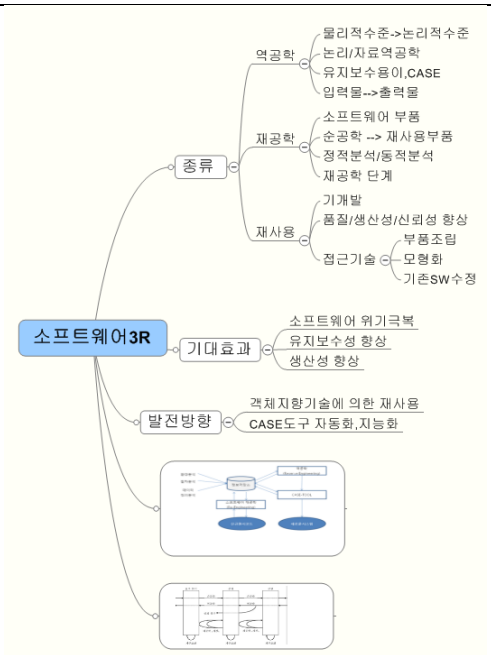
온달의 IT세상 – REVIEW

Niceit.tistory.com

소프트웨어 3R(역공학/재공학/재사용)



REVIEW

2009.04.18

➢ 목차

I. 3R 개요
II. 3R 개념
III. 역공학(Reverse Engineering)
IV. 재공학(Re-Engineering)
V. 재사용(Reuse)
VI. 3R의 발전방향

April 18, 2009

I. 3R개요

I. 3R개요

가. 정의

소프트웨어 개발 후 유지보수 관리를 용이하게 하기 위한 3가지 방법

나. 3의 종류

- 역공학(Reverse Engineering)
- 재공학(Re-Engineering)
- 재사용(Reuse)

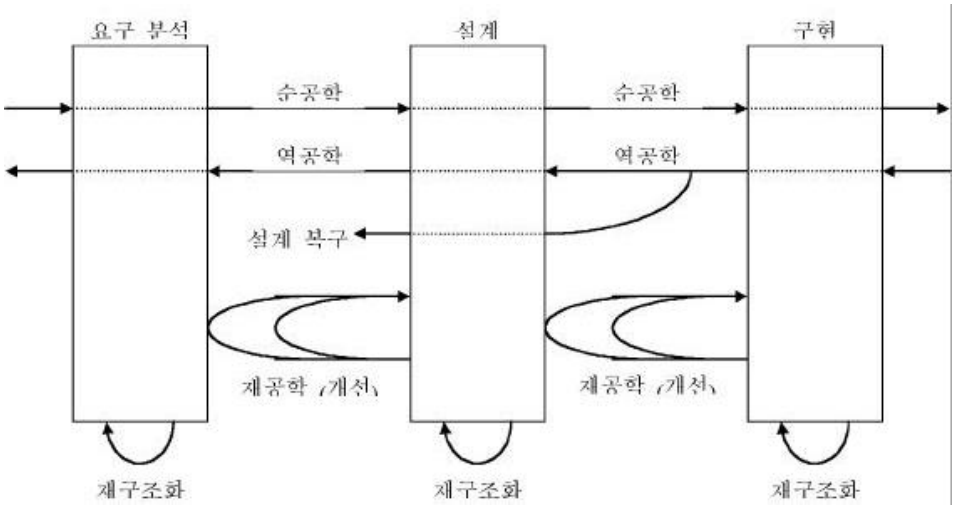
다. 등장배경

- 소프트웨어 위기 극복 – 신규 업무개발 적체/유지보수의 어려움
- 소프트웨어 개발 생산성 향상 요구
- 끝없는 요구 및 변경

II. 3R개념

II. 3R개념

가. 3R개념도



나. 순공학

- 고수준의 추상적 개념 → 물리적 실현(구체화)
- 요구분석 → 설계 → 구현

다. 역공학

- 물리적 실현체에 의존하여 고수준의 추상적 개념으로의 데이터의 기술과 프로그램 논리(업무규칙)의 추출, 표준화,문서화,설계복원

라. 재구조화

- 코드 구조를 개량하기 위하여 프로그램 기능을 변경하지 않은 상태에서 소스코드의 재편성 및 개선

III. 역공학(Reverse Engineering)

III. 역공학(Reserve Engineering)

가. 정의

- 자동화된 도구의 도움으로 물리적인 수준의 소프트웨어 정보를 논리적인 소프트웨어 정보의 서술로 추출하는 프로세스
- 예) 하드웨어 분야(완제품 → 설계를 해독해내는 작업)

나. 역공학의 이점

- 상용화된 소프트웨어의 분석을 도와줌
- 시스템과 일치하는 문서생성으로 시스템의 정보 획득
- 기존 시스템의 자료분석으로 유지보수가 용이해짐
- 유지보수자가 설계 수준에서 시스템에 대한 향상 작업을 가능케 함
- 기존 시스템 정보를 Repository에 보관하여 CASE-TOOL의 사용을 가능케 함

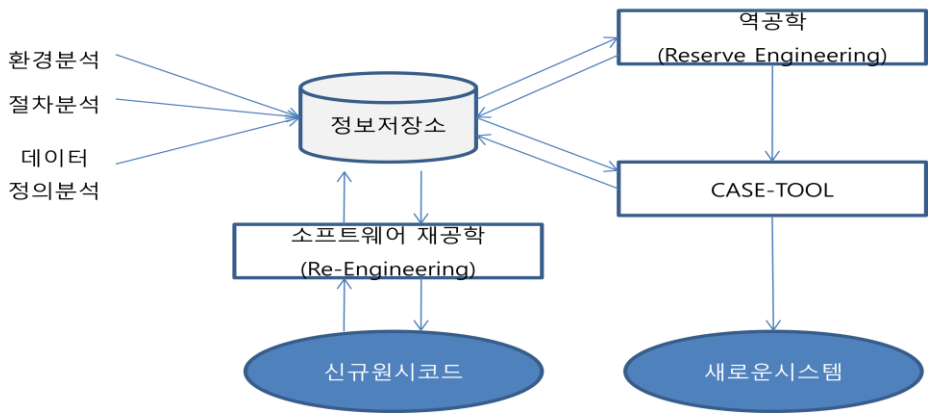
다. 역공학에 사용되는 입력물과 그에 따른 산출물

입 력 물	출 력 물(산출물)
원시코드, 목적코드, 작업제어절차, 라이브러리, 디스크 디렉토리, 텍스트자료, 데이터베이스 구조, 입출력 형태와 자료, 기타 분석을 도와주는 각종 문서	구조도, 자료사전, 자료흐름도, 제어흐름 그래프, 개체 관계도, 자료흐름 그래프

라. 역공학의 종류 및 사용도구

논리역공학	자료역공학
- 원시코드로부터 정보를 뽑아내어 물리적인 설계정보를 얻어내어 Repository에 저장 - 원시코드 → 정보추출 → 물리 설계정보	- 기존 데이터베이스를 수정하거나 새로운 데이터베이스 관리 시스템으로 전이하는 역할 수행

마. 역공학,재공학,CASE정보 저장소 관계



IV. 재공학(Re-Engineering)

IV.재공학(Re-Engineering)

가. 정의

역공학, 재구조화에 의해 추출된 소프트웨어 부품을 이용하여 다시 순공학을 구현함으로써 재사용 부품으로 만들어 내는 핵심 역할 기술

나. 목적

- 현재 시스템의 유지보수 및 향상
- 시스템의 이해와 변형을 용이하게 하며 유지보수비용 및 시간 절감
- 표준 준수를 통하여 CASE-TOOL사용 용이

다. 재공학 단계

원시코드 추출	원시코드나 데이터베이스 정보 등에서 필요 정보 추출 및 저장
역공학 단계	정보 저장소의 정보들로부터 새로운 정보인식, 그들의 성질과 선택사항 결정
시스템 향상과 검증	시스템 분석가가 정보구조,정보흐름과 같은 응용분야에 대한 명세단계 정의를 향상
순공학 단계	응용시스템을 정의하는 명세를 구현하는 과정
설계와 최적화	파일과 데이터베이스 설계, 프로그램 명세서를 이용하여 최적화 설계로 기능 향상
원시코드의 생성	정보저장소로부터 구현 단계의 설계 내역을 바탕으로 원시코드 생성

라. 재공학 분석방법

정 적 분 석	동 적 분 석
- 코드의 라인수, 기능점수 - IF나 GOTO문의 수 - 수행되지 않는 라인 - 사용되지 않는 변수 - 비구조적 코드의 비율 - 복잡도 - 기타 품질 척도	- 프로그램의 수행경로(PATH) 수 - 기존 코드의 분석 및 이해 소요시간 - 데이터 개체나 항목을 검색 수정하는 라인들 - 하나의 변수 값을 바꿀 때 나타나는 프로그램의 반응

V. 재사용(Reuse)

VI. 소프트웨어공학 3R의 발전방향

V. 재사용(Reuse)

가. 정의

이미 개발되어 그 기능 및 성능, 품질을 인정 받았던 소프트웨어의 전체 또는 일부분을 다시 사용하여 새롭게 개발되는 소프트웨어의 품질과 생산성 및 신뢰성을 높이고 개발 기간과 비용을 감소 시켜주는 소프트웨어 위기극복 방안

나. 재사용의 원칙

- 범용성, 모듈성(최소한의 결합도, 최대한의 응집도)
- 하드웨어/소프트웨어(OS/DBMS) 독립성, 자기 문서화
- 기타(일반성,신뢰성,효율성,간편성,구현성,정교성,이식성)

다. 기술적 접근방법

방 안	설 명
부품조립	- 부품전략, 빌딩블럭조립, 결합방법 - 소프트웨어부품 → 필요한 부품만 선별하여 결합
모 형 화	- 패턴 재사용 - 목표 SW의 일반모형 → 매개변수 적용에 따른 S/W생성
기존SW수정	- 일명 전체 전략(Whole Approach) - 기존 SW → 새로운 SW로 발전 - 전체 SW모습 정립→새모듈 개발→기존모듈 수정/삭제 - 부작용에 대비(Side Effect)

라. SW 재사용의 효과

- 품질/생산성 향상
- 개발기간과 비용의 감소
- 시스템 및 구축 방법에 대한 지식의 공유(산출물)

VI. 소프트웨어 공학 3R의 발전방향

- 90년대 다운사이징 + 역공학, 재공학 → 객체지향기술에 의한 재사용
- 재사용 가능한 컴포넌트 개발의 확산 → 생산성/유지보수성 향상에 기여
- CASE도구 등의 사용 확산 → 설계 내용의 DB화 → 재사용
- CASE도구의 지능화, 자동화