

웹 애플리케이션 개발에서의 개인 프로세스 데이터 수집 및 분석을 위한 도구

신현일*, 최호진*

*한국정보통신대학교 공학부

e-mail : {linugee, hjchoi }@icu.ac.kr

Collecting and Analyzing Personal Software Process Data in Web Application Development

Hyunil Shin*, Hojin Choi*

*Information and Communications University, School of Engineering

요 약

Personal Software Process (PSP)에서 개인 개발자들이 수행해야 되는 측정 및 분석 활동이 제시된다. 측정 및 분석 활동의 일관되고 지속적인 수행으로 개발자들은 소프트웨어 품질 향상, 보다 정확한 예측, 개인 역량의 객관적인 평가, 개인 프로세스의 정량적 관리 등의 이득을 얻을 수 있다. 측정 및 분석 활동에서 신뢰성 있는 데이터의 수집이 무엇보다 중요하다. 하지만, 데이터 수집의 높은 오버헤드(overhead)와 컨텍스트 스위칭(context switching)의 문제로 인하여 신뢰성 있는 데이터의 수집에 많은 어려움이 존재한다. 이러한 어려움을 감소시키기 위해 자동 데이터 수집 및 분석 기법이 연구되어 왔다. 본 논문에서는 기존의 자동 데이터 수집 및 분석 기법을 확장하여 웹 애플리케이션 개발에서의 개인 프로세스 데이터를 자동 수집 및 분석 하는 도구를 설명한다.

1. 서론

체계적이고 지속적인 소프트웨어 개발 프로세스의 측정 및 분석 활동은 프로젝트 관리, 프로세스 개선, 소프트웨어 품질 향상에 있어서 중요한 요소 중의 하나로 알려져 있다 [1, 2]. “측정할 수 없는 것은 예측할 수도 관리할 수도 없다 [10].”라는 말이 있듯이 측정과 분석 활동은 공학 분야에서의 핵심 활동으로 강조되고 있다. 프로젝트 관리 관점에서 측정과 분석 활동을 통한 객관적인 정보의 획득으로 인해 얻을 수 있는 이득에는 효과적인 의사소통, 프로젝트 목적의 추적, 문제의 조기 식별 및 대처, 대안절충 의사결정, 의사결정의 정당화가 있다 [2]. 이러한 이득을 얻기 위해서는 조직, 프로젝트, 팀, 개인 레벨 모두에서 측정 및 분석 활동이 유기적으로 수행되어야 한다. 즉, 개인 레벨에서 측정 및 분석 활동은 팀 레벨에서의 측정 및 분석 활동의 기반이 되며, 다시 이는 조직 및 프로젝트 레벨에서 측정 및 분석 활동의 기반이 된다. 이렇듯 모든 레벨에서의 유기적인 측정 및 분석 활동을 위해서는 개인 레벨에서의 지속적인 측정

및 분석 활동이 무엇보다 중요해 지며, PSP[3]에서 개인 프로세스의 측정 및 분석 기법이 제시된다.

소프트웨어 개발자 개인의 소프트웨어 개발 역량을 향상시키기 위한 방법으로서 PSP 가 개발되었다 [3]. PSP 는 개인 개발 프로세스의 중요성을 강조하며 개인 프로세스 정의 및 수행, 데이터 수집 및 분석, 소프트웨어 크기와 개발 시간 예측, 계획 및 계획 추적, 설계/코드 리뷰, 소프트웨어 품질 관리에 대한 다양한 기법들을 제시한다. 개발자들이 이러한 기법들을 지속적으로 사용한다면 소프트웨어 품질 향상, 보다 정확한 예측, 개인 역량의 객관적인 평가, 개인 프로세스의 정량적 관리 등의 이득을 얻을 수 있을 것이다. 이러한 이득을 얻기 위해서는 무엇보다 개인 프로세스 상의 데이터를 수집하고 분석하는 일이 중요하게 되고, 데이터 수집 및 분석 활동에서는 신뢰성 있는 데이터 수집이 무엇보다 중요해진다. 수집된 데이터가 개인이 어떻게 개발 프로세스를 수행하였는지를 반영할 수 있어야 그것으로부터 올바른 분석을 할 수가 있다. 부정확한 데이터가 수집되거나 필요한 데이터가 수집되지 않는다면 수집된 데이터로부터 개인 프로세스가 어떻게 수행되었는지를 파악하기가 어

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2006-(C1090-0603-0032))

럽게 된다. 즉, 개인 프로세스 수행을 정확하게 반영할 수 있는 신뢰성 있는 데이터 없이는 측정 및 분석 활동을 올바르게 수행할 수 없다.

그러나 데이터 수집의 높은 오버헤드와 컨텍스트 스위칭의 문제로 신뢰성 있는 데이터를 수집하는 데 많은 어려움이 있는 걸로 알려져 있다 [4, 5, 6, 7]. 높은 오버헤드 때문에 소프트웨어 개발 일정을 맞추기에도 힘든 상황에서 당장에 이익을 가져다 주지 않는 데이터 수집 활동에 시간을 사용하기 어렵게 만든다. 개발자가 소프트웨어 개발과 데이터 수집이라는 두 가지 일을 계속적으로 반복함으로써 발생하는 컨텍스트 스위칭은 개발자가 보다 중요한 업무인 소프트웨어 개발에 장시간을 집중하기 어렵게 한다. 이러한 문제점을 해결하기 위해 개인 프로세스 데이터를 자동으로 수집 및 분석하는 도구가 개발되었다 [4, 5, 6]. 데이터 수집 및 분석 도구의 핵심으로는 개발에 사용되는 다양한 도구들에 플러그인 형태로 붙어 필요한 데이터를 자동으로 수집하는 소프트웨어가 있다. 개발 도구, 문서 도구, 빌드 도구, 테스트 프레임워크, 형상 관리 시스템 등의 플러그인 소프트웨어들이 개발 시간, 결함, 소프트웨어 크기, 테스트 커버리지 (coverage) 등의 데이터를 자동으로 수집하고, 수집된 데이터를 서버로 보낸다. 개발자들은 서버에 접속하여 수집된 데이터와 수집된 데이터에 대한 다양한 분석 결과를 볼 수 있다.

소프트웨어 개발에서 개발 환경은 개발하려는 소프트웨어 특성, 개발 팀의 특성, 상위 조직의 요구 사항 등에 따라서 달라진다. 데이터 자동 수집 및 분석에 필요한 플러그인 소프트웨어와 수집하는 데이터는 다양한 개발 환경에 맞게 개발 될 필요가 있다. 본 논문에서는 웹 애플리케이션 개발에서 개인 프로세스 데이터를 자동 수집 및 분석 해 주는 도구를 소개한다. 현재 이 도구는 Java 를 사용하는 웹 애플리케이션 개발을 지원한다. 기존 도구들도 웹 애플리케이션 개발에서의 개인 프로세스 데이터를 자동 수집할 수 있지만 웹 애플리케이션의 특성이 충분히 고려되지 않았기에 다음의 데이터를 자동 수집하는 데 어려움을 가지고 있다.

- Java Server Page (JSP), html 등의 웹 개발 관련 파일 작성 및 수정에 사용한 시간
- 컴파일 시점이 아닌 프로그램 실행 시점에 나타나는 JSP 컴파일 에러
- 웹 브라우저를 이용한 웹 애플리케이션 테스트 시간

본 도구는 위 데이터의 자동 수집을 지원할 뿐만 아니라 다음의 기능들을 제공함으로써 데이터 자동 수집 및 분석 기법을 확장한다.

- 파일 수정을 수반하지 않는 작업에 사용된 시간 자동 수집
- 요구사항, 설계, 구현, 테스트의 개발 단계
- 프로그램 실행 중 발생하는 에러 자동 수집

본 논문의 구성은 다음과 같다. 2 장에서는 본 도구

에서 자동 수집하는 데이터의 종류와 데이터 수집 기법을 설명하고 3 장에서는 도구의 구현에 대해서 기술한다. 관련 연구와 결론 및 향후 연구를 각각 4 장과 5 장에서 기술한다.

2. 데이터 수집

1) 개발 시간

개발자는 설계, 구현, 테스트, 회의, 코드/설계 리뷰, 결함 제거 등 다양한 활동에 사용한 시간을 지속적으로 기록해야 한다. PSP 에서는 개발자들에게 Time Recording Log 를 이용하여 분 단위로 개발에 사용한 시간을 기록하도록 하고 있다. 이론적으로는 이러한 방법을 통해 시간 사용을 정확하게 파악할 수 있지만 앞서 언급한 오버헤드와 컨텍스트 스위칭의 문제로 인해 이러한 방법을 실무에서 사용하기는 쉽지 않다. 이러한 어려움을 감소시키기 위해 개발자가 작성 및 수정하고 있는 파일의 크기를 주기적으로 (예, 30 초마다) 비교함으로써 개발 시간을 자동으로 수집하는 방법이 Hackstat 에서 소개되었다 [8]. 이렇게 함으로써 소스 코드, 설계 문서 등의 개발 관련 파일의 작성 및 수정에 사용한 시간을 자동으로 수집할 수 있다.

본 도구는 이러한 방법과 함께 파일 수정을 수반하지 않는 작업에 사용된 시간을 자동으로 수집한다. 기술 문서 보기, 웹 검색, 프로그램 테스트 시간 등이 이러한 작업에 해당된다. 예를 들어, 개발자는 소스 코드 작성을 하면서 API (Application Programming Interface) 문서, 설계 및 요구사항 문서를 빈번하게 참조한다. 이러한 작업에 사용한 시간을 자동으로 수집하기 위해서 개발자가 사용하고 있는 도구를 계속적으로 감시한다. 즉, 도구 내에서의 마우스 움직임, 버튼 클릭, 스크롤바 움직임, 키보드 입력 등의 이벤트를 감지함으로써 도구 사용 시간을 자동으로 수집한다. 도구 사용 시간을 수집함으로써 개발자가 개발 관련 문서를 참조하는 데 사용한 시간을 파악할 수 있다.

프로그램 테스트 시간도 이 방법으로 자동 수집된다. 사용자 관점에서 웹 애플리케이션 테스트는 주로 웹 브라우저를 이용하여 수행되기 때문에 웹 브라우저의 사용 시간을 자동 수집하여 테스트 시간을 수집할 수 있다. 웹 브라우저는 웹 검색, 테스트, 문서 보기(예, Java API 문서)의 작업에 사용되기 때문에 웹 브라우저 사용 시간이 어느 작업에 해당되는 것인지 구별할 필요가 있다. 이를 위해 웹 브라우저에 로드된 문서의 사이트를 이용한다. 예를 들어, 웹 애플리케이션은 <http://localhost:8080/> 과 같은 특정 주소 내에서 실행되기 때문에 이 주소 내에서의 웹 브라우저 사용 시간은 테스트 시간이 된다.

또한, 본 도구는 수집된 개발 시간의 개발 단계를 지정할 수 있게 한다. PSP 는 개발 시간을 요구 사항, 설계, 구현, 테스트의 개발 단계별로 수집되는 것을 요구한다. 개발 단계를 수집하기 위해서는 본 도구는 개발자의 개입을 필요로 한다. 즉,

2) 결합

[표 1]에 본 도구에서 수집하는 결합의 종류와 각 결합 종류에 포함된 정보가 나타나 있다.

컴파일 에러는 일반적으로 컴파일 시점에 생성되고 문법 오류를 나타낸다. 그러나 JSP 파일에 포함된 Java 소스 코드는 컴파일 시점이 아닌 실행 시점에 생성된다. JSP 페이지에 대한 요청이 발생하는 시점에서 해당 JSP 파일이 컴파일 되고, 이 때 컴파일 에러의 유무를 알 수가 있다. 현재 도구는 Eclipse 에서 Java 웹 애플리케이션 개발을 지원하며 Eclipse 에서 나타나는 Java 파일의 컴파일 에러와 JSP 파일의 컴파일 에러를 자동으로 수집한다.

유닛 테스트 실패는 JUnit 과 같은 유닛 테스트 프레임워크의 실행을 감지함으로써 자동 수집한다. 현재 도구는 JUnit 실행을 감지하여 유닛 테스트 실패를 수집한다.

런타임 에러는 프로그램 실행 중에 발생하는 에러로써 Java 에서는 대부분 예외(Exception)로 나타난다. 본 도구는 프로그램 실행 중 발생하는 Java 예외를 관련 정보와 함께 자동 수집한다. 관련 정보에는 NullPointerException, SQLException, IOException 등의 예외 종류, 예외가 발생한 위치 (Java 소스 파일명과 라인 번호), 예외 트레이스(trace)가 포함된다.

표 1. 결합 데이터

결합 종류	컴파일 에러	유닛 테스트 실패	런타임 에러
Phase	컴파일 또는 프로그램 실행	유닛 테스트	프로그램 실행
Description	문법 오류 서술	예외 트레이스	예외 트레이스
Resource	소스 파일명과 라인번호	유닛 테스트 파일명	소스 파일명과 라인번호
Defect sub-type	문법 오류	Java 예외 종류	Java 예외 종류

3) 소프트웨어 크기

소프트웨어 크기를 자동으로 수집하기 위해서는 LOCC [9]와 같은 코드 라인 카운팅 도구의 실행 결과를 자동으로 수집한다. LOCC 는 Java 파일(.java)만을 지원하기 때문에 JSP, HTML 의 크기를 측정하기 위해서 LOCC 를 수정(계획 중)하여 사용한다.

3. 구현

[그림 1]에 나와 있듯이 본 도구는 데이터 자동 수집을 담당하는 Sensor, Sensor 에서 수집된 데이터를 Server 로 보내는 Client Manager, 수집된 데이터를 분석하는 Server, 데이터 저장에 사용되는 Database 로 구성된다. 개발자는 Web Browser 이용하여 차트 형태 등으로 수집된 데이터에 대한 다양한 분석 결과를 볼 수 있다. 분석 결과는 메일을 통해서 개발자에게 주

기적으로(예, 메일) 전달될 수도 있다.

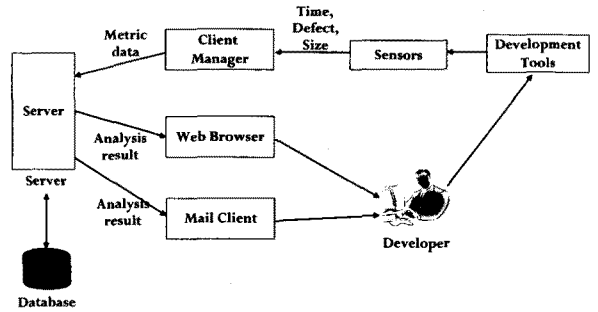


그림 1. 도구 아키텍처

[그림 2]에는 웹 애플리케이션 개발에 필요한 도구들과 그러한 도구들에서 필요한 데이터를 수집하는 센서들이 나타나 있다. 현재 도구는 Java, Eclipse 를 이용한 웹 애플리케이션 개발을 지원하며 웹 브라우저로는 Internet Explorer 을 지원한다. 센서에는 개발 관련 파일의 작성 및 수정에 사용한 시간을 수집하기 위한 Eclipse Sensor, MS Word Sensor, Internet Explorer 과 Acrobat Reader 의 도구 사용 시간을 수집하기 위한 Tool Monitor 가 포함된다. 또한 본 도구는 개발 단계를 지정하기 위해 Phase Selector 를 제공하여 개발자가 개발 단계를 선택할 수 있게 한다. 컴파일 에러, 유닛 테스트 실패, 런타임 에러와 소프트웨어 크기는 Eclipse Sensor 에서 수집된다.

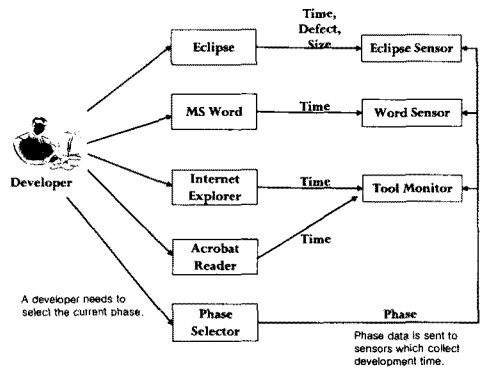


그림 2. 데이터 자동 수집 아키텍처

4. 관련 연구

1) Hackstat [4, 8]

Hawaii 대학의 Collaborative Software Development Laboratory 에서 Philip M. Johnson 교수가 중심이 되어 오픈 소스로 개발된 데이터 자동 수집 및 분석 도구이다. 이 도구는 개발자 개인 데이터 및 팀 데이터를 자동 수집하고 수집된 데이터에 대해 다양한 분석을 제공함으로써 소프트웨어 개발 과정에서 발생한 문제를

점을 파악하고 해결책을 찾는데 도움을 준다. PSP 가 소개된 초창기부터 대학교 교과 과정으로 PSP 를 학생들에게 가르치고 적용함으로써 축적된 다양한 경험과 PSP 데이터를 바탕으로 PSP 측정 및 분석 활동 지원을 목표로 Hackstat 도구를 개발하였다. 개발자들이 사용하는 다양한 도구(예, Eclipse) 또는 형상 관리 시스템(예, Current Versions System)에 플러그인 형태로 삽입된 다양한 센서들(예, CVS Sensor, Eclipse Sensor)이 필요한 데이터를 자동으로 수집하고, 수집된 데이터를 서버를 통해 XML Database 에 저장된다. 개발자와 관리자는 웹 브라우저를 이용하여 수집된 데이터에 대한 분석을 할 수 있다. Hackstat 도구는 주로 시간 경과에 따른 데이터의 변화를 보여주는 다양한 차트를 제공한다.

2) PROM (Pro Metrics) [5]

PROM 은 소프트웨어 제품과 프로세스 메트릭을 자동으로 수집하는 데이터 수집 및 분석 도구이다. 수집된 데이터는 PSP 메트릭, 프로시저와 객체 지향 메트릭, 워드 프로세서를 이용한 문서 작업 등 코딩 작업 이외의 활동을 추적하기 위한 특별 메트릭을 포함한다. PROM 은 다음의 네 가지 컴포넌트로 이루어져 있다.

- PROM Probes: 개발 관련 도구에 플러그인 형태로 부착되어 관련 메트릭을 수집하는 PROM plug-in 과 운영체제의 시스템 호출을 감시하는 PROM Trace 를 포함한다.
- PROM Transfer: 수집된 데이터를 플러그인으로 부터 받고, 필요한 프리프로세싱(pre-processing) 을 한 후, 프리프로세싱된 메트릭 데이터를 PROM Server 로 보낸다.
- PROM Server: PROM Database 에 데이터를 저장하는 역할을 하며 분석 및 프로젝트와 사용자 관리 등의 기능을 웹 서비스를 통해서 제공한다.
- PROM Database: 사용자와 프로젝트 관련 정보, 수집되는 메트릭 데이터와 분석 결과를 저장한다.

개발자와 프로젝트 관리자는 웹 페이지를 통해서 PROM Server 에 원하는 종류의 분석을 요청할 수 있다. 예를 들어, 코드의 길이, 복잡도, 커플링(coupling) 을 나타내는 그래프, 한 프로젝트에 쓰여진 전체 시간 등 원하는 분석을 웹 인터페이스를 통해서 PROM Server 에 요청할 수 있다. 또한 제 3 자(third-parties)가 제작한 분석도구를 이용해서 원하는 분석을 할 수 있게 하는 메커니즘을 제공한다.

앞서 언급한 것처럼 본 도구는 Hackstat, PROM 에서 사용된 데이터 자동 수집 및 분석 기법을 확장하여 웹 애플리케이션 개발에서의 개인 프로세스 데이터를 수집 및 분석을 포함한다.

5. 결론

본 논문에서는 기존 데이터 자동 수집 및 분석 기법을 확장하여 웹 애플리케이션 개발에서의 개인 프로세스 데이터를 자동 수집 및 분석 하는 도구를 설명하였다. 현재 핵심 기능들이 구현되었고, 필요한 센서들이 개발 중에 있다. 본 도구의 사용을 통해 데이터의 일관되고 지속적인 수집이 가능해져 신뢰성 있는 데이터를 획득할 수 있고, 신뢰성 있는 데이터의 분석을 통해 소프트웨어 품질 향상, 보다 정확한 예측, 개인 역량의 객관적인 평가, 개인 프로세스의 정량적 관리 등의 이득을 얻을 수 있을 것이다.

향후 연구로서 웹 애플리케이션 개발 이외의 임베디드 소프트웨어 개발, 데이터베이스 설계 등 다양한 개발 환경에 맞는 데이터 자동 수집 및 분석 기법 연구가 요구된다.

참고문헌

- [1] M. B. Chrissis, M. Konrad and S. Shrum, CMMI - Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003.
- [2] J. McGarry, D. Card, et al., Practical Software Measurement, Addison Wesley, 2002.
- [3] W. S. Humphrey, PSPSM: A Self-Improvement Process for Software Engineers, Addison-Wesley, 2005.
- [4] P. M. Johnson, H. B. Kou, J. M. Agustin, C. Chan, C. A. Moore, J. Miglani, S. Zhen, and W. E. Doane. Beyond the personal software process: Metrics collection and analysis for the differently disciplined. In Proceedings of the 2003 International Conference on Software Engineering, Portland, Oregon, May 2003.
- [5] Sillitti A., Janes A., Succi G, Vernazza T., "Collecting, Integrating and Analyzing Software Metrics and Personal Software Process Data", EUROMICRO 2003, Belek-Antalya, Turkey, 1 - 6 September 2003.
- [6] Raymund Sison, David Diaz, Eliska Lam, Dennis Navarro, Jessica Navarro, "Personal Software Process (PSP) Assistant," apsec, pp. 687-696, 12th Asia-Pacific Software Engineering Conference (APSEC'05), 2005.
- [7] P.M. Johnson, A.M. Disney, "A critical analysis of PSP data quality: Results from a case study", Journal of Empirical Software Engineering, December 1999.
- [8] Johnson, P.M.; Hongbing Kou; Agustin, J.M.; Qin Zhang; Kagawa, A.; Yamashita, T., "Practical automated process and product metric collection and analysis in a classroom setting: lessons learned from Hackstat-UH," Empirical Software Engineering, 2004. ISESE '04. Proceedings. 2004 International Symposium on, vol., no.pp. 136- 144, 19-20 Aug. 2004
- [9] LOCC, <http://csdl.ics.hawaii.edu/Tools/LOCC>
- [10] T. DeMarco. Controlling Software Projects, Yourdon Press, 1982.