3주차
2차시

06

소프트웨어공학

소프트웨어 개발 생명주기 모형 2

1 프로토타입 모형

2 나선형 모형

3 4GT 모형

01
프로토타입 모형

1. 프로토타입 모형의 개요

- 사용자의 요구사항을 정확히 파악하기 위해 실제 소프트웨어에 대한 시제품(prototype)을 미리 만들어 최종 결과물을 예측하는 모형

- 시제품을 사전에 만들어 결과물 예측

- 프로토타입 : 요구 분석 단계에서 사용

- 프로토타입의 평가 후 → 개발 승인 → 본격적인 개발

01
프로토타입 모형

2. 프로토타입 모형의 특징

- 폭포수 모형의 단점을 보완 → 사용자 요구 사항 미리 파악 가능
- 시제품은 사용자와 시스템 사이의 인터페이스에 중점을 두어 개발
- 시스템의 일부 혹은 시스템의 모형을 만드는 과정 → 요구된 소프트웨어의 일부 구현 → 추후 구현 단계에서 사용될 골격 코드
- SDLC에서 유지보수가 없어지고, 개발 단계 안에서 유지보수가 이루어짐
- 발주자나 개발자 모두에게 공동의 참조 모델 제공
- 구축할 시스템의 요구사항이 불명확한 경우 → 시제품으로 사용자 요구사항 도출
- 개발자와 사용자 간의 오해 요소 감소

01
프로토타입 모형

3. 개발 순서

요구 분석 → 빠른 시제품 설계 → 시제품 구축 → 고객 평가 → 시제품 조정 → 구현

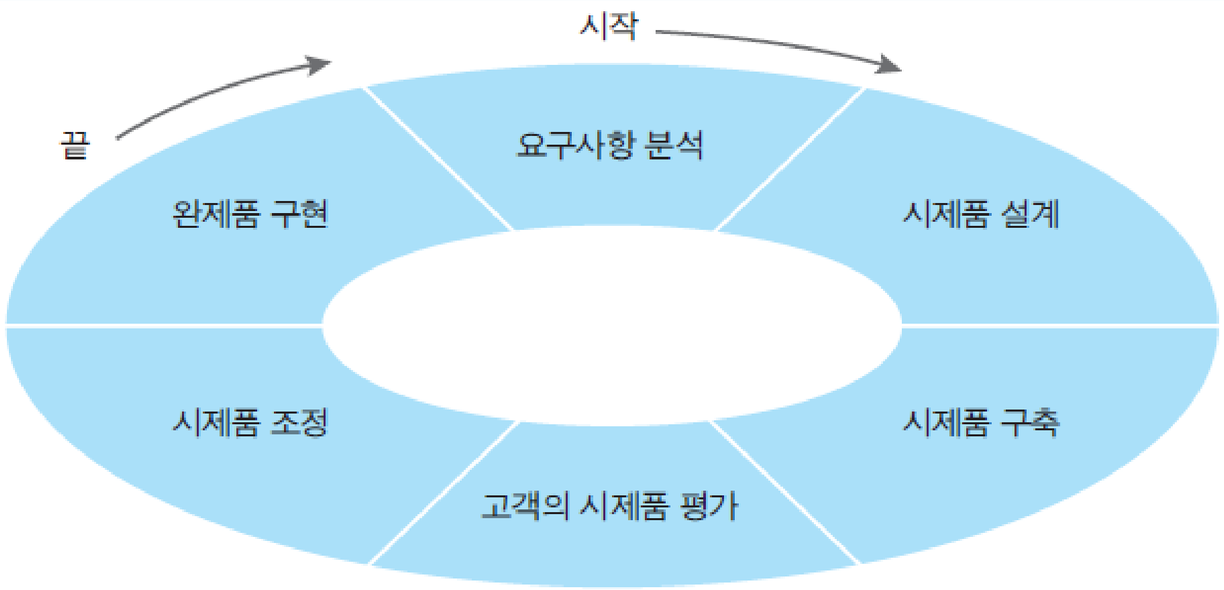


그림 3-2 프로토타입 모형 개발 순서

01
프로토타입 모형

4. 개발 순서

- **요구 분석**
  - 폭포수 모델의 요구사항 분석단계와 유사
    - ✓ 고객으로부터 받은 일부의 요구사항만 정의
    - ✓ 완전치 않은 요구사항에 대하여 윤곽 설정
    - ✓ 추가적인 정의가 필요한 부분은 시제품이 개발된 후 계속 정제

- **시제품(프로토타입) 설계**
  - 사용자들 입장에 초점
  - 시제품 개발의 목표가 확립되고 시제품에 포함될 시스템의 기능 선택
  - 시제품에 포함되는 것과 시제품에서 배제되어야 하는 것이 무엇인지 규명하는 것은 중요

01
프로토타입 모형

4. 개발 순서

- **시제품 구축 단계**
  - 성능, 다른 시스템과의 인터페이스 등에 대한 것은 중요하지 않음
  - 오류를 관리하고 다루는 면은 무시되거나 기초 수준 정도로 구현
  - 시제품의 신뢰도와 프로그램 품질 수준은 떨어짐
  - 이 단계의 목표는 '어떻게 하면 시제품을 빨리 만들 수 있겠는가' 이다.

- **고객의 시제품 평가 단계**
  - 프로토타입 모형의 가장 중요한 단계
  - 시제품은 고객에 의해 평가되고, 개발될 소프트웨어의 요구사항을 구체적으로 조정
  - 이를 통해 요구사항의 오류를 발견하고 규명
  - 추가할 요구사항 발견 가능

01
프로토타입 모형

4. 개발 순서

- **시제품 조정 단계**
  - 사용자가 원하는 것을 만족시키기 위해 시제품에 대한 조율이 필요
  - 시제품이 변경 방법/내용 결정 후 다음 단계의 시제품이 빠르게 제작
  - 시제품은 다시 고객에게 평가되는 순환 → 고객 요구사항 만족할때까지 계속

- **완제품 구현 단계**
  - 이 단계의 목표는 원하는 시스템을 개발하는 것
  - 만약 시제품을 버리고 새 시스템을 개발해야 한다면, 이 단계는 완전한 폭포수 모형의 생명주기를 따르거나 4세대 기법(4GT)의 사용 가능

01
프로토타입 모형

5. 장단점

장점	단점
• 사용자 요구사항 충실히 반영 • 개발자와 사용자의 오해 해소 • 생각하지 못하였던 기능과 서비스가 발견 • 완전하지 못하지만 작동하는 시스템을 만들어 가능성과 유용성을 사용자 제공 • 프로토타입은 사용자와 개발자 모두에게 공동의 참조 모델 제공	• 시제품이 실제 소프트웨어와 차이가 발생할 경우 사용자에게 혼란 • 단기간에 제작해야 하기 때문에 비효율적인 언어나 알고리즘을 사용 가능성 • 시제품 폐기 시 비경제적 • 소프트웨어 개발에 많은 시간이 소요되며, 보고서 등 출력물 증가

02
나선형 모형

1. 나선형 모형의 개요

- 나선형 모형(Spiral Model, 점직적 모형)
  - 보헴(Boehm)이 제안
  - 폭포수 모형 장점 + 프로토타입 모형 장점 + 위험 분석(risk analysis)
  - 나선을 따라 돌듯이 여러 번의 소프트웨어 개발 과정
  - 점진적 모형 →점진적으로(프로토타입을 지속적으로 발전시켜) 완벽한 최종 소프트웨어를 개발하는 것

- 시스템을 개발하면서 생기는 위험 최소화 주 목적

02
나선형 모형

2. 나선형 모형의 특징

- 고객과의 의사소통(communication)을 통해 계획 수립과 위험분석,
  구축 고객 평가의 과정을 거쳐 소프트웨어를 개발

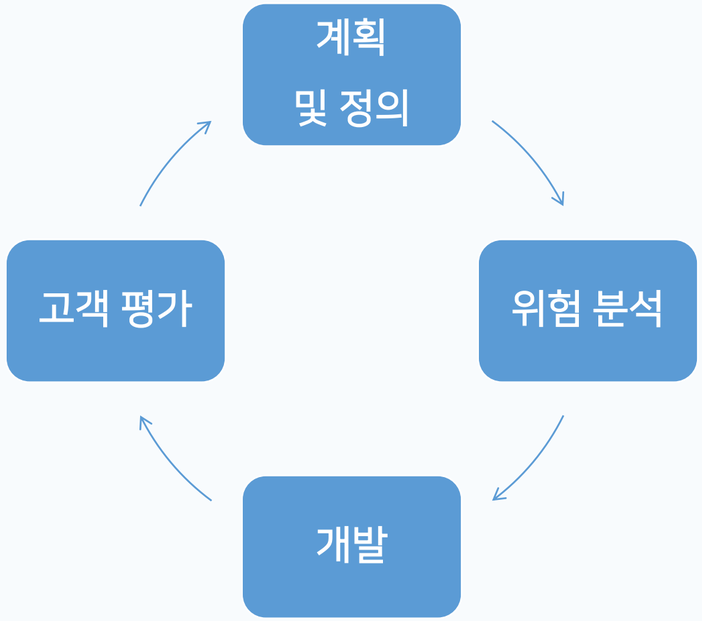
- 가장 큰 장점 : 위험 분석 단계
  - 기술과 관리의 위험요소들을 하나씩 제거 → 완성도 높은 소프트웨어

- 반복적인 작업을 수행하는 점증적 생명주기 모델

- 비용이 많이 들거나 시간이 많이 소요되는 대규모 프로젝트나 큰 시스템을
  구축할 때 유리

02
나선형 모형

3. 개발 순서



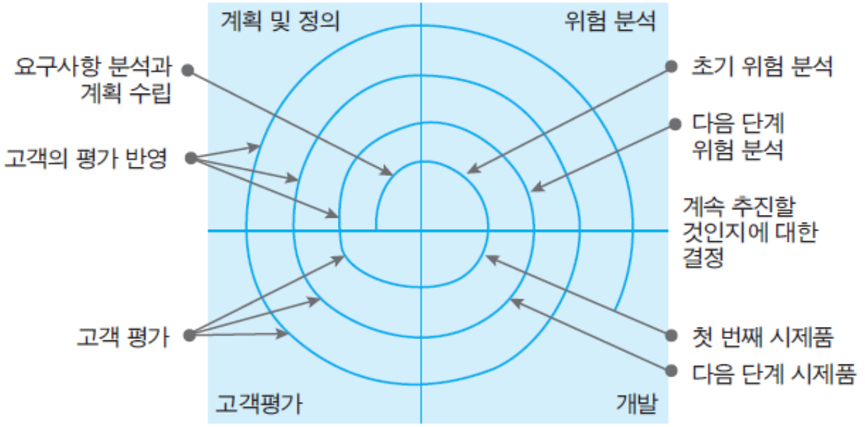


그림 3-3 나선형 모형의 개발 순서

02
나선형 모형

4. 장단점

장점	단점
• 나선형 모델은 비용이 많이 들고 시간이 오래 걸리는 큰 시스템을 구축해 나가는 데 가장 현실적인접근 방법 • ex) 초고속 보통신망 개발, 큰 국책사업, 대형사업 • 성과를 보면서 조금씩 투자하여 위험 부담을 줄일 수 있는 이상적 방법	• 모델 자체가 앞의 두 모델보다 더 복잡하여 프로젝트 관리 자체를 어렵게 만들 가능성이 많다. • 많은 고객을 상대로 하는 상업용 제품에 적용하기 힘들다.

03
4GT 모형

1. 4GT(4th Generation Techniques) 모형의 개요

- 4GT 모형(4세대 기법)
  - 사용자와 개발자가 쉽게 접근하고 사용할 수 있는 CASE를 비롯한 자동화 도구, 4세대 언어(4th Generation Language) 등을 이용하여 개발자가 조사한 요구 사항 명세서로부터 원시 코드를 자동으로 생성할 수 있게 해주는 모형

- CASE(computer-aided software engineering)
  - 컴퓨터 프로그램의 개발에서, 계획에서 문서화까지의 모든 공정을 자동화하고 공학적 관점에서 구축하기 위해 컴퓨터를 이용하도록 설계된 소프트웨어의 총칭
  - 컴퓨터를 이용한 소프트웨어공학
  - 컴퓨터 시스템의 응용 프로그램 설계와 작성을 자동화할 수 있도록 도와주는 각종 프로그램, 기법 및 기타 개발 툴이 제공되어 있는 작업 환경을 의미

03
4GT 모형

1. 4GT(4th Generation Techniques) 모형의 개요

- 4GT 모형(4세대 기법)
  - CASE의 특징
    - ✓ 개발도구와 개발 방법론이 결합된 것
    - ✓ 시스템 개발과정의 일부 또는 전체를 자동화
    - ✓ 정형화된 구조 및 메커니즘 → 소프트웨어 개발 적용 → 소프트웨어 생산성 향상

- 자동화 도구
  - 사람이 사용하는 고급 언어로 요구사항이 명시하면 소프트웨어 제품으로의 생성 해주는 도구

03
4GT 모형

1. 4GT(4th Generation Techniques) 모형의 개요

- 장단점

장점	단점
• 설계 단계 단축 • 4세대 언어를 사용하므로 원시 코드를 자동으로 생성	• 중소형 소프트웨어 개발에 적합하나, 대규모 소프트웨어 개발은 부적합 [자동화로 인해 단축된 시간보다 분석, 설계 단계 등에서 더 많은 시간을 필요]