Technology Explanation

Development of a next-generation quality data management system

FURUKAWA Akira

Abstract

In recent years, Digital Transformation (DX) has been attracting attention worldwide. All sorts of industries, including manufacturing, finance, and information technology, are working on DX.

However, few companies in Japan have achieved adequate results with DX, and they lag behind their U.S. counterparts. One reason for this is the existence of legacy systems. Legacy systems are becoming less maintainable due to increased system complexity and data bloat, which increase system operating costs. In addition, it is difficult to improve and integrate these systems with other systems to drive DX, so there is a need to break away from legacy systems.

KYB also has many legacy systems and spends many man-hours on their operation and management. It is also difficult to utilize data linked with other systems to promote DX.

To solve these problems, we developed a next-generation quality data management system by renewing a legacy quality data management system. We believe this next-generation system will be a revolutionary low-cost system in terms of both cost and man-hours required for operation and management, and have developed the system based on this concept.

This paper describes the functions of the developed system, the migration method from the legacy system, and the development and operation system for continuous operation of the system.

1 Introduction

In recent years, DX has been attracting attention worldwide. All sorts of industries, including manufacturing, finance, and IT, are working on DX. Many companies in Japan have also launched DX initiatives. The proportion of these companies reached 69.3% in FY2022[1].

However, few companies in Japan have achieved customer value creation or business model transformation through DX, while their simple digitization has had some effect. They lag behind their U.S. counterparts in promoting DX in terms of both human resources and technology. There are several possible reasons for the lag in DX. Three typical causes are:

[1] Organizational problems
[2] Lack of human resources
[3] Existence of legacy systems

Regarding item [1], management has failed to present a direction to work on DX as a management strategy. There is no organization-wide scheme to involve employees in DX activities.

For [2], Japan faces a shortage of human resources that can be mobilized to use digital technology. In order to promote DX, it is necessary to effectively utilize digital technologies such as cloud and AI [Note 1].

Item [3] means that the country has many legacy systems, which consist of old technologies and mechanisms. These legacy systems have become bloated, complicated, and turned into a black box, making it difficult to make improvements to drive DX. In addition, legacy systems require a number of IT engineer man-hours to operate, causing another problem that these engineers do not have time to make improvements.

Therefore, it is essential to first eliminate these causes in order to promote DX.

KYB Corporation also has activities to promote DX. As a function to promote DX, the Digital Transformation Improvement Dept. was established in 2019[Note 2] and has been working on various efforts. First of all, the company has built the KYB-IoT Platform on the Amazon Web Services (AWS) cloud itself as a base to utilize data[2]. On this base, we are working on the development of a preventive equipment maintenance system as DX in the production field[3], and the application of AI technology to the development of mechanical components of shock absorbers (SA) as DX in the product development field[4]. In addition, the company is working to develop human resources skilled in digital technology[5]. A company-wide scheme for promoting DX is almost ready.

However, the company has not yet launched an effort to break away from the legacy systems and still spends many man-hours on their operation and management[Note 3]. Another problem with these systems is that they were built for individual optimization. It is therefore difficult for them to use data and develop new functions in conjunction with other systems to promote DX.

To solve these problems, we renewed the conventional quality data management system, which was one such legacy system, to develop a next-generation quality data management system. We believe this next-generation system will greatly improve the failure/disaster control and provide diversified analysis by linking with other multiple systems. This will be a revolutionary low-expenditure system in terms of both cost and man-hours required for operation and management. Specifically, we have developed the next-generation quality data management system based on the following concepts:

[1] Develop a cloud-based system that is resilient to failures and disasters and reduce operational and management time.

[2] Reduce developer workload with low-code development tools and eliminate reliance on human skills for post-development maintenance.

[3] Identify and resolve problems earlier by detecting system abnormalities (error handling and illegal operation).

[4] Facilitate data linking between different systems through data aggregation and integration.

[5] Automate the system configuration management with IaC[Note 4] and testing and deployment with CI/CD[Note 5], preventing update errors and eliminating reliance on human skills for administration and development.

Note 1) An acronym for Artificial Intelligence.

Note 2) The Digital Transformation Improvement Dept. and IT Planning Dept. were unified into the Digital Transformation Improvement Division in April 2023.

Note 3) KYB has traditionally developed itself various systems in the production field. These systems have been operated by the Digital Transformation Improvement Dept.

Note 4) An acronym for Infrastructure as Code. Refers to building system infrastructure using codes.

Note 5) An acronym for Continuous Integration/Continuous Delivery. Refers to an approach to software development that uses automation in the processes of building, testing, releasing, and deploying software through the use of dedicated tools to achieve efficient development, labor savings, and earlier reflection in the production environment.

2 Overview and Challenges of the Conventional System

This section describes the conventional quality data management system. This system collects and accumulates quality information, including product processing conditions and test measurements collected by equipment and testers. Through the information, operators can identify changes and trends in measurements and use them for daily quality control and traceability in case of failure. This system has been introduced to six sites inside and outside Japan that produce power steering and/or CVTs[Note 6]. More than 20 years have passed since the first system was introduced at one of these sites.

Note 6) An acronym for Continuously Variable Transmission.

2.1 System Configuration

Fig. 1 shows the block diagram of the conventional system. The system puts together files that record quality data collected from various equipment units on an on-premises server, partially formats the data on the server, and stores it in a relational database (RDB) for centralized management. The RDB can be accessed with dedicated software developed in-house, allowing operators to view performance data and register master data necessary for the system. The RDB is based on an Oracle database. The software used to view performance data and register master data was developed in-house using Microsoft Visual Basic .NET.

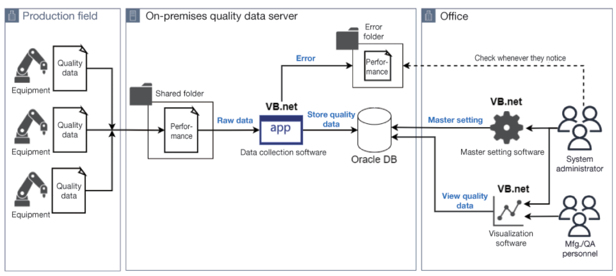

Fig. 1 Block diagram of the conventional system

2. 2 Challenges of the Conventional System

The conventional system has the following challenges:

[1] Increased man-hours and costs to operate and manage the system
- Man-hours and costs have increased due to regular server updates and maintenance and responding to aging issues.
- RDB licensing and maintenance costs have increased.
- Many man-hours have been spent adding functions and fixing problems due to the absence of the original developers and the inadequacy of documents.

[2] Inadequate failure/disaster response
- The system will inevitably shut down if the server or RDB fails due to the lack of a redundancy of the system.
- The system may not be recoverable in case of a natural disaster because there is no backup in a geographically remote location.

[3] Difficulty in promoting the use of data
- It is difficult to link the system with others because it was built for individual optimization.
- The technology is outdated with inefficient processing because the system was designed more than 20 years ago.

3 Requirements

This section describes the requirements for a new system to be viable by solving the challenges of the conventional system:

[1] The system should be built on the KYB-IoT Platform along with a corresponding AWS-based cloud.

[2] System reconstruction should be made to provide an optimal mechanism, rather than just cloud migration.

[3] Conventional core functions should be maintained, while those that cannot be easily used should be improved.

[4] Horizontal expansion and function addition/maintenance should be easy.

[5] The RDB should be transferred from the commercial DB to an OSS[Note 7)] based database.

Requirement [2] needs some additional information. AWS proposes a set of cloud migration strategies called "the seven Rs"[6]. These migration strategies are listed in Table 1 in order of difficulty. Among them, the most difficult strategy, namely Refactor, is what we are trying to achieve in this development project. This strategy requires a long time for development but can be expected to bring the most effect from cloud migration.

Table 1 Seven cloud migration strategies

Migration strategy	Outline
Refactor	Redesign applications to take full advantage of cloud-centric functions.
Replatform	Move specific components to a cloud-based service.
Repurchase	Perform cloud migration by purchasing SaaS and application packages.
Rehost	Move the conventional on-premises configuration directly to cloud infrastructure.
Relocate	Move the conventional on-premises configuration directly to the VMWare Cloud on AWS.
Retain	Keep the system as-is without cloud migration.
Retire	Retire the system.

Note 7) An acronym for Open-Source Software. Refers to free software whose source codes can be freely modified and distributed.

4 Development of a New System

4. 1 System Configuration

Fig. 2 shows the block diagram of the newly developed system.

While the production field is untouched, the following field, surrounded by a red line, has been reconstructed.

A Microsoft Azure virtual server is used to aggregate quality data files collected from various equipment units. This is the only IaaS[Note 8)] among the resources built in the cloud. This was because data collection without changing the processing on the equipment side required a Windows server as for the conventional system. Since KYB has built several Windows servers on Azure, this system also has an Azure-based server for integrated management. This server is mainly used to cache quality data files and then transfer them directly to AWS.

The process of storing collected quality data for centralized management has been built on AWS. Most of the processing uses AWS managed services, although their details are not shown in the Figure. This reduces the administrator's workload and makes the system configuration resilient to failures and disasters.

For reference, Fig. 3 shows a comparison of RDB management between the conventional and new systems.

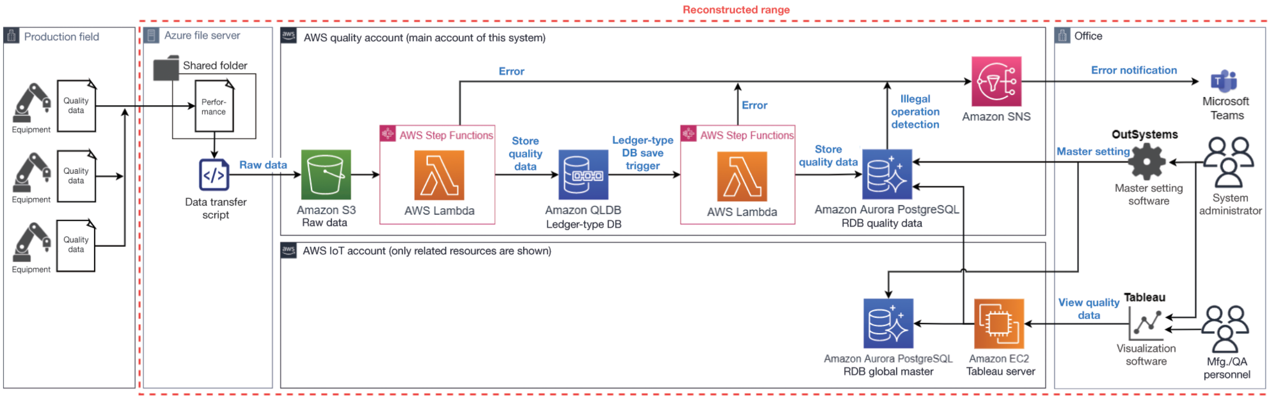

Fig. 2 shows the block diagram of the newly developed system.

The new system eliminates the management of servers, networks, and DB middleware, substantially reducing the administrator's workload.

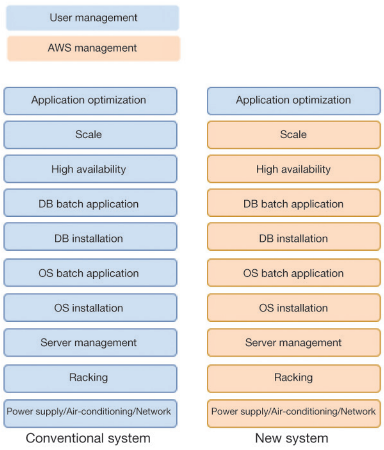

Fig. 3 Comparison of responsibilities for RDB building, operation, and management

We have redeveloped a visualization software that allows users to view collected quality data and a software for setting master data required by the system by using low-code development tools[Note 9)] called Tableau[Note 10)] and OutSystems[Note 11)]. Both tools enable easy software development with a small number of source codes mainly through drag-and-drop actions, which significantly reduces the development man-hours from the conventional level. For more information about the developed software, see Section 4.5.

This development project consists of six main specific activities as below. Each is described in the following sections.

[1] Data Migration
[2] Data storage and management
[3] Detection of errors and illegal operation
[4] Development of visualization software
[5] Development of master setting software
[6] Improvement of data linking

Note 8) An acronym for Infrastructure as a Service. Refers to cloud services that provide infrastructure.

Note 9) A tool used to develop software with minimal source codes.

Note 10) A Business Intelligence (BI) tool provided by Salesforce that enables visual analysis with a variety of graphs and charts.

Note 11) A low-code development platform provided by OutSystems that enables development of various applications including Web and mobile applications.

4.2 Data Transfer

Before the new system is put into operation, the data accumulated in the conventional system must be transferred to the new system. Instead of directly transferring the data from the conventional configuration, this development project transferred the data with the following modifications:

[1] The database type was changed from Oracle to OSS PostgreSQL.
[2] The configuration and names of schemas and tables were changed.

Reasons for change [1]:

- Choosing the OSS DB reduces the licensing and maintenance costs.
- The transfer was relatively easy because the PostgreSQL has many data types and functions that are highly compatible with those of Oracle.
- The PostgreSQL can deliver high performance.

Reasons for change [2]:

- The conventional DB needed to be reviewed in terms of configuration, as its bloated data caused lower performance.
- The data storage method was unified with a view to future integration with other systems.

The data transfer process is shown in Fig. 4.

First, the on-premises Oracle data was transferred to Amazon RDS for Oracle. After partial data processing, AWS Database Migration Service (DMS) was used to transfer all data to Amazon Aurora PostgreSQL (Aurora PostgreSQL). Additional data processing and conversion was also performed on Aurora PostgreSQL to complete the data transfer in the intended configuration.

This data transfer method involves a simple procedure that transfers all the data at once but requires the system to interrupt some functions during the data transfer to avoid updating data other than the data being transferred. Of course, there are other methods where the system only needs to be interrupted for a short time, but the transfer work will be correspondingly complicated and difficult. Finally, we used this data transfer method with the simple procedure because this system could accept several days of system interruption.

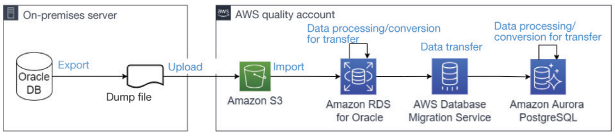

Fig. 4 Data transfer process

4. 3 Data Storage and Management

Quality data collected from the equipment is important information used to prove the quality and performance of the products. In this development project, the data storage system is designed to eventually store data in two databases: Amazon Quantum Ledger Database (QLDB) and Aurora PostgreSQL. QLDB is a full managed ledger-type database where data, once stored, cannot be physically modified even by the administrator. Aurora PostgreSQL is a high performance, managed RDB built by AWS for use in the cloud that achieves higher performance and availability than regular RDBs.

Storing quality data in the QLDB prevents physical alteration of the data, thus providing a tamper-proof data management system. However, QLDB is primarily intended to be used for data storage and management and can hardly provide such complex data retrieval as seen in the conventional linking of multiple information sets. We then designed the system to store quality data also in RDB Aurora PostgreSQL so that users can access Aurora PostgreSQL to retrieve data. The master data needed to operate the system is also managed in Aurora PostgreSQL so that it can be modified as circumstances require.

The two different types of databases are used in this way to ensure data management that achieves both tamper prevention and data retrieval.

4. 4 Notification of Processing Errors and Illegal Operation

Quality data is checked for proper format and information before it is stored in the databases. The data storage process takes place only after the data has passed the final check. Of course, the conventional system also provided this kind of data checking function, which, in case of error, moved the corresponding file to an error data folder. However, the conventional system did not notify the occurrence of errors, resulting in a delayed response in some cases.

For the new system, we then developed a processing function to notify Microsoft Teams (Teams) of the occurrence of errors. Fig. 5 shows an example of screens containing messages notified to Teams[Note 12)]. These messages allow the user to identify in which file for which equipment the error has occurred. In addition, clicking on a URL on the message screen will display the master setting software described later. Using the software, the user can download the file with the error to respond to the error. This error notification and response mechanism has been used to solve the conventional problems.

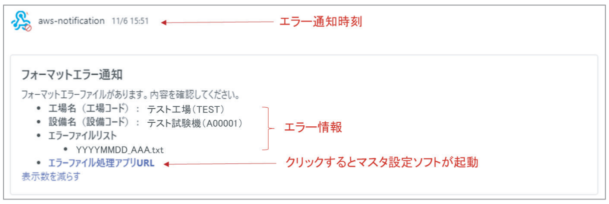

Fig. 5 Example of format error messages to Teams

The notification function has also been deployed to make Aurora PostgreSQL tamper-proof. As described in section 4.3, data stored in QLDB cannot be tampered with even by the administrator, so no specific measures are required. However, the data that the user sees is the RDB on Aurora PostgreSQL. If data in the DB is manipulated, it is a problem. On the other hand, Aurora PostgreSQL provides strict access management that prevents data modification by general users, which means that data tampering never occurs during regular use. However, if the administrator's credentials are stolen and used to access Aurora PostgreSQL, data can be tampered with.

Then we designed the system to notify Teams that any quality data stored in Aurora PostgreSQL has been modified or deleted, or any operation that could lead to data addition or illegal connection has been performed by an irregular user or from an irregular location. Fig. 6 shows an example of screens containing messages notified to Teams[Note 13)]. These messages allow the user to identify who performed what operation for what, when and where. Based on this information, the user can determine whether or not an illegal operation has occurred and then take action if necessary. In addition, the user can compare the data with the data stored in the QLDB, if necessary, to determine whether or not data tampering has occurred and to determine the true value of the data.

Note 12) The messages shown in the Figure are only examples and may differ from the actual ones.

Note 13) The messages shown in the Figure are only examples and may differ from the actual ones.

Fig. 6 Example of DB tampering messages to Teams

4. 5 Development of Visualization Software

To visualize the collected quality data, we developed a visualization software using Tableau. Below are some of the screens of the developed software.

4. 5. 1 Measurement Data Display Screen

This screen shows how the collected quality data changes over time using graphs and tables (Fig. 7)[Note 14]. From this screen, the user can determine the trend of quality data for daily trend management, including identifying abnormal data or data that may become abnormal in the long run.

Note 14) Some information that should not be disclosed has been intentionally deleted or shaded.

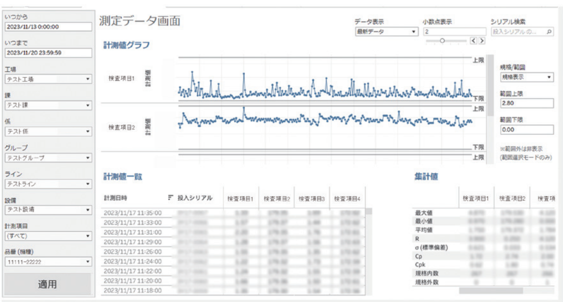

Fig. 7 Measurement data display screen

4. 5. 2 Histogram Display Screen

This screen displays the histogram of collected quality data (Fig. 8)[Note 15]. On this screen, the user can check the variation of the quality data to identify any abnormal data for daily trend management.

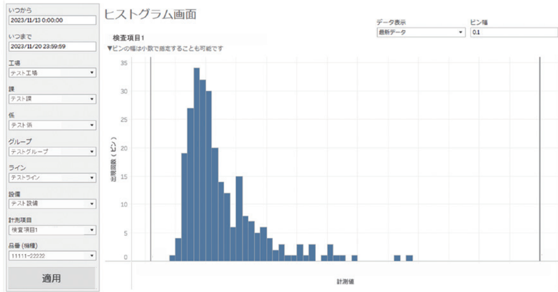

Fig. 8 Histogram display screen

Note 15) Some information that should not be disclosed has been intentionally deleted or shaded.

4. 5. 3 Traceability Search Screen

This screen provides an at-a-glance view of the assembly and subassembly information associated with a product's serial number (Fig. 9)[Note 16]. If a problem occurs, the user can identify the related parts from the serial number to determine the scope of the problem.

Fig. 9 Traceability search screen

Note 16) The data shown in the Figure are only examples and may differ from the actual ones.

4. 6 Development of Master Setting Software

We have developed the master setting software using OutSystems to allow registration of master data necessary for system operation and to support correction of any data format errors. Below are some of the screens of the developed software.

4. 6. 1 Quality Inspection Items Master Registration Screen

This screen is used to register inspection items for specific quality data to be collected (Fig.10)[Note 17].

Registration can be done on this screen to initiate the collection of new data or to change inspection items in the middle of a data collection session.

Note 17) The data shown in the Figure are only examples and may differ from the actual ones.

Fig. 10 Quality inspection item master registration screen

4. 6. 2 Form Error File Processing Screen

This screen is used to identify files that were found to be in error during the format check process prior to saving data, and then to re-upload or delete these files if necessary, as described in Section 4.4 (Fig. 11)[Note 18]. The screen provides an at-a-glance list of erroneous files, from which any file can be selected and downloaded. The user can examine the file to correct errors and re-upload the file. The master setting software provides access rights for each screen, and this screen can only be accessed by designated administrators. This screen also records the operation history. An incorrect operation, if any, can be then traced.

Note 18) The data shown in the Figure are only examples and may differ from the actual ones.

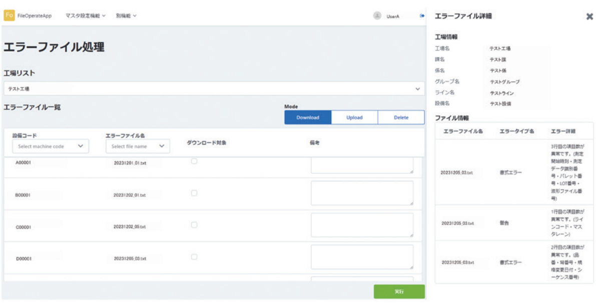

Fig. 11 Format error file processing screen

4. 7 Improvement of Data Linkage

In the conventional quality data management system, the master information was managed by individual systems. Even if different systems collected information about the same line, the systems could hardly link their data with each other because their line control units were different. We then created a global master table to manage the master information of plants, lines, equipment, and others that may be used by multiple systems, and built a mechanism for centralized management (Fig. 12). This mechanism allowed different systems to obtain information from the same line and equipment, thus improving the data linkage between them.

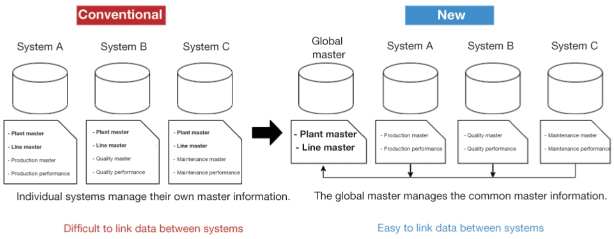

Fig. 12 Improvement of data linkage

At the time of writing this paper, this newly developed global master system has not yet achieved data linkage with other systems. However, the link will be made not only with the quality data collected by this system, but also with manufacturing, equipment maintenance, and other information to enable diversified data analysis. This is expected to contribute to higher quality and productivity.

5 Efforts for Continual Development and Operation Scheme

5. 1 Multi-account Cloud Operation

KYB has started to provide comprehensive services with multiple systems built on AWS. To ensure that the systems with different access rights under different service environments are operated securely, the company provides multi-account operation by assigning a separate AWS account to each application.

Fig. 13 illustrates this multi-account management and operation scheme. AWS Organizations is used to provide integrated account management, preventing the omission of minimal settings (such as log collection for security and auditing) for certain accounts. In addition, AWS IAM Identity Center is used to provide single sign-on so that each account can only be accessed with appropriate privilege.

Multiple accounts were created individually because this system handles quality data that requires even tighter access control. In addition, different accounts have been assigned to the development and production environments with different access rights. For example, developers are allowed to read and write data with the development environment account but are not allowed to write with the production environment account. These separate accounts are used to ensure proper access control to prevent developers from accidentally shutting down the production system and other problems.

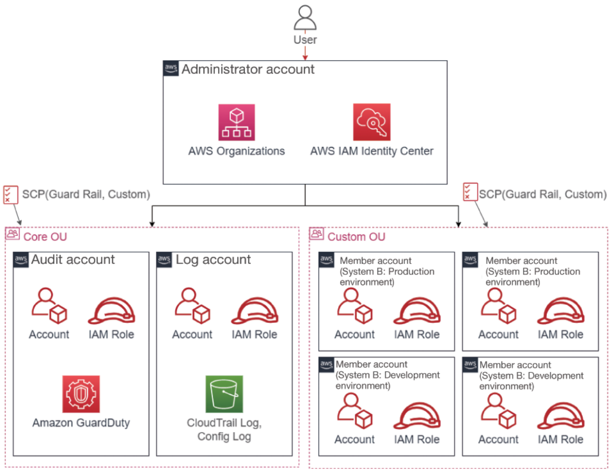

Fig. 13 Multi-account management and operation

5.2 System Development with IaC

In view of future developability and maintainability, this development project uses IaC for system development. In system construction, the use of IaC increases the efficiency of reuse, reduces human errors due to manual execution, facilitates version control, and realizes a CI/CD mechanism that enables automation of processes including testing to deployment. (Version control and automation of processes up to deployment will be explained in the next section).

In this development project, workflows and applications built with AWS Step Functions, AWS Lambda, or other services (including the data storage described in Section 4.4) were created using the Serverless Framework, while databases, networking, and security-related infrastructure were built using HashiCorp Terraform.

5.3 Automation of Testing and Deployment with CI/CD

Using IaC facilitates version control. In this development project, a source code version control tool called GitLab is used to control versions of the applications and infrastructure described in Section 5.2. GitLab includes a CI/CD feature that is used for testing of source code pushed into the Git repository and for deployment to AWS Clouds. Fig. 14 shows the source code flow from creation to deployment to AWS Clouds Note 19).

The first step is to create a source code. A feature branch that contains the current version of the source code is created in Gitlab, and work is performed in that branch. When the work is done, the source code is pushed to the feature branch, and then a merge request is made to the develop branch. At this point, GitLab

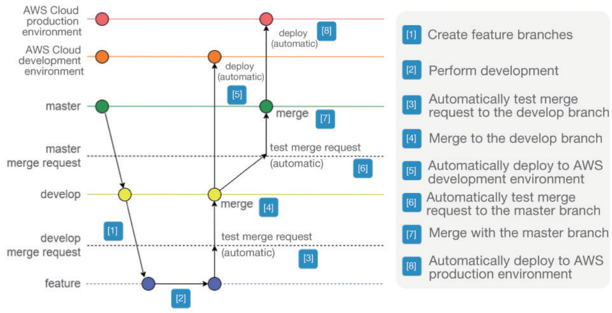

Fig. 14 Workflow from development to deployment using GitLab

automatically verifies the source code to make sure there are no problems. After that, the administrator performs a manual check, and if there are no problems, performs the merge. At the same time as the merge, the deployment task to the AWS Cloud is automatically executed, and the applications and infrastructure described in the source code are finally deployed to the development environment in the AWS Cloud. In addition, when merging from the develop branch to the master branch, automated testing and deployment tasks are executed in the same way, and finally the applications and infrastructure are deployed to the production environment in the AWS Cloud. By performing the above sequence of steps from development to deployment to the AWS Cloud, we can eliminate work errors and reduce the number of man-hours required.

Note 19) The Company's Git management is based on Git Flow Note 20). We merge the develop branch into the master branch without using the release branch.

Note 20) One of the workflows for effective version control and development proposed by Vincent Driessen.

6 Prospects

This new quality data management system was actually implemented at one site and put into full operation as of December 2023. It has not yet produced any remarkable results in this short period of use, but it has gradually started to show effects, including a reduction in the number of man-hours for management due to the elimination of on-premises servers and a reduction in the number of man-hours for daily work due to added functions. From now on, the system will be expanded horizontally to the remaining five sites, completing the migration from the legacy quality data management system.

In addition to this system, KYB has many other legacy systems. By leveraging the insights gained from this development project, we will sequentially perform the cloud migration or re-architecture of these systems to complete the exit from all kinds of legacy systems.

In fact, just breaking away from legacy systems will not facilitate DX. By simultaneously achieving data linkage with other systems and building a data analysis base using AI, we will activate the use of data and promote DX, contributing to higher productivity and higher quality.

7 Concluding Remarks

This development project has successfully built the next-generation system to greatly improve the failure/disaster control, provide diversified analysis by linking with other multiple systems, and revolutionarily reduce both cost and man-hours. As a result, this new system has resolved many problems of the conventional legacy system, gaining a beachhead from which to promote DX.

However, the migration from the legacy system is only a preparation for DX improvement. We will continue to pursue DX improvement by laying the groundwork and developing functions for continuous data utilization.

Finally, I would like to take this opportunity to express my sincere gratitude to all those in related departments who have provided substantial support and cooperation for this development project.

References

1) Information-technology Promotion Agency, Japan (IPA): “DX White Paper 2023”, p.9 (February 2023).
2) NAITO: “AI x IoT Initiatives in the Production Technology Domain at KYB”, KYB Technical Review No.60 (April 2020).
3) FURUKAWA, ISASHI: “Development of an Equipment Predictive Maintenance System”, KYB Technical Review No.63 (October 2021).
4) OHUCHIDA, MIYAUCHI, SAGEHASHI: “Use of AI Technology in Development of SA Elements”, KYB Technical Review No.66 (April 2023).
5) MIYAUCHI, ISASHI, TAKINO: “Initiatives to Develop Digital Personnel”, KYB Technical Review No.66 (April 2023).
6) Amazon Web Services, Inc.: “What is Cloud Migration?” https://aws.amazon.com/what-is/cloud-migration/?nc1=h_ls (Accessed on December 12, 2023).

Author

FURUKAWA Akira

Joined the company in 2005.
System Development Sect., Digital Transformation Improvement Div.
Engaged in building of cloud-based IoT platforms and development of production systems.