

소프트웨어 공학 (Software Engineering)

**유지보수
(Maintenance)**



In this chapter ... (1/3)

유지보수(Maintenance)

- 많은 소프트웨어는 개발이 끝이 아니라, 그때부터 시작이다.
- 전자레인지의 경우 새로운 기능을 추가할 수 있다고 생각하지 않으나, 소프트웨어는 주물주물하면 새로운 기능을 넣을 수 있다고 생각한다.
- 자동차의 관리와 유사하게, 소프트웨어의 유지보수도 “해당 제품을 값어치 있게 사용하기 위한” 매우 중요한 요소이다.

We will cover ...

- 유지보수의 소개
- 유지보수 작업 과정
- 형상 관리
- 역공학
- 리엔지니어링
- 유지보수 도구



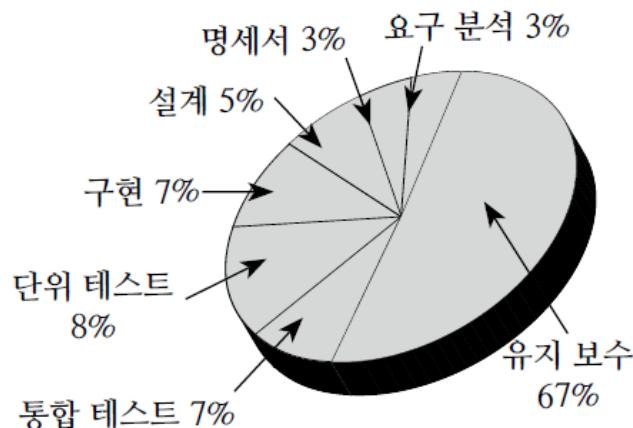


In this chapter ... (2/3)

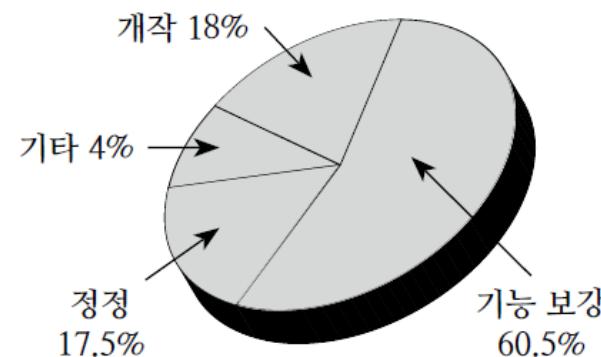
유지보수(Maintenance)

많은 소프트웨어는 개발이 끝이 아니라, 그때부터 시작이다.

- 소프트웨어가 인수 설치된 후 일어나는 모든 작업
- 소프트웨어가 유용하게 활용되는 기간
→ 소프트웨어가 살아있는 기간
- 소프트웨어 유형에 따라 비용이 많이 들 수 있다



(a) 소프트웨어 생명 주기 전체에서 차지하는 유지보수 노력



(b) 유지보수 유형 분포



유지보수가 어려운 이유

- 소프트웨어의 특성: Invisibility, Complexity, Changeability

- Old code

(예전 소프트웨어는 기능, 작동에만 관심이 있었을 뿐, 설계, 문서화, 테스트 등이 체계적으로 이루어지지 않았음)

- 관리 부재: 소프트웨어 유지 보수에 대한 개념 부족

(예: 국내에선 무상 유지 보수 기간의 비용 무시, 선진국의 경우 매우 중시),
인력 교체에 따른 불가피한 관리 부재





In this chapter ...

유지보수(Maintenance)



10.1 유지보수의 소개



10.2 유지보수 작업 과정



10.3 형상 관리



10.4 역공학



10.5 리엔지니어링



10.6 유지보수 도구



정의

- 결함을 고치거나, 성능을 높이거나, 새로운 기능을 추가하거나, 변경된 환경에 적응시키기 위하여 배포 후 수정하는 작업



변경의 이유

- 버그 제거
- 운영 환경의 변화 – 하드웨어, 플랫폼, 시스템 형상의 변화
- 정부 정책, 규례의 변화
- 비즈니스 절차의 변화
- 미래 문제를 배제(예방)하기 위한 변경



교정형 유지보수(corrective maintenance) – 정정(correction)

- 발견된 오류의 원인을 찾아 계획적으로 문제해결

적응형 유지보수(adaptive maintenance) – 개작(adaptation)

- 새로운 자료나 운영체제, 하드웨어 환경으로 이식

완전형 유지보수(perfective maintenance) – 기능 개선(enhancement)

- 성능이나 유지보수성을 개선하기 위한 변경

응급형 유지보수(emergency maintenance)

- 응급처치하기 위한 무계획적 유지보수





In this chapter ...

유지보수(Maintenance)

10.1 유지보수의 소개

10.2 유지보수 작업 과정

10.3 형상 관리

10.4 역공학

10.5 리엔지니어링

10.6 유지보수 도구



(1) 프로그램 이해

- 문서, 코드 reading
- 프로그램의 구조 파악
- Domain knowledge 습득
- 변수와의 관계, 서브루틴 사이의 관계 파악
- 코드 안에 숨겨진 의미(semantic) 파악
- 분석도구(call graph, cross-reference table 등), 디버깅 도구(tracer) 사용

(2) 변경 파악과 분석

- 변경이 불가피한 이유와 요구를 이해
- 변경의 영향과 소요 비용, 리스크 분석



(3) 형상 변경 관리

- 특정 컴포넌트의 변경이 다른 컴포넌트에 영향을 줄 수 있음
- 변경의 이해 당사자에게 알리고 피드백

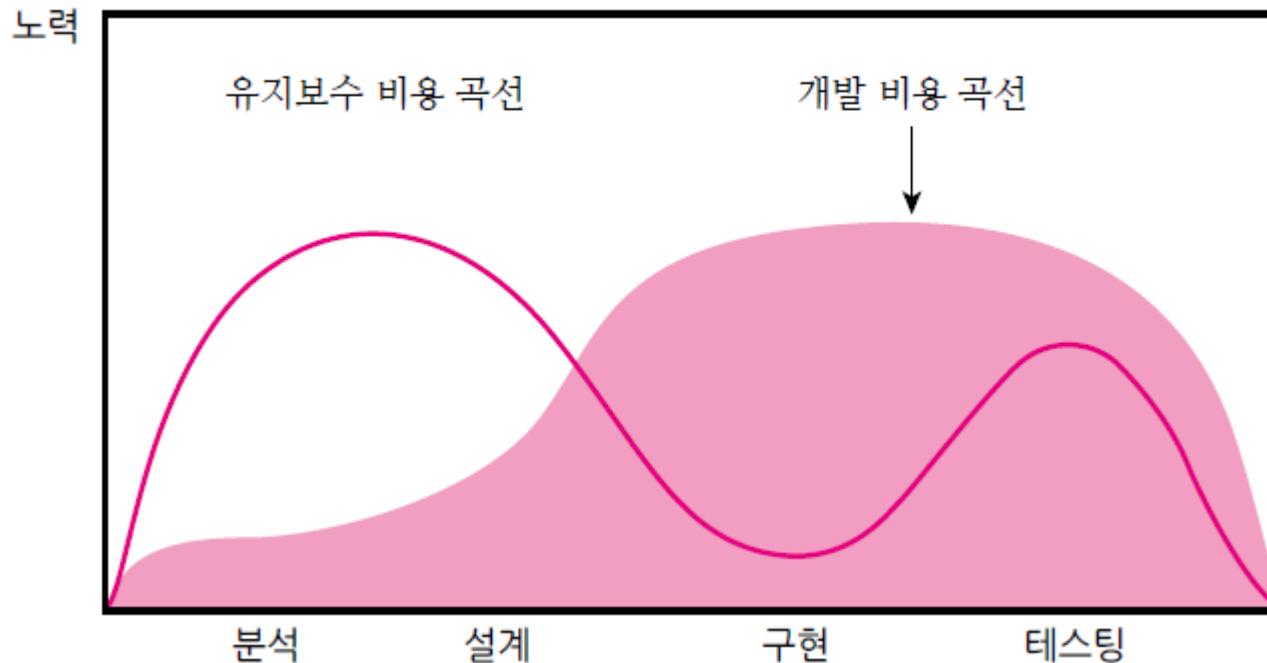
(4) 변경 구현, 테스팅, 설치

- Code change
- Change effect 분석



통합적이고 이해 중심적

- 개발은 코딩 중심의 작업, 유지보수는 이해 중심의 작업
- 각 작업이 포괄적이며 단발로 시행 → 다양한 기술이 필요



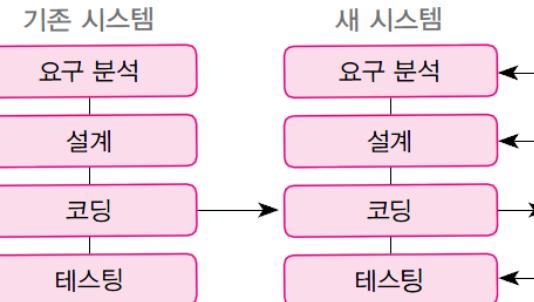


유지 보수 프로세스 모델

유지보수(Maintenance)

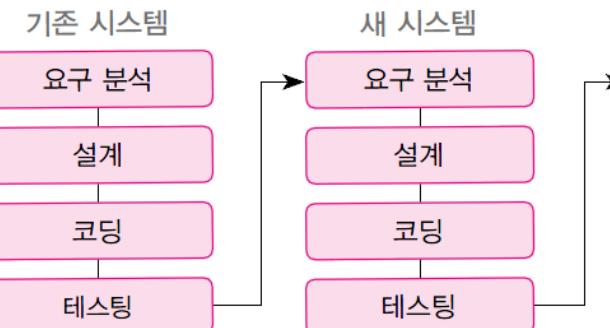
즉시 수정 모델

- 임시방편적인 유지보수 모델
- 문제가 일어나면 가능한 빨리 해결



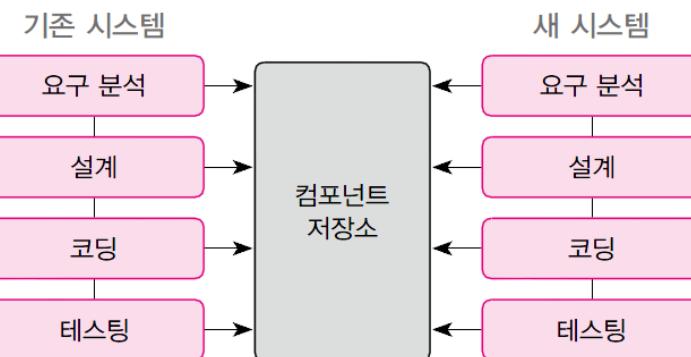
반복적 개선 모델

- 변경이 전체 생명주기 단계에서 반복
- 변경은 새 시스템의 개발 사이클을 시작



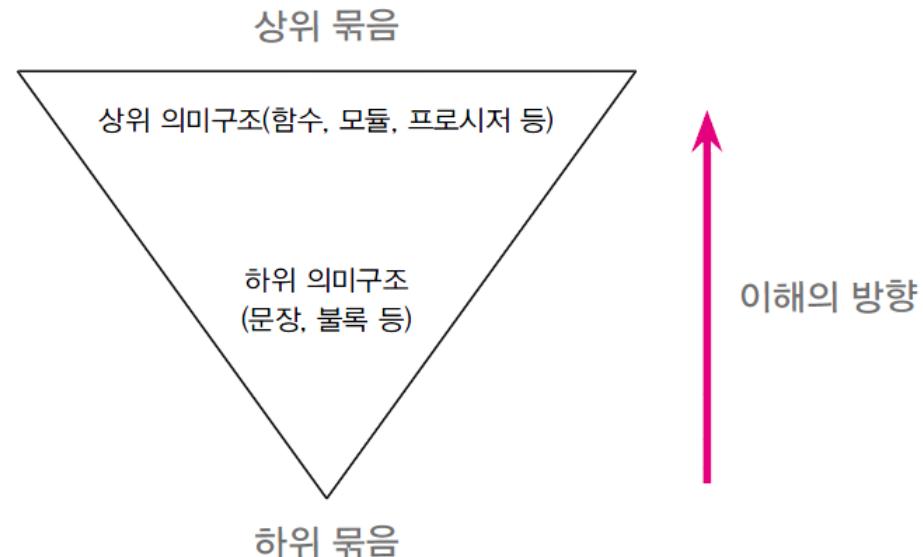
재사용 중심 모델

- 재사용 컴포넌트를 분류 및 관리
- 재사용 컴포넌트를 선택 맞춤화





- ④ 프로그램 이해는 원시 코드로부터 설계나 명세를 추출하여 멘탈 모델로 표현하는 작업 (코드 묶음을 추상화 → 의미 도출)
- ④ 개발 프로세스와 정반대의 순서
- ④ 프로그램 이해에 동원되는 원리
 - 상향식(bottom-up)
 - 묶음화(chunking)
 - ➔ 원시코드에서 의미 있는 묶음을 발견하고, 계속 더 큰 구조로 올라가 시스템 전체를 이해할 때 까지 반복





In this chapter ...

유지보수(Maintenance)

10.1 유지보수의 소개

10.2 유지보수 작업 과정

10.3 형상 관리

10.4 역공학

10.5 리엔지니어링

10.6 유지보수 도구



- ④ 형상 관리(Configuration Management): 개발 주기 동안 생성된 문서를 관리하고 소프트웨어 시스템과 컴포넌트의 상태를 추적하는 작업
- ④ 문서와 결과물에 대한 변경이 잘 조정되지 않는다면 불일치 발생
- ④ 클래스 변경 후 의존 클래스를 업데이트 하여야 함
- ④ 하드웨어에 적용되었던 전통적 원리를 소프트웨어 개발에 적용



베이스라인

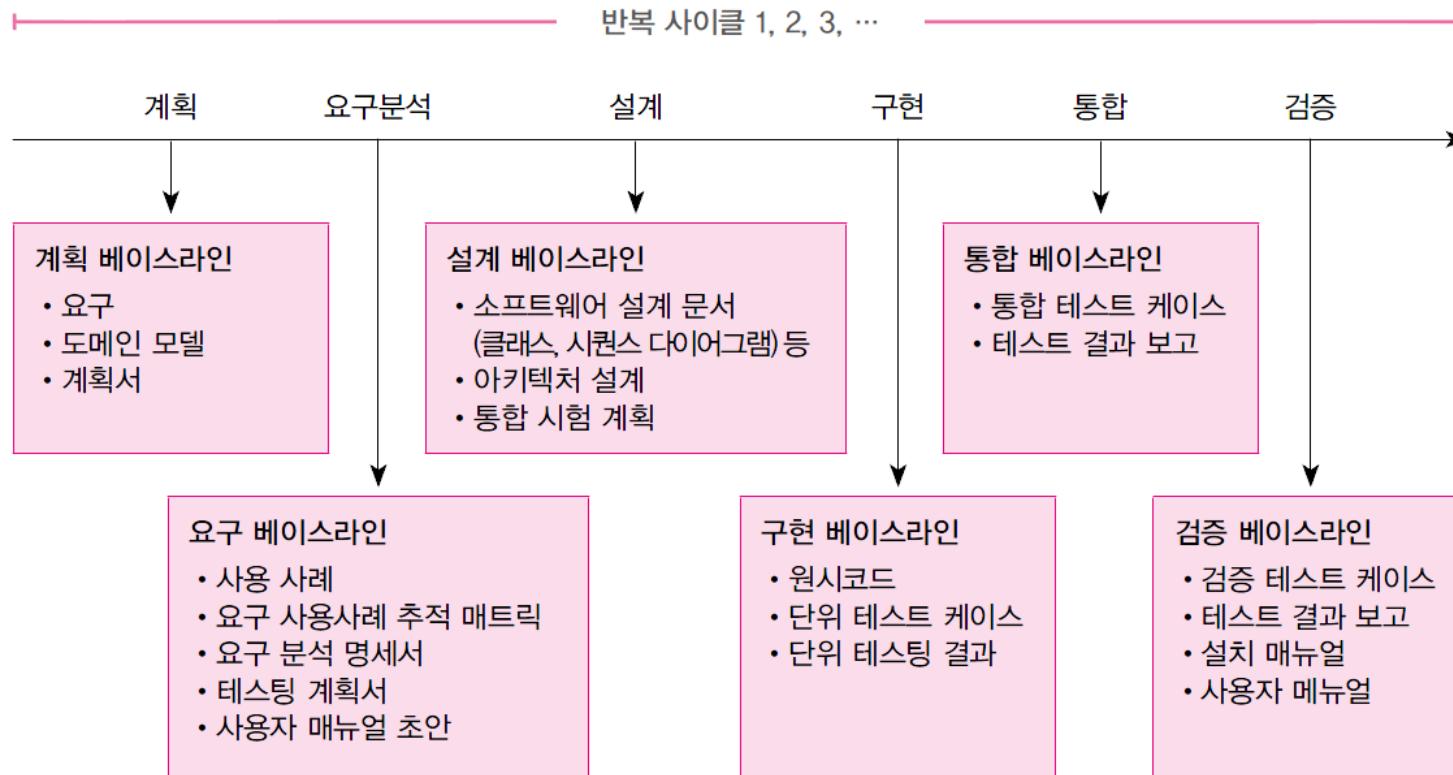
- 요구 단계, 설계 단계 등 각 단계의 종료 후 문서를 체크하는 시점을 마일스톤이라 함
→ 소프트웨어 형상관리에서는 마일스톤을 베이스라인이라 부름
- 정의: 소프트웨어 형상 항목(configuration item)의 집합으로 구성

베이스라인의 관리 목적

- 프로젝트의 중요한 상태 정의
- 프로젝트가 특정 상태에 이르렀는지를 나타냄
- 계속되는 개발, 유지보수 작업의 기준
- 형상 항목에 대한 변경을 제어하는 메커니즘



베이스라인과 형상 항목

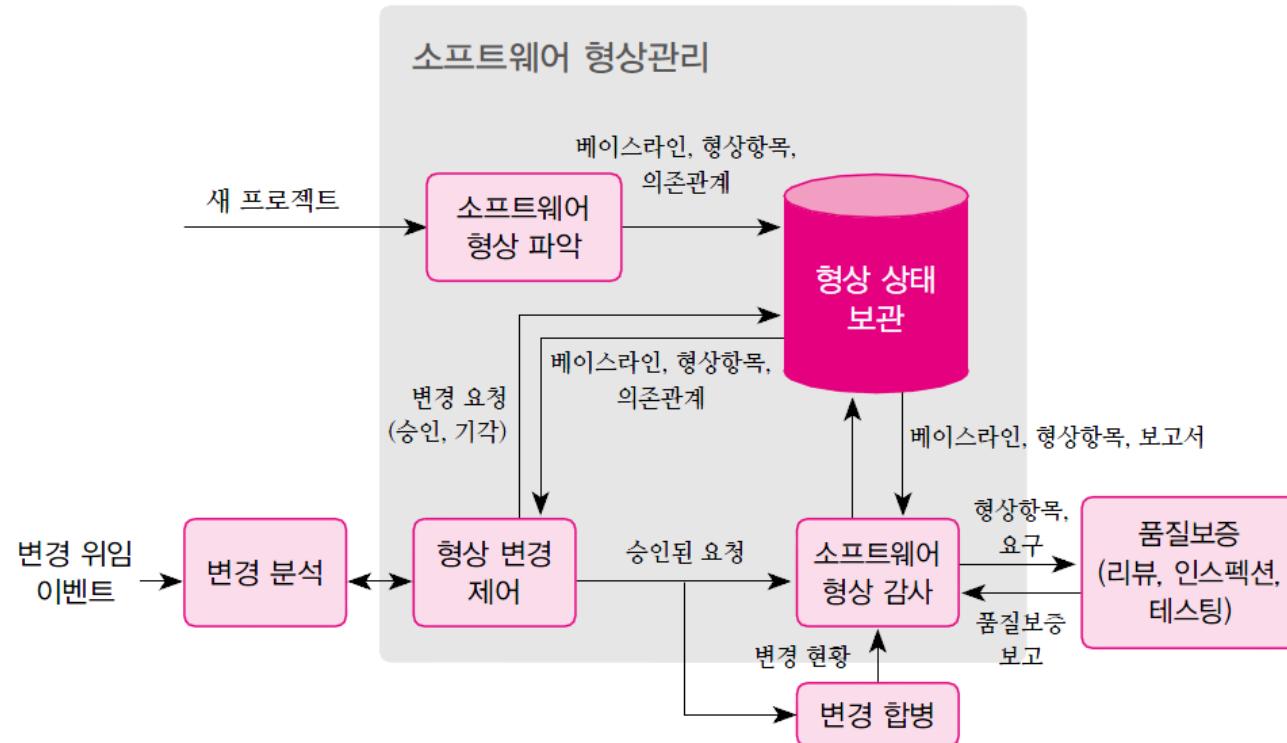




- ④ 소수의 개발자가 한 장소에서 일한다면 형상관리는 불필요
- ④ 시스템을 개발하는 많은 팀과 개발자들이 협력하고 동기화 할 필요
- ④ 여러 버전을 유지하여야 할 경우
 - 다양한 고객을 만족시키기 위한 제품을 유지하기 위해



- ▣ **소프트웨어 형상 파악**: 새 프로젝트 시작 시, 베이스라인과 형상 항목 등을 파악
- ▣ **형상 변경 제어**: 변경 요청 시 형상 항목에 대한 변경을 관리
- ▣ **소프트웨어 형상 감사**: 요청된 변경들의 적절성 검증
- ▣ **소프트웨어 형상 상태 보관**: 형상 정보 추적 및 유지





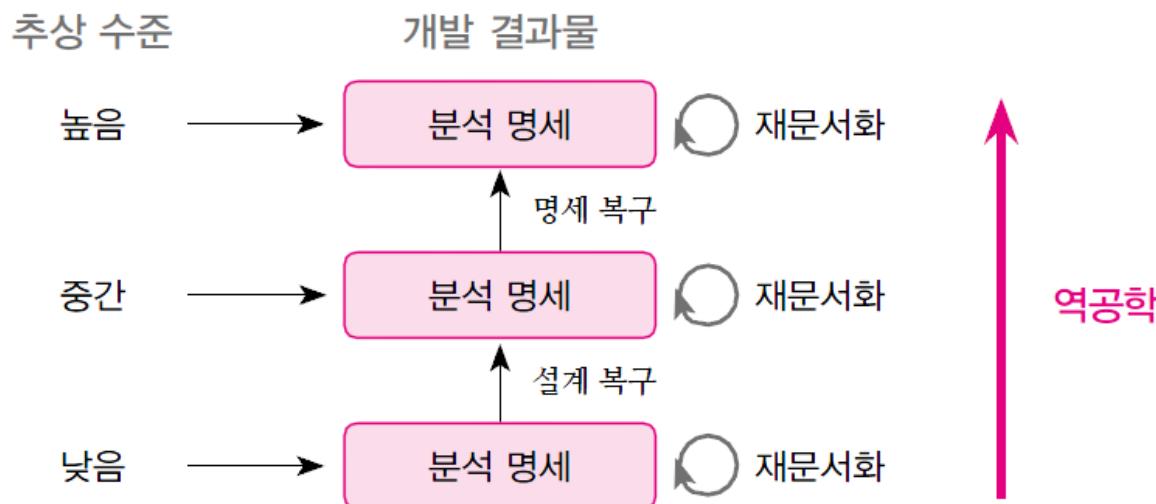
In this chapter ...

유지보수(Maintenance)

- 10.1 유지보수의 소개
- 10.2 유지보수 작업 과정
- 10.3 형상 관리
- 10.4 역공학
- 10.5 리엔지니어링
- 10.6 유지보수 도구



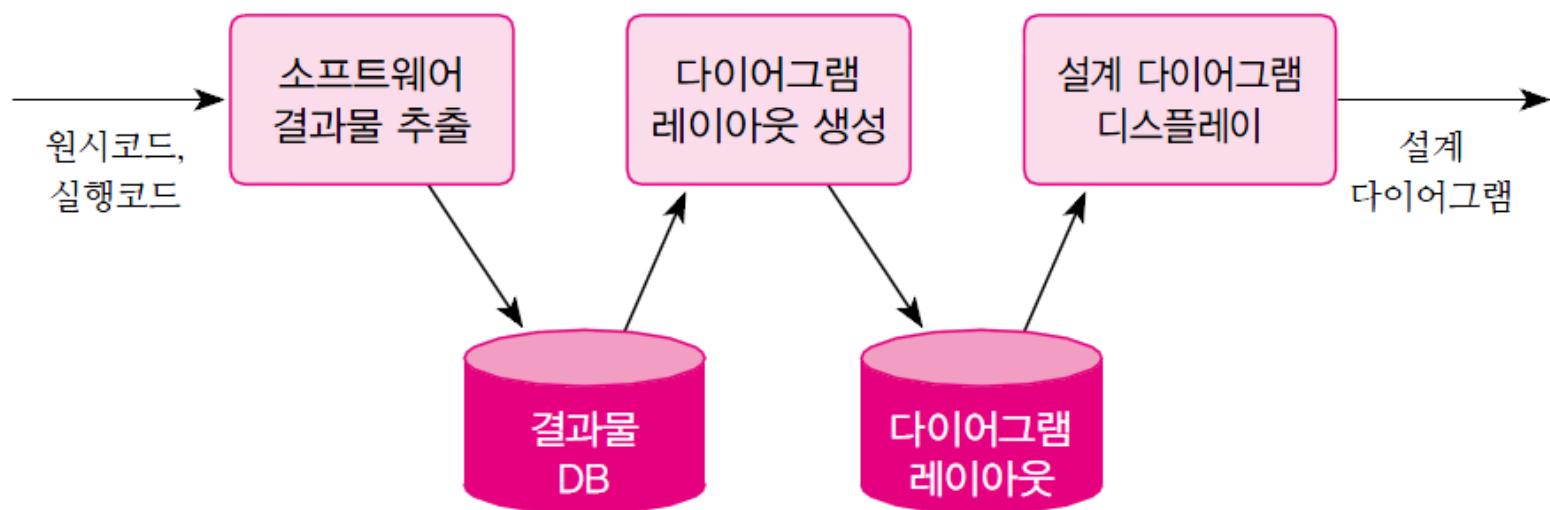
- 원래 의미: 하드웨어 분야에서 완제품으로부터 설계를 해독해내는 작업
- 정의: 대상 시스템을 분석하여 시스템의 컴포넌트와 관계를 찾아내어 같은 수준의 다른 표현이나 더 높은 수준의 표현으로 만드는 작업
 - 설계, 요구 명세, 문제 정의를 회복해 나가는 과정으로 개발 과정의 역방향 프로세스
 - 프로그램의 추상 수준을 점증적으로 복구해 나가는 과정





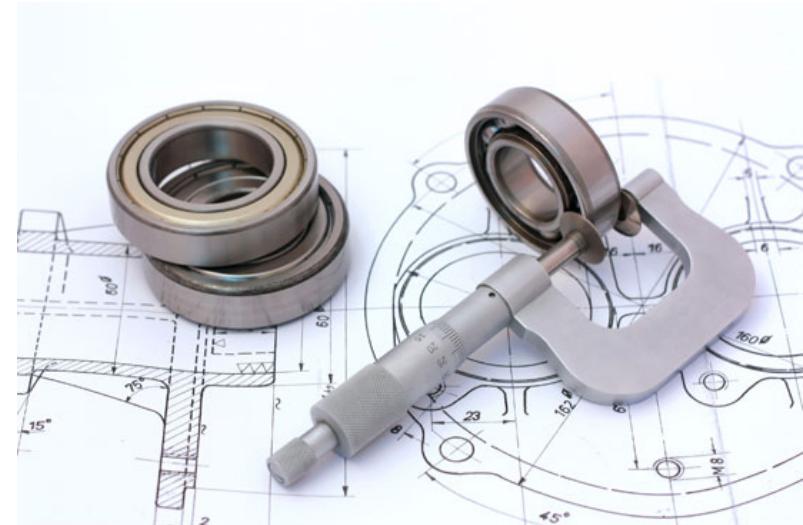
원시코드에서 소프트웨어 결과물들을 추출하는 것

역공학 도구의 구성





- ▣ 복원된 다이어그램은 다음 여러 방면에 사용
- ▣ 프로그램 이해: 소프트웨어의 구조, 기능, 동작 이해 용이
- ▣ 정형적 분석: 소프트웨어에 존재할 수 있는 문제를 감지
- ▣ 테스트 케이스 생성: 흐름도의 경로 – 경로 테스트 케이스에 도움
- ▣ 리엔지니어링 – 다음 섹션에서 설명





In this chapter ...

유지보수(Maintenance)

- 10.1 유지보수의 소개
- 10.2 유지보수 작업 과정
- 10.3 형상 관리
- 10.4 역공학
- 10.5 리엔지니어링
- 10.6 유지보수 도구

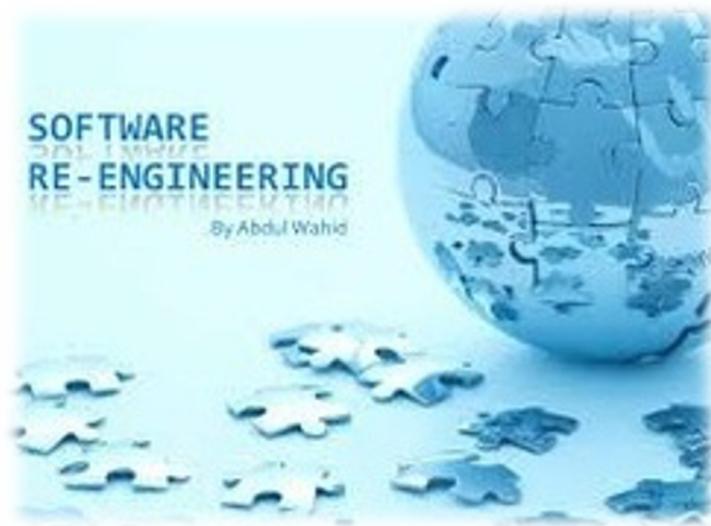


④ 시스템 또는 컴포넌트를 재구조화 하는 과정

- 유지보수에 비용이 많이 들어감 → 소프트웨어 시스템 구조가 나빠짐
- 유지보수 비용을 줄이기 위해 시스템의 재구조화가 필요함

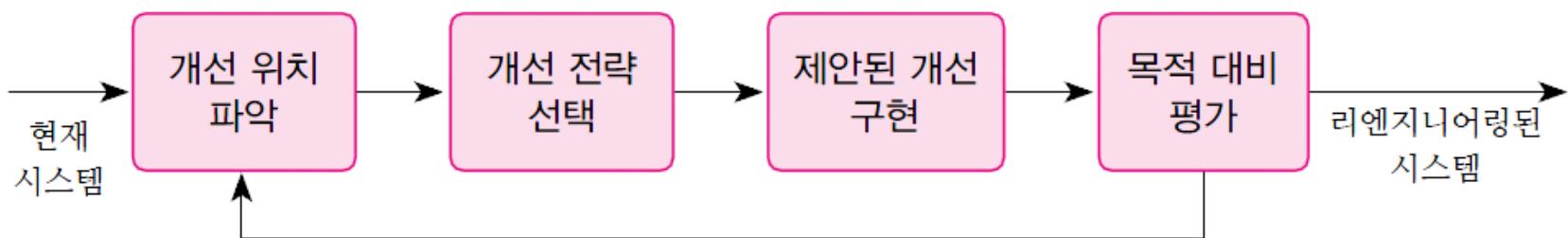
⑤ 리엔지니어링의 목적

- 소프트웨어 아키텍처 개선
- 소프트웨어의 복잡도 경감
- 변경에 대한 적응성 개선
- 성능, 효율성, 자원 유용성 개선
- 소프트웨어 시스템의 유지보수 개선





- ④ 개선이 필요한 위치 파악
- ④ 개선 전략을 선택 → 설계로 볼 수 있음
- ④ 제안된 개선의 구현 → 코딩/단위 테스팅/리그레션 테스팅
- ④ 목표를 기준으로 시스템 평가





In this chapter ...

유지보수(Maintenance)

10.1 유지보수의 소개

10.2 유지보수 작업 과정

10.3 형상 관리

10.4 역공학

10.5 리엔지니어링

10.6 유지보수 도구



도구의 사용은 시간과 노력을 대폭 감소시킬 수 있음

역공학 도구

- 메트릭 측정 도구
- 성능 측정 도구
- 정적 분석 도구
- 변경 효과 분석 도구
- 형상 관리 도구
- 리그레션 테스팅 도구

Aqua Data Studio 15.0.0 [CompanyRevenueAdvanced.xjs]*

```
File Edit Server Query Automate Query Builder Visual Analytics ER Modeler Tools DBA Tools Window Help
File System C:\SVN-AquaProjects Company Revenue AquaScripts CompanyRevenueAdvanced.xjs CompanyRevenueBasic.xjs Servers MySQL 6.0 - 38.conn User Files Custom Jar Example Database Exporter Financial Examples Master Slave Compare Math Solutions Multiserver Script Execute Random Database Generator C:\SVN-website

Editor [CompanyRevenueAdvanced.xjs]* X
16 // Specify the entry point URL. Set this to either dashboardURL or runURL depending
17 // the report in ADServer.
18
19
20 1 line deleted
21
22
23 var urlEntryPoint = aqua.project.dashboardURL + "/";
24 try
25 {
26     buildToolbar();
27     buildFormActions(aqua.request, aqua.response); //aqua response
28     processRequest(aqua.request); //aqua response
29 }
30 catch (err)
31 {
32     aqua.response.write("A server error occurred. Please contact your server admin");
33     aqua.console.println(err.lineNumber + ":" + err);
34 }
35
36
37
38
39
40 function processRequest(request)
41 {
42     if (request.getParameter("downloadExcel") != null)
43     {
44         // Create an excel file and download
45     }
46 }
```

File Edit Server Query Automate Query Builder Visual Analytics ER Modeler Tools DBA Tools Window Help

File System C:\SVN-AquaProjects Company Revenue AquaScripts CompanyRevenueAdvanced.xjs CompanyRevenueBasic.xjs Servers MySQL 6.0 - 38.conn User Files Custom Jar Example Database Exporter Financial Examples Master Slave Compare Math Solutions Multiserver Script Execute Random Database Generator C:\SVN-website

Editor [CompanyRevenueAdvanced.xjs]* X

16 // Specify the entry point URL. Set this to either dashboardURL or runURL depending
17 // the report in ADServer.
18
19
20 1 line deleted
21
22
23 var urlEntryPoint = aqua.project.dashboardURL + "/";
24 try
25 {
26 buildToolbar();
27 buildFormActions(aqua.request, aqua.response); //aqua response
28 processRequest(aqua.request); //aqua response
29 }
30 catch (err)
31 {
32 aqua.response.write("A server error occurred. Please contact your server admin");
33 aqua.console.println(err.lineNumber + ":" + err);
34 }
35
36
37
38
39
40 function processRequest(request)
41 {
42 if (request.getParameter("downloadExcel") != null)
43 {
44 // Create an excel file and download
45 }
46 }

38 | 1:0 INS PC 142 : 244 : 2,513 MB



In this chapter ...

유지보수(Maintenance)

- 10.1 유지보수의 소개
- 10.2 유지보수 작업 과정
- 10.3 형상 관리
- 10.4 역공학
- 10.5 리엔지니어링
- 10.6 유지보수 도구



Homework #5

유지보수(Maintenance)