

Web기반 ERP 커스터마이징을 위한 백엔드 개발도구의 유용성 연구

정훈¹, 이강수^{2*}

¹더존비즈온 미래기술연구소 수석연구원, ²더존비즈온 미래기술연구소 연구소장

A Study on the Usefulness of Backend Development Tools for Web-based ERP Customization

Hoon Jung¹, KangSu Lee^{2*}

¹Researcher, DOUZONE Future Technology Research Institute

²Director, DOUZONE Future Technology Research Institute

요약 최근 ERP 시스템이 Web 환경으로 변화하고 업무의 복잡도가 증가함에 따라 프로젝트의 실패위험이 증가되고 있다. 이를 해결하기 위한 방안으로 로우코드 플랫폼 개발도구가 활용되고 있으나 UI 중심이라 한계점이 존재한다. 이를 극복하기 위하여 프론트개발 뿐만아니라 ERP 개발과정에서 생산되는 다양한 개발소스의 빅데이터를 활용하여 백엔드 비즈니스 서비스까지 쉽고 빠르게 개발이 가능한 백엔드 개발도구가 요구되고 있다. 또한 기존 ERP 제품 내에 포함된 개발도구는 초급개발자 및 중급개발자의 입장에서는 진입장벽이 높아서 많은 학습시간이 필요로 한다. 본 논문에서는 이러한 단점을 해결하기 위해 쿼리 작성 시간 단축, 서비스 기반 단위테스트를 위한 테스트용 데이터 자동 바인딩, 소스 코드 품질 검사 등 ERP 개발도구에게 필요한 요구사항을 바탕으로 백엔드 개발도구를 구현하여 적용한 결과 개발자별 스킬과 역할에 맞는 ERP 개발을 위한 개발도구의 활용성을 높여 맞춤형 개발도구 기능을 제공함으로써 기존 ERP 내에 포함된 개발도구의 한계점을 극복하는 방안을 연구하고자 한다.

주제어 : Web, 로우코드, 백엔드 개발도구, ERP 커스터마이징, 개발생산성, 플랫폼

Abstract The risk of project failure has increased recently as ERP systems have been transformed into Web environments and task complexity has increased. Although low-code platform development tools are being used as a way to solve this problem, limitations exist as they are centered on UI. To overcome this, back-end development tools are required that can be developed quickly and easily, not only from the front development but also from a variety of development sources produced from the ERP development process, including back-end business services. In addition, the development tools included within existing ERP products require a lot of learning time from the perspective of beginner and intermediate developers due to high entry barriers. To address these shortcomings, this paper seeks to study ways to overcome the limitations of existing development tools within the ERP by providing customized development tool functions by enhancing the usability of ERP development tools suitable for each developer's skills and roles based on the requirements required by ERP development tools, such as reducing the time required for querying, automatic binding of data for testing for service-based units, and checking of source code quality.

Key Words : Web, Low code, Back-end development tools, ERP customization, Development productivity, Platform

*This paper was supported by the fund of World Class 300 R&D Project in 2019.

*Corresponding Author : KangSu Lee(duzon.lee@gmail.com)

Received October 22, 2019

Accepted December 20, 2019

Revised December 4, 2019

Published December 28, 2019

1. 서론

1.1 연구의 배경 및 필요성

최근 4차 산업혁명 시대로 접어들면서 빅데이터를 활용한 AI, 스마트 공장, CLOUD 이용의 활성화 등 기업 IT 환경이 급변하고 있다. 이에 따라 ERP 시스템도 빅데이터(Big-data)를 활용하여 업무 프로세스를 자동화하고 간소화하여 외부 시스템, 자체의 Legacy 시스템과의 연계 등을 통해 효율성을 높이고자 차세대 ERP로 고도화하는 수요가 늘어나고 있다. 그러나 과거 시스템과는 다르게 다양한 규격의 PC와 스마트 디바이스 등장과 리눅스, 맥, 윈도우즈 등 다양한 OS, 그리고 연계대상도 내부 Legacy 시스템뿐만 아닌 금융기관, 공공기관, 협력업체, 고객 등으로 다양해지고 있다[1].

이와 같이 복잡하고 다양한 IT 환경을 지원하기 위해서는 기존과 같이 특정 OS에 국한되는 애플리케이션보다는 멀티브라우저에서 이용 가능한 Web Platform 기반의 ERP가 선호 되고 있으며, 이미 많은 벤더사가 Web 기반으로 전환하고 있다[2,3]. 또한 기업의 특성상 다양한 규모, 산업, 비즈니스 방식의 차이 등으로 업무 프로세스의 복잡도(complexity)는 증가하고, 자동화를 통한 업무효율 제고를 위하여 패키지로만으로는 대응이 어렵기 때문에 대부분의 ERP를 도입하는 기업은 패키지에 커스터마이징을 추가하는 방식으로 도입하고 있으나 커스터마이징 자체가 프로젝트의 리스크를 증가시키므로 효율적 대응이 중요하다[4].

애플리케이션을 둘러싼 환경은 빠르게 변화하고 있다. 그래서 새로운 많은 기능의 추가나 변경 등 요구사항도 그만큼 다양함에 따라 제품 개발 및 테스트가 너무 길어지면 고객의 불만 증가로 피드백도 받기 어려워진다. 이런 리스크를 사전에 제거하기 위하여 빠른 개발 플랫폼(Rapid Development Platform)의 일종인 Low Code(로우코드)가 대안이 될 수 있다.

Panoram Consulting Group의 2013년 ERP 보고서에서 2012년 조사된 172개 기업 중 59%가 예산을 초과했고 약 60%기업이 기대이익이 50%이상 감소하였고 53%의 프로젝트가 계획된 기간을 초과 되었음을 보여준다. 국내에서는 정보통신산업진흥원의 2017년 소프트웨어 공학백서에 따르면 2016년 국내 SW프로젝트 납기. 비용준수율이 11.8%로 조사되었고 납기와 비용준수를 위한 야근이 필요하다는 경우가 92.8%로 높게 나타났고 최근 주 52시간 근로시간 규제로 프로젝트 실패위험이 점점 증가하고 있다[5].

기업의 IT환경이 복잡하고 빠른 변화에 신속하게 대응하고 IT프로젝트의 실패위험을 줄이기 위해서 ERP 개발과 유지보수를 용이하게 하고 ERP 고도화에 필연적 수반되는 많은 커스터마이징에 대한 문제의 해결은 SW 개발생산성을 높여야 한다. 특히 Web환경에서의 ERP시스템은 상대적으로 Front-End 개발이 표준화와 패턴화가 되어 있으나 Back-End 개발은 복잡도가 높고 표준화의 어려움으로 이 것을 해결할 수 있는 요구가 많다[6].

본 논문에서는 로우코드 기반 개발 개념을 활용하여 백엔드 개발도구의 구현사례를 연구를 통해 개발생산성을 높이는 방안을 모색하고자 한다.

2. 선행 연구

2.1 ERP System 도입과 성공 요인

ERP System은 4단계별로 발전되어 왔다. 제1단계는 1970년대 자재소요계획(MRP : Material Requirement Planning)의 발전과, 제2단계는 1980년대 제조 자원 계획 (MRP II : Manufacture Resource Planning)의 발전과, 제3단계는 1990년대 이후 전사적 자원관리 (ERP : Enterprise Resource Planning)의 발전과, 제4단계는 2000년대 이후부터 확장형 ERP로 발전되어 왔으며, 앞으로의 ERP System이 발전하는 방향은 기업 간의 업무 통합(integrated work)을 통한 Best Practice를 지향하는 통합업무솔루션으로서 E-비즈니스로 발전되어 갈 것이다[7].

일반적으로 ERP System은 최신의 IT기술을 활용하여 하나의 패키지로 구성된 통합된 시스템이다. 이러한 ERP시스템에는 비즈니스 프로세스에 대한 리엔지니어링 개념이 내재되어 있다. 따라서 ERP 시스템을 도입은 선도기업의 업무 프로세스를 이용할 수 있기 때문에 ERP System 도입을 통해 업무 프로세스에 대한 리엔지니어링(reengineering)의 효과를 얻을 수 있다. 하지만 이러한 효과는 유사규모의 유사업종의 ERP System을 도입 시에만 패키지화된 소프트웨어로 적용할 수 있고 그 외 나머지는 커스터마이징 개발 과정이 필요하다. 커스터마이징 개발 과정에서 많은 시간과 예산이 추가로 소요되며, 사용자를 위한 교육과 특별한 유지관리도 필요하다[8].

각 기업마다 업무관행이 달라 상품의 출고 시기나 매출의 인식 등 다양한 부분에서 각기 서로 다른 특성(nature)이 나타날 수 있다. 이런 특성들을 잘 이해하고 수용하는가와 규격화되지 않는 업무 프로세스를 System

에 적절하게 적용하는지가 ERP System 구축 프로젝트의 성공을 좌우하게 된다. 특히 국내에서 사용되는 외산 ERP 패키지(package)는 대부분의 업무관행이나 법규의 국가 간 차이로 인하여 ERP 시스템의 프로젝트가 성공하기 위해서는 구축 과정에서 발생하는 커스터마이징(customizing)에 대한 문제를 해결해야 한다[9,10].

대부분 ERP System 도입 시 핵심 성공요소는 커스터마이징 개발이라 할 수 있으며, 성공적으로 도입한 ERP System도 지속적인 프로세스 개발뿐만 아니라 지속적인 유지보수를 통한 개선작업도 기업 경쟁력에도 중요한 영향을 미치게 된다[11,12].

2.2 로우코드 플랫폼 기반 개발

로우코드(Low Code) 플랫폼 기반 개발은 복잡하고 어려운 코딩 과정을 단순화하여 소프트웨어를 신속, 정확하게 개발(development)하여 빌드(build), 배포(deploy)하는 일련의 개발 환경을 말한다[13].

로우코드 개발은 새로운 개념은 아니고 Delphi, Visual Basic이나 PowerBuilder와 같은 고속 애플리케이션 개발 툴은 이미 오래전부터 있었고, 이러한 툴들을 이용하여 소프트웨어 개발생산성을 지속적으로 높여 왔다. 그러나 구성 위주의 과거의 로우코드 개발 솔루션들은 기획, 개발, 디버깅, 테스트, 실행, 그리고 배포 등 소프트웨어 개발 라이프사이클의 다른 단계들과 단절되어 능률화하지 못했다. 그 결과 충분한 테스트를 거치지 않은 로우코드 솔루션기반 소프트웨어는 출시하면 할수록 개발 프로세스는 더 복잡해지고 시간과 비용이 더 많이 소요되었다. 사용하기 쉽다는 이유로 로우코드 솔루션을 선택했던 비개발자들에게는 버그가 많은 프로젝트를 경험해야만 했다.

애플리케이션을 둘러싼 환경은 빠르게 변화하고 있다. 여러 가지 새로운 기능 추가나 변경 등 개발요구사항도 그만큼 다양하다. 베타 사용 기간이나 테스트 제품 개발 시간이 너무 길어지면 사용자의 불신으로 제대로 된 피드백도 받기도 어려워진다. 이럴 때 고속 개발 플랫폼의 일종인 로우코드 플랫폼 개발이 해결책이 될 수 있다.

로우코드 개발플랫폼을 이용하여 직접 코딩이 줄더라도 기본 프로그래밍 지식은 필요하다. 그리고 로우코드 개발플랫폼의 개발자(developer) 중에 전문개발자와는 다른 기술 역량이나 경험을 지닌 사람도 많다. 로우코드 개발플랫폼을 통해 데이터 분석하거나 소프트웨어 관리 책임자들이 간단하지만 필수적인 기능 개발에 참여할 수 있다. 관련 업무와 조직이 요구하는 것을 알 수 있다면

프로그래밍 교육을 별도로 받은 적이 없는 직원이라도 애플리케이션의 기능을 추가하고 개선점을 찾아내거나 사용자 경험을 바로 개선할 수 있다.

로우코드 개발 방식의 이점은 개발의 가치(value)와 속도(speed)이다. 빠른 시간 내에 애플리케이션을 선보이고, 빠르게 피드백을 받아 애플리케이션을 수정, 변경할 수 있어 개발, 빌드, 배포 속도가 빨라진다. 특히 숙련되고 개발기술이 높은 전문개발자의 경우엔 최상의 자동화하고, 생략 가능한 부분은 생략하여 개발 작업의 효율성을 크게 높일 수 있다. 개발자가 필수기능을 직접 추가하여 테스트하고, 최종적으로 제품에 새로운 기능을 반영하는 속도를 높이는 것이 로우코드 개발플랫폼의 최대 장점이다.

3. ERP 개발도구의 요구사항

3.1 빠른 비즈니스 서비스 생성

ERP System을 커스터마이징 해야 하는 경우 개발자들 간의 비표준화된 다양한 코드, 취약점을 가진 코드, 담당 모듈별로 다른 기준의 완성도 등의 문제점들이 나타나고 있다. 이러한 문제 해결을 위한 자동화된 비주얼 IDE(통합 개발 환경, Integration Development Environment)를 이용해 로우코드 기반의 개발 절차 도입이 필요하다. 이를 통해 개발 속도뿐만 아니라 표준화된 애플리케이션 구조를 유지하여 관리의 용이성을 높인다.

3.2 간편한 서비스 테스트

테스트는 사용자의 기대 수준과 요구사항(requirement)에 맞게 구현되어 수행되는지를 확인하고 결함(defect)을 발견하는 것이다. ERP 애플리케이션 소스 코드의 특정 모듈이 설계된 대로 정확히 작동하는지 모든 함수(function)와 메서드(method)에 대한 테스트가 수행되어야 한다.

3.3 신속한 코드 품질 검사

소스 코드 정적 분석을 통해, Runtime 시 장애를 유발할 수 있는 코드를 사전 검출하고 소스 코드 연계를 통해 코드 인스펙션(code inspection)에 검출된 소스 코드를 바로 확인하고 조치를 취해야 한다.

3.4 ERP 빅데이터 활용

ERP 애플리케이션 내에는 다양한 정보들이 생성되며, 적재되고 있으나, 기존 로우코드 플랫폼 기반 개발도구와 ERP 제품이 제공하는 개발도구는 활용성이 부족하다. 이를 개선하기 위한 기능 도출이 필요하다.

4. ERP 백엔드 개발도구의 구현

4.1 개요

Fig. 1의 다이어그램에서 보는 것처럼 ERP 백엔드 개발도구(DEWS-BD)는 개발자 개발환경과 ERP 거버넌스 도구(Meta-Q)로 구성된다. 이를 통해 ERP 개발도구의 요구사항을 구체화하기 위해 빠른 비즈니스 서비스 생성을 위해 마법사, 테스트 기반 편집기, 소스코드 정적 분석기, ERP 빅데이터 활용하는 거버넌스 도구를 제공한다.

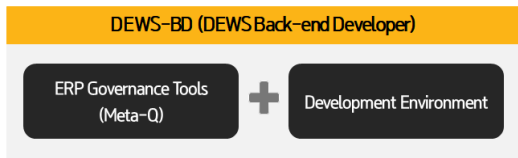


Fig. 1. ERP Backend Development Tool Diagram

개발자 개발환경은 비즈니스 프로그램 개발, 단위테스트, 코드품질검사 등 개발 Life-cycle 전반에 대한 구현을 위한 지원 도구를 제공한다.

ERP 거버넌스 도구 (Meta-Q)는 ERP 애플리케이션의 다양한 정보를 바탕으로 개발자 개발환경을 위한 환경 및 구현 정보를 제공한다. 예를 들어, 추천권리 정보, 소스 코드 정적 분석 결과 정보 등을 제공함으로써 개발환경의 자동화 기능을 향상시킨다.

4.2 마법사(Wizard) 기반 신규 생성

ERP 애플리케이션 구현 항목은 신규 프로젝트, 모델 Class, DBIO Mapper 파일, DAO Class, 서비스 Class 등이다. Table 1과 같이 ERP 애플리케이션의 구현 항목을 보다 쉽고, 빠르게 개발과 표준화된 구조를 위해 로우코드 기반의 자동화된 도구를 제공한다.

로우코드 개발 방식의 이점은 가치와 속도이다. 이를 위하여 ERP 애플리케이션의 구조 패턴을 분석하여 스킴레톤(skeleton) 프로그래밍 기법이 적용된 더미 코드(dummy code)를 자동으로 생성한다. ERP 애플리케이션의 구현 항목 중 서비스 Class는 Table 2와 같이 클라

이언트 사용자 인터페이스(UI)의 처리 패턴에 따라 다양하며, 개발자는 처리 패턴에 부합되는 유형을 선택하여 개발하고자 하는 서비스 구현체의 기본 골격을 가지는 프로그램 소스코드를 자동으로 생성한다.

Table 1. Implementation Items for ERP Applications

Impl.	Explanation
Project	Divided into online program and batch program
Model Class	JavaBeans object for data exchange between layers
DBIO Mapper file	Persistence framework that handles developer-specified SQL, stored procedures, and more
DAO Class	Abbreviation for Data Access Object, which is an object that actually accesses the DB. It is used for efficient connection management and security.
Service Class	Implement processing logic for ERP business services

Table 2. Processing Patterns in the User Interface (UI)

Division	Explanation
Pattern A1	Generate only lookup code that handles Single Grid data type
Pattern A2	Generate lookup, save (add, modify, delete) code to handle Single Grid data types
Pattern B	Generate inquiry, save (add, modify, delete) codes to handle horizontal Master and Slave Grid data types
Pattern C	Generate inquiry, save (add, modify, delete) code to handle vertical Master, Slave Grid data type
Pattern D	Generate lookup, save (add, modify, delete) codes to handle horizontal Master Grid and Slave Form data types
Pattern E	Generate lookup, save (add, modify, delete) codes to handle vertical Master Grid and Slave Form data types

Fig. 2는 서비스 Class의 패턴별 구현 로직을 자동으로 생성하기 위한 방안으로써 신규 서비스 생성 마법사 기능을 활용하여 ERP 애플리케이션의 사용자 화면(user interface)유형에 맞는 구현 항목들을 선택하여 검색, 저장 등 상세화를 위한 로직을 처리한다. 로직을 처리하는 코드는 개발프레임워크에 따라 다르지만 입출력 파라미터 정보를 바탕으로 등록(creation), 조회(retrieve), 수정(update), 삭제(deletion)의 실제 구현 로직은 적용된 개발프레임워크의 템플릿에 맞게 코드가 생성된다.

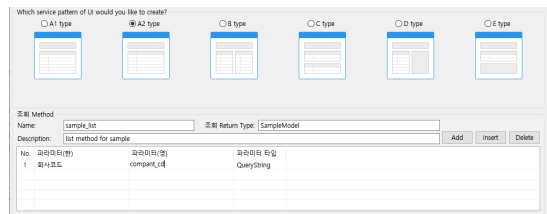


Fig. 2. Create method by service class and pattern

4.5 ERP 빅데이터 활용하는 거버넌스 도구

ERP 거버넌스 도구(Meta-Q)는 Fig. 6처럼 ERP 애플리케이션의 다양한 리소스를 수집, 분석, 적재 과정을 통해 저장소(repository)에 지속적이며 영구적으로 보관된다.

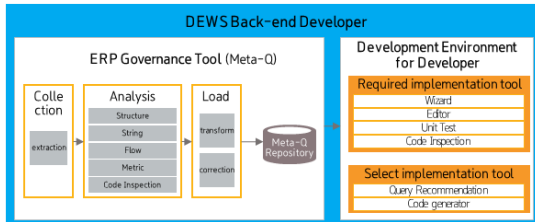


Fig. 6. ERP Governance Tool (Meta-Q) Diagram

각 수집 대상은 특성에 따라 고유한 룰 규칙을 적용하여 정보를 분석하여 적재한다.

예를 들어, ERP 애플리케이션의 구조 분석을 위해서 소스 및 실제 애플리케이션의 분석 정보를 바탕으로 레이어(layer) 간의 호출 관계 정보를 만들 수 있다. 그리고 개발자가 쿼리(query) 작성 시 검색 키워드에 따라 ERP System이 이미 수행된 쿼리 데이터 중 가장 적합도가 높은 ERP 내의 모든 쿼리를 찾아서 제공해 준다. 이것은 ERP 내의 수많은 데이터 중 소스 쿼리뿐만 아니라 실제 수행된 모든 쿼리를 ERP 거버넌스 도구(Meta-Q)가 분석하고 제공한다.

쿼리는 개발자들이 ERP 애플리케이션을 커스터마이징할 때, 사용빈도가 가장 높고 많은 시간이 소요된다. 이를 위하여 개발자 개발환경을 위하여 쿼리 추천 정보를 제공함으로써 보가 쉽게 빠르게 쿼리를 작성하도록 한다.

Fig. 7과 같이 '품목정보'라는 키워드로 질의하면, 최적화 룰에 의해 적합도가 높은 쿼리 소스 및 실제 수행된 쿼리 정보를 찾아서 개발자에게 제시한다. 개발자는 이를 활용하여 구현하고자 하는 쿼리를 작성하여 완성한다.

The screenshot shows the 'Query Recommender' interface. A search box contains '품목정보'. Below, a table lists recommended queries with columns: No, 개발도수, 쿼리(MyData), 실행력, 코멘트, 서비스, 서비스명, and API. The first query is highlighted.

Fig. 7. Query recommendation by goodness of fit

테스트 데이터는 서비스의 단위테스트를 위해 중요한 정보이지만, 테스트마다 입력 시 불편함이 가중된다. 이를 해소하기 위하여 과거 수행된 테스트 데이터를 재사

용한다.

Fig. 8은 로컬 PC와 개발서버에서 수행된 테스트 데이터를 원클릭으로 재사용하게 한다. 이것은 ERP System이 개발되고 테스트되거나 운영될 때, ERP 거버넌스 도구(Meta-Q)는 이미 학습이 된 지속적인 분석을 정보를 가지고 있다. 특히 해당 특정 서비스의 테스트를 위해 가장 적합한 테스트 데이터를 추출하여 자동으로 바인딩 시켜준다. 이를 통해 개발자는 화면과 연결하지 않거나 최소화함으로써 개발 생산성을 높일 수 있으며, 테스트 기반 개발을 수행한다.

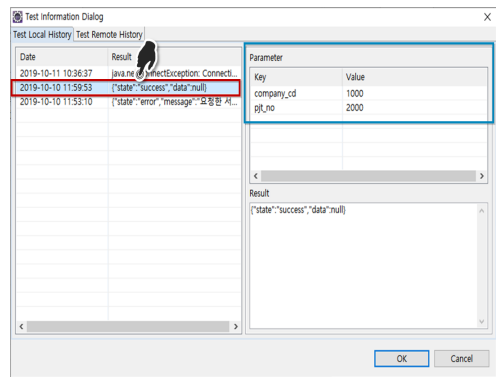


Fig. 8. One-click reuse of test data

5. ERP 백엔드 개발도구 활용 만족도 평가

논문에서 연구 및 구현된 로우코드 기반 ERP 백엔드 개발도구의 활용 만족도 평가를 위해 2019년 8월 1일부터 11월 18일까지 Table 4와 같이 D사 ERP 제품개발팀 과 D사 ERP 구축팀의 128명을 대상으로 설문조사를 통해 개발자의 경력 및 등급별 활용성의 만족도를 분석하였다. 초급(Beginner)은 4년 미만, 중급(Middle class)은 4년 이상 12년 미만, 고급(Advanced)은 12년 이상 16년 미만, 특급(Express)은 16년 이상으로 개발자 등급을 분류하였다.

Table 4. Development career and rating

Development Grade	Number of developers
Beginner	71
Middle class	33
Advanced	6
Express	18
Total	128

Table 5는 개발자 등급별로 개발도구로써의 기능성(Functional), 편의성(Convenience), 신뢰성(Responsibility)의 만족도 분석 결과이다. 5단계 척도로 평가 문항을 사용하였다.

Table 5. Developer Satisfaction Analysis Results

Analysis	Func.	Conv.	Resp.
Very satisfied	32	28	28
Satisfaction (yes)	44	44	45
Usually	50	52	51
Dissatisfied (not)	0	2	2
Very dissatisfied (very not)	2	2	2
Total	128	128	128

Fig. 9는 개발도구의 기능 품질에 대한 전반적인 만족도의 분석 결과이며, 97%가 기능 품질에 만족하는 결과를 나타낸다.

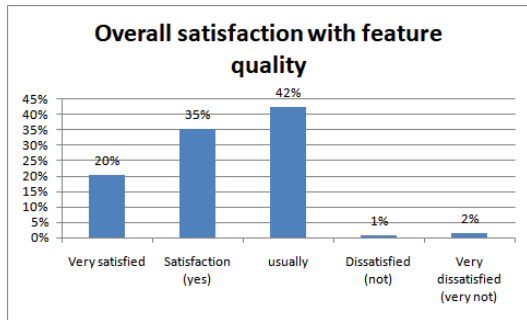


Fig. 9. Overall satisfaction with feature quality

Fig. 10은 개발도구의 주요 기능에 대한 활용 만족도 분석 결과이다. 분석 항목은 마법사(wizard) 기능, 편집기(editor) 기능, 테스트(test) 기능, Code Inspection과 쿼리추천을 포함한 부가기능(add-on)이다.

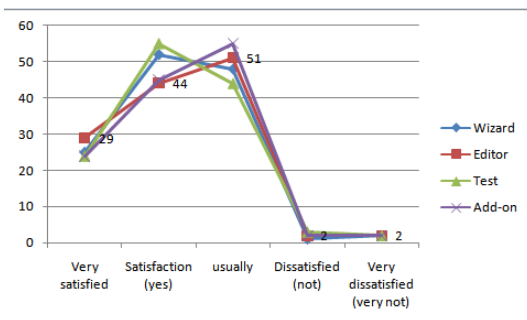


Fig. 10. Major functional satisfaction analysis result

Fig. 11은 개발도구의 주요 기능에 대한 전반적인 만족도의 분석 결과이며, 96.9%가 백엔드 개발도구가 갖춰야 할 주요 기능을 바탕으로 요구사항을 만족하는 결과를 나타낸다.

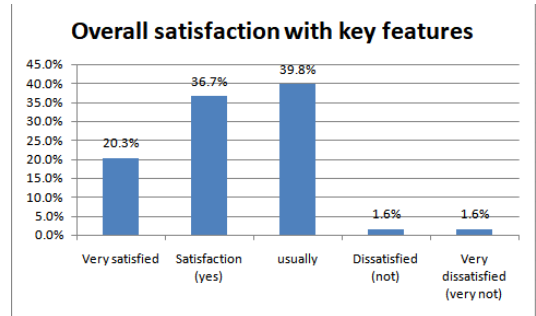


Fig. 11. Overall satisfaction with key features

Table 6은 현재까지 구현된 로우코드 플랫폼 기반의 개발도구와 비교하였으며, DEWS-BD는 Web 기반 ERP 개발을 위한 필수 요구사항을 모두 만족한다.

Table 6. Comparison with existing development tools

Technology elements	Classification	DEWS-BD	DWKit	Applan	KissLOW	Mendix	OutSystems
Backend Service Development	JAVA language support	support	Not supported	Not supported	Not supported	Not supported	Not supported
	Backend Service Development	support	Not supported	Not supported	Not supported	Not supported	Partially support
Development productivity	Generation wizard	support	support	support	support	support	support
	JAVA editor	support	Not supported	Not supported	Not supported	Not supported	Not supported
	Unit test	support	Not supported	Not supported	Not supported	Not supported	Not supported
	Code Inspection	support	Not supported	Not supported	Not supported	Not supported	Not supported
	Linkage Integration	support	Not supported	Not supported	Not supported	Not supported	Partially support
Ease of use	Easy installation	support	Partially support	Partially support	Partially support	Partially support	Partially support
	Developer Needed Features	support	Partially support	Partially support	Partially support	Partially support	Partially support

Table 7은 기존 Global ERP 제품에 포함된 개발도구와 비교하였으며, DEWS-BD는 여러 차별적인 기능요소들을 포함하고 있음을 알 수 있다.

Table 7. Global ERP Development Tool Comparison

Technology elements	Classification	DEWS-BD	SAP	Oracle JD Edwards	Microsoft Dynamics
Backend Service Development	JAVA language support	support	Partially support	Partially support	Not supported
	Backend Service Development	support	support	support	support
Development productivity	Generation wizard	support	support	support	support
	JAVA editor	support	Not supported	Not supported	Not supported
	Unit test	support	Partially support	Partially support	Partially support
	Code Inspection	support	Not supported	Not supported	Not supported
	Linkage Integration	support	support	support	support
Ease of use	Easy installation	support	Partially support	Partially support	Partially support
	Developer Needed Features	support	Partially support	Partially support	Partially support

Fig. 12는 ERP 백엔드 개발도구의 종합적인 만족도로써 기존 개발도구와 비교되며 개발도구의 요구사항을 바탕으로 본 논문에서 새롭게 구현된 개발도구의 기능 품질과 주요 기능에 대한 전체적인 만족도의 분석 결과이다. 설문조사에 참가한 개발자 중 96.9%는 기존 개발도구에 비해 기능 품질과 주요 기능에 대해 요구사항을 충족하여 ERP 개발을 위해 지속적인 이용과 사용 확대를

위해 고려되고 있음을 알 수 있다.

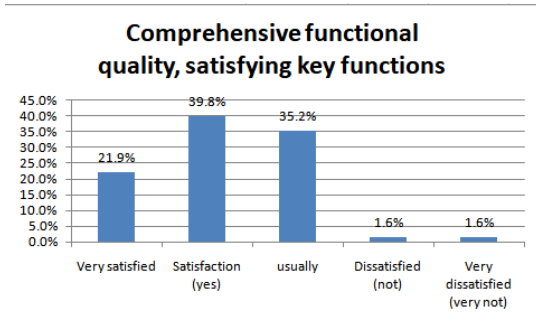


Fig. 12. Comprehensive feature quality, key features

Table 8은 개발자의 경력과 등급에 따른 개발도구의 만족도를 나타내며, 초급개발자와 특급개발자 외에 개발도구에 대한 만족도가 각각 71명중 48명으로 67.6%와 18명중 12명으로 66.7%가 만족하는 것으로 나타나 개발도구의 유용성이 높음을 알 수 있다.

Table 8. Satisfaction by Grade

Grade	Very satisfied	Satisfaction (yes)	usually	Dissatisfied (not)	Very dissatisfied (very not)	Total
Express	8	4	6			18
Advanced	2	1	4			7
Middle class	9	7	15	1		32
beginner	32	16	20	1	2	71
total	51	28	45	2	2	128

6. 결론

본 논문에서 구현한 개발도구(DEWS-BD)가 기존 로우코드 플랫폼 기반 개발도구와 ERP 제품에서 제공하는 개발도구의 가장 큰 차이점은 ERP 내에 적재되는 빅데이터를 이용한다. 이 빅데이터를 활용하여 마법사, 편집기, 테스트, 코드품질검사 등을 자동화 하기 위한 기초 데이터로 재사용 된다.

본 논문에서 제안한 방식을 토대로 D사 ERP 제품개발팀과 D사 프로젝트의 ERP 애플리케이션의 커스터마이징에 투입한 개발자를 대상으로 개발도구의 사용의 만족도와 개발의 용의성을 조사하였다. 그 결과 첫째, 동일한 코딩 패턴이 적용되어 손쉽게 코딩 표준이 가능하였고 둘째, 예기치 못한 버그가 발생하는 확률이 낮아졌으며 셋째, 단시간의 교육만으로 단시간에 개발이 가능하였다. 모든 개발자에게 개발생산성을 높여주는 아주 유용한 개발도구를 확인하였다.

앞으로 추가적인 연구를 통해 개발도구를 시각화

(visualization)하고 소스코드를 용이하게 추적가능하게 하고 테스트를 자동화함으로써 보다 개발 생산성을 향상시킬 수 있다. 또한 개발되는 모든 ERP 개발소스를 재사용하여 다양한 개발패턴으로 빅데이터 화하여 고도화 된다면 ERP개발에 있어서 백엔드 개발도구의 새로운 패러다임이 될 것이다.

REFERENCES

- [1] K. S. Lee & C. S. Leem. (2018). A Case Study on Implementation of UI Development Tool for Web Environment ERP System. *Journal of the Korea Convergence Society*, 10(1), 13-24. DOI : 10.15207/JKCS.201910.1.013
- [2] J. H. Jeon & S. Y. Lee. (2006. 10). Trend and Prospect of the Web 2.0 Technology. *Electronic Communication Trend Analysis*, 21(5).
- [3] J. H. Jeon & S. Y. Lee. (2012). Trends on Standardizations of HTML5 based Web Platform Technology Trends on Standardizations of HTML5 based Web Platform Technology. *Electronics and Telecommunications Trends*, 27(4), 83-95.
- [4] S. H. Ha, J. D. Choi & S. M. Bae. (2014). A Case Study of ERP Customizing for a Make-to-order Company. *Entrer Journal of Information Technology*, 13(2), 63-76.
- [5] K. S. Lee. (2019). *A Study of Structural Relationship Between Organization Competce, ERP Customization Method, and Performance*, Doctoral dissertation. Yonsei University, Seoul.
- [6] M. S. Shin, K. J. Jeong & B. C. Kim. (2016). Research on Web standards compliance and professional level of domestic Web Developer. *Journal of Digital Convergence*, 14(4), 30-42.
- [7] K. H. Lee, Y. J. Jeong, K. Y. Lee, J. S. Kim I. S. Kim & C. D. Lim. (2011). Technology Trend of Web-based SW Platform Technology Trend of Web-based SW. *Electronics and Telecommunications Trends*, 26(5), 66-73.
- [8] J. S. Lee. (2008). The Development Process and Direction Suggestion of ERP (Enterprise Resource Planning). *The Journal of Korean Institute of Information Technology*, 6(3), 192-199.
- [9] K. S. Bae, S. H. Chang, H. J. Park & S. H. Jung. (2013). *An Empirical Study on the Effect of ERP Solution on Accounting Information*. *Proceedings of Academic Society of Global Business Association*. Seoul. Thesis for Master's Degree in Seoul School Integrated Sciences & Technologies. Seoul.
- [10] J. W. Han. (2012). *A Study on the effects of ERP in small and midium size company*. Master's Thesis, Kyungpook National University, Dae-gu.

- [11] K. D. Choi. (2000) *An Empirical Study on Critical Success factors in Implementing ERP System: A system Development Life Cycle Perspective*, The Graduate School, Kwang-won University, Ph. D. Dissertation, Chun Cheon.
- [12] J. S. Kim & S. H. Leam. (2002). A Study on the Analysis of ERP System Selection Factors. *Journal of Information Technology Applications & Management*, 9(1), 1-28
- [13] Isaac Sacolick. (2018). *Why developers should rethink low code platforms*, InfoWord(Online), <http://www.itworld.co.kr/news/112329>
- [14] Ministry of Government Administration and Home Affairs, Korea Internet & Security Agency. (2017). *A guide of software development security for e-government SW development operator*, 16-20.
- [15] Ministry of Public Administration and Security. (2011). *Software Development Security Guide*, MOIS [Online]. <https://www.mois.go.kr>.
- [16] Korea Internet & Security Agency. (2012). *Software Development Security (Secure Coding) Guide*. KISA [Online]. <http://www.kisa.or.kr>.

정 훈(Hoon Jung)

[정회원]



- 1997년 2월 : 경희대학교 수학과(이학사)
- 2005년 2월 : 한국외국어대학교 경영정보대학원 소프트웨어공학 (공학석사)
- 2018년 2월 ~ 현재 : 더존비즈온 미래기술연구소 수석연구원

- 관심분야 : ERP, 기술아키텍처, 개발도구, Cloud, SW품질
- E-Mail : pomsjung@naver.com

이 강 수(KangSu Lee)

[정회원]



- 1991년 2월 : 영남대학교 경영학과(학사)
- 2011년 8월 : 연세대학교 공학경영(공학석사)
- 2019년 8월 : 연세대학교 기술정책(공학박사)
- 1996년 9월 ~ 현재 : 더존비즈온 미래기술연구소 연구소장

- 관심분야 : ERP, Cloud, 개발 도구, 개발생산성, 경영혁신
- E-Mail : duzon.lee@gmail.com