# Migration Integration Services in CPU Replacement Software Development

**Yuuichi Ikeda** **Naoya Tonooka**

OKI IDS Co., Ltd. (OIDS) is a design company that primarily develops video, audio, and communications equipment, as well as industrial equipment. In the past, the company has been contracted to develop new devices to replace discontinued devices in existing products, but this has often required a significant amount of time and cost to ensure mass production quality during the evaluation process after the replacement.

In recent years, demand for replacements following the discontinuation of a device has been increasing, and it was necessary to establish a development process that would both shorten development time and ensure mass production quality.

## What is Device Replacement Development?

Device replacement development refers to the development of a CPU replacement due to the discontinuation of the CPU running in an existing mass-produced product.

Device replacement development requires functional and performance compatibility with the existing product, but in many cases, the system is aged and the original design documents no longer exist. In such cases, development begins with unclear functional and performance requirements, posing an issue with the significant amount of time required to validate equivalence with the existing product after the replacement.

## Establishing Proprietary OIDS Process

Leveraging OIDS' past experience with device replacement development, the company established a proprietary software development process for efficiently replacing aged systems with next-generation devices, even when design assets are no longer available. This development process makes it possible to rebuild a system into one that maintains the same functionality and performance as the existing system. The proprietary OIDS development process is being promoted externally through web seminars in an effort to expand software contracts.

This article introduces OIDS' Device Replacement Development Process for software, applying measures to avoid and mitigate issues/risks that are expected to arise during device replacement development.

## Reason for Reverse Engineering

In today's business environment, system updates are essential as technology continues to evolve. Existing software and systems must constantly evolve, at times to maintain functionality and efficiency and at other times to meet new business demands. Future-proof replacement development is key to taking the business to the next level. Reverse engineering plays a crucial role in this pre-development process.

At OIDS, reverse engineering is used to gain a deep understanding of existing software and systems and perform detailed functional analysis in the early stage of development. This process fully unlocks the value of existing technology and provides insights that form the basis for future development plans. Successful replacement development relies on understanding the potential and limitations of existing assets and how to integrate and utilize them in new solutions.

Market trends indicate that software lifecycles are shortening and technology choices are diversifying, forcing organizations to upgrade their systems more quickly and flexibly. OIDS' Reverse Engineering Service helps clients maintain their competitive advantage and overcome technical barriers in this dynamic market environment.

OIDS' expertise lies in uncovering the role and value of legacy systems, paving the way for a smooth transition to the next generation. From deep understanding of software architectures to feature identification, compatibility assessment, and optimal update strategy formulation, reverse engineering at the early stage lays down a solid foundation for wise investment decisions and sustainable system development.

## Issues in Device Replacement Development

This section presents four issues faced in device replacement development.

**(1) Lack of Original Design Documents or Absence of Device's Designer**

In many cases, the design documents for the existing product are lacking or not saved in electronic format. Furthermore, if the designer of the device has retired or been transferred from the project, their technical knowledge and design intent may not be passed on. As a result, time is lost investigating the design purpose, risking significant delays in the device replacement process. In such a situation, after replacement measures are taken, continuing mass production of the key equipment and launching it into the market will be difficult.

**(2) Original Design Is Outdated and Source Code Cannot Be Reused As-Is in New Device**

Often, the existing software design is too outdated to be directly reused in the new device. Therefore, extensive refactoring and redesign of the source code is required, increasing development cost and time.

Compatibility issues also arise, requiring significant effort for operational verification and debugging. As a result, it becomes difficult to guarantee that the replacement product's functionality and performance are equal or better than the existing device.

**(3) Insufficient Documentation and Understanding of Existing System**

There are cases where documentation for an existing system is insufficient, making it difficult to fully understand the overall system structure and operation. This makes switching to new devices difficult and an extensive amount of time is required for testing and validation.

**(4) Eliminating Duplicate Codes and Streamlining Source Code**

Source code that has undergone repeated maintenance and revisions over the years tends to contain many duplicate and redundant sections. This reduces code readability and increases the risk of bugs and maintenance difficulties. Performance optimization also becomes an issue when migrating to new devices.

## Devised Solution: N-Process™ *1)

Previously, the device replacement development process was carried out using a V-shaped process, based on the V-model[1)] software development methodology. This development process is shown in **Figure 1**. To address the aforementioned issues required an improvement development stage in the latter half of the process to ensure mass production quality.

**<Previous Development Process: V-Process>**

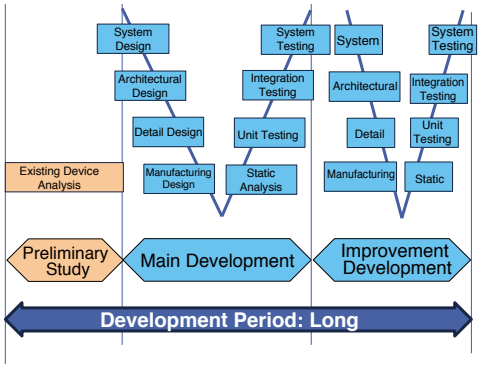


**Figure 1. Previous Development Process**

To shorten the development time, OIDS devised its own method referred to as the N-Process. In the N-Process, reverse engineering is carried out during the preliminary study, and the development details are thoroughly examined. This makes it possible to replace the existing device with a system that satisfies the functions and performance of that existing device without the improvement development that was performed in the latter half of the previous process. The new development process is shown in **Figure 2**.

**<New Development Process:*N-Process*>**

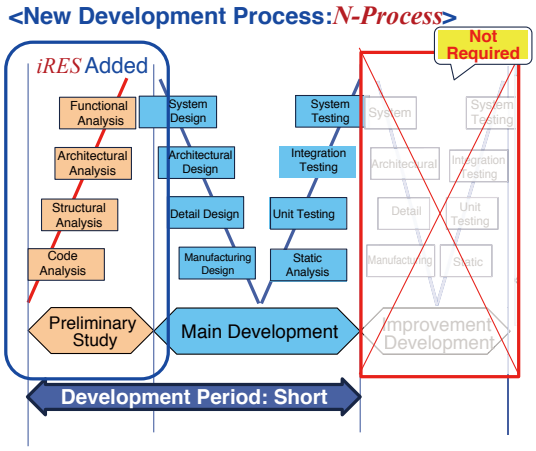


**Figure 2. New Development Process**

*1) N-Process is a trademark of OKI IDS Co., Ltd.
*2) iRES is a trademark of OKI IDS Co., Ltd.

As shown in **Figure 2**, addition of the reverse engineering process iRES™*2) before the V-Process (main development) enables the development details for the device replacement to be thoroughly examined, making it possible to replace and develop a system that satisfies the functions and performance of the existing device without the improvement development stage.

## Details of New Development Process "N-Process"

Details of the new N-Process is given below.

### (1) Accurate Reproduction of Design Intent: Detailed Analysis Through Reverse Engineering

In the N-Process, reverse engineering is performed in advance for a detailed analysis of the existing device's design intent. An overview of the reverse engineering method is shown in **Figure 3**.

First, the source software's characteristics, properties, and evaluation criteria are identified to establish a development baseline. This allows the original design to be accurately reproduced, preventing development delays, even if the design document or designer is not available.

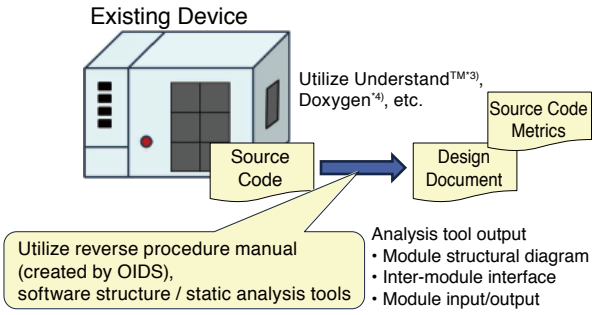


**Figure 3. Reverse Engineering Method**

### (2) Streamlining and Optimization: Eliminating Duplicate Codes and Checking Compatibility

Based on the information obtained through reverse engineering, duplicate sections in the source code are eliminated and streamlined. The goal of this phase is to improve the efficiency and readability of the code base without compromising the existing functionality. Equivalence is thoroughly assured to ensure that functionality and performance are equivalent to the existing version after the replacement.

Next, the optimal replacement device is selected and compatibility with the new device is thoroughly checked. Focus is placed on maintaining the same functionality and performance as the existing device to assure equivalent replacement (**Figure 4**).

If improvements in functionality and performance are desired, a separate phase is carried out after the equivalence assurance is completed, in which redundant code is deleted while development is carried out to achieve functionality and performance that exceeds the existing device.

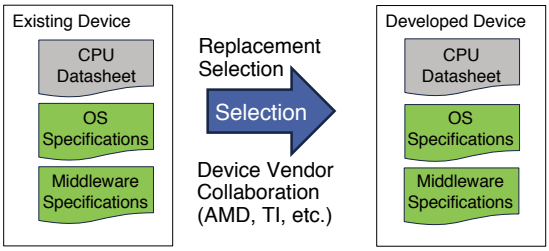


**Figure 4. Selection of Replacement Device**

### (3) Building an Evaluation Environment and Validating Equivalence: Validating Functional Equivalence BetweenExisting and Developed Device

An evaluation environment capable of simulation using actual machine code is built, and the same evaluation data is used for both the existing and developed devices (**Figure 5**). The evaluation environment is used to thoroughly validate the functional equivalence between the existing and developed devices, ensuring equivalent functionality and performance after the replacement. Adjustments to the control methods associated with device change are also made to ensure equivalent functionality and performance as the existing device. With this process, assurance of equivalence is established.

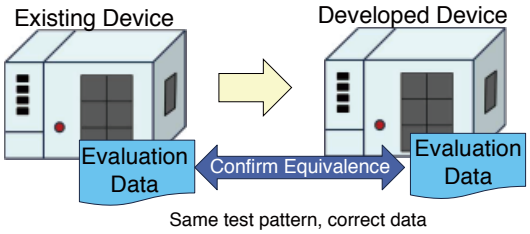


**Figure 5. Building Evaluation Data and Validating Equivalence**

*3) Understand is a source code analysis tool and is a trademark of Scientific Toolworks, Inc (SciTools). Japanese license provided through TechMatrix Corporation. https://www.techmatrix.co.jp/product/understand/index.html (in Japanese)

*4) Doxygen: A free source code documentation tool. https://sourceforge.net/projects/doxygen/

### (4) Promoting Development with V-Process: Effectively Utilizing Reverse Engineering Data

In the V-Process portion of the N-Process, design documents and evaluation data created through reverse engineering are thoroughly analyzed and examined to identify potential issues in advance so that occurrence of problems is minimized. Furthermore, the created design documents and evaluation data can be reused during the main development. An example of this reuse is shown in **Figure 6**.

This eliminates the need for the improvement development stage after the main development. Eliminating the unnecessary process significantly shortens the development time and builds a development plan that enables clients to quickly bring their equipment to market.

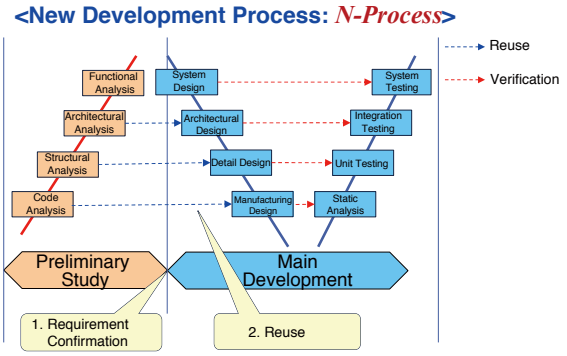


**Figure 6. Reuse of Design Documents and Evaluation Data**

## Future Plans

This development process was established in fiscal year 2024, and OIDS is currently accumulating development results. The process will be applied to more projects and the results will be evaluated to provide feedback to the development process for continuous improvement. ◆◆

## References

1) "V model," Free Encyclopedia Japanese Wikipedia, August 1, 2024, 20:06 UTC

## Authors

**Yuuichi Ikeda**, Development Department, OKI IDS Co., Ltd.

**Naoya Tonooka**, Development Department, OKI IDS Co., Ltd.

## TIPS【Glossary】

**N-Process**
OKI IDS' proprietary software development process specialized for device replacement development.

**iRES (intelligent Reverse Engineering Service)**
Preliminary study portion of the N-Process described above.

**V-Process**
V-model software development methodology.