

소프트웨어 벤치마크 테스트 프로세스 개발

오영배*, 김재웅**, 정영은**, 신석규**
*수원여대 컴퓨터응용학부, **한국정보통신기술협회 소프트웨어시험센터

Development of a Software Benchmark Test Process

Young Bae Oh, Jae Woong Kim, Young Eun Jung, Seok Kyu Shin
Suwon Women's College, Telecommunications Technology Association
E-mail : yboh@suwon-c.ac.kr, {jwkim, yejung, skshin}@tta.or.kr

요약

제품으로서의 소프트웨어가 급속히 보급되면서 소프트웨어 제품 품질 평가의 필요성이 증가하고 있다. 소프트웨어는 하드웨어와 다른 무형의 제품으로서 기술 및 품질에 대한 표준화와 시험 평가 방법의 도출이 어렵다. 최근에 개발된 소프트웨어 품질 모델과 도메인별 제품 표준화 노력으로 소프트웨어 제품 평가 방법이 개발되고 있다. 이와 더불어 제품을 비교 평가하여 소비자에게 제품을 선택할 수 있는 기준을 줄 수 있는 벤치마크 테스트의 필요성이 제기되고 있다.

본 연구에서는 하드웨어와 비교되는 소프트웨어 벤치마크 테스트의 특성을 살펴보고 본 연구에서 개발한 벤치마크 테스트를 위한 표준 프로세스 개발 결과를 제시하고 개발된 프로세스의 각 단계별 활동에 대하여 논한다.

1. 서론

벤치마크 테스트는 하드웨어 부문에서 시행되어 오고 있다. 하드웨어는 기술 및 부품 등이 비교적 표준화가 잘 되어있어서 테스트 항목 도출 및 기준 설정이 용이하다. 그러나 소프트웨어는 무형의 제품으로서 기술 및 품질에 대한 표준화가 매우 어려운 분야이다. 따라서 소프트웨어의 품질을 평가하는데 애로를 많이 겪고 있으며 생산자나 소비자 모두 제품의 비교 평가 및 품질 신뢰도를 파악하기 힘들다. 그러나 최근의 산업은 하드웨어보다는 소프트웨어가 점점 주류를 이루어 가고 있고 소프트웨어 제품 벤치마크의 필요성이 증대되고 있다. 따라서 벤치마크의 표준화된 절차와 평가 기준을 개발하여 소비자는 물론 생산자의 요구에 부응할 필요가 있다.

소프트웨어 제품의 평가는 소프트웨어 프로세스 평가와는 사뭇 다르다. 지난 10년간 SEI(Software Engineering Institute)의 CMM(Capacity Maturity Model)[1] 등 관련 모델과 표준의 제정 등 소프트웨어 프로세스 평가 분야는 활발한 작업을 하였으

나 소프트웨어 제품 평가 분야는 그렇지 못했다. 그 두 가지 이유는 다음과 같다.

- 소프트웨어 구매자의 입장에서 제품의 평가 보다는 프로세스의 평가가 예측능력을 높힐 수 있다는 판단

- 소프트웨어 제품 평가 기법의 미비

소프트웨어 제품 평가 기법이 미비한 이유는 이것이 어렵다는 것이다. 그렇지만 제품 평가가 불가능한 것은 아니며 소프트웨어 테스트, 검사 및 측정 메트릭 개발과 평가 절차의 확립으로 평가 기법을 개발할 수 있으며 소프트웨어 제품 평가를 통하여 그동안 프로세스 평가를 통한 제품 품질에 대한 간접적 평가에서 직접적인 제품 평가가 가능하게 되는 것이다.

그동안 하드웨어 부문에서 벤치마크 테스트가 시행되어 온 것은 하드웨어의 기술 및 부품 표준화가 잘 되어 있어 제품간 비교 평가가 용이하다는데 기인하고 있다. 소프트웨어 부문에서는 미들웨어, DBMS 등이 표준화가 되어 있고 이 부문에서 벤치마크 테스트가 시행되어 오고 있다. 어플리케이션 소프트웨어는 소프트웨어의 특성 상 하드웨어에

비해 변경이 용이하며(Flexible) 매우 다양한 영역(Domain)이 존재하고 동일 영역의 소프트웨어도 표준화가 되어 있지 않아 공통성을 찾기가 어렵기 때문에 제품을 비교 테스트하는 벤치마크 테스트가 어려운 요인이 된다.

[표 1] 벤치마크 테스트의 난이도

구분	표준화	난이도
하드웨어	높음	낮음
미들웨어	중간	중간
어플리케이션 소프트웨어	낮음	높음

2. 관련연구

“벤치마크는 여러 가지 컴퓨터간의 상대적인 성능을 판단하기 위한 표준” 이라고 정의한다[2]. 즉 벤치마크는 주로 컴퓨터 시스템의 성능을 비교 평가하는 도구로 활용되고 있으며 주요한 벤치마크 사례는 다음과 같다.

1) SPEC (System Performance Evaluation Cooperative)

워크스테이션급 서버를 제조하는 업체들 간의 컨소시엄으로 추진되어 SPEC[3]이 출범하였다. SPEC은 최신 컴퓨터 시스템의 성능 평가를 위해 표준화된 벤치마킹 테스트와 평가표를 작성하여 기준을 세우고, 이러한 벤치마크를 유지하고 승인하는 역할을 계속해 오고 있다.

SPEC의 벤치마크는 인텔이나 AMD와 같은 CPU제조업체들에 의해 자주 사용되며, 과학이나 기술적인 컴퓨팅 분야에서도 많이 사용된다. 최근에는 자바 어플리케이션을 실행하는 서버의 성능을 시험하는 벤치마크를 개발하였다.

[표 2] 현재 시행하고 있는 주요 벤치마크

SPEC 벤치마크 CODE	벤치마크 내용
SPEC CPU2000	CPU 성능 벤치마크
SPEC JBB2000	자바 어플리케이션을 실행하는 서버 벤치마크
SPEC JVM98	자바 머신 벤치마크
SPEC MAIL2001	메일서버 벤치마크
SPEC WEB99	웹 서버 벤치마크

2) TPC (Transaction Processing Performance Council)

트랜잭션 프로세싱 관련 업체들과 공급업체들 간 컨소시엄으로 TPC[4]가 출범하였다. 많은 관련 대기업들이 참여하여 사실상 대형 트랜잭션 기반 시스템의 표준이라 할 만큼 성장하였다. TPC테스트는 까다로운 절차를 보유하고 있어서 Sybase와 Compaq 등의 일부 회사는 TPC테스트 팀을 별도로 운영하고 있다.

TPC는 객관적이고 입증 가능한 TPC 성능 데이터를 보유하고 벤치마크를 보급하고 있으며, 트랜잭션 처리과정과 데이터베이스 벤치마크를 정의하기 위한 비영리 기관으로 발전하고 있다.

TPC는 대형 트랜잭션 기반 시스템의 성능을 측정하기 위한 표준으로서 TPC테스트의 결과는 TPC에 의해 감사된다.

[표 3] 주요 표준 목록

- TPC-B : Database server test
- TPC-C : OLTP (OnLine Transaction Processing)
- TPC-H : Decision Support for Ad Hoc Queries
- TPC-w : Web e-Commerce
- TPC-R : Decision Support for Business Reporting

3) ISBSG Data Repository

호주에 본부를 두고 있는 ISBSG(International Software Benchmarking Standards Group)[5]는 소프트웨어 프로젝트가 어떻게 수행되었는지에 대한 지식을 축적하여 분석과 연구를 통해 과거의 경험으로부터 교훈을 얻고 소프트웨어 개발과정에서의 교훈을 전파할 목적으로 설립되었다.

주로 프로젝트의 효과적 수행을 위하여 다음 단계를 거쳐 벤치마크를 해준다.

가) 프로젝트 양식 작성 : 프로젝트 내용을 ISBSG의 지정된 양식(On-line, Off-line)에 따라 작성한다.

나) 프로젝트의 제출 : 프로젝트 내용을 e-mail, 소포(디스켓, 하드카피)로 ISBSG에 제출한다.

다) 저장소 검색 : ISBSG는 제출한 프로젝트의 내용을 검토한 후, ISBSG내의 저장소로부터 이와 유사한 프로젝트가 있었는지를 찾는다. 개발 플랫폼과 랭귀지 유형에 맞는 기존 프로젝트를 찾아 비교 대상으로 삼는다.

라) 프로젝트의 측정 : ISBSG에서 세운 기준에 부합하는 형식에 따라, 방정식, 대조, 비교의 방법을 이용하여 측정한다. 이러한 측정 기준은 많은 항목을 보유하고 있으며, 한 가지 방식에 의존하지 않고 미시측정과 거시측정 양쪽의 조화를 이용하여 최대한 정확한 결과를 증명한다.

마) 프로젝트의 보고 : 프로젝트의 측정과 평가가 완료되면 그 결과를 되돌려 준다. 이 때, 프로젝트 데이터의 비밀은 보존된다.

바) 프로젝트의 저장 : 완료된 프로젝트는 ISBSG의 저장소에 저장된다. 추후에 이와 비슷한 사례의 선례로 참고된다.

4) SCOPE (Software Certification On Program in Europe)

ESPRIT 프로젝트 SCOPE[6]는 처음으로 소프트웨어 제품평가의 인증제도를 세우기 위한 국제적인 시도라 할 수 있으며, 시험 평가 분석의 결과로 개발되었으며, '평가자 가이드'로 문서화되어 ISO에 제출되었다. 이것이 지금의 ISO/IEC 14598-5 평가 자동 프로세스로 채택되어 발간되었다.

비록 인증제도를 세우지는 못하였지만, 이 프로젝트의 주요 성과는 현재 전 세계적으로 널리 채택되어 평가의 기초로 사용되고 있는 소프트웨어 품질평가의 기본틀을 제시한 것이라고 할 수 있다.

5) MicroScope

델타의 DELTA 소프트웨어 공학 (Delta Software Technology)에서 1991년 소프트웨어 평가에 대한 접근방식을 제시하였다. MicroScope[6]는 SCOPE의 결과를 결과로 기본으로 하여 ISO/IEC 9126과 ISO/IEC 14598-5를 따르고 있다. MicroScope는 여러 상황에서 사용되고 있는데, 일반적으로 규정된 외부 표준이나 규제를 따르고 있는지를 기술하고 문서화의 수준과 소프트웨어 제품의 안전성이 만족스러운지를 확인하는 것이다. 평가는 인증 의뢰인과 DELTA사이에 소프트웨어 제품의 어느 품질 특성을 고려하고 어느 평가 모듈을 사용할 것인지에 대한 합의를 바탕으로 한다.

MicroScope의 평가 항목은 체크리스트 형식으로 되어있으며, ISO/IEC 9126의 6가지 품질특성을 모두 망라하는 12개의 모듈로 구성되어 있다. 제품의 중대성에 따라 관련 특성 각각에 대하여 4가지 가능한 수준에서 평가가 수행된다.

상기의 사례에서 SPEC은 하드웨어 벤치마크이며 TPC는 미들웨어이고 ISBSG는 프로세스에 대

한 벤치마크이며, SCOPE과 MicroScope는 벤치마크라기 보다는 소프트웨어 제품 평가라 할 수 있다. 즉 본격적인 어플리케이션 소프트웨어 수준의 벤치마크 사례는 찾아보기 어려운 실정이다.

3. 벤치마크 테스트 프로세스 개발

본 연구는 미들웨어를 포함한 어플리케이션 소프트웨어의 벤치마크 테스트를 하기 위해 도메인 독립적인 벤치마크 프로세스를 정의하는 방법을 개발하였다. 본 프로세스에는 소프트웨어 벤치마크를 하기 위한 전체적인 과정을 안내하고 있으며, 도메인을 선정하고 도메인별 벤치마크 항목을 도출하기 위한 절차를 포함하고 있어 소프트웨어 벤치마크 테스트를 위한 지침으로 활용하기 위한 것이다.

본 프로세스의 특징은 다음과 같다.

- 도메인에 독립적인 벤치마크 방법론 제공
- 시험원의 작업 매뉴얼로 활용 가능
- 소프트웨어 인증 테스트 프로세스와의 일관성 유지
- 산출물 문서화의 과대화 방지
- 작업절차와 산출물 목차와의 일관성 유지
- 특정 도메인을 대상으로한 방법론 적용사례 제시
- 특정 도메인들에 대한 벤치마크 기법 제시

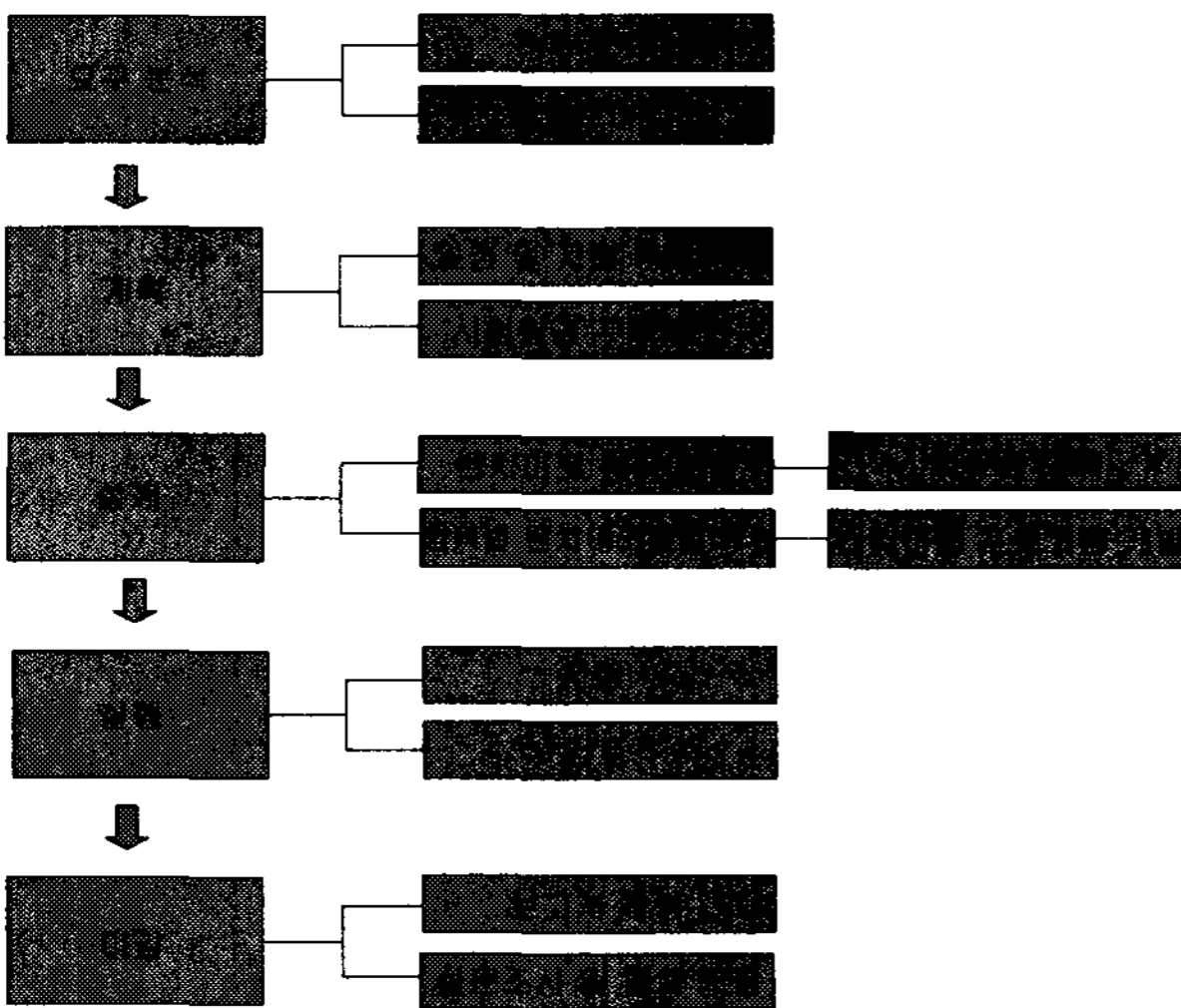
본 방법론이 제시하는 벤치마크 프로세스는 작업 절차와 산출물로 구성되며 작업 절차는 단계, 활동 및 작업으로 구분되고, 산출물은 산출물 양식과 입력 양식으로 구성된다. 작업 절차의 단계, 활동, 작업은 작업의 대, 중, 소 분류가 된다. 산출물 양식은 각 단계별로 작성해야하는 문서화의 양식을 정의한다. 입력 양식은 해당 작업에 입력으로 들어와야 하는 양식을 정의한다. 본 방법론은 특정 도메인에 대한 방법론 적용 사례를 포함하고 있으며, 이것은 본 방법론을 처음 적용하는 시험원이 방법론 적용의 가이드로 활용할 수 있다. 또한 몇몇 도메인에 대한 벤치마크 기법을 제시하고 있으며, 이것은 벤치마크 기법을 개발하는데 가이드로 참고할 수 있다.

본 방법론의 구성은 다음과 같다.

- 절차서 : 5개의 단계, 16개의 활동, 48개의 작업
- 절차서의 구성 :
 - 목적
 - 활동구조
 - 주요기법

- 주요 산출물
- 주요 입력물
- 타 단계 및 활동과의 관계
- 점검사항
- 산출물 : 5개의 산출물 양식, 10개의 입력 양식
- 산출물의 구성 : - 정의
 - 항목 설명
 - 입력물
 - 관련 기법
 - 지침 및 고려사항
- 적용사례 : 1개 영역
- 기법 : 6개 영역
- 기법의 구성 : - 평가 기준
 - 벤치마크 항목선정
 - 벤치마크 기법
 - 벤치마크 시나리오
 - 테스트 케이스

본 방법론이 제시하는 벤치마크 프로세스는 [그림 1]과 같다.



[그림 1] 벤치마크 테스트 프로세스

절차서의 5개 단계별 활동 내용은 다음과 같다.

- 1) 요구분석
 - 벤치마크 할 소프트웨어에 대한 사용자의 요구사항을 충분히 이해하고 요구사항을 명확히 하기 위해 사용자 요구사항에 대한 분석 작업을 수행한다.
 - 이를 바탕으로 벤치마크 할 소프트웨어의 환경과 항목을 정의한다.
 - 실제 벤치마크를 할 수 있는지를 검증하기 위해 프로토타입 단계를 포함하여 설계단계에 앞서

준비해야 할 조건을 파악한다.

- 2) 계획
 - 요구분석 단계에서 나온 결과물들을 토대로, 계약서에 명시된 바에 따라 일정 및 자원 투입 계획을 구체적으로 산정한다.
 - 투입되어야 할 인력의 기간을 일정에 따라 날짜를 기입하고, 필요한 자원을 적절한 시기에 맞추어 공급하도록 한다.
 - 필요한 경우 교육을 받도록 계획하며, 전문가 그룹을 구성하도록 한다.
 - 시험 환경, 시험 도구 및 S/W 설정 계획을 수립한다.

- 3) 설계
 - 도메인별 전문가 그룹을 구성하여 시험 대상 주요 항목을 선정하며, 비교대상 제품별로 특징적인 부가항목을 선정한다. 벤치마크 항목이 확정되면 항목별 벤치마크 항목별 측정기준 및 공식을 도출하여 메트릭을 작성한다.
 - 벤치마크 항목 선정에 따라 시험 환경을 구축하고 시험 도구 및 소프트웨어를 설치한다.
 - 벤치마크 항목별로 벤치마크 형태, 수행순서 및 방법을 개발한다.
 - 벤치마크 항목별 특성에 따라 벤치마크 기법, 시나리오, 벤치마크 프로그램, 스크립트 및 테스트 케이스를 작성한다.

- 4) 실행
 - 벤치마크 시험을 실시한다. 벤치마크 시험은 기능성, 비기능성 및 도구적용 시험으로 구분이 되며 비교대상 제품별로 동일한 조건에서 시험을 실시하고 시험결과가 도출되면 메트릭에 따른 측정치를 분석하고 벤치마크 결과를 비교한다.
 - 시험결과를 도출하면 시험결과에 대한 문제점을 정리 검토하고, 필요에 따라 보완시험을 하며 결과를 정리한다.

- 5) 마감
 - 벤치마크 절차, 방법 및 비교분석의 결과를 정리 검토한다.
 - 벤치마크 결과에 대한 고객 의견을 조사한다.
 - 벤치마크 시험과정 및 결과를 저장한다.

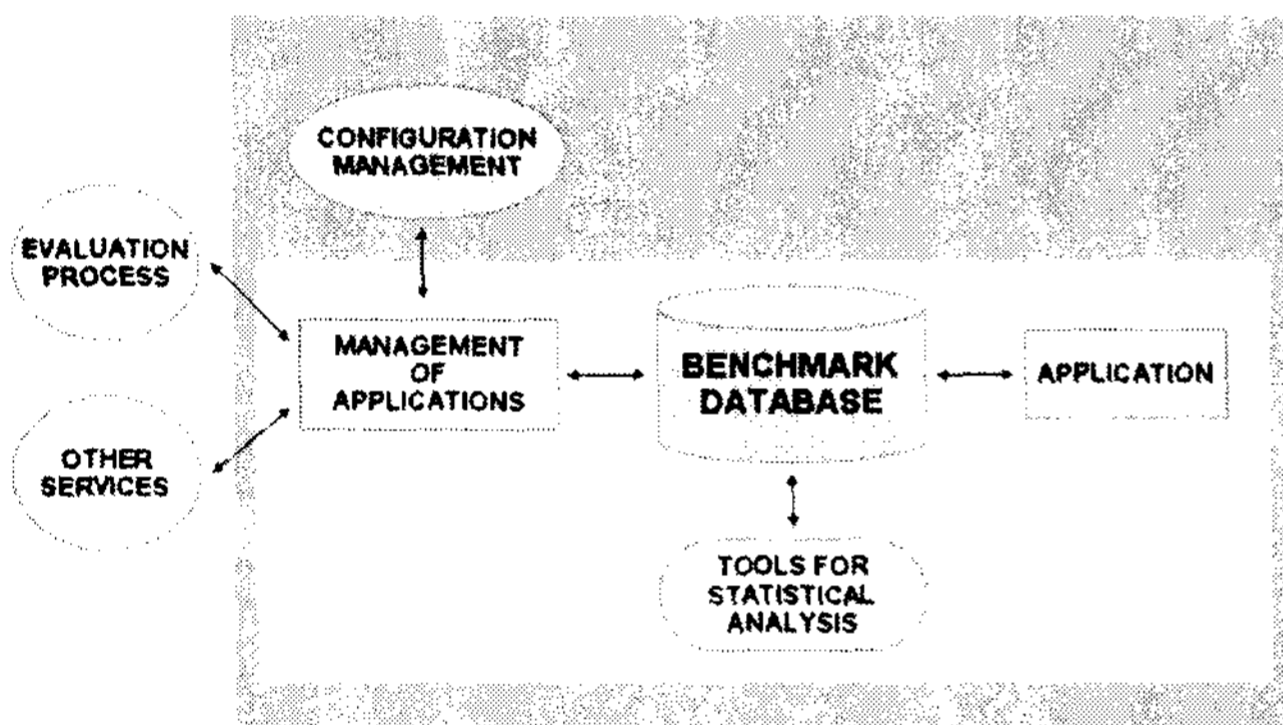
4. 결론

본 연구에서는 도메인별 실용화 방안에 초점을 둔 소프트웨어 제품 벤치마크 테스트 프로세스를 개발하였다. 따라서 도메인에 독립적인 일반 모형

개발 및 문서화, 테스트 절차 개발에 초점을 두었다. 기존의 소프트웨어 제품 시험 인증 기법과 일관성을 유지하는 수준에서의 실용성 및 적용방안을 개발하였다.

그동안 개발한 벤치마크 테스트 기법을 수정, 보완하고 파일럿 스터디를 통한 기법 및 가이드를 완성하여 벤치마크 방법론으로 완성할 예정이며 또한 본 방법론의 도메인별 적용을 통하여 실용성을 검증할 예정이다.

본 연구결과는 국내의 산업체에 적용하여 제도화하고 향후 해외 유수의 소프트웨어 제품 평가 기관과의 교류를 통한 기술 확보 및 협력 방안이 필요하며, 업체에 적용한 히스토리 데이터의 저장, 분류 및 평가 활용을 위한 데이터베이스 구축 및 도구의 개발이 필요할 것으로 본다.



[그림 2] 벤치마크 자동화 환경

참고문헌

- [1] Paulk, M.C., Weber, C.V., Curtis, B., Chrissis, M.B., The Capability Maturity Model: Guidelines for Improving the Software Process, Addison Wesley, 1995.
- [2] Price, W.J., "A Benchmark Tutorial", IEEE Micro, Oct. 1989
- [3] SPEC, SPEC CPU2000, <http://www.spec.org>, 2002
- [4] TPC, TPC-C (OLTP), <http://www.tpc.org>, 2002
- [5] ISBSG, ISBSG Data Repository Demographics, <http://www.isbsg.org.au>, 2002
- [6] Bache, R., Bazzana G., Software Metrics for Product Assessment, McGraw-Hill, 1993.