정 도균, 이 남용
jdk@beans.soongsil.ac.kr
nylee@computing.soongsil.ac.kr

소프트웨어공학 연구실
숭실대학교 대학원 컴퓨터학과

- 연구 목적
- 연구 범위
- 연구 방법
- 관련 연구
- 컴포넌트 재사용성 모델
- 결론 및 향후 연구과제
- 참조문헌

- 컴포넌트 기반의 시스템 개발에서의 SW 재사용성 측정을 위한 접근방법에 대해 논의
- 컴포넌트 개발 및 컴포넌트 기반 개발에 있어서의 재사용성 개념모델을 제안
- 컴포넌트 개발 및 컴포넌트 기반 개발과 관련된 생명주기 초기단계에서의 SW 재사용과 관련된 가능성을 제시

- 컴포넌트 개발 및 컴포넌트 기반 개발에서의 소프트웨어 재사용성 모델 정의와 분류 체계 확립

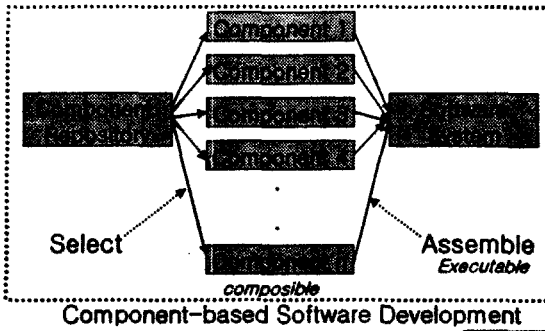
- 기존 소프트웨어 품질평가 기법에 있어서의 재사용과 관련된 모델에 대해서 자세히 알아보고 컴포넌트 기반 개발에 있어서 기존 소프트웨어 품질 평가기법에서의 재사용성 모델의 적용의 한계를 기술, 컴포넌트의 특성을 고려하여 적용가능한 기법 및 요소를 파악하여 확장

- Standish Group의 1999년 CHAOS Report 개발 프로젝트의 26% 정도만이 주어진 예산과 시간 안에 품질이 보장된 제품을 개발하는데 성공
- 소프트웨어 위기를 극복할 수 있는 한가지의 해결책 → 컴포넌트 소프트웨어
- Yourdon E. : 컴포넌트의 사용은 개발사이클의 70%, 개발비용의 80%의 단축

❑ 컴포넌트 : 부품을 조립해서 제품을 만들어 내는 것처럼 부품화된 SW들을 조립하여 완성된 소프트웨어를 구성하는데 이러한 독립된 단위기능의 소프트웨어 부품

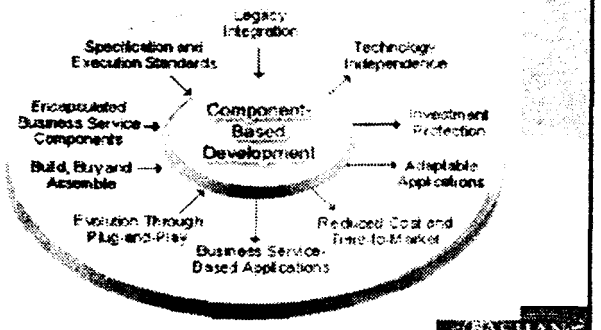
❑ 컴포넌트 방식 소프트웨어 구성

02-02-24



Component-based Software Development

02-02-24



02-02-24

❑ 재사용성
- "프로그램 또는 프로그램의 부분을 다른 어플리케이션에서 재사용할 수 있는 정도. 즉, 프로그램이 수행하는 패키징과 범위에 관련되어 재사용되어지는 정도" [IEEE90]

❑ 재사용성은 재사용 가능한 코드보다 좀 더 광범위한 개념

02-02-24

❑ "재사용 가능한 컴포넌트의 가능성"

❑ 재사용성은 재사용 가능한 컴포넌트들로부터 SW 시스템을 개발하는데 기초를 두는 SW개발 접근방법

02-02-24

❑ Denvanbu의 재사용 척도

$R_b(S) = [C_{noreuse} - C_{reuse}] / C_{noreuse}$

$C_{noreuse}$: 재사용없이 S를 개발하는 비용이다.
C_{reuse} : 재사용을 이용, S를 개발하는 비용

$0 <= R_b(S) <= 1$

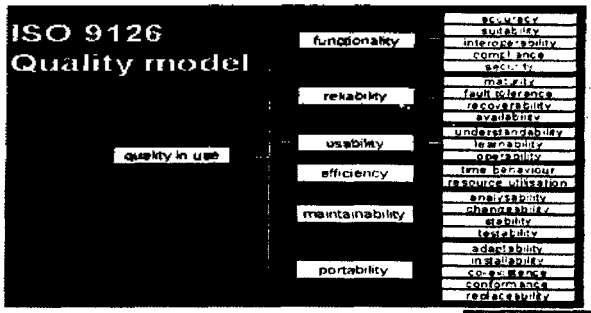
02-02-24

❑ Basili의 재사용 수준

$R_{lev} = OBJ_{reused} / OBJ_{built}$

OBJ_{reused} : 시스템에서 재사용된 객체들의 수
OBJ_{built} : 시스템에서 만들어진 객체들의 수

02-02-24



02-02-24

❑ ISO/IEC 9126 포괄적으로 품질 특성 정의
❑ 소프트웨어 품질은 "소프트웨어 생산물의 품질을 표현하고 평가할 수 있는 속성"이라고 정의 [ISO/IEC 9126]
❑ 6가지 품질 요소 : **기능성, 신뢰성, 사용성 효율성, 이식성, 유지보수성**
❑ ISO 표준에서는 품질 평가 프로세스 모형만을 제시, 방법에 대한 언급 없음

02-02-24

❑ 4개의 재사용성 Factor, 각 Factor에 대한 Criteria, 각 Criteria에 대한 Metrics
- **호환성(Portability)** : 다른 컴퓨터 시스템으로 부품이 전환 되기 쉬운 정도 – Modularity, Environment indepence
- **유연성(flexibility)** : 다른 기능적 요구를 위하여 부품을 개작하거나 변경하여 다른 환경에서 사용하기 용이한 정도 – Generality, Modularity
- **이해용이성(Understandability)** : 프로그래머가 컴포넌트를 이해하기 쉬운 정도 – Code Complexity, Self Descriptiveness, Documentation quality, Module complexity
- **적합성(Confidence)** : 새로운 컴포넌트가 특정 시간 내에 결함 없이 수행되는 주관적 가능성 – Model complexity, Observed reliability, Error tolerance

02-02-24

Criteria	Metrics
Generality	Generality Checklist
Modularity	Code / number of methods
Environment Indepence	Machine-dependent code / executable code System-dependent code / executable code
Code Complexity	Cyclomatic complexity
Self Descriptiveness	Comments / Source Code Self-Descriptiveness Checklist
Documentation Quality	Documentation / Source Code Documentation Checklist
Module Complexity	Fan-in, Fan-out
Reliability	Total number of test Number of observed errors
Error tolerance	Error tolerance checklist

02-02-24

❑ 품질과 재사용성에 대한 지시자로서 4개의 메트릭스를 추천

- **열람 되어진 횟수** : 누군가가 재사용 모듈로 고려한 횟수
- **재사용 되어진 횟수** : 누군가가 실제적으로 모듈을 재사용한 횟수
- **복잡도** : 코드의 복잡도, 기본적으로 McCabe 복잡도에 기반
- **문제점 보고서의 수** : 모듈에서 발견되어진 결점의 수

❑ 문제점

02-02-24

❑ Bieman, Karunanithi는 수정없이 그대로 재사용, 수정하여 재사용하는 형태에 대하여 재사용성을 나타내는 Metric를 제시

02-02-24

정적인 구조

디자인/코드를 구체화한다

어플리케이션 흐름을 포함하지 않는다.

어플리케이션에 포함되어질 수 있는 특정 기능성을 제공한다.

동적인 구조

프로세스/아키텍처/디자인/구현을 구체화한다.

Semi-Finished 어플리케이션.

어플리케이션의 흐름을 포함한다.

02-02-24

❑ 환경독립/플랫폼 바이너리 독립
❑ 위치 투명성
❑ 구현과 분리된 인터페이스
❑ 자기 설명적인 인터페이스
❑ Self-Descriptiveness
❑ 보편적인 응용 컴포넌트 프레임워크
❑ Mix and Match
❑ Plug and Play
❑ 캡슐화(Inheritance, Aggreation, Association)

02-02-24

❑ 어플리케이션 빌딩 블록
❑ 퍼시스턴스 지원
❑ 독립적이만 최소한의 특정 의존성 허용
❑ 블랙박스 또는 화이트 박스 배포
❑ 교체성
❑ 실행 타임시 다중 컴포넌트 오브젝트 지원

02-02-24

❑ 컴포넌트 기반 소프트웨어의 개발은 OOP를 기반으로 하지만 명세, 구축, 조립 지향적

❑ 컴포넌트의 특성이 기존의 OOP와는 다르기 때문에 새로운 재사용성 모델이 필요

❑ 또한 기존 제시되어진 소프트웨어의 품질과 재사용 평가기법은 매우 일반적으로 정의

02-02-24

❑ 재사용성에 대한 개념을 특징 지을 수 있는 방법을 제공

❑ 재사용성을 측정할 수 있는 방법을 제공

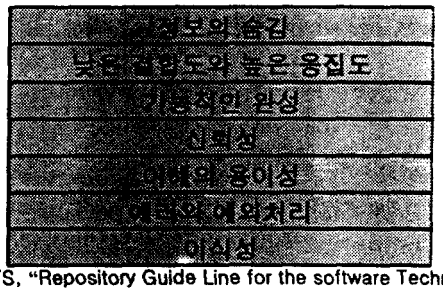
02-02-24

❑ 기존의 소프트웨어 재사용성 모델 수용

❑ 컴포넌트의 특성을 고려하여
적용 가능한 기법 및 요소를 파악

❑ 컴포넌트의 특성을 고려하여 확장

02-02-24



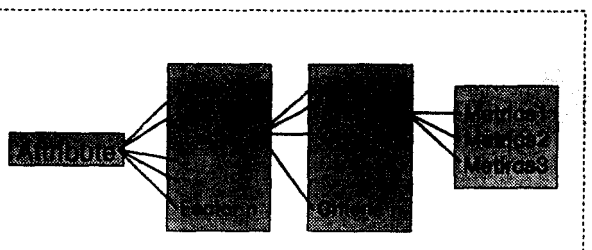
STATS, "Repository Guide Line for the software Technology for Adaptable, Reliable Systems(STARS) Program

02-02-24

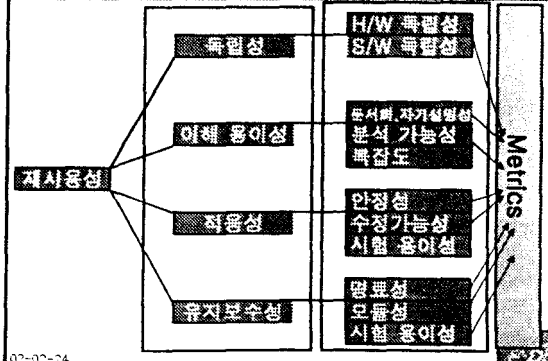
❑ 개발자나 관리자에게 질문이 주어짐.
❑ Factor를 생성
❑ 각 Factor를 그들의 목적을 만족시키는 행위들의 집합으로 분해. (Criteria)
❑ Metrics는 각 재사용 비즈니스에 의해 결정되어야 함.

02-02-24

❑ Factor/Criteria/Metrics(FCM) 모델은 IEEE와 ISO에 의해 기본적인 SW평가방법으로써 사용
❑ SW의 속성들은 Factor들로 분해
❑ 각 Factor들은 Criteria들로 분해
❑ 각 Criteria들은 Metrics에 의해 측정될 수 있음.



02-02-24



02-02-24

❑ **독립성** : 한 SW가 HW/SW에 영향을 받지 않고 다른 시스템이나 환경으로 쉽게 옮겨질 수 있는 정도

❑ **이해 용이성** : 논리적 개념과 적용성을 이해하기 위한 사용자의 노력과 관련된 SW의 속성

❑ **유지보수성** :사용중인 프로그램의 에러를 식별하고 수정하는 데 요구되는 노력의 정도로서 유연성을 의미

❑ **적용성** : 다른 시스템 규약과 사용자 요구에 만족하는 정도

02-02-24

❑ **수정용이성** : 수정 결점 제거 또는 환경변화에 요구되는 노력과 관련된 SW의 속성

❑ **시험용이성** : 프로그램이 그것의 의도된 기능을 수행함을 보장하기 위해 프로그램을 시험하는데 드는 노력의 정도, 수정한 소프트웨어를 확인하기 위하여 요구되는 노력과 관련된 SW의 속성

❑ **모듈성** :소프트웨어를 기능적으로 독립된 여러 개의 블랙박스로 어느 정도 모듈화 되었는지의 정도

02-02-24

❑ **S/W, H/W 독립성** : 이식성 = 1 - ET/ER

ET : 시스템을 목표환경으로 옮기는데 필요한 자원의 양
ER : 주어진 환경에서 시스템을 만드는데 필요한 자원의 양

❑ **자기 설명성**

Comments/Source Code,
Self-Descriptiveness Checklist

❑ **이해 용이성**

Code Complexity, Self Descriptiveness
Documentation Quality, Module Complexity

❑ **모듈성** : m_CBO, m_LCOM, m_RFC

02-02-24

❑ 컴포넌트 재사용성 모델은 어떻게 재사용 가능한 컴포넌트들을 개발할 것인가와 현재 존재하는 SW사이에서 재사용 가능한 컴포넌트을 식별하는데 직관적 도움을 준다.

❑ 컴포넌트-기반 소프트웨어 재사용성 모델을 이용한 정량적인 수치나 데이터로 측정할 수 있는 척도의 개발이 필요

❑ 재사 모델의 검증

❑ 컴포넌트 기반 SW (재사용성)시험 기술의 개발.적용

02-02-24

- Devanbu, P., et al., "Analytical and Empirical Evaluation of Software Reuse Metrics," Technical Report, Computer Science Department, University of Maryland, August 1995.
- Karlsson, Even-Andre, Guttorm Sindre, and Tor Stalhane, "Techniques for Making More Reusable Component", REBOOT Technical Report #41, 7 June 1992
- NATO, "Standard for Management of a Reusable Software Component Library," NATO Communications and Information Systems Agency, 16 August 1991
- STARTS, "Repository Guidelines for the Software Technology for Adaptable, Reliable Systems(STARS) Program," CDRL Sequence Number 060, 15 March 1989
- Pressman, R.S Software Engineering : A Practitioner's Approach 5th Edition McGraw-Hill, 2 June 2000

02-02-24

- Basili, V.R., L.C. Briand, and W.M Thomas, "Domain Analysis for the Reuse of Software Development Experiences," Proc. Of the 19th Annual Software Engineering Workshop, NASA GSFC, Greenbelt, MD, December 1994
- Nam-Yong Lee, "Object-Oriented Technology and Software Reuse", KOMG [White Paper], 1999. 10. 29
- 배두환, e-Business를 위한 CBD개발방법, www.component.or.kr
- Kyungju Kim "Enterprise CBD" COOL Software Korea
- 권오천, 신규상, 전진욱, "컴포넌트 기반 개발기술 검토 및 동향", ETRI IT정보센터, 1999. 06. 05
- 김수동, 류성열, "웹 기반 소프트웨어 품질 평가 기술 개발", ETRI S/W공학연구원 연차보고서 11, 2001
- Yourdon, E., "Software Reuse," Application Development Strategies, vol. 6, no 12, December 1994, pp 1-16

02-02-24