

Chapter 02

소프트웨어 개발 프로세스



쉽게 배우는
소프트웨어 공학

01 소프트웨어 개발 프로세스의 이해

02 소프트웨어 프로세스 모델의 이해

03 주먹구구식 모델

04 선형 순차적 모델

05 V 모델

06 진화적 프로세스 모델

07 나선형 모델

08 단계적 개발 모델

09 통합 프로세스 모델

10 애자일 프로세스 모델

요약

연습문제



Section 01 소프트웨어 개발 프로세스의 이해

1. 일상에서의 프로세스 의미

■ 프로세스

- 일을 처리하는 과정 또는 순서

(예 1) 공장에서 자동차, 세탁기 등이 조립되어 완제품이 되는 과정

(예 2) TV요리 프로에서 요리사가 맛있는 요리를 만드는 과정



그림 2-1 요리 프로세스

2. 프로세스의 정의

■ 프로세스

- 일이 처리되는 과정이나 공정

즉, 주어진 일을 해결하기 위한 목적으로 그 순서가 정해져 수행되는 일련의 절차

■ 프로세스를 따랐을 때의 효과의 예

- 요리 레시피 활용하면?
- 세탁기 사용설명서 활용하면?
- 화면 지시에 따라 OS 설치하면?

→ 목적 달성

3. 소프트웨어 개발 프로세스(2)

■ 프로세스의 목적

- 이전에 얻은 노하우를 전달 -> 시행착오 감소 -> 빠르게 적응
- guide 역할

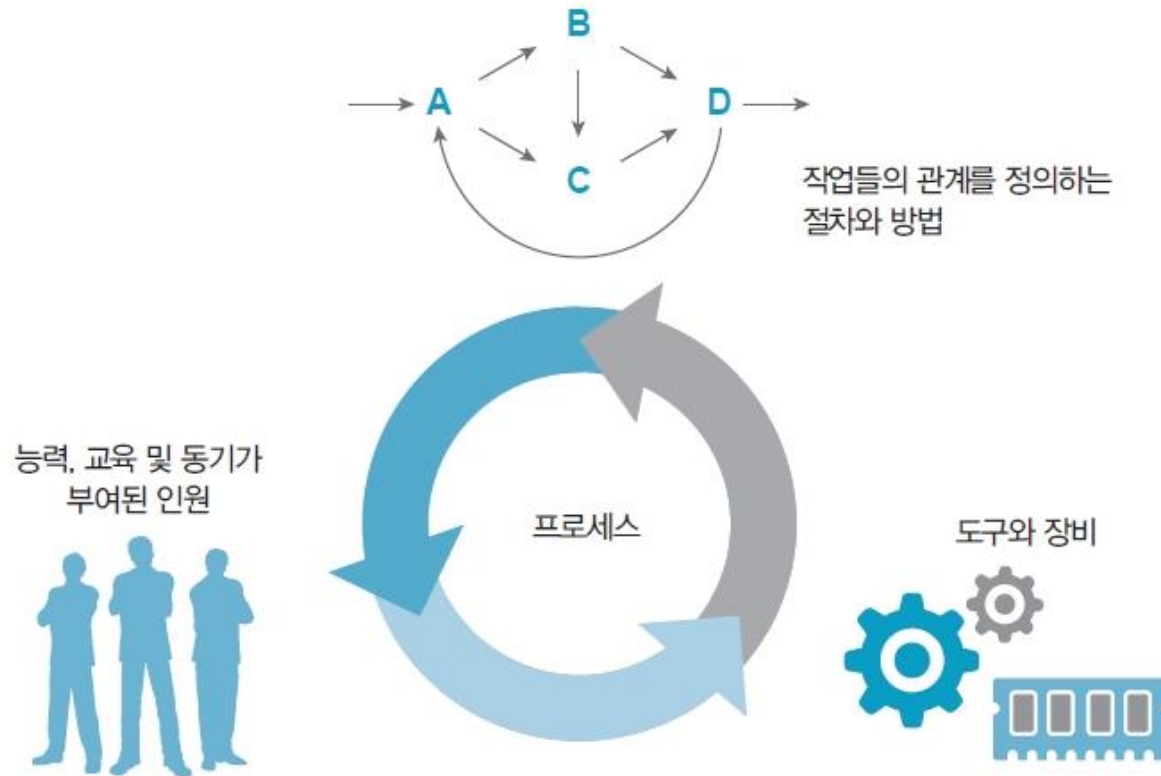


그림 2-2 프로세스의 3요소



Section 02 소프트웨어 프로세스 모델의 이해

1. 소프트웨어 개발 과정

■ 작은 규모의 소프트웨어 개발 과정

- 개집 짓는 일에 비유

■ 대규모의 소프트웨어 개발 과정

- 빌딩 짓는 일에 비유



그림 2-3 개집 설계와 빌딩 설계

2. 소프트웨어 프로세스 모델(1)

■ 소프트웨어 프로세스 모델의 정의

- 소프트웨어 개발 생명주기(SDLC Software Development Life Cycle)
- SW를 어떻게 개발할 것인가에 대한 전체적인 흐름을 체계화한 개념
- 개발 계획 수립부터 최종 폐기 때까지의 전 과정을 다룸
- 순차적인 단계로 이루어 짐

■ 소프트웨어 프로세스 모델의 목적

- 공장에서 제품을 생산하듯이 소프트웨어 개발의 전 과정을 하나의 프로세스로 정의
- 주어진 예산과 자원으로 개발하고 관리하는 방법을 구체적으로 정의
- 고품질의 소프트웨어 제품 생산을 목적으로 함



Section 03 주먹구구식 모델

1. 주먹구구식 모델

- Build and fix 모델, code and fix 모델, 즉흥적 소프트웨어 개발 모델
- 주먹구구식
 - 주먹으로 구구셈을 따지던 방법에서 유래한 말
 - 정확한 앞뒤 계산 없이 일을 대충 처리할 때 쓰는 말
- 소프트웨어 개발에서의 주먹구구식 모델
 - 공식적인 가이드라인이나 프로세스가 없는 개발 방식
 - 요구 분석 명세서나 설계 단계 없이 간단한 기능만을 정리하여 개발하는 형태
 - 일단 코드를 작성하여 제품을 만들어본 후에 요구 분석, 설계, 유지보수에 대해 생각

2. 주먹구구식 모델의 개발 단계

- ① 첫 번째 버전의 코드를 작성하여 제품을 완성한다.
- ② 작성된 코드에 문제점이 있으면 수정하여 해결한다.
- ③ 문제가 없으면 사용한다.

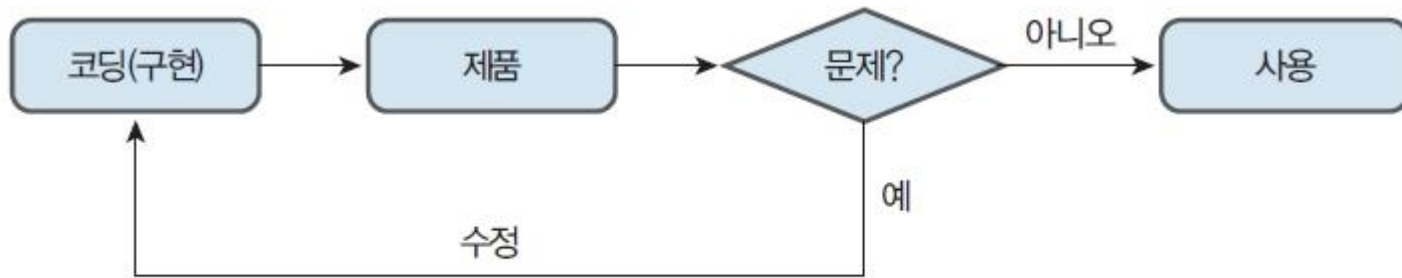


그림 2-4 주먹구구식 모델

3. 주먹구구식 모델의 사용 및 단점

■ 주먹구구식 모델의 사용

- 개발자 한 명이 단시간에 마칠 수 있는 경우에 적합
- 대학 수업의 한 학기용 프로젝트 정도

■ 주먹구구식 모델의 단점

- 정해진 개발 순서나 각 단계별로 문서화된 산출물이 없어 관리 및 유지보수가 어렵다.
- 프로젝트 전체 범위를 알 수 없을 뿐더러 좋은 아키텍처를 만들 수도 없다.
- 일을 효과적으로 나눠 개발할 수도 없으며, 프로젝트 진척 상황을 파악할 수 없다.
- 계속적 수정으로 인해 프로그램의 구조가 나빠져 수정이 매우 어려워진다.



Section 04 선형 순차적 모델

1. 선형 순차적 모델

- Linear sequential 모델, waterfall 모델, Classic life cycle



그림 2-5 폭포수 모델

3. 폭포수 모델의 장점

- 관리의 용이
- 체계적인 문서화
- 요구사항의 변화가 적은 프로젝트에 적합

4. 폭포수 모델의 단점

- 각 단계는 앞 단계가 완료되어야 수행할 수 있다.
- 각 단계의 결과물이 완벽한 수준으로 작성되어야 다음 단계에 오류를 넘겨주지 않는다.
- 사용자가 중간에 가시적인 결과를 볼 수 없어 답답해할 수 있다.



그림 2-6 폭포수 모델의 단점



Section 05 V 모델

1. V 모델

- 폭포수 모델 + 테스트 단계 추가 확장
- 산출물 중심(폭포수 모델) vs 각 개발 단계를 검증하는 데 초점(V 모델)

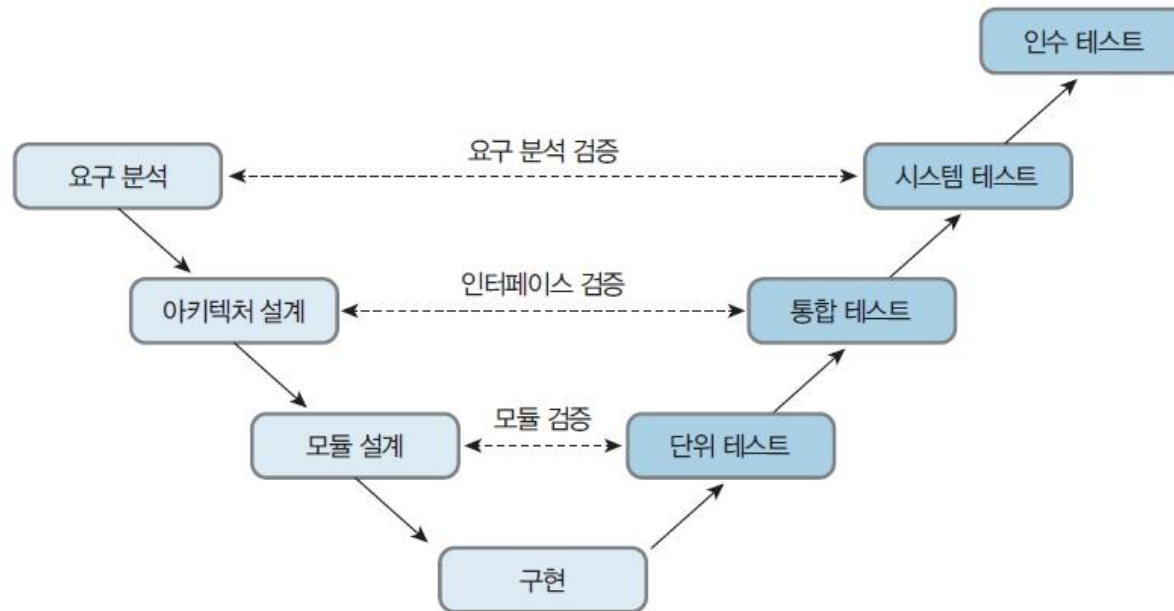


그림 2-7 V 모델



Section 06 진화적 프로세스 모델

1. 진화적 프로세스 모델의 등장 배경

- 선형순차적 모델의 대표: 폭포수 모델
- 진화적 프로세스 모델의 대표: 프로토타입 모델

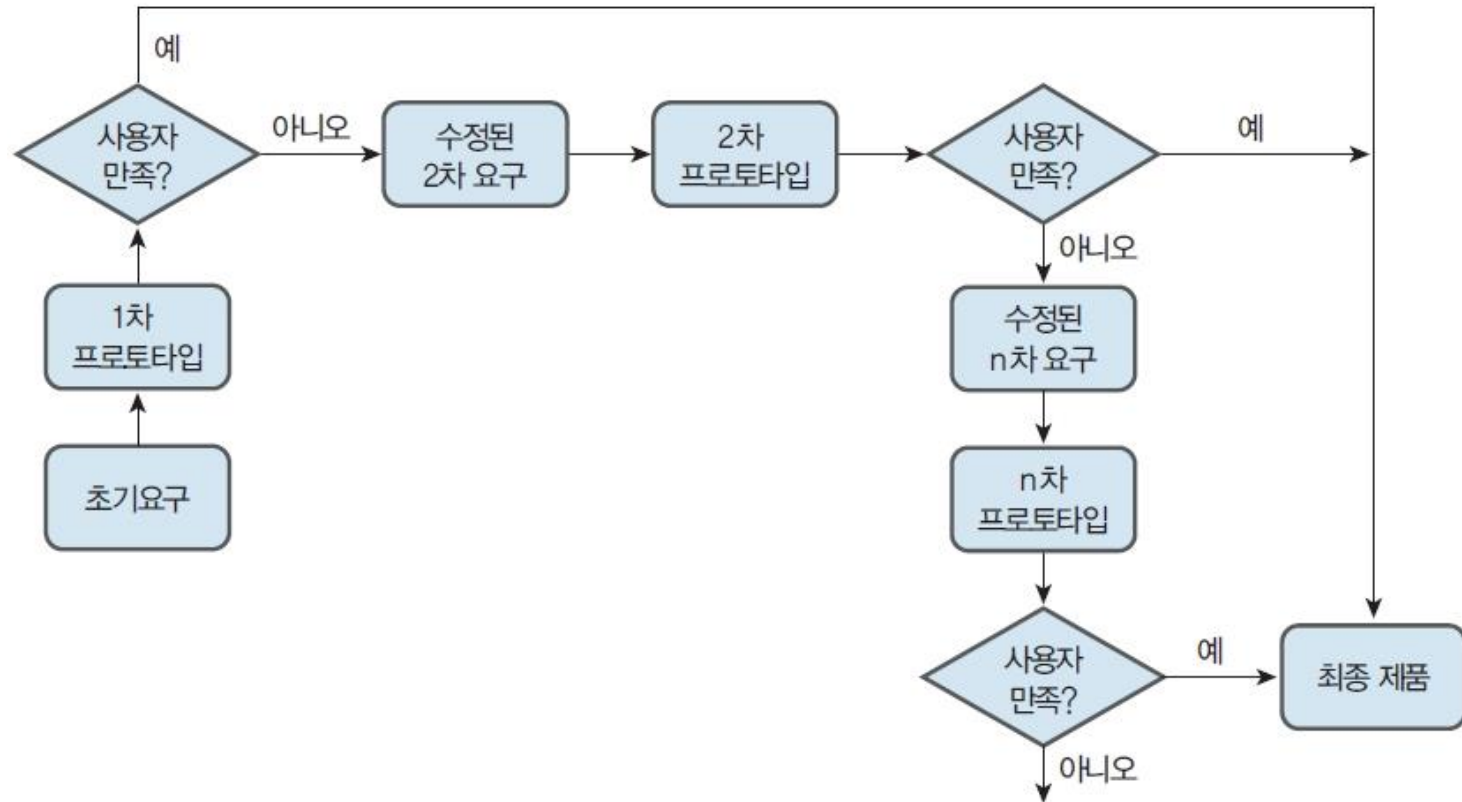


그림 2-8 진화적 프로세스 모델

1-1 프로토타입

■ 프로토타입

- 대량 생산에 앞서 미리 제작해보는 원형 또는 시제품으로, 제작물의 모형

■ 소프트웨어 개발에서의 프로토타입

- 정식 절차에 따라 완전한 소프트웨어를 만들기 전에 사용자의 요구를 받아 일단 모형을 만들고 이 모형을 사용자와 의사소통 하는 도구로 활용



그림 2-9 프로토타입의 예 : 아파트 모델하우스

1-2 프로토타입 모델의 개발 생명주기

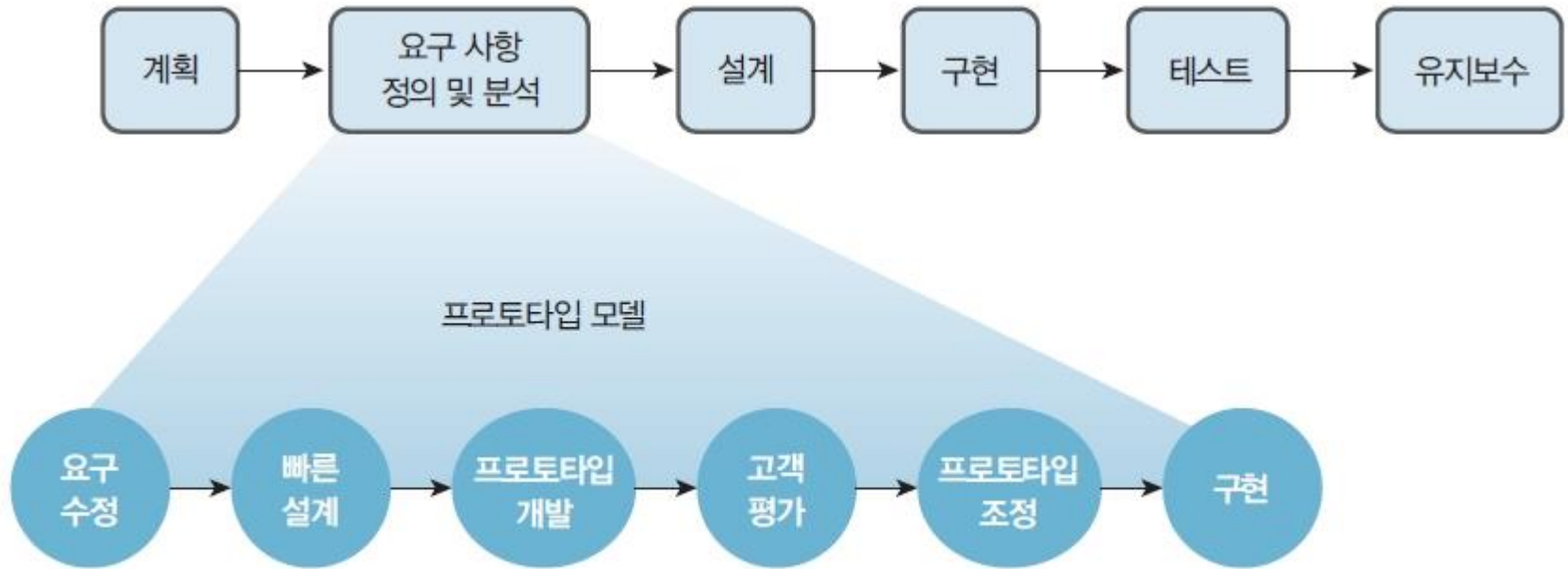


그림 2-10 프로토타입 모델의 개발 생명주기

2. 실험적 프로토타입 모델

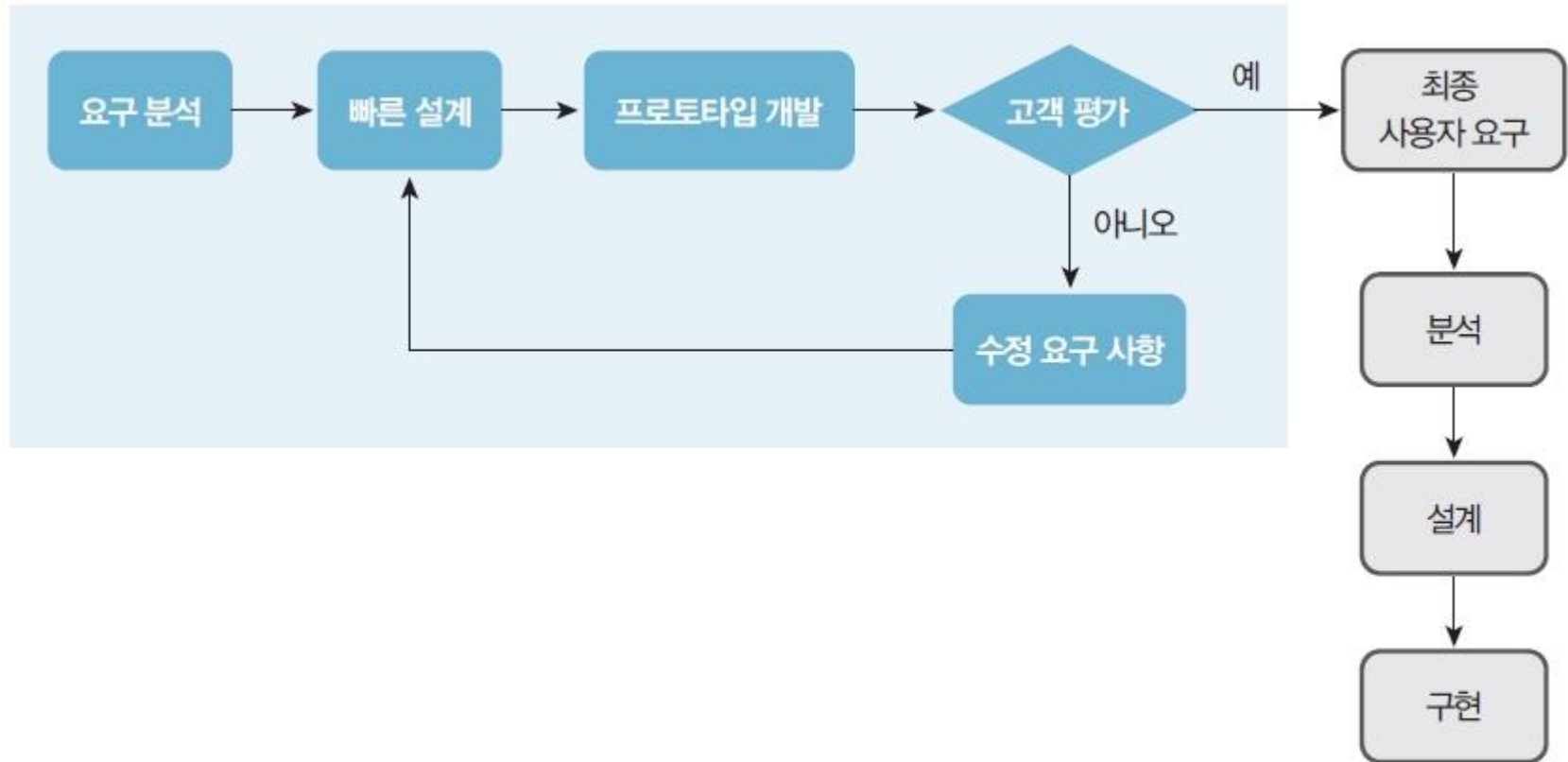


그림 2-11 실험적 프로토타입 모델 절차

3. 진화적 프로토타입 모델

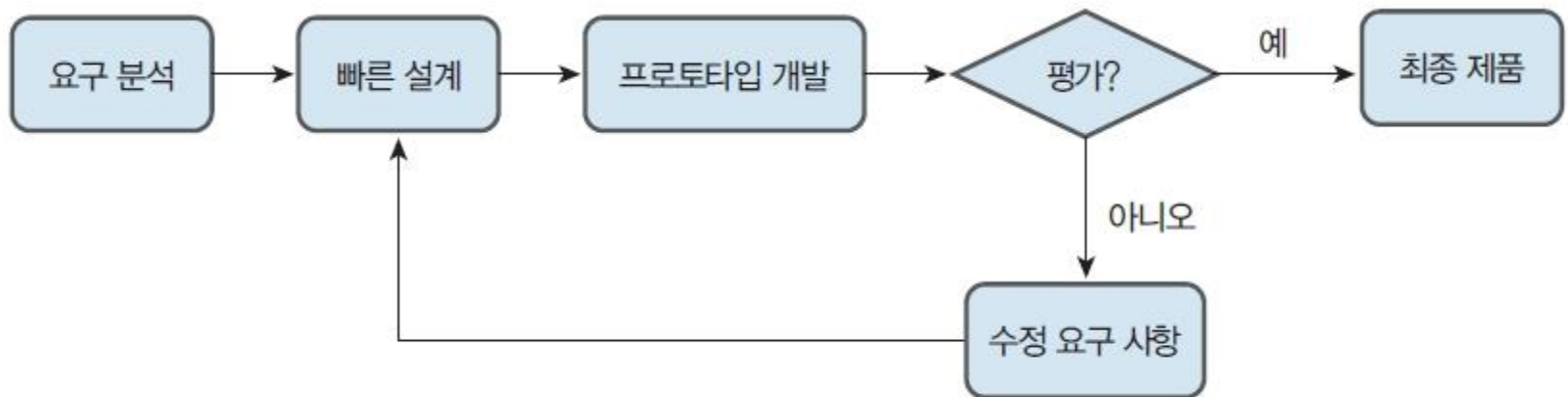


그림 2-12 진화적 프로토타입 모델 절차

4. 프로토타입 모델의 개발절차(1)

① 요구사항 정의 및 분석

- 1차 개략적인 요구 사항 정의 후 2차, 3차, ... n차를 반복하면서 최종 프로토타입 개발

② 프로토타입 설계

- 완전한 설계 대신, 사용자와 대화할 수 있는 수준의 설계
- 입출력 화면을 통한 사용자 인터페이스 중심 설계

③ 프로토타입 개발

- 완전히 동작하는 완제품을 개발하는 것이 아님
- 입력 화면을 통한 사용자의 요구 항목 확인
- 출력 결과를 통해 사용자가 원하는 것인지 확인



Section 09 통합 프로세스 모델

1. 통합 프로세스 모델

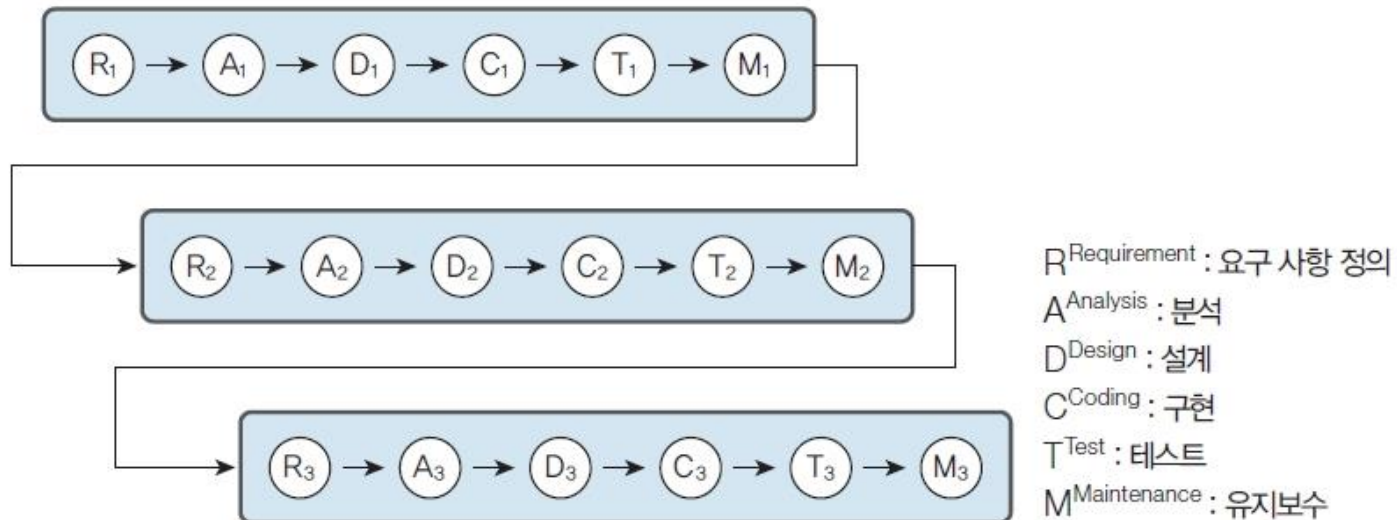
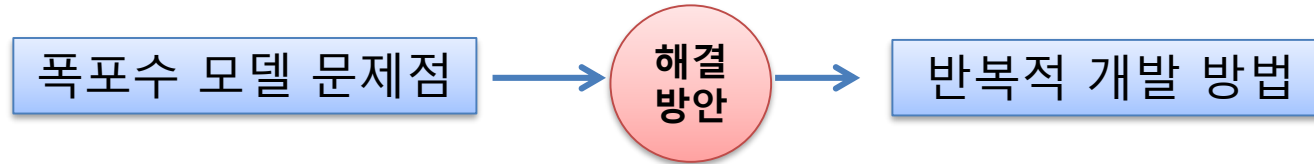


그림 2-17 반복적 개발 방법

2. 통합 프로세스(UP) 모델

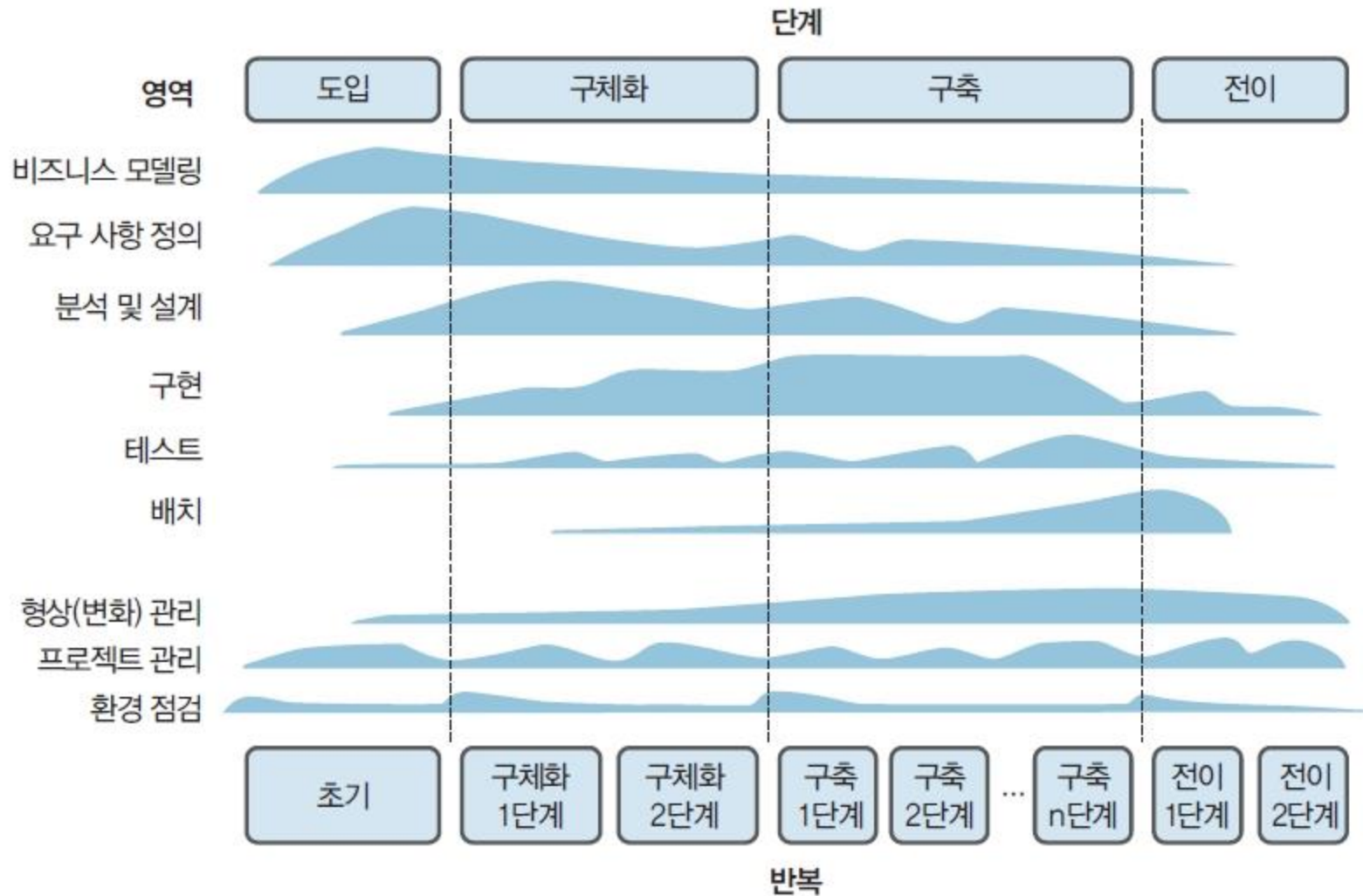


그림 2-18 통합 프로세스^{UP} 모델

3. 통합 프로세스(UP) 방법

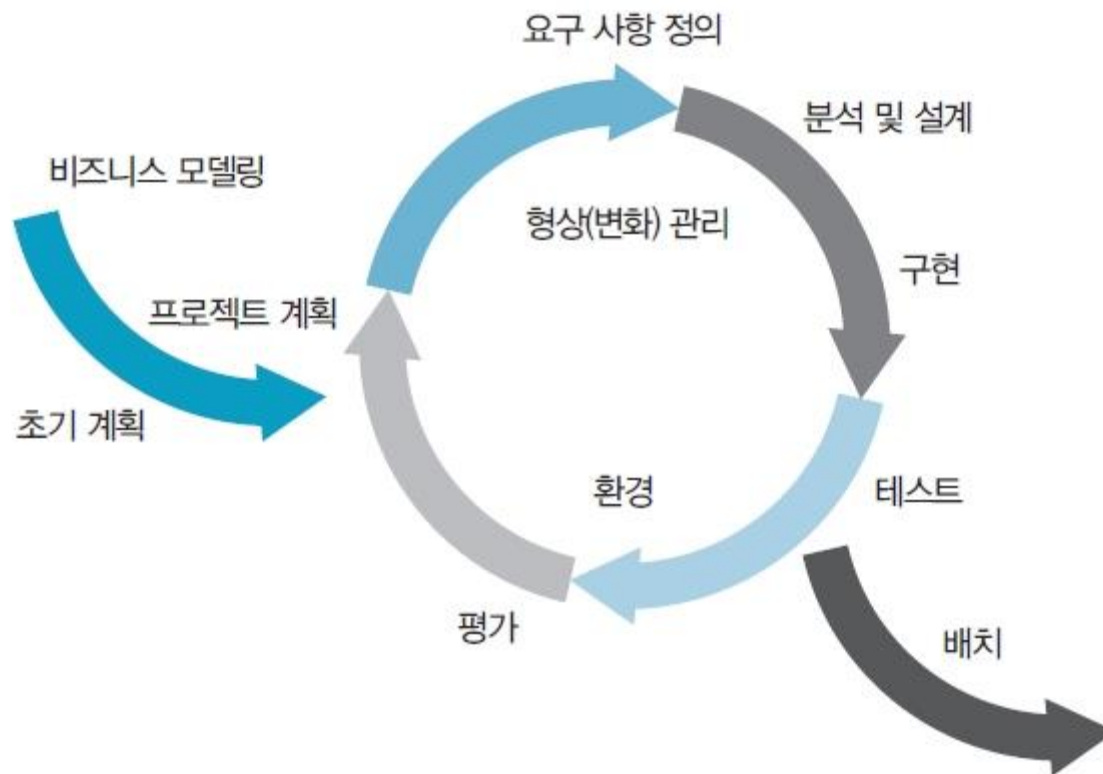


그림 2-19 통합 프로세스^{UP} 방법

4. 통합 프로세스^{UP} 모델의 절차

- ① 도입 단계 inception phase
- ② 구체화 단계 elaboration phase
- ③ 구축 단계 construction phase
- ④ 전이 단계 transition phase
- ⑤ 도입/구체화/구축전이 단계의 공통 작업

① 도입 단계inception phase

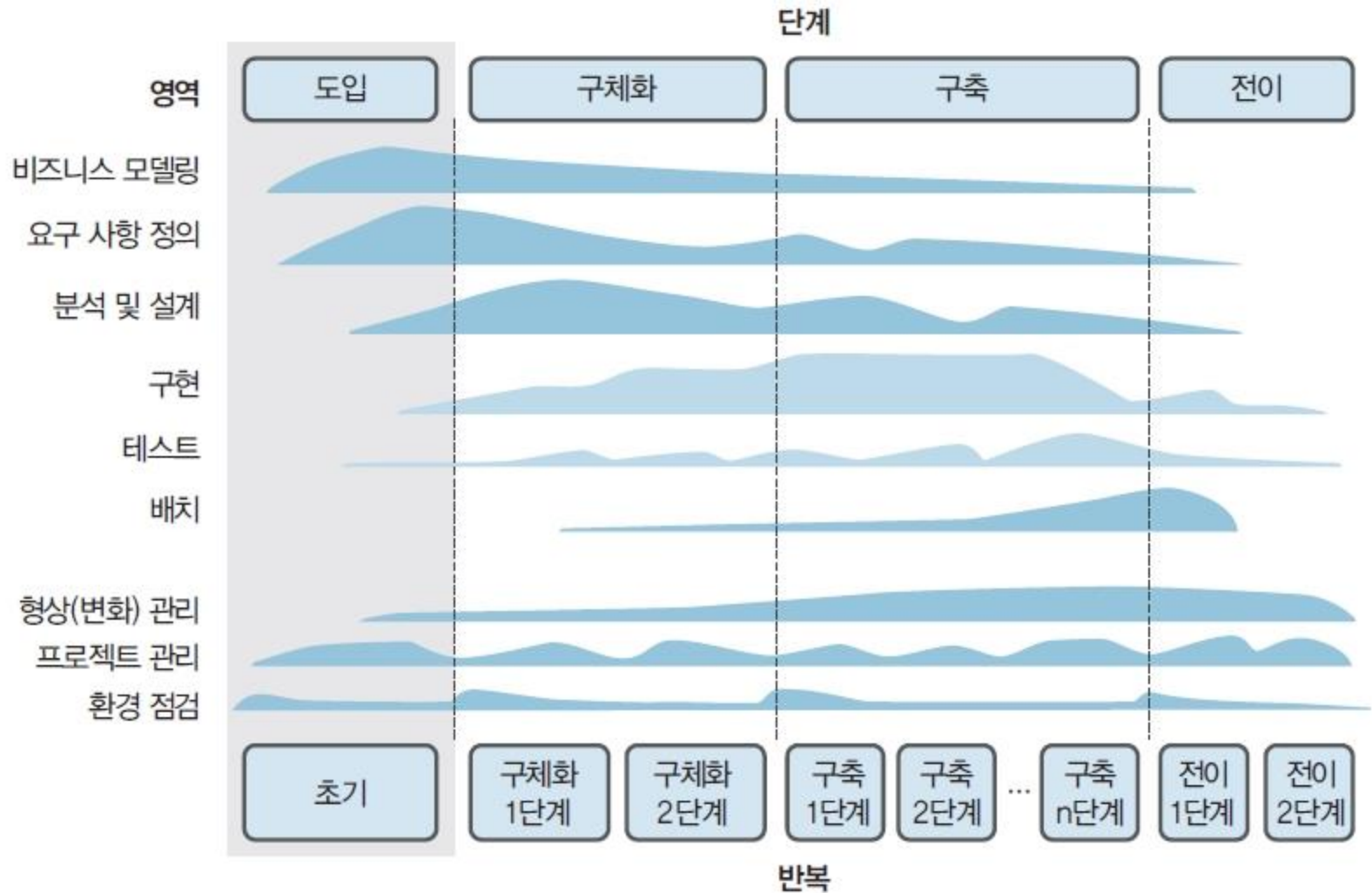


그림 2-20 통합 프로세스 모델의 도입 단계

② 구체화 단계elaboration phase

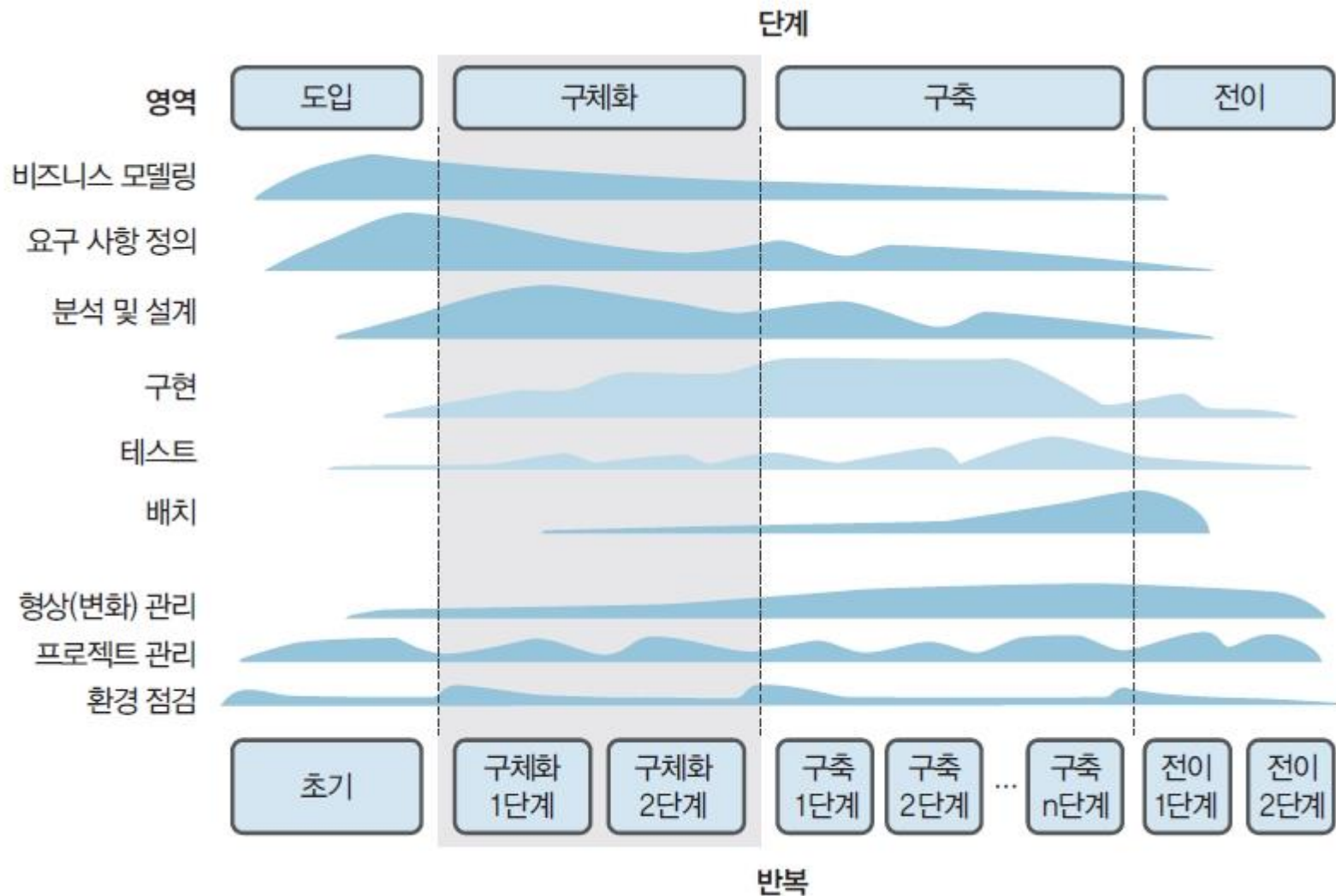


그림 2-21 통합 프로세스 모델의 구체화 단계

③ 구축 단계 construction phase

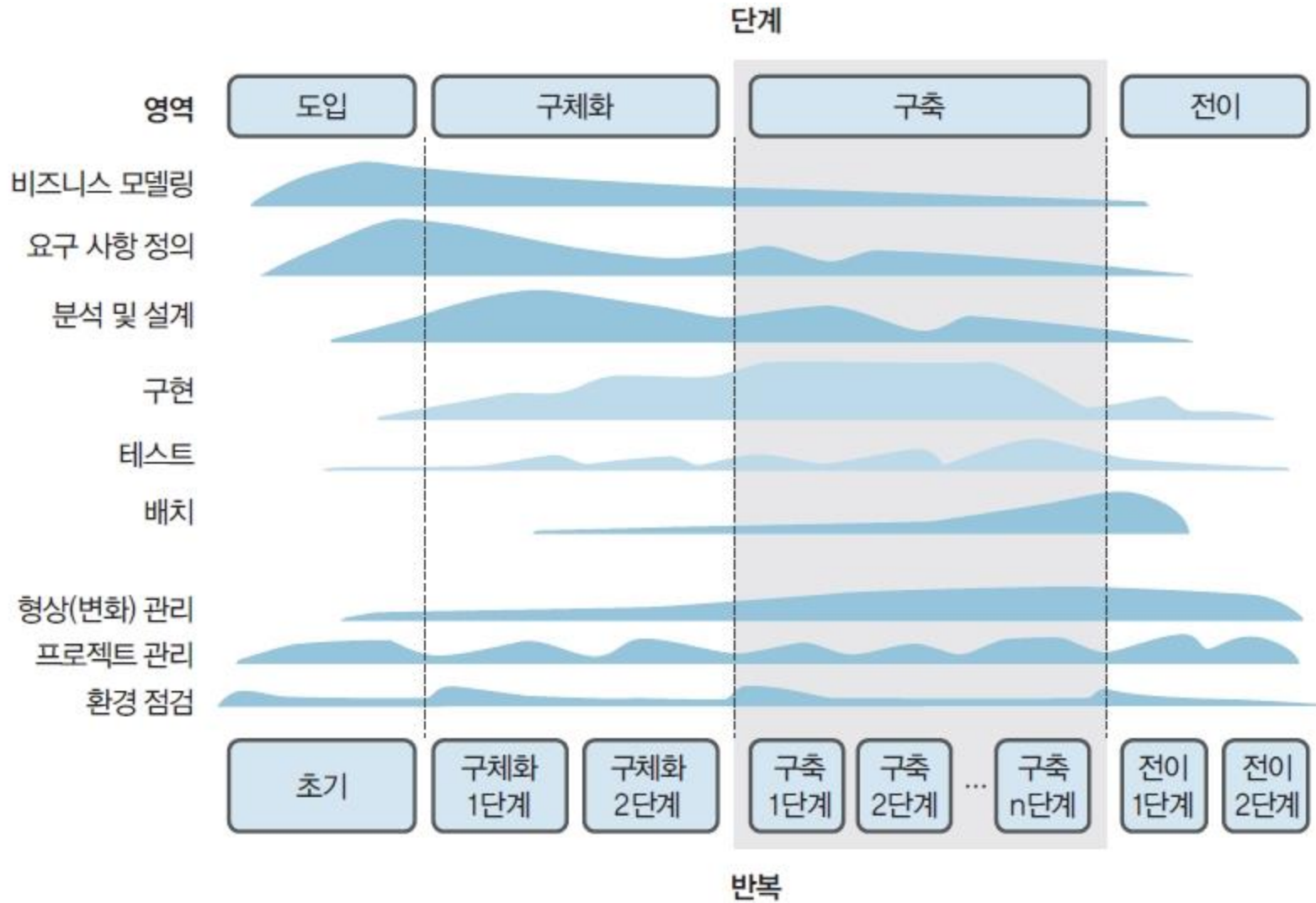


그림 2-22 통합 프로세스 모델의 구축 단계

④ 전이 단계 transition phase

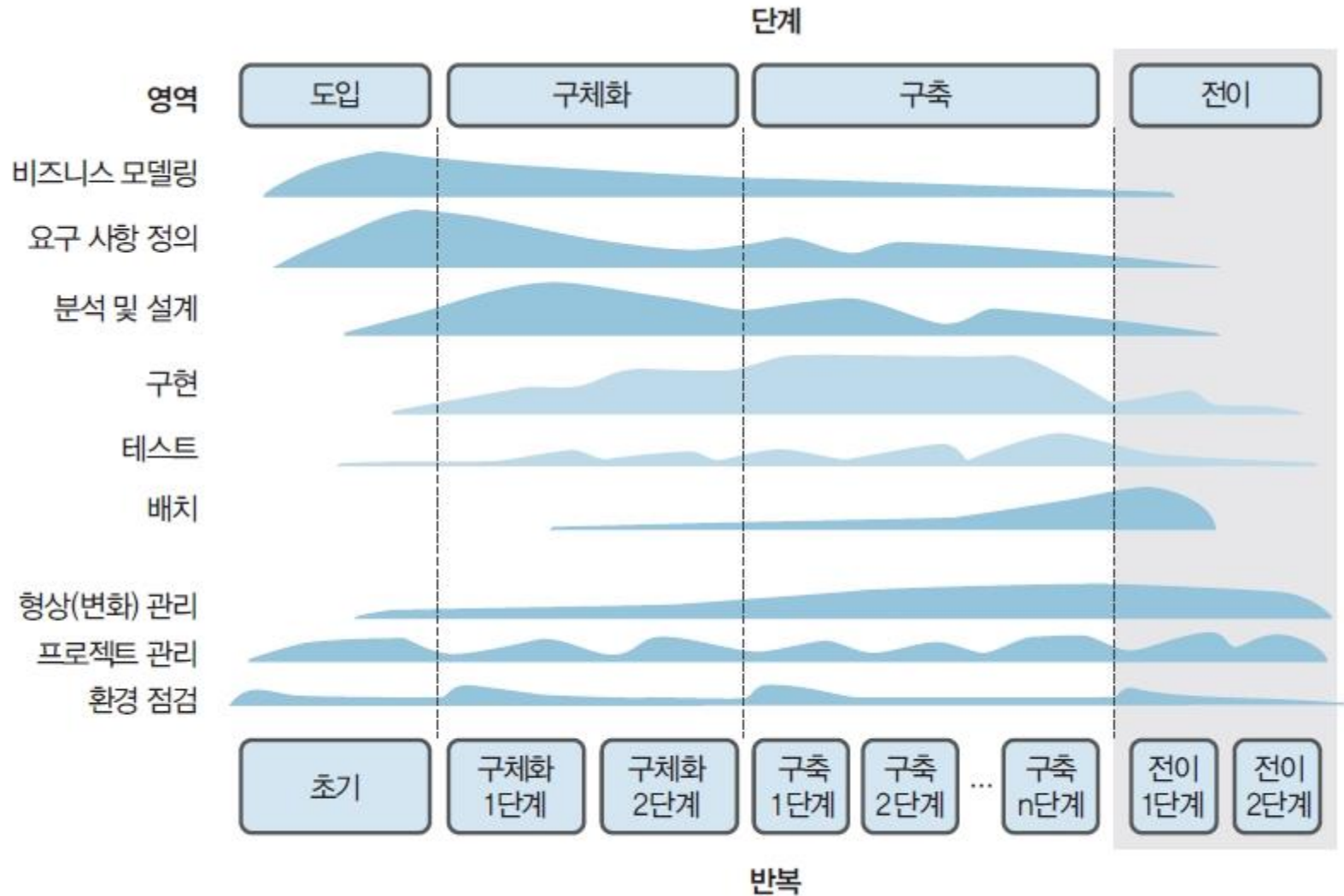


그림 2-23 통합 프로세스 모델의 전이 단계

⑤ 도입/구체화/구축전이 단계의 공통 작업

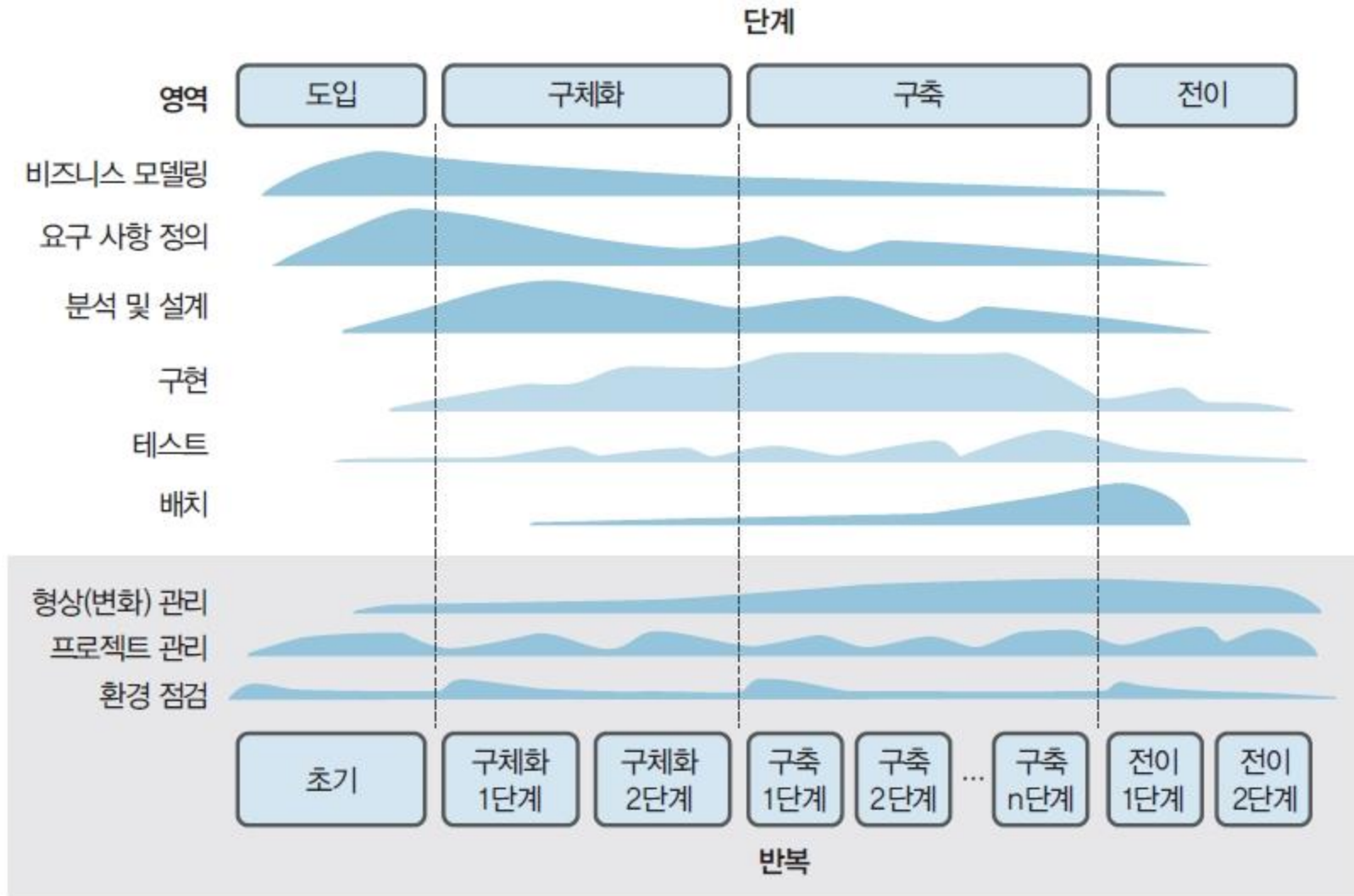


그림 2-24 통합 프로세스 모델의 공통 작업



Section 10 애자일 프로세스 모델

1. 애자일 프로세스 모델

■ 애자일(agile)

- '날렵한', '민첩한'

■ 애자일 프로세스 모델

- 고객의 요구에 민첩하게 대응하고 그때그때 주어지는 문제를 풀어나가는 방법론

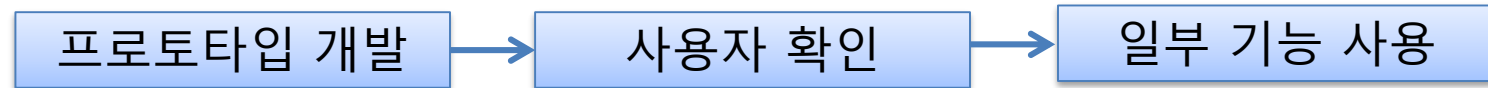
■ 애자일의 기본 가치(애자일 선언문)

- 프로세스와 도구 중심이 아닌, 개개인과의 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

1-1 애자일 프로세스 모델의 이해

■ 애자일의 개발 방법

- 반복적인 개발을 통한 잦은 출시를 목표로 함



3. 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)

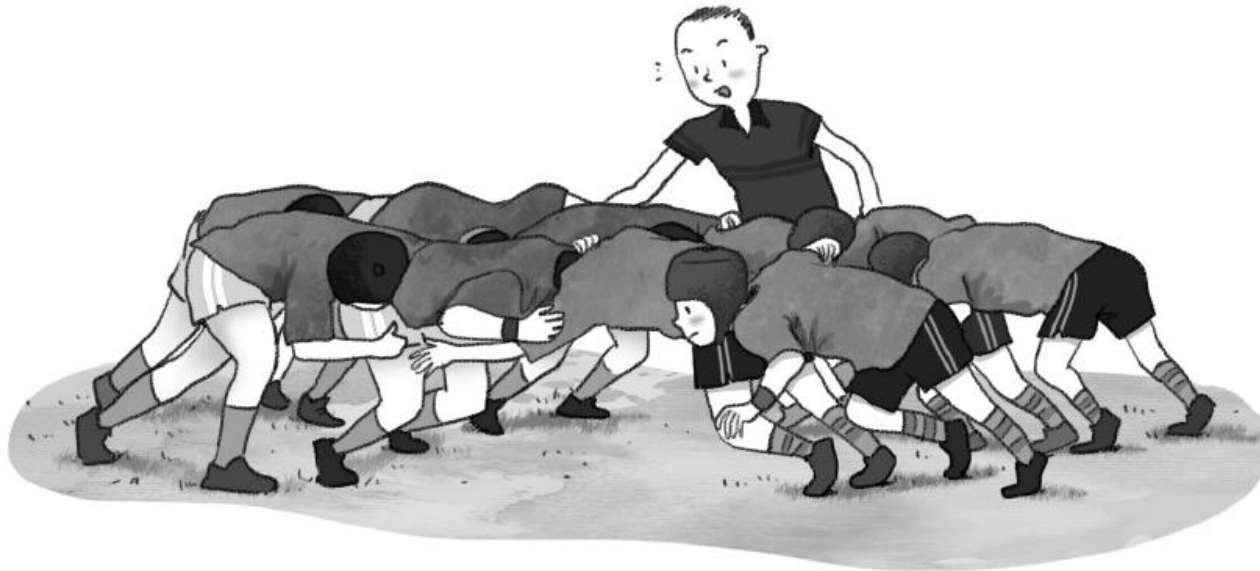


그림 2-25 럭비 경기의 스크럼 대형

4. 스크럼 방식의 진행 과정

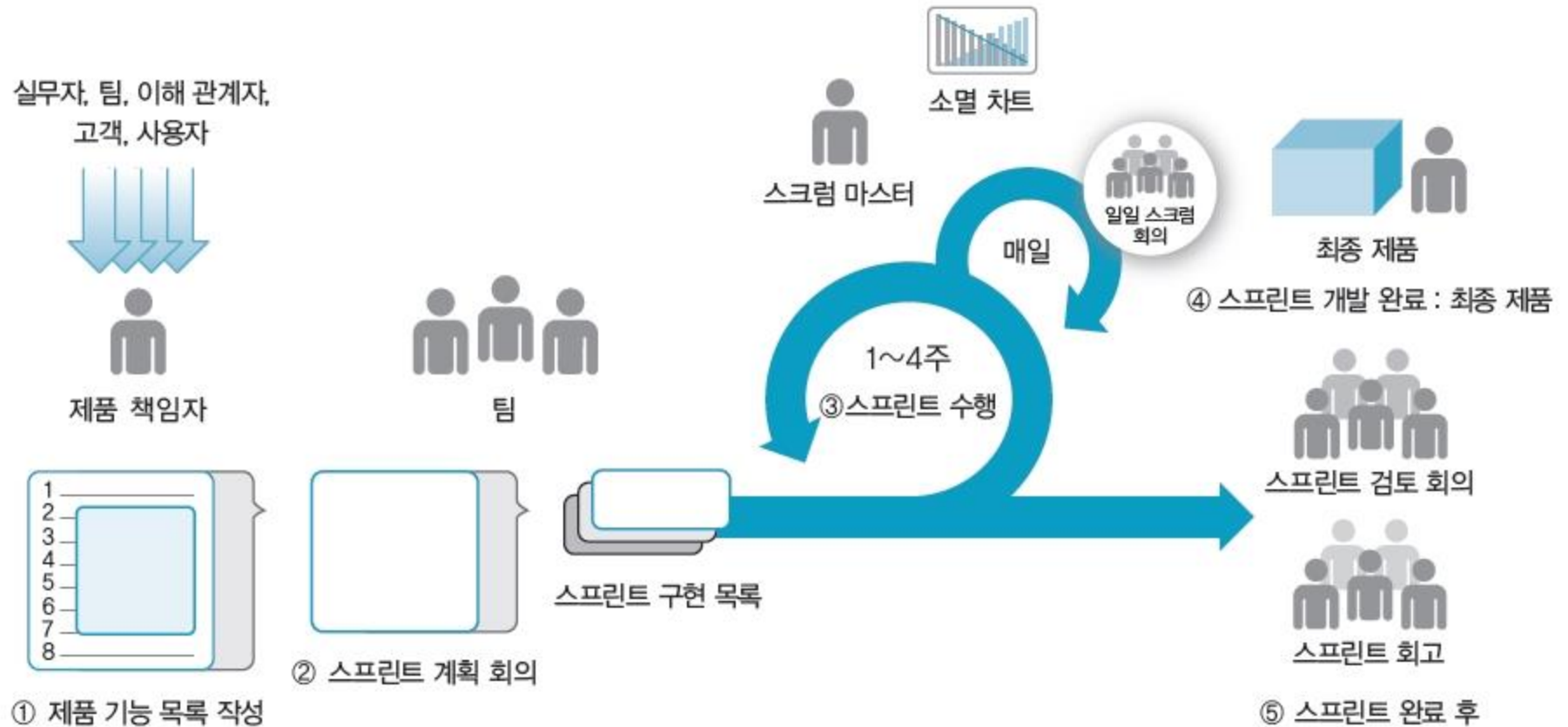


그림 2-26 스크럼 방식의 진행 과정