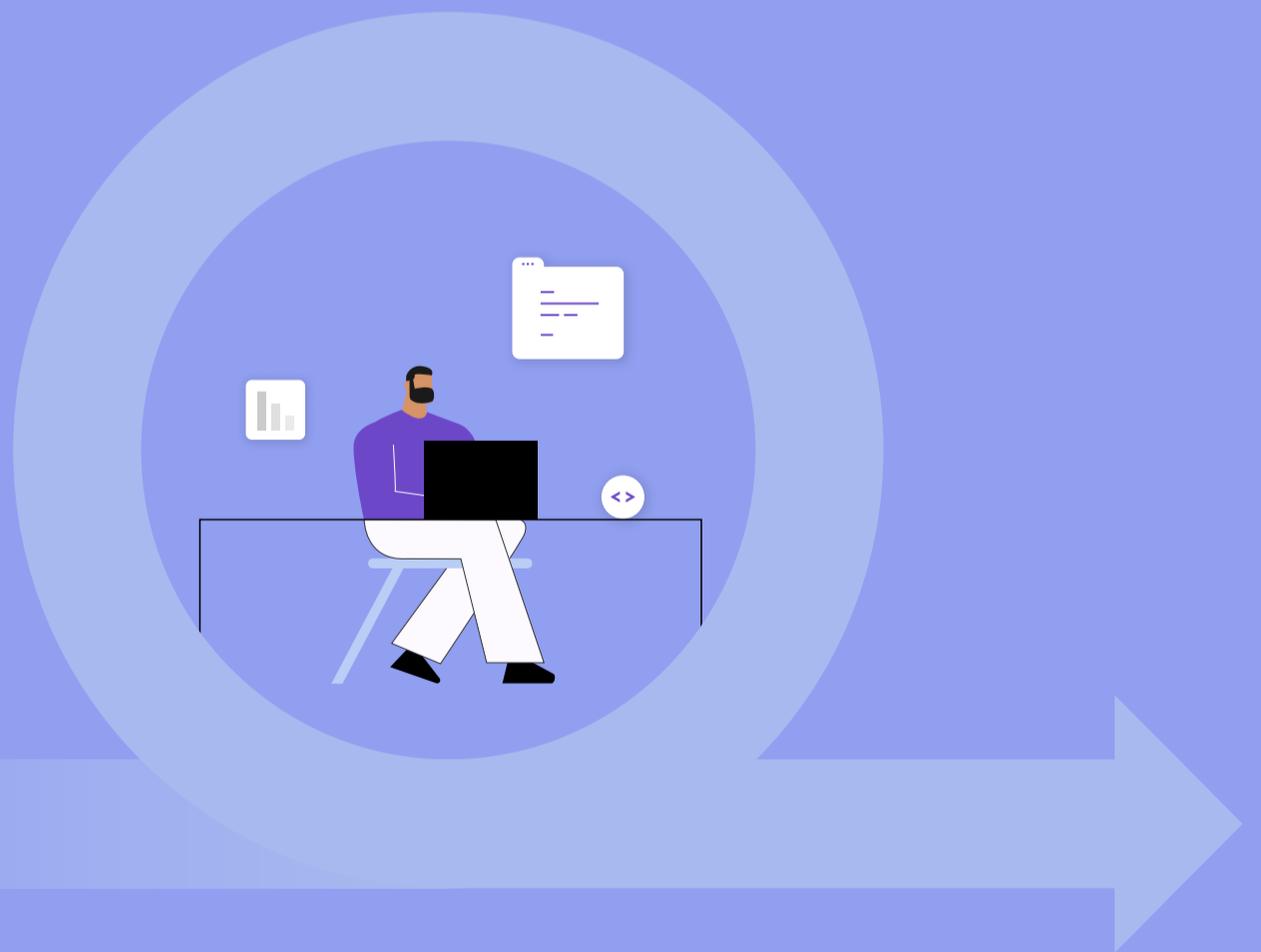


Japan

Mitsubishi Electric Corporation, Kamakura Works

Whitepaper



GeneXus™



Why did the Information Systems Department, which used to develop the system entirely in-house, focus on low-code development and decide to introduce it?

Mitsubishi Electric Corporation, Kamakura Works (hereinafter referred to as “Kamakura Works”) has been developing all the IT systems used in the office by itself, making full use of the latest technology at the time. We asked its members how they introduced GeneXus, a low-code development method, and G.RAD.E to further increase productivity from 2018.

Every system was made in-house except for the company's communal system:

The Information System Section of the Manufacturing Management Department of Kamakura Works, which we spoke with this time, is responsible for the maintenance and management of all the PCs, servers and networks used in the plant, as well as the development and maintenance of most of the systems used in the plant. Mitsubishi Electric Information Network Co., Ltd. is famous not only as a system integrator for Mitsubishi Electric, but also as an **IT system integrator for a wide range of industries.**

The Kamakura Works is not a division that only manufactures Mitsubishi Electric products. It is highly independent from the head office, and the scope of work is very similar to that of a normal company, including management of orders from the sales department, production management, and personnel task management, so this section develops and maintains all the IT systems operated at the Kamakura Works in addition to the ones that are common to the entire company.

Naturally, we are constantly improving and upgrading the system on a proposal basis in cooperation with each department to improve business operations.



Information System Section, Manufacturing Management Department, Kamakura Works, Mitsubishi Electric Corporation (from left) Hiromi Higashi, Team Leader, Toshiaki Nakamura, and Shimpei Katsumoto, Section Chief.

Problems that arose in the department that had chosen the latest development base:

Since we think about and promote business solutions on our own rather than leaving it to the vendors, we have always sought and selected the optimal solution for the development infrastructure and language to match the appropriateness of the system to be developed.

As a result, some of these systems, which were initially the latest technology, have become legacy systems as technology advances. Recently, the loss of support for Oracle Developer 2000, which we have been using since 2000, has been an issue.

Recently, we have been developing with Java, but for further productivity improvement, we investigated Low-Code development, which we had been focusing on for a long time.



Mitsubishi Electric Information Network Corporation
(from left) Mr. Toyotaka Nakamura, Mr. Tomoaki Takano,
Mr. Gen Ohno, Mr. Hideki Enomoto.

Select GeneXus from a number of low-code development platforms:

From the information we gathered on the Internet, we chose a development platform that met the security constraints inherent in the Kamakura factory, such as the inability to use the cloud, and in the end, GeneXus and another visual modeling tool were the two finalists. The deciding factor was the licensing system.

“GeneXus charges per developer’s license, whereas visual modeling tools charge per program developed,” he said. “For someone like us who develops a lot of systems, GeneXus, which charges by the number of developers, is more cost effective.”

In addition, the ability to reuse the SQL, stored procedures and other assets of the system currently in use was also a key point.

In addition, the freedom of choice of operating environment and language was an important factor. It was also good to have multi-platform support and the ability to choose the language to be generated. The sustainability of being able to migrate to a new environment with the same source code, without being overwhelmed by future OS or database changes, is very attractive.

With the mixture of various DB and OS versions, it is not practical to modify each program individually, as it would be a lot of work.

«GeneXus charges per developer’s license, whereas visual modeling tools charge per program developed. For someone like us who develops a lot of systems, GeneXus, which charges by the number of developers, is more cost effective.»

Explore templates for further productivity gains:

Kamakura Works introduced GeneXus in 2018. The following year, we started to develop the system in earnest, and as of July 2020, we have released seven systems.

At that time, we happened to be able to exchange ideas with another user company at a seminar by a tool vendor called Genexus Day.

The company advised us that we could increase our productivity by using a template product like WorkWithPlus rather than using GeneXus on its own. So we asked the company to explain to WeING

Co., Ltd., who had requested development support for GeneXus.

At that time, WeING Co., Ltd. had just released a template infrastructure product **G.RAD.E** (GeneXus Rapid Application Development Environment) utilizing WorkWithPlus and received this introduction.

We were interested in G.RAD.E, its applicability to practice and its productivity improvement, and decided to try and evaluate it in a business that was scheduled to be released on the premise of Java development at the time.

Actual productivity going up in G.RAD.E and future challenges:

The results of the evaluation of the system that was planned to be released in JAVA and developed in G.RAD.E are shown in **Table 1**. Table 1 shows the evaluation results for each theme.

As far as the evaluation sheet is concerned, some of the screens are more productive and some of the functions are not so productive.

As we developed it with zero knowledge, it took us a little time to learn and get used to it, but the development time in the table includes that.

We feel that it could have been done more effectively by someone who is more familiar with it.

Migrating the current system to GeneXus in exactly the same way would take a lot of time, but if you develop it using the templates provided, it will work.

If you're not familiar with how to choose a template, there will be rework and trial and error. **I feel that once I get used to it, my productivity will improve.**

Personally, we're under the impression that it's going to be at least 50% less than Java. It's convenient to be able to automatically create parts of the code that would have to be implemented in Java, especially the standard output part.

We asked them about the challenges of using G.RAD.E. It was difficult to create an environment with the security constraints

unique to Kamakura Seisakusho. Standardization of development and release procedures is necessary, but we need to prepare for G.RAD.E.. We are groping for a way to change the traditional waterfall approach and development methodology to one that is best suited for GeneXus + G.RAD.E. Establishing a methodology for team development is a challenge for the future.

Table 1

No.	Target Screen	Type of Screen	Java Estimated man-hours (person - days)	Number of man-hours (person - days)
1	Payout History Search	List Search Main-Sub screen	5.5	1
2	Location number master maintenance	Mastermente List Search	4.5	0.3
3	PWB Design Request System Menu	Menu	1	0
4	PWB Design Dependencies List	List Search	5	4
5	PWB Design Dependencies	Details List Search	15	13
6	Attachment Screen (PWB Design Dependencies)	Sub-screen	5.5	1
7	Department name selection screen (PWB Design Dependencies)	Sub-screen	5	1
8	Show past comments (PWB Design Dependencies)	Sub-screen	3	1
			44.5	21.3

Find out how GeneXus
can do the same for your company.

info@genexus.com

