

# Legacy Modernization for Continuous Use of Information Systems

*Hideaki Tanino*

### Abstract

Large-scale mainframe systems are becoming more and more complex and their internal structure are becoming unknown, which leads to higher maintenance, development, and system renewal costs. Migration to open systems can reduce these costs, although migration to a completely different platform is not easy. This article explains the difficulties in using open systems for rehosting of large-scale mainframe systems and presents solutions using automated system development techniques to address such difficulties.

*Keywords: legacy modernization, rehosting, mainframe*

### 1. Introduction

Mainframe computers were at their peak in the 1980s; the volume of domestic shipments was approximately 3500 units, and the value of total shipments amounted to over 1 trillion yen. Since the middle of the 1990s, however, mainframes have been overwhelmed by the wave of open-source computers, which has resulted in production falling to less than one-tenth the number at its peak (**Fig. 1**). The number of small/mid-scale mainframes (less than 250 million yen per unit) has been decreasing especially rapidly. At one point, it seemed that at this rate mainframes would be replaced by open-source computers, but production began picking up in the last one or two years. The number of large-scale mainframes (more than 250 million yen per unit) is now on the increase. This indicates that mainframes will continue to be used for the time being.

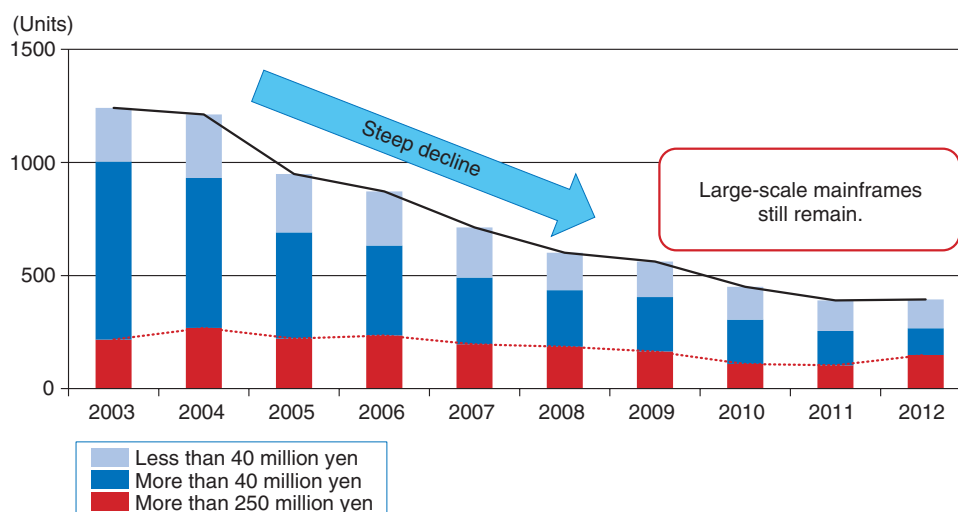
### 2. Issues with legacy systems

NTT DATA uses a large number of the remaining mainframe systems in Japan. Therefore, we are facing a variety of issues with mainframes. The most prominent issue is that system operating costs are

high. This is partially due to the expensive hardware that is required and also due to the large and complicated applications, which after repeated modifications over a long period of time, require a lot of investigation and testing even for small changes. In terms of systems running on mainframes in Japan, batch processing (to output many different formats) is particularly complicated and large in scale, and in order to ensure precise operations, detailed applications need to be created. In addition, one effect of the so-called *year 2007 problem*, in which many baby boomers reached the mandatory retirement age of 60, is that many mainframe experts have retired, and systems are becoming *black boxes*, meaning that we have less knowledge of their inner structures, which makes these issues even more serious.

### 3. Common pitfalls

An open source-based development method called *rehosting* has received attention as a way to reduce operating costs for systems that use mainframes. Rehosting converts applications that have been used on mainframes and also converts job control language (JCL)/job net (which describes the processing sequence) so that they can be operated and reused on



Source: Graph created based on data calculated by JEITA (Japan Electronics and Information Technology Industries Association)

Fig. 1. Changes in production by scale of mainframe computers.

open systems. By directly using currently used assets, rehosting is aimed at achieving rapid migration to open-source systems at low cost and without a change in quality. At first, only systems with a few hundred programs were convertible. Then, as the target systems increased in size, rehosting was successfully applied to enable the use of open-source systems for large-scale systems with more than ten thousand programs.

We have often observed that rehosting requires higher costs and longer periods than initially expected. An investigation shows that the major cause of cost overruns is overconfidence in converting applications. In a conversion project at NTT DATA, a misunderstanding that a new system can be created by merely converting applications meant that we did not pay enough attention to the proper redesign for integration (Fig. 2). Particularly in batch processing, system operations and the application infrastructure need to be redesigned. Several tens of thousands of JCLs and job nets are required just for batch processing. To ensure normal operation of the entire system, we need to take into consideration the processing order, boot conditions, processing and backup time, and method to recover from a failure for each JCL and job net. This requires detailed work that is truly like threading a needle. If we try to run these on a new platform with different non-functional requirements, there may be no problem in program unit testing, but a significant number of failures can occur in integra-

tion testing. Consequently, we have to continue to play an endless game of whack-a-mole.

Another major cause for the higher costs and longer periods than expected is a less than accurate estimation due to the difficulty in understanding the current status. Systems that have been constructed over a 20–30 year period contain various irregular and temporary modifications. A partial investigation of systems cannot ensure highly accurate estimation because some irregularities and risks may be overlooked. In the worst case, we have to check applications one by one, and the cost and time for doing this outweigh the advantage of rehosting.

#### 4. Automation solution applying N-shaped development

The solution is not to reuse converted applications but to create design specifications using existing assets including reliable source codes and to reuse these design specifications. In the first steps, we obtain correct specifications in order to accurately understand the current specifications of the system. In subsequent steps, we fully automate the development process of new systems to accelerate development and achieve optimum system performance. This process is called the *N-shaped process* because a process for creating design specifications for reuse is added to a preliminary phase prior to the normal V-shaped development process (Fig. 3).

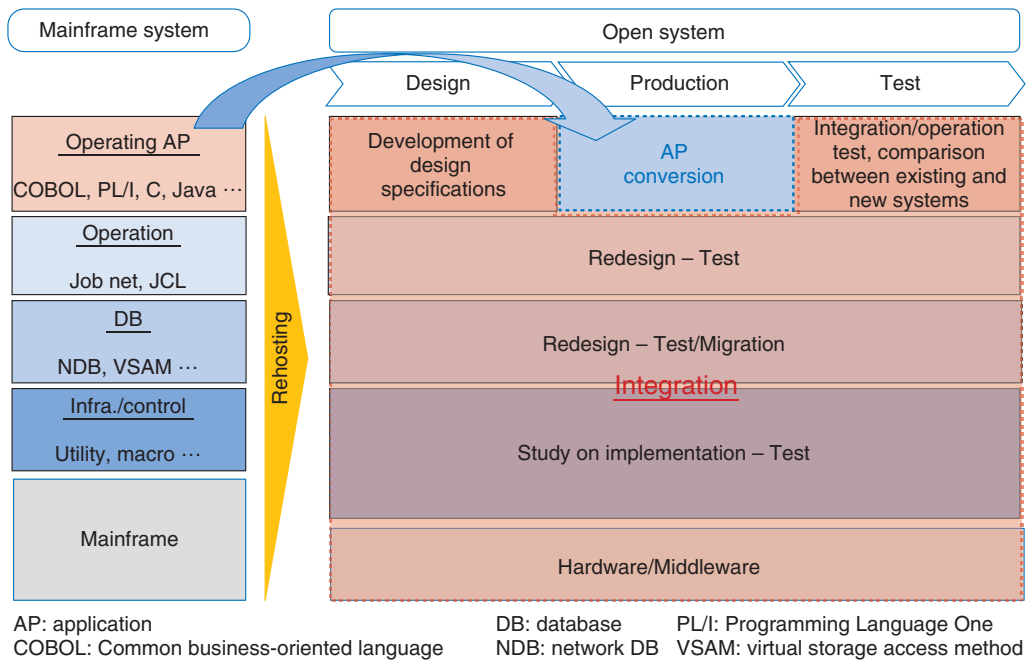


Fig. 2. Integration is easily overlooked during rehosting.

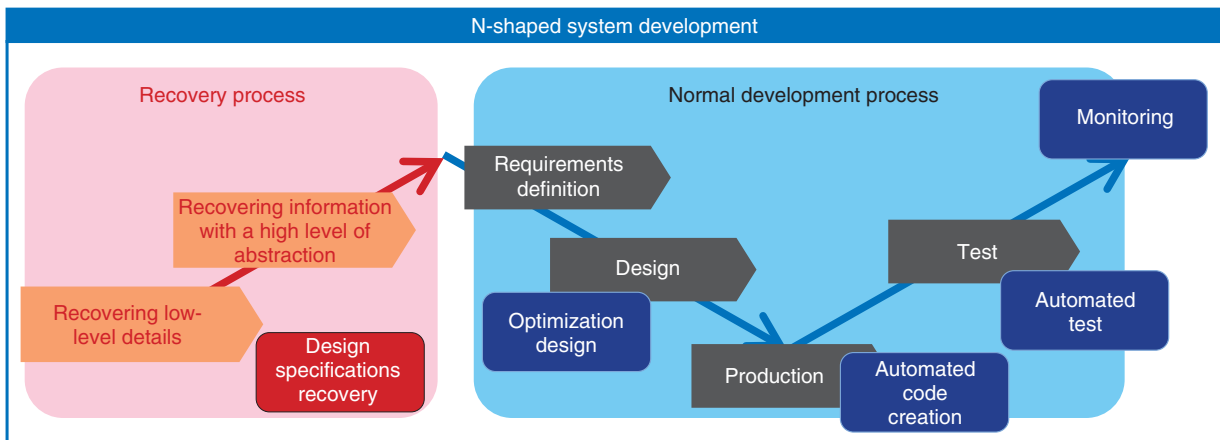


Fig. 3. N-shaped system development for successful rehosting.

The first stroke of the letter N shows the first phase of N-shaped development. This is the recovery process, in which we visualize and understand the system and its current operations. We take full advantage of reverse engineering technology here because visualization of business logic and application structures can be achieved automatically. Next, to enable humans to understand the relation between the operations and the system, we give operational semantics

to the visualized structure and data. For example, for online processing, we give consistent traceability and semantics starting from the screens used by system users to applications and databases (DBs) that do the actual processing. For batch processing, semantics is given on a batch basis for easy understanding by humans; furthermore, the semantics for batch processing is arranged in multiple layers, and in the top-level layers, names and descriptions are given using

working language. This process for understanding operations and systems enables us to overview the whole system and understand its details.

Understanding data is as important as understanding the processing. In order to correctly name and define every piece of data used in the system, we have to clarify the derived logic of data after classifying the relation between data and understanding the static structure. Reverse engineering technology can also be used here. Automatically creating a CRUD (create/read/update/delete) matrix\* enables us to identify data and derived logic. The final goal of this phase is to reveal operational requirements. Once these requirements are revealed, we can achieve the ideal design in the next system design process, independent of the system's current structure and processing method.

The remaining V portion of the letter N shows the second phase in the N-shaped development: the development process. In this phase, the deliverables obtained in the aforementioned process of visualizing and understanding operations serve as inputs to the upstream design process of the next system. A standardized input format leads to smooth automation of system development. In addition to automated source code generation using TERASOLUNA ViSC and IDE and automated testing using TERASOLUNA RACTES, a variety of tool groups can be used for ensuring high-speed development [1]. For the implementation of new requirements and optimization, TERASOLUNA DS can be used to automatically check the consistency in the design phase. With the use of TERASOLUNA Simulator [2], we can even check the validity of the specifications with a high degree of accuracy in the early phases. Taking full advantage of automation tools by following an N-shaped development process that reuses design specifications is the fastest way to renew systems and secure development.

### 5. Modernization through optimization

In contrast to rehosting, N-shaped development, in which the system is redesigned, can solve the issues that the current system faces. One of the issues that large-scale mainframe systems face is limited online service time due to a large amount of batch processing, which makes it difficult to understand the operating status in real time. This is because a shortage of computer power at the time of development means that nightly batch processing is implemented as much as possible, as opposed to online processing, which

places greater importance on response time. To maximize online and real-time processing operations, the processing needs to be designed after classifying the operational requirements for creating data. An optimization considering the aspect of redesigning the processing in N-shaped development is also required.

Data management is also an important issue. If a network database (NDB) is used in the mainframe system, we need to change to an RDB (relational database) in open systems. At the time of that change, if a table structure is constructed with the NDB structure left mostly as-is, there may be frequent table join operations and data accesses, causing a performance issue due to inefficient DB access. With N-shaped development, optimized table design is ensured in the operational requirements, preventing the data structure from causing problems.

Even in highly advanced systems, if the design specifications are not maintained, and there is any difference between source codes and design specifications, the system can become a legacy system again (Fig. 4). Incomplete design specifications may lead to black-box systems and lack of experts, resulting in more complicated systems and higher operating costs. Even a small change would require a large workload and a lot of time. For system renewals, it is important to update the design specifications according to the rules. It is difficult, however, when there is a limited amount of time and limited human resources involved. One solution is to have computers automatically check the consistency between design specifications and check if the changes made are compliant with the rules. If design specifications and programs used for system renewal can be checked and changed anytime, we can detect failures in the upstream process, significantly contributing to improved QCD (quality, cost, delivery). A continued checking process cannot be achieved by humans, whose work may involve oversights and omissions. Automated implementation by computers that is independent from humans is required.

### 6. Automated development to break a negative spiral

Many of the mainframes are also called legacy systems, and there has been a delay in the transfer of human resources and the introduction of new production technologies. The difficulty of system renewal is

\* CRUD matrix

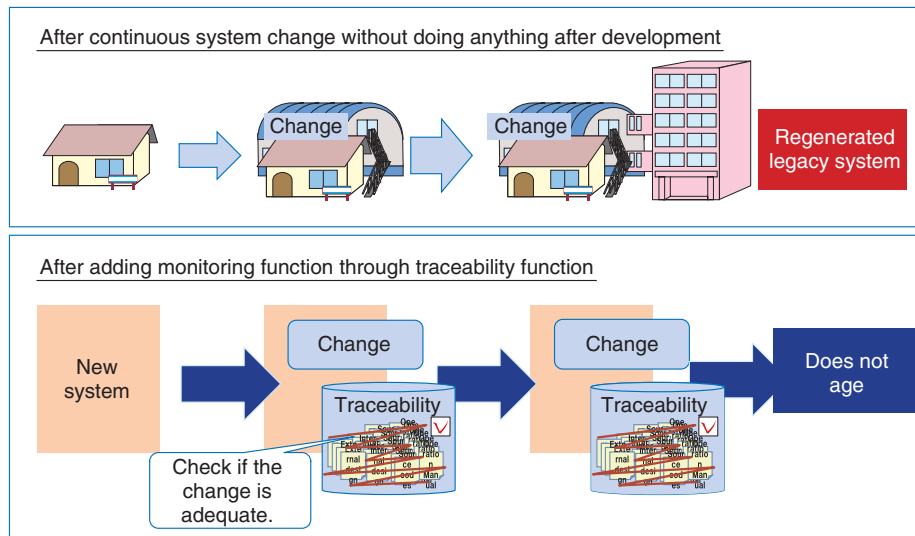


Fig. 4. Preventing rehoused system from becoming legacy system again.

increasing more and more due to its large scale and the increasing complexity, lack of experts, and lack of investment. If any additional issues with scale, complexity, or lack of experts are found, system renewal may be in vain. Case studies have shown that if system maintenance is continued for 10 years, the scale becomes one-and-a-half times larger than the initial scale, and the amount of time required to introduce new products and services can be two to four times larger than the initial amount. In order to break this negative spiral and continuously use information systems, it is becoming more important to switch to

automated development and an automated checking process that is independent from human work.

### References

- [1] T. Azuma, "Automation Technology for Software Development at NTT DATA," NTT Technical Review, Vol. 12, No. 12, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201412fa2.html>
- [2] T. Kaneko, "From Labor Intensive to Knowledge Intensive—Realizing True Shift to Upper Phases of Development with TERASOLUNA Simulator," NTT Technical Review, Vol. 12, No. 12, 2014. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201412fa3.html>



**Hideaki Tanino**

Manager, Senior Consultant, Research and Development Headquarters, NTT DATA Corporation.

He graduated from Tokuyama Technical College in 1991. He joined NTT DATA in 1991 and evaluated DBMS, software development tools, and Internet services. From 2000 to 2010, he developed systems for the Tokyo Stock Exchange as well as credit card and telecom systems. He is currently developing legacy modernization processes and tools. He is a PMI (Project Management Institute)-certified PMP (Project Management Professional).