

Software Product Lines in the OMRON Group

HARADA Shintaro, NIWA Toru, AKAMATSU Yasuyuki and TAGUCHI Masahisa

Software is becoming larger and more complex mainly due to higher functionality of the product, software implementation of functions what used to be realized by hardware. As a result, increasing development and quality costs, frequent schedule overruns of software are becoming a business issue.

To resolve this issue, we focused on Software Product Lines technology, which is one of commonalization, reuse technology. We developed technology for effective deployment, and trained expert human resources to lead the development. As a result, we deployed the technology to eight product groups in five years with each achieving the deployment purpose.

1. Introduction

The size and complexity of software for embedded products are increasing because of various factors. Such factors include enhanced product functionality, product systematization, and softwarization of conventional hardware-implemented functions.

The same applies to the OMRON Group as well. The volume of software development has been increasing exponentially for our flagship products, such as control equipment, FA systems, healthcare and medical equipment, automotive electronic components, social infrastructure systems, and environment-related equipment. This has given rise to operational problems, such as soaring development costs, longer development periods, and increased quality costs.

In response to these problems, we have addressed software process improvement (SPI) activities to improve quality, cost, and delivery (QCD) performance. For example, improvement activities based on the Process Reference Model (CMMI[®]) have been promoted at individual business companies under the Group to achieve continuous improvement in QCD performance.

Meanwhile, as shown in Fig. 1, each of OMRON's flagship product families undergoes major product renewal, including hardware, at approximately five-to-ten-year intervals. This major product renewal is followed by the variant development of the product families to expand the product lineup. Such is the development cycle at OMRON. Therefore, the QCD performance during the whole product family life cycle is significantly influenced by the software architecture built at the time of new product development and the framework for proceeding with variant development.

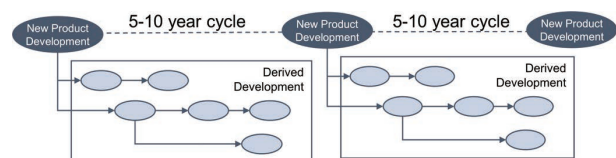


Fig. 1 New product development and variant development

Hence, in addition to conventional process improvement activities, we turned our attention to Software Product Line (SPL) engineering. This engineering tool strengthens the software architecture at the time of new product development and the framework for proceeding with variant development with the whole life cycle taken into consideration.

2. Software Product Lines (SPLs)

Software Product Line (SPL) engineering is one of the many engineering tools for standardization and reuse. It is a comprehensive approach to arrangements and techniques that views products from a product-line perspective and improves the QCD of and customer satisfaction with each product as a constituent of the product line.

It is said that when product variants are derived from SPL-based development and the number of product variants exceeds a certain numerical value (breakeven point), their cost effectiveness will be equal to or greater than that achievable with the conventional product-by-product development approach, as shown in Fig. 2¹⁾. In addition, the market responsiveness of products and the market share thereof will be improved if product variations are efficiently derived.

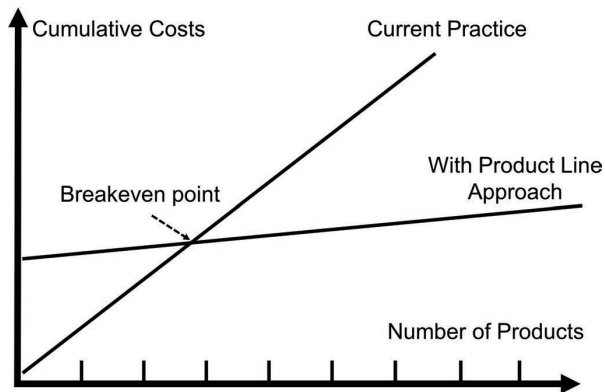


Fig. 2 SPL-based cost effectiveness

SPL engineering consists of the following three activities:¹⁾

- (i) Core asset development
This is intended to develop the core assets for wider use in product development as a preparation for rapid continuous development of product variants.
- (ii) Product development
This is intended to develop product variants efficiently and effectively, making full use of core assets, while responding to customer requests.
- (iii) Management
This is intended to support core asset development and product development from the organizational and technical perspectives to ensure normal startup and operation of the whole process.

SPL engineering exhibits effectiveness when these three activities are implemented systematically and continuously. Adoption of an SPL approach means that the conventional approach to development, which involves modifications to the existing software for each product development project, will be replaced with, as shown in Fig. 3, an approach that develops and utilizes core assets for derivation of product variants.

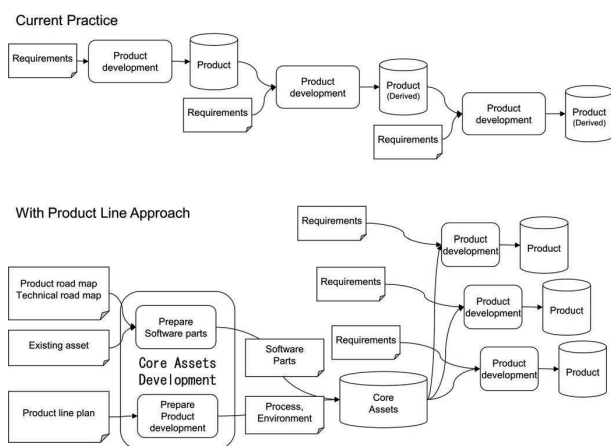


Fig. 3 SPL-based development

There are many reports on examples, mainly European and American, of remarkable effects resulting from the adoption of SPL approaches. To give an example of successful efficiency improvement, Cummins Corporation (USA) reduced the product cycle time for a diesel engine control system from 250 person-months to several person-months. There is also a case report from Japan, according to which Toshiba maintains and utilizes core assets for power plant monitoring and control systems²⁾.

We considered that by adopting an SPL approach for software development for new products, we would be able to achieve a revolutionary improvement in the overall QCD performance, including the subsequent variant development. The problem is, however, that unlike examples of SPL activities for specific product families at other companies, there are few examples of company-wide SPL projects covering more than one business domain. Hence, we had to devise a whole new SPL approach that best serves the needs of the OMRON Group.

3. Challenges

As already explained, major product renewal takes place once every several years, and individual development departments do not have frequent experience with new product development. Therefore, it was feared that even if individual development departments made SPL development efforts for their new product development, meaningful knowledge would not sufficiently accumulate to develop effective and efficient activities for all OMRON Group companies. Moreover, factors, such as reassignment of personnel, increased the uncertainty as to whether or not developers with SPL experience would be assigned to subsequent new product development. Accordingly, as viewed from the perspective of the handover of technical knowledge and skills, it was obvious that some measures had to be implemented.

Thus, considering that we would be able to bring about improvements efficiently and effectively with SPL development skills accumulated and SPL-versed human assets developed based on the knowledge collected by the Headquarters Functional Department from all employees with new product development experience, we addressed the following two tasks:

- (i) Accumulation of knowledge earned through SPL experience by all OMRON Group companies
- (ii) SPL expert human assets training at Headquarters Departments

4. Measures

Based on SPL-related technologies and processes generally practiced outside our group, we added our own knowledge to the knowledge accumulated from all OMRON Group

companies and edited and compiled the combination of the former and the latter into SPL Technical Materials.

Aiming to build human assets as quick as possible, we also established an SPL expert training framework for expert human assets training. SPL experts are expected to lead development department members and solve problems associated with new product development, using the SPL Technical Materials, thereby improving the QCD over the whole product family life cycle. This is intended to lead to the establishment of a condition in which there will always be continuous efforts to improve the competence of SPL experts and to enhance the SPL Technical Materials both quantitatively and qualitatively. The next and subsequent chapters describe specific examples of the above-mentioned activities.

5. SPL Technical Materials

As for SPL-related technologies and processes, Carnegie Mellon University’s Software Engineering Institute (CMU/SEI) defined a framework for software product line practice consisting of 29 practice areas in 2002³⁾. Moreover, in 2013, a reference model was defined in ISO/IEC 26550: “Software and systems engineering—Reference model for product line engineering and management”⁴⁾.

We reorganized technologies and processes based on those defined by CMU/SEI in the 29 practice areas and the reference model presented in ISO/IEC 26550. Then, while taking into consideration the reality of development practices, viewpoints regarded as of particular importance were added to allow effective implementation of SPLs across the OMRON Group. Moreover, knowledge earned through SPL experience was also incorporated as appropriate.

Using these SPL Technical Materials, we solved problems associated with SPL expert training (to be explained later) and SPL implementation and successfully implemented SPLs in effective ways.

The following sections outline the practice areas on which we concentrated our efforts in this project. These are areas of technologies and processes insufficiently covered by the conventional approach to development and are parts that we particularly elaborated over and over again to ensure that this project best serves the needs of OMRON.

5.1 Component development (Variability Implementation Guide)

What matters for successful SPL implementation is to identify differences between the original product and its variants as variabilities beforehand and to perform design and implementation with allowances made for product variations

or changes due to future upgrades. In other words, it must be clearly specified when and how to perform implementation for individual variabilities (binding time and binding method).

This selection of the implementation method will significantly affect the maintainability and scalability of the product. Therefore, this selection is of extreme importance. In fact, however, implementation methods could take a wide variety of forms. In addition, their selection was left to the discretion of designers, and there was no established set of standardized selection criteria.

Typical items that may significantly affect the variability implementation methods for many of OMRON’s embedded products include the required memory size (which affects the product cost), the number of binary types (which affects the production efficiency), and code readability (which affects the efficiency of variant development). Based on examples of past development projects, the relationships between the status of these items and the implementation methods suitable therefor were defined and incorporated into a Variability Implementation Guide (Table 1).

Table 1 Variability Implementation Guide (excerpt)

Binding timing: during development	
Implementer	Developer
Main methods of implementation	Compiler switches Make File or the like for selection of files to be used
Required memory size	Small
Number of binary types	Large
Code readability	Low
Binding timing: during manufacturing and installation	
Implementer	Producer
Main methods of implementation	Writing of settings Change of hardware configuration
Required memory size	Large
Number of binary types	Small
Code readability	—
Binding timing: during use	
Implementer	User
Main methods of implementation	Selection by installer User-defined settings
Required memory size	Large
Number of binary types	Small
Code readability	—

We summarized for our own use these judgment criteria into a comparative table between C/C++ and MBD, which are frequently used to develop software for embedded products. Now, this Implementation Guide serves as a set of selection criteria, thereby enabling individual designers to select appropriate variability implementation methods.

5.2 Architecture evaluation (ATAM)

The quality of the initial software architecture critically affects whether or not software built for new product development can remain reusable in the subsequent long-term variant development. In a development process with the focus placed on individual product development projects, however, it is quite usual that the mid-to-long term ease of modification (maintainability and portability) is not sufficiently examined.

Therefore, we comparatively examined several techniques as to whether or not they could systematically evaluate software architectures to be used over the mid-to-long term at the initial stage of software development. In conclusion, we decided to introduce the Architecture Tradeoff Analysis Method (ATAM), which is an architecture evaluation technique developed by CMU/SEI.

As shown in Fig. 4, the ATAM is a technique used to perform architectural analysis to identify factors detrimental to the achievement of business goals, and summarizes them into a list of risks based on a specific scenario (quality requirements) related to the quality attributes (processing speed, availability, maintainability, security, portability, etc.) for achieving the business goals⁵⁾. With the inclusion of business goals viewed from the perspective of the whole product life cycle, an architecture can be evaluated for reusability in the subsequent long-term variant development.

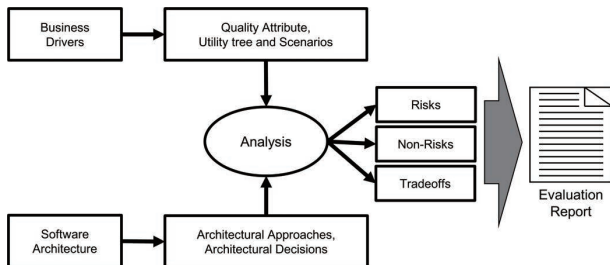


Fig. 4 Overall image of the ATAM

At the same time, however, the ATAM is a technique that may require an evaluation period of approximately one month, depending on the size and complexity of the relevant software architecture or the scale and scope of the stakeholders. Then, without compromising an evaluation that starts from quality attributes, which is the point of this technique, we made the whole evaluation process lighter and systematized it as the OMRON version of ATAM. For example, the conventional technique required the involvement of a wide range of stakeholders, including sales and operational personnel, to make an architecture better understood; in the new technique, an evaluation task needs only a small number of personnel, including a development leader, a product leader, and a software

design leader to suit the characteristics of product development projects at OMRON. By narrowing the objectives down to as few as possible, the number of activity steps has also been reduced from the conventional 20 activity steps in four phases to nine activity steps in three phases.

Similarly, OMRON’s version of the Software Architecture Analysis Method (SAAM) was also established as a systematic technique that gives priority to ease of change over other quality attributes.

Using these techniques, we performed a verification of SPL-based development and confirmed that the architecture designed at the time of new product development was fit for future reuse. Additionally, when planning the subsequent variant development, we used these techniques to determine the necessity of changes to the architecture.

5.3 Configuration management

In order that core assets built for new product development remain usable in the subsequent long-term variant development, a framework is necessary that enables systematic maintenance of core assets. Without this framework, haphazard changes would be added to software every time variant development occurred. As a result, core assets would end up degraded, whereby SPLs would lose their originally intended effects.

As a countermeasure, it is advisable to establish a Change/Configuration Control Board (CCB) responsible for controlling and managing changes to the core assets. To facilitate the establishment and operation of this CCB, we specified the typical processes involved and the organizational framework required, along with judgment criteria for changes, and implemented them in forms suitable for individual development departments. Fig. 5 shows the judgment criteria specified for use in the determination of the acceptability of changes, together with the development processes subsequent to individual judgments. Currently, there are several departments operating a CCB to maintain their core assets.

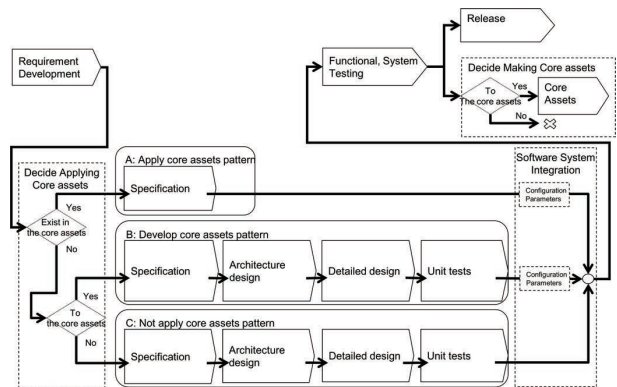


Fig. 5 Judgments on changes to core assets

5.4 Practice pattern

The 29 SPL practice areas are divided from the technical perspective and hence difficult to grasp project tasks along the time axis. As a result, engineers with limited SPL experience were often unsure which specific steps should be taken towards implementation, thus causing activities to fall behind schedule.

To enable anyone to put SPLs into effective practice, it was desired to make project tasks understood in chronological order to help proceed with activities. Then, headquarters staff experienced with SPL implementation into product families teamed up and worked together and systematized the flow of utilization of the 29 practice areas typical of OMRON (OMRON SPL Patterns). For example, the team sorted out the relationship between, and the flow of, the typical practice areas in each of the four phases (merchandise planning, core asset development, product development, and review) shown in Fig. 6 and summarized the points of activities at the respective phases. Fig. 7 shows as an example the flow of launching an SPL.

As a result, our SPL implementation procedures have been made more systematic to allow even members with less SPL implementation experience to provide appropriate support. Moreover, with the addition of knowledge from other projects, highly experienced members now can implement SPLs more efficiently and effectively than before.

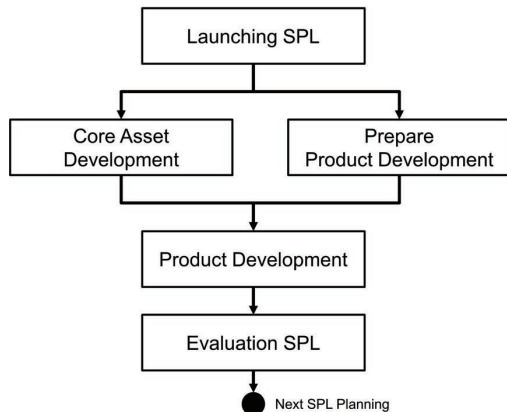


Fig. 6 OMRON SPL Pattern (overall image)

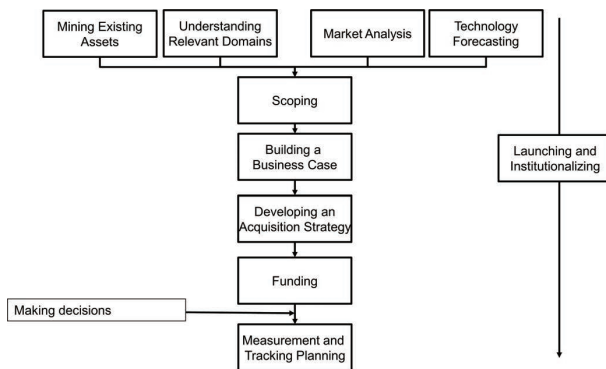


Fig. 7 OMRON SPL Pattern (launching of an SPL)

6. SPL expert training framework

We developed SPL experts as human assets capable of efficiently applying SPLs to development departments while accumulating technical expertise and established a pool of these people at the Headquarters Functional Department. To serve as an SPL expert, a person is required to have not only SPL skills but also liaison skills with development departments. They are also expected to have capabilities as an in-house consultant, such as presenting proposals for SPL techniques, building consensus, and deploying skills in forms suitable for individual development projects. Then, as shown in Fig. 8, a set of requirements to be met to qualify as an SPL expert has been defined as an SPL expert training framework from the three perspectives of core skills (consulting skills), technical skills (SPL expertise), and experience.

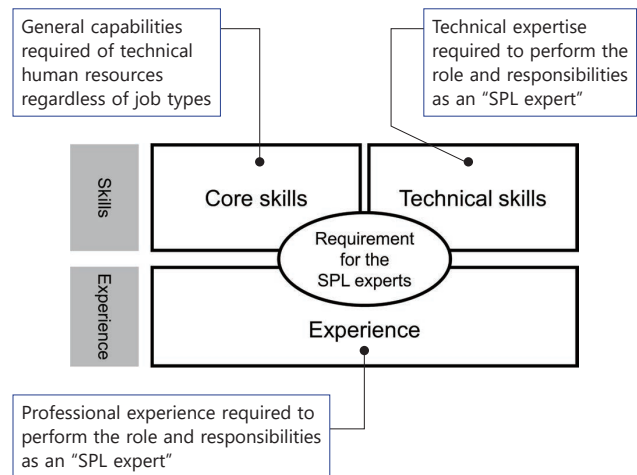


Fig. 8 SPL expert training framework

In addition, the levels of ability as SPL experts have been defined as *beginner*, *intermediate*, and *advanced*, and detailed training programs have also been established.

Based on this framework, individual trainees are allowed to set their own targets and turn the cycle of plan, act, evaluate, and review. At the same time, they are organizationally assigned to OJT matching their training targets and abilities. In this way, we have successfully realized efficient human asset development.

Additionally, we developed and continuously implemented training programs for enhancing individual skill elements. Using this training framework, we succeeded, as planned, in technical expertise enhancement at the individual trainee level as well as at the organizational level. As a result, seven individuals have qualified as SPL Advanced experts as shown in Fig. 9.

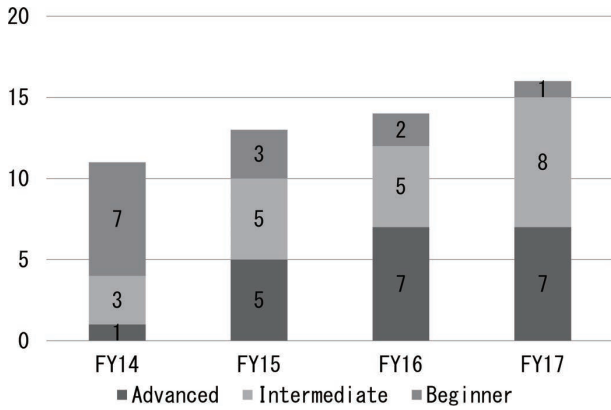


Fig. 9 Number of certified SPL experts

7. Achievements

Launched full-scale in 2013, our training framework produced 16 SPL experts (including seven advanced SPL experts) by 2017. In addition, the knowledge accumulated through their actual SPL implementation activities has been compiled into an approximate total of 400 pages of SPL Technical Materials.

Fig. 10 shows the average skill scores of the SPL experts belonging to the Headquarters Functional Department. Their core skills, technical skills, and total ability are rated by scores from 0 to 3 and shown with changes over time. As shown in this figure, we successfully and continuously improved our expertise at the organizational level through the operation of the SPL expert training framework. In addition, despite the fact that not a few SPL experts were reassigned to other departments and many new members joined from other departments each year during the period, the department has continuously maintained and enhanced its collective technical expertise.

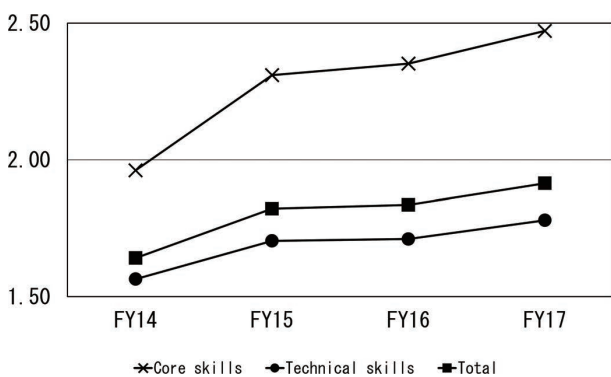


Fig. 10 Changes over time in the average skill scores of the Headquarters Functional Department

Accordingly, we implemented SPLs into eight product families with successful results. For example, an SPL expert team consisting of headquarters staff took part in a product development project from the planning stage of a product family and worked together with the development department

members to determine the feasibility of the SPL based on ROI, performed domain analyses, and established the methods of core asset building and maintenance. This led to the following achievements:

- A 30 percent reduction in the variant development cost from the conventional level
- Significant improvement in production efficiency through the consolidation of six software variants into one
- Acquisition of business negotiations through early-stage customer demonstrations based on pre-arranged core assets

We believe that this project was effective for the following reasons: it contributed to the achievements made through the application of SPLs to the development of eight new product families in five years, and the practical experience thus obtained enabled quick development of SPL expert human assets and allowed the establishment of the SPL Technical Materials in a short period of time.

8. Conclusion

For OMRON’s embedded products, new product development typically occurs at five- to ten-year intervals and is followed by variant development. Adopted as a technical solution for significantly improving the QCD performance of these developments, SPL techniques are applied in forms suitable for OMRON and with some successful results.

On the other hand, two new issues came to our attention. One is that the presence or absence of a software architect with high expertise critically affects the success or failure of any new and large-scale development project and hence, of course, the success or failure of SPL implementation. The other is that more sophisticated variant development practices, including suppliers, must be defined and rolled out because suppliers often play the central role in variant development. New improvement activities are already underway to address these two challenges.

The role of software in products is expected to become larger with advancements in technologies, such as IoT, AI, and robotics. Along with this trend, QCD performance improvement in software development will pose more important business challenges than before. We will continue with innovation in software development technology to ensure OMRON’s further business growth.

References

- 1) Clements, P.; Northrop, L. *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2002, 563p.
- 2) Software Product Line Conferences. “Product Line Hall Of Fame”. <http://splc.net/hall-of-fame>, (accessed 2018-12-14).

- 3) Clements, P.; Northrop, L. et al. "A Framework for Software Product Line Practice, Version 5.0". http://www.sei.cmu.edu/productlines/frame_report/index.html, (accessed 2018-12-14).
- 4) ISO/IEC 26550: 2013. Software and systems engineering —Reference model for product line engineering and management.
- 5) Clements, P.; Kazman, R.; Klein, M. Evaluating Software Architectures: Methods and Case Studies. Addison-Wesley, 2002, 323p.

About the Authors

HARADA Shintaro

SPI & SPL Integrated Tasks Promotion Dept.
Design & Engineering Process Technology Center
Global Manufacturing Innovation H.Q.
Speciality: Software Engineering

NIWA Toru

SPI & SPL Integrated Tasks Promotion Dept.
Design & Engineering Process Technology Center
Global Manufacturing Innovation H.Q.
Speciality: Software Engineering
Affiliated Academic Society: IEICE

AKAMATSU Yasushi

Technology Dept. 3
Technology Development Division H.Q.
Industrial Automation Company
Speciality: Software Engineering

TAGUCHI Masahisa

SPI & SPL Integrated Tasks Promotion Dept.
Design & Engineering Process Technology Center
Global Manufacturing Innovation H.Q.
Speciality: Software Engineering

CMMI is a registered trademark or trademark of CMMI Institute USA in the United States and other countries.
The names of products in the text may be trademarks of each company.