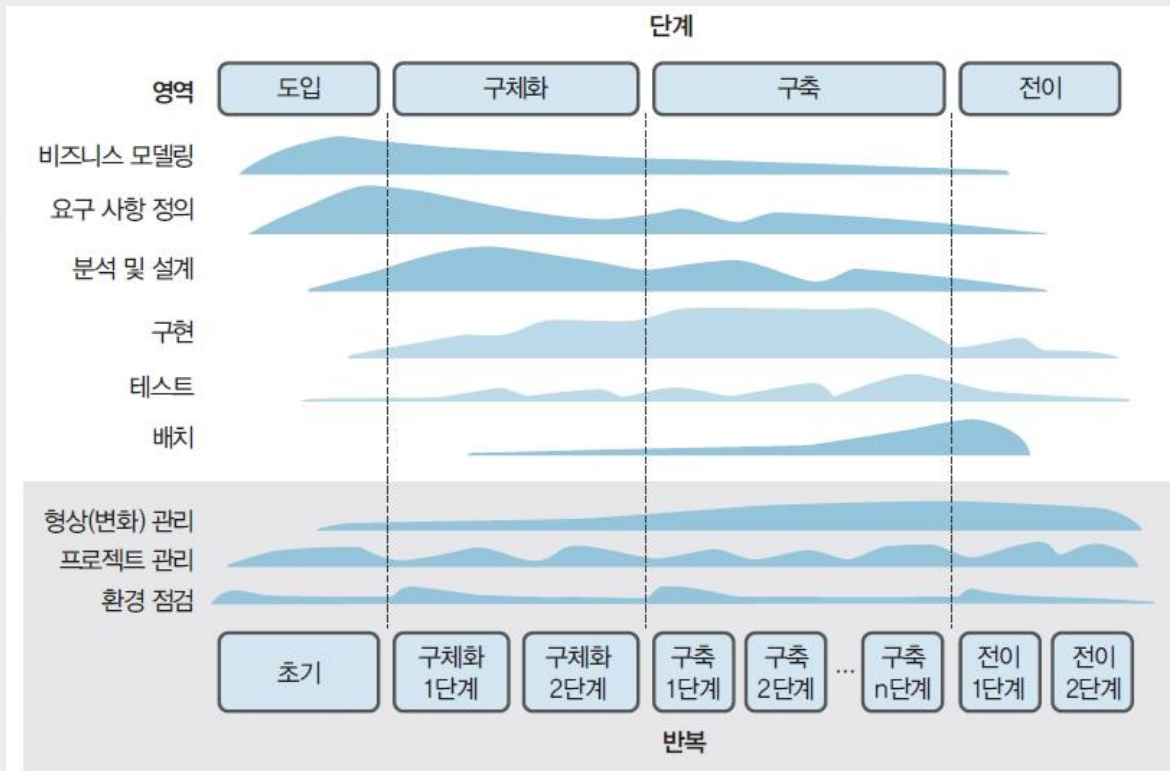


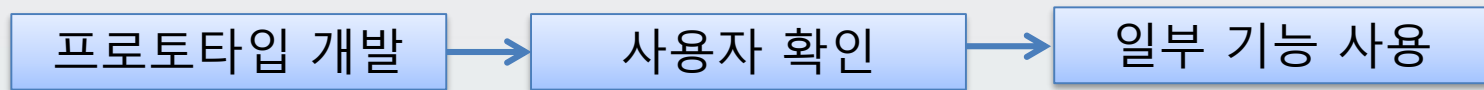
그림 2-24 통합 프로세스 모델의 공통 작업

10. 애자일 프로세스 모델

- 애자일(agile)
 - ‘날렵한’, ‘민첩한’
- 애자일 프로세스 모델
 - 고객의 요구에 민첩하게 대응하고 그때그때 주어지는 문제를 풀어나가는 방법론
- 애자일의 기본 가치(애자일 선언문)
 - 프로세스와 도구 중심이 아닌, 개개인과 상호 소통 중시
 - 문서 중심이 아닌, 실행 가능한 소프트웨어 중시
 - 계약과 협상 중심이 아닌, 고객과의 협력 중시
 - 계획 중심이 아닌, 변화에 대한 민첩한 대응 중시

11. 애자일 프로세스 모델의 이해

- 애자일의 개발 방법
 - 반복적인 개발을 통한 잦은 출시를 목표로 함



12. 애자일 방법과 폭포수 모델의 비교

구분	애자일 방법론	폭포수 모델
추가 요구 사항의 수용	추가 요구 사항을 수용할 수 있는 방법의 설계	추가 요구 사항을 반영하기 어려운 구조
릴리스 시점	수시로 릴리스	최종 완성된 제품을 릴리스
시작 상태	시작 단계는 미흡, 점차 완성도가 높아짐	시작 단계에서의 완성도가 매우 높음
고객과의 의사소통	처음부터 사용자의 참여 유도, 대화를 통한 개발 진행	사용자와 산출물의 근거 중심, 대화 부족
진행 상황 점검	개발자와 사용자는 개발 초기부터 진행 상황 공유	단계별 산출물에 대한 결과로 개발의 진척 상황을 점검
분석/설계/구현 진행 과정	하나의 단계 또는 반복 안에 분석/설계/구현 과정이 모두 포함되어 동시에 진행	분석/설계/구현 과정이 명확
모듈(컴포넌트)통합	개발 초기부터 빈번한 통합.문제점을 빨리 발견하고 수정하는 방식	구현이 완료된 후에 모듈 간의 통합 작업을 수행

13. 애자일 개발 방법론(스크럼)

■ 스크럼 개발 프로세스

- 소프트웨어 개발보다는 팀의 개선과 프로젝트 관리를 위한 애자일 방법론
- 경험적 관리 기법 중 하나
- 구체적인 프로세스를 명확하게 제시하지 않음
- 개발 팀(조직)을 운영하는 효율적인 운영 방식(지침)



그림 2-25 럭비 경기의 스크럼 대형

14. 스크럼 방식의 진행 과정

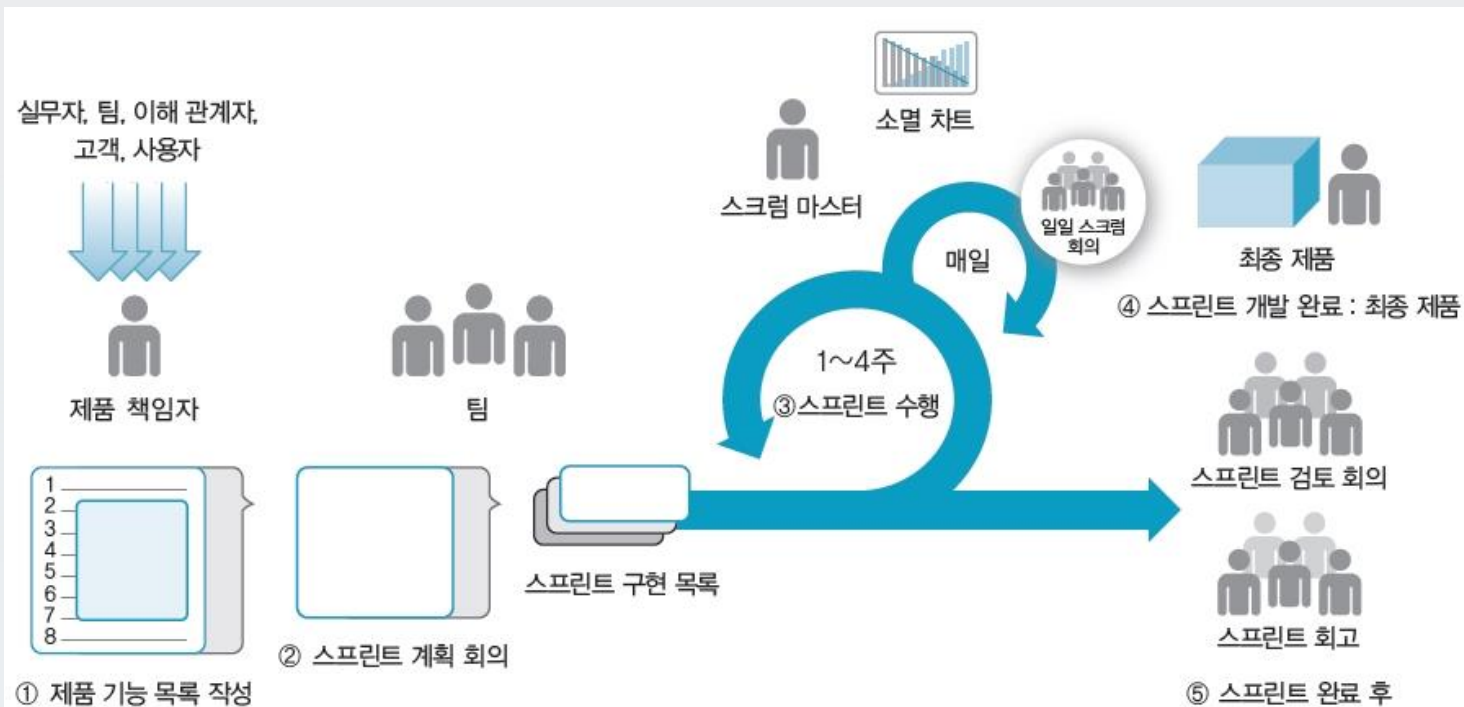


그림 2-26 스크럼 방식의 진행 과정

15. 스크럼 방식에서 사용되는 용어(1)

- 제품 기능 목록product backlog 작성
 - 우선순위가 매겨진 사용자의 요구 사항 목록

표 2-3 제품 기능 목록의 예 : 소프트웨어 공학 원고 목록

순위	요구 사항 목록	요구 사항 내역	작업 소요 기간	작업 월
1	품질(9장)	프로세스 품질과 제품 품질에 대해 기술한다.	30일	2015. 12.
2	테스트(8장)	테스트의 종류를 분류하고 단계별로 설명한다.	40일	2016. 1.
3	요구 분석(4장)	사용자 요구 사항을 정의하는 방법에 대해 기술한다.	20일	2016. 2.
...	30일
9	소프트웨어 개발 프로세스(2장)	개발 프로세스의 종류에 대해 기술한다.	20일	2016. 4.
10	소프트웨어 공학 소개(1장)	소프트웨어 공학의 일반적인 내용을 기술한다.	10일	2016. 5.

16. 스크럼 방식에서 사용되는 용어(2)

- 사용자 스토리user story작성 및 스토리 포인트story point 산정
 - 사용자 스토리
 - 메모지 한 장에 구현할 기능을 사용자 관점에서 사용자 언어로 작성한 사용자 요구 사항
 - 스토리 포인트
 - 사용자 스토리를 수행하는데 걸리는 상대적인 개발 기간(시간)



그림 2-27 사용자 스토리

17. 스크럼 방식에서 사용되는 용어(3)

■ 사용자 스토리 user story

- 제품 기능 목록에 정의된 사용자 관점에서의 기능
- 사용자에게 가치를 평가 받을 수 있도록 기능을 표현한 것
- 보통 작은 인덱스 카드를 사용해 필요한 것만 짧게 표현
- 고객의 요구 사항을 문서화한 것이라기보다는 표현했다고 보는 것이 적합
- 유스케이스보다 작은 규모
- 사용자 스토리는 반복을 마치면 사라지지만 유스케이스는 개발 기간 동안 지속
- 사용자와 충분히 대화하여 세부 사항을 구체적으로 서술
- 테스트를 통해 스토리가 완료된 것을 확인
- 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 스토리

18. 스크럼 방식에서 사용되는 용어(4)

- 스프린트sprint
 - ‘전력 질주’
 - 작업량으로 볼 때 그렇게 많지 않고, 개발 기간도 짧다.
 - 작은 단위의 개발 업무를 단기간 내에 전력 질주하여 개발한다는 뜻



그림 2-28 단거리 달리기 같은 스프린트

19. 스크럼 방식에서 사용되는 용어(5)

■ 스프린트의 예

계획: 소프트웨어 공학 원고 작성, 총기간(1년), 1장/(10일~40일)

- 스프린트 = 반복 주기 = 10일~40일

표 2-4 스프린트 구현 목록의 진척 관리

제품 기능 목록	세부 작업 항목(일)	1 일	2 일	3 일	4 일	5 일	6 일	7 일	8 일	9 일	10 일	11 일	12 일	13 일	14 일	15 일	16 일	17 일
품질(9장)	ISO/IEC 9126(5일)	1	1	1				1	1									
	ISO/IEC 14598(4일)				1	1	1						1					
	ISO/IEC 12119(3일)									1	1	1						
	CMMI(5일)													1	1	1	1	1
총 남은 시간		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

20. 스크럼 방식에서 사용되는 용어(6)

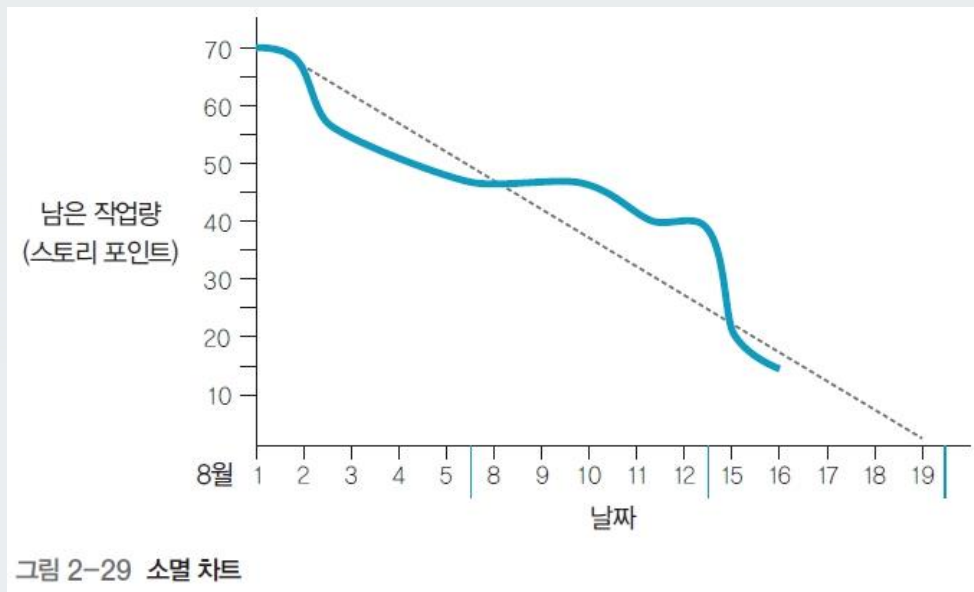
- 스프린트 구현 목록sprint backlog
 - 각각의 스프린트 주기에서 개발할 작업 목록
 - 세부 작업 항목과 작업자, 예상 작업 시간 등에 관한 정보를 작성

표 2-5 스프린트 구현 목록의 예

작업 목록	세부 작업 항목	예상 작업 시간	작업자
품질	제품 품질의 ISO/IEC 9126에 대해 기술한다.	5일	
	제품 품질의 ISO/IEC 14598에 대해 기술한다.	4일	
	제품 품질의 ISO/IEC 12119에 대해 기술한다.	3일	
	프로세스 품질의 CMMI에 대해 기술한다.	5일	
테스트	블랙박스 테스트에 대해 기술한다.	4일	
	화이트박스 테스트에 대해 기술한다.	4일	

21. 스크럼 방식에서 사용되는 용어(7)

- 소멸 차트burndown chart
 - 시간이 지남에 따라 소멸되고 남은 것을 표현
 - 계획 대비 작업이 어떻게 진행되고 있는지를 날짜별로 남은 작업량으로 표현



22. 스크럼 방식에서의 회의(1)

- 스프린트 계획 회의(sprint planning meeting)
 - 전체적인 스프린트 계획 회의
 - 가장 높은 순위의 항목에 관심
 - 그 배경과 목표에 대해 팀원들과 토의
 - 제품 책임자의 의도 파악
- 세부적인 스프린트 계획 회의
 - 우선순위가 높은 항목의 구현 방법에 대한 구체적인 작업 계획을 세움
 - 결정된 개발 항목에 대한 스프린트 구현 목록 작성
 - 정해진 작업 수행 소요 시간 추정

23. 스크럼 방식에서의 회의(2)

- 일일 스크럼 회의daily scrum meeting
 - 매일, 서서, 짧게(15분 정도) 함
 - 진행 상황만 점검하고, 스프린트 작업 목록을 잘 개발하고 있는지 확인
 - 모든 팀원이 참석하고, 한 사람씩 어제 한 일을 얘기
 - 한 사람씩 오늘 할 일과 문제점 및 어려운 점 정도만 얘기
 - 매일 완료된 세부 작업 항목을 완료 상태로 옮겨 스프린트 현황판 업데이트
 - 개별 팀원에 대한 진척 상태를 확인
 - 그날의 남은 작업량을 소멸 차트에 표시

24. 스크럼 방식에서의 회의(3)

- 스프린트 현황판task board

- 개발 팀의 개발 현황(진척도, 남은 작업, 진행 속도)을 나타냄



그림 2-30 스프린트 현황판의 예

- 최종 제품finished work

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

25. 스크럼 방식에서의 회의(4)

- 스프린트 검토회의sprint review
 - 하나의 스프린트 반복 주기(2~4주)가 끝났을 때 생성되는 실행 가능한 제품에 대해 검토
 - 스프린트 목표를 달성했는지 작업 진행과 결과물을 확인
 - 전체 흐름을 확인하여 비즈니스 가치를 점검
- 스프린트 회고sprint retrospective
 - 스프린트에서 수행한 활동과 개발한 것을 되돌아 봄
 - 개선점은 없는지, 팀이 정한 규칙이나 표준을 잘 준수했는지 등을 검토
 - 문제점을 확인하고 기록하는 정도로만 진행
 - 추정 속도와 실제 속도를 비교해보고, 차이가 크면 그 이유를 분석
 - 프로세스 품질은 측정하지 않음

26. 스크럼 방식에서의 회의(5)

- 배포 목록 release backlog

표 2-6 배포 목록

장	내역	작업 기간	스프린트 주기
1장	소프트웨어 공학의 개요	10일	스프린트 1
2장	소프트웨어 개발 프로세스	20일	스프린트 2
3장	계획	20일	스프린트 3
총 남은 시간		100일	
배포 스프린트			총 3스프린트
배포 날짜			2016년 8월 30일

27. 스크럼 방식의 진행 절차

표 2-7 스크럼 방식의 진행 절차

단계	수행 목록	내용
1	제품 기능 목록 작성	<ul style="list-style-type: none">• 요구 사항 목록에 우선순위를 매겨 제품 기능 목록 작성
2	스프린트 계획 회의	<ul style="list-style-type: none">• 스프린트 구현 목록 작성• 스프린트 개발 시간 추정
3	스프린트 수행	<ul style="list-style-type: none">• 스프린트 개발• 일일 스크럼 회의• 스프린트 현황판 변경• 소멸 차트 표시
4	스프린트 개발 완료	<ul style="list-style-type: none">• 실행 가능한 최종 제품 생산
5	스프린트 완료 후	<ul style="list-style-type: none">• 스프린트 검토 회의• 스프린트 회고• 두 번째 스프린트 계획 회의

28. 제품 책임자, 스크럼 마스터, 스크럼 팀의 역할

표 2-8 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.

29. 스크럼 방식의 장점

- 실행 가능한 제품을 통해 사용자와의 충분한 의견 조율 가능
- 일일 회의를 통한 팀원들 간의 신속한 협조와 조율 가능
- 일일 회의 시 직접 자신의 일정 발표를 통한 업무 집중 환경 조성
- 다른 개발 방법론들에 비해 단순하고 실천 지향적
- 팀의 문제를 해결할 수 있는 스크럼 마스터의 능력(역할)
- 프로젝트 진행 현황을 통한 신속하게 목표와 결과 추정 가능, 목표에 맞는 변화 시도 가능

30. 스크럼 방식의 단점

- 추가 작업 시간 필요
 - 반복 주기가 끝날 때마다 실행 가능하거나 테스트할 수 있는 제품을 만들어야 하기 때문
- 일일 스크럼 회의를 15분 안에 마쳐야 함
 - 길어지는 회의 시간으로 인한 작업의 방해
- 투입 공수 불 측정에 따른 효율성 평가 불가
 - 투입 공수 불 측정으로 인해 얼마나 효율적으로 수행되었는지 모름
- 프로세스 품질 평가 불가
 - 프로세스 품질 미 평가로 인한 품질 관련 활동이 미약하고 품질의 정도를 알 수 없음

3주차

소프트웨어공학 계획

