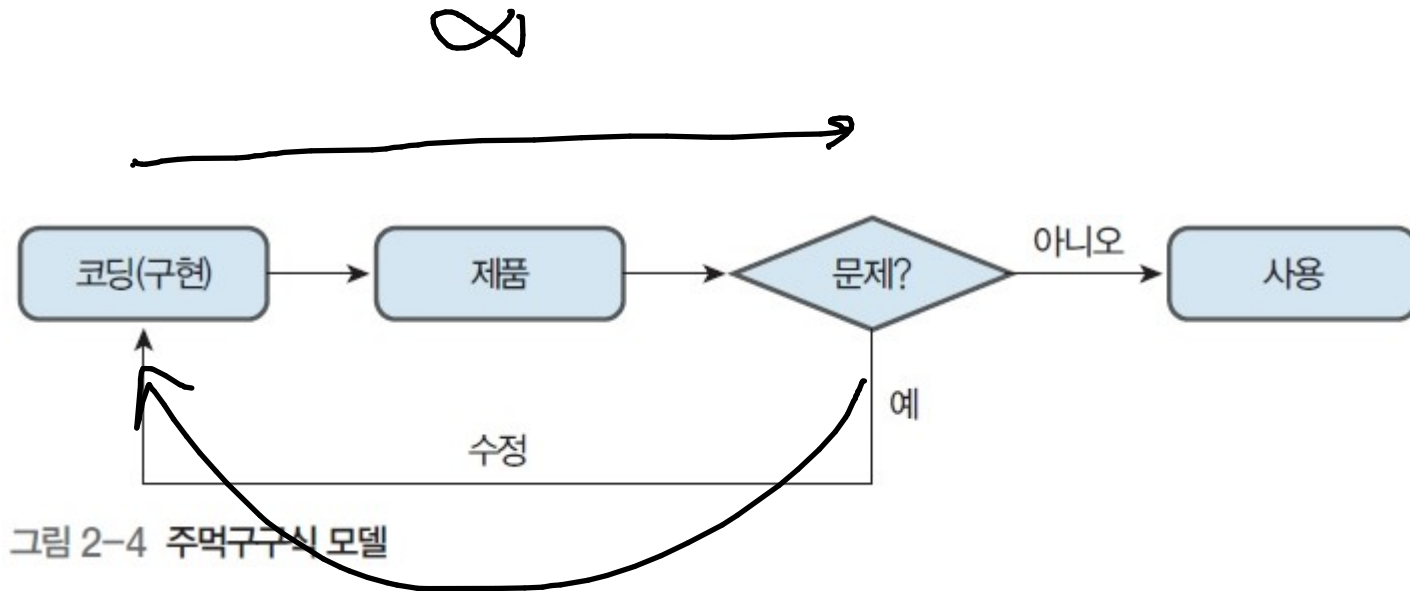


2. 주먹구구식 모델의 개발 단계

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



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.

* 짜보고 고치기 모델 (Code & Fix)

- **짜보고 고치기 모델** : 거의 쓸모 없는 모델임 . 그럼에도 불구하고 흔히 사용하고 있는 모델 .
- 특정 소프트웨어개발생명주기 모델을 선정하지 않은 경우는 바로 “짜보고 고치기 모델”인 가능성 농후 .
- 프로젝트 계획 수립 활동을 하지 않고 진행하고 있는 프로젝트의 경우 - ‘짜보고 고치기 모델’ 사용 중임 .
- 일정이 촉박되면 -> ‘미친 듯 코딩하기 (code-like-hell approach)’로 변질 .
- 만들 제품에 대한 전반적인 개념만으로 출발 .
- 공식적 명세서가 있을 수도 있고 없을 수도 있음 .
- 비공식적 단계들을 조합하여 적당하다 싶으면 마구잡이로 무엇이든지 사용 .



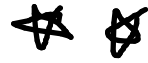
* 짜보고 고치기 모델 (Code & Fix) 의 장점과 적용

- ^{load} 짜보고 고치기 모델의 장점 두 가지
 - (1) 부하가 없음 - 여러 활동 (계획수립 , 문서화 , 품질보증 , 표준확립) 에 시간 들이지 않고 곧바로 구현에 투입
 - (2) 비전문가가 사용 - 누구든 사용
 - 적절한 프로젝트 : 한번 만들어보고 버릴 목적의 아주 작은 프로젝트 (작은 개념 확인용 프로그램 , 잠깐 쓸 Demo, 1 회성 Prototype 등)
-
- 그 외 프로젝트에서는 위험한 모델 : 부하는 없으나 진행상황 평가 방법도 없음 . 완성할 때까지 무작정 진행 . 품질평가나 위험인지 방법이 없음 .



Section 04 선형 순차적 모델

1. 선형 순차적 모델



- Linear sequential 모델, waterfall 모델 (= 폭포수 모델), Classic life cycle (= 고적전 생명주기)



그림 2-5 폭포수 모델

2. 폭포수 모델의 개발 절차

* 명물 단어

- 계획단계 (3 장에서 자세히 다룸) planning
- 요구분석 단계 (4 장에서 자세히 다룸) requirement analysis
= 분석 단계
- 설계 단계 (5-6 장에서 자세히 다룸) design
- 구현 단계 (7 장에서 자세히 다룸) implementation
- 테스트 단계 (8 장에서 자세히 다룸) test
- 유지보수 단계 (10 장에서 자세히 다룸) maintenance

* 폭포수모델 (Waterfall Model)

- 모든 생명주기모델의 조상
 - 다른 효과적 생명주기모델의 기초
 - 질서 정연한 단계의 개념
 - 매 단계 끝에 다음단계로 넘어갈 준비가 되었는지 결정하는 검토과정 있음.
- 검토와 문서가 나뉜다.
- 문서를 기반으로 한다. (매 단계 문서화)
 - 연속적이지 않고, 단계가 서로 겹치지 않는다.

3. 폭포수 모델의 장점

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

정확함

- 제품 정의가 안정적이고, 익숙한 기술 방법론을 쓰고자 할 때 효과적 : 오류를 비용이 적게 드는 초기에 발견 가능하고, 개발자들이 바라보는 요구사항 안정성을 얻을 수 있음.
- 기존 제품을 판올림 (version up) 하는 유지보수성 버전 개발이거나, 새 플랫폼 (platform) 에 이식할 경우 적합.
- 모든 계획을 초반에 세우기 때문에 계획수립에 드는 부하를 최대한 줄일 수 있다.
- 요구사항이 안정적이면서, 시스템 자체가 복잡한 프로젝트에 적합 (그 이유는 프로젝트가 갖는 복잡성을 순차적으로 풀어갈 수 있기 때문에 유리).
- 품질 요구가 비용과 일정 요구에 비해 클 때 적합 (중간 단계에서 변경이 없으므로 오류발생 여지가 줄어듦)
- 기술력이 떨어지거나 경험 부족 개발팀에 특히 효과적 (프로젝트 노력 낭비를 줄일 수 있는 구조)

품질 > 일정

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.

단계가
끊어져있다.



그림 2-6 폭포수 모델의 단점

강의제

* 폭포수모델의 단점 강. 안걸 명하 /

- 생명 주기가 끝날 때까지 실제적인 결과물을 보여주지 못함. (단, 문서인식 가능자는 진행상황 인식가능)
현대 비즈니스 요구에 부합 X
- 설계와 구현을 시작하기도 전인 프로젝트 초기에는 요구사항을 완벽하게 명시하기가 어렵다.
- 폭포수 모델의 **가장 큰 문제점** : 유연하지 않다는 점. (프로젝트를 시작할 당시에 요구사항을 완벽하게 명시해야한다) -> 이 모델은 현대 비즈니스 요구 (프로젝트 후반에 가장 많은 기능을 구현한 개발자를 높이 평가)에 부합되지 않음.
- MS 사의 Roger Sherman 의 지적 : “진짜목표는 프로젝트 초반에 약속한 내용달성이 아닌 시간과 자원이 허용하는 한 최고량 달성이다 (95)”
생각
- 이모델은 다음단계에서 이전단계로 복원이 어렵다. (= 연어 생명주기 모델 ; Salmon Life cycle Model) * 복원이 불가능하지는 않으나 어려움.
- 작성할 문서와 최신정보 유지를 위한 문서관리가 너무 많다. 최신 정보 유지 시 “문서관리”만 전업이 되는 꼴 (패속개발의 저해 요인)
- 마지막 순간 까지 진행 상황을 거의 보여주지 못한다 (실제 개발이 늦지 않음에도 느리다는 인식을 줄 우려)