Agile-based Low-code Software Development for Nuclear Big Data Processing System

Gwi-sook Jang*, Taek-kyu Kim, Jae-Kwan Park, Seo-Ryong Koo

Autonomous Operation Research Division, Korea Atomic Energy Research Institute, 989-111, Daedeok-daero, Yuseong-gu, Daejeon 305-353, Republic of Korea
Corresponding author: gsjang@kaeri.re.kr

1. Introduction

The nuclear sector also needs to undergo digital transformation. An essential element of digital transformation is a new digital approach, which must be redesigned and designed according to the new environmental change. The biggest challenge for digital transformation is the lack of time, skills and knowledge to deliver digital workflows to meet new demands. So no-code and low-code are emerging as the mainstream of big data application development methods. The reason why low code is attracting attention is because it generally uses a visual programming interface, so the entry barrier is low and the realization speed is fast. In addition, it is advantageous in terms of safety because only functions that have been verified in advance are used.

2. Agile-based low-code development method

2.1 Visual Programming Language

If you use a big data analysis tool, you can quickly find a data analysis method that is suitable for big data characteristics and problem-solving before implementation, and also allows you to quickly and easily check the verification and implementation feasibility. In general, machine learning implementation languages used to analyze big data can be classified into programming languages, script languages, and visual programming (VPL) languages, and the characteristics of each language are shown in Table I [1]. In particular, VPL is a method that implements functions using graphic symbols rather than characters, so that modules can be easily installed and used, and intuitive results can be checked through node manipulation. Types of VPL include KNIME (Konstanz Information Miner), Orange, MS Azure Machine Learning, and UDAS (Unified Data Analysis Suite), and the characteristics of each tool are shown in Table II [1].

This study uses the VPL-based Orange data mining tool that is easy to build on a Python-based nuclear power plant big data processing system, quickly configures various functions with graphic symbols, and can check the results. Existing big data analysis tools, as commercial programs, are difficult to use for free and take a long time to get used to, but Orange is a component-based data mining framework that enables easy big data analysis with just a few clicks. The advantage of Orange is its graphical interface, allowing you to focus on data analysis instead of coding. Add-ons that enhance the mining of machine learning components and external data are also substantial.

Table I: Types of Machine Learning Implementation Language

	Programming Language C/C++,JAVA,C#	Script Language Python, R, Lua, LISP	Visual Programming Language(VPL) KNIME, Orange, MS Azure
Form	Compile Required	Interpreting	Graph based
Module Dependency	Low		High
Development Speed	Slow		Fast
Learning Difficulty	Difficult		Easy

Table II: Features of each VPL Type

KNIME	Enterprise-grade open source data analysis platform • Using Java and Eclipse platform • Modular workflow • Designed for teaching
Orange	**Data visualization and analysis platform for experts and beginners** **• Python based** **• Intuitive user interaction** **• High and simple extensibility**
MS Azure	Cloud-based data predictive and analysis service • Web based • Service deployment support

2.2 Application of Data Mining Technology

Data mining analyzes data from various perspectives and combines the results into useful information. Data mining identifies patterns and correlations hidden in massive data with statistical methods and gives them value. VPL, which is one of the machine learning implementation languages, is a tool that is usually based on data mining techniques. If you use VPL based on data mining technology like Orange, you can implement the intended functional requirements without actually coding and check the results, saving cost and time.

Since the nuclear big data processing system is a data-driven system that collects, refines, stores, and analyzes vast amounts of nuclear data, it is important to extract valid data through appropriate analysis for each data characteristic.

In fact, data mining technology for the nuclear big data processing system performs functions such as big data integration, transformation, and data reduction. If we use the VPL tool based on this, we can check the function without a hardware-based prototype in the system concept and requirements design stage before the detailed design of the big data processing system. In

addition, the result can be efficiently verified through comparison with the detailed design result later.

2.3 Comparison between Waterfall and Agile Model

The most commonly used model in software development are waterfall model and agile model. The waterfall model is a traditional linear development model that collects all requirements, creates a plan, and progresses the project step by step. Agile model is a development model that improves quality by repeating execution and results in small units for given requirements. Table III shows the differences between the two models.

2.4 Agile-based Low-code Development Method

In the development stage of implementing ideas into actual functions, new problems are discovered, and the results are sometimes different than expected. So, in the IT industry, they make a minimum functional product and test it quickly. Even if it is simple and sloppy, it is necessary to make it quickly and verify it. Therefore, in this study, we propose to implement the actual waterfall model after applying the agile model in the Rx data driven system concept and system requirement analysis stage as shown in Fig. 1. The result of applying the agile model is used as an input for the requirement analysis stage of the waterfall model. In addition, the results of each detailed design stages can be verified by comparing it with the results of the agile model based VPL. The detailed verification method is shown in Fig.1.

Table III: Comparison between Waterfall and Agile

	Waterfall Model	Agile Model
Scope	The scope is set in advanced, conditions are fixed	The scope is not fixed, can be changed within the schedule and budget
Features	Large projects with clear procedures and hierarchies	Small and medium-sized projects that frequently try new things to minimize risk
Requirements	Requirements can be added or changed only within the milestone because it goes through the stages of collecting and comprehensively analyzing all requirements	It is possible to reflect changing requirements during the entire project period
Realization Method	A linear way of planning according to the requirements and putting in and developing the necessary resources	Requirements cannot be predicted in advance, so iterating execution and results from minimal discussion of what to build and how to develop to results
Development Method	Creating software step by step	Iteratively confirms the process of developing prototypes and reflecting feedback according to function priorities
Integration Method	The final result is integrated at the end	Iterating releases for each version to find, solve, and fix problems
Test	Final testing and defect fixes prior to release	Integrate from the beginning and always find and improve potential flaws and problems

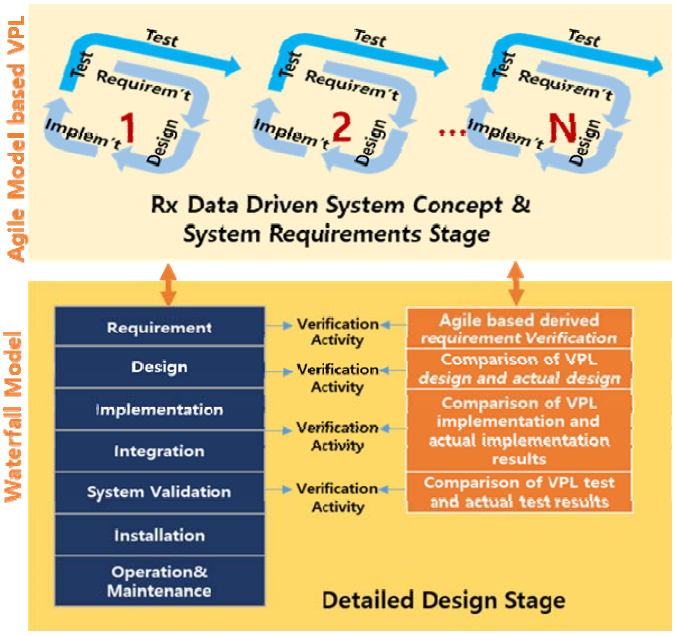

Fig. 1. Agile-based low-code software development for nuclear big data processing system

2.5 Application Results

Fig. 2 shows the results of applying the agile based low-code using the data mining tool Orange according to Fig. 1. Fig.2 shows an application example of agile model that starts with a small function of a nuclear big data processing system and checks the result by gradually adding functions. In addition, it shows that the detailed design results can be compared with the functional results using Orange tool.

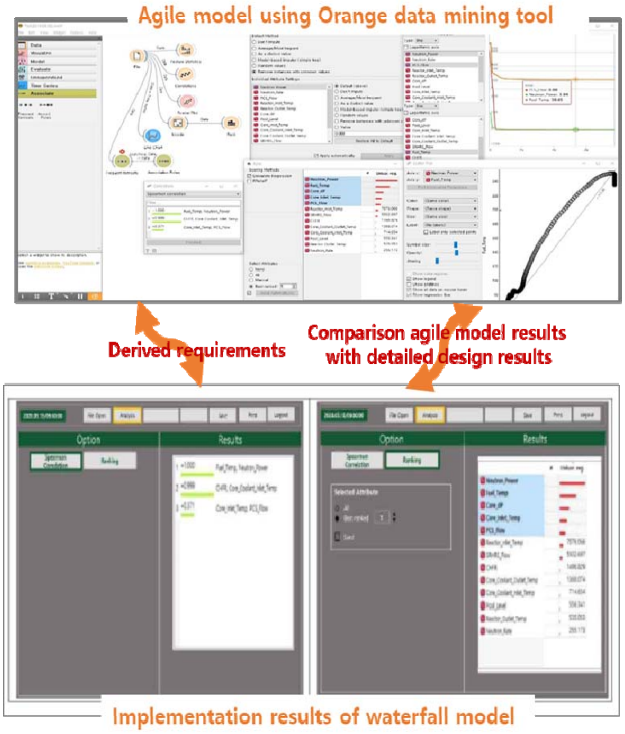

Fig. 2. Application results of agile-based low-code software development for nuclear big data processing system

3. Conclusions

In this study, agile-based low-code software development method was proposed as shown in Fig.1 for a nuclear big data processing system. This proposed method applies the VPL-based agile model to the nuclear data-driven conceptual design and requirements analysis stage, which is the previous stage of the waterfall model applied to the detailed design of a big data processing system. In addition, the results of each software life cycle can be verified by comparing them with the results of the agile model. In this study, the possibility was confirmed as shown in Fig.2 by using the Orange data mining tool based on VPL.

REFERENCES

[1] G. S. Jang, Evaluation of Feature Selection Method for Big Data at Korean Nuclear Power Plant and Verification of its Validity, BIEN 2021, 2021.