2

Toward the Improvement of Quality and Reliability in Information Systems Construction – A Study of “Business Rules” and Requirements Engineering in the Upstream Process–

TOSHIAKI KUROKAWA *(Affiliated Fellow)*

1 Introduction

Information systems assume a critical role as the basis of effective activities at various organizations and as a social infrastructure. Establishing and proliferating technologies to build safe and reliable information systems promptly and inexpensively is an indispensable element of the safe and competitive society. Information systems are serving not only as the source of industrial strength in general but also as the foundation for overall research and development capabilities.

In other words, degradation in the ability to construct information systems may cause inconveniences in civic life as well as poor international competitiveness. Japan should therefore make relentless efforts to maintain and improve its level of system construction technology.

It is widely granted that the key to improving quality and reducing cost in system construction lies in the upstream phase of system construction, or more specifically, in the concept stage in which the requirements for the system are identified. That is to incorporate a mechanism to select key functions for the system and to enhance the system’s quality in the upstream process. In Japan, however, rather little emphasis seems to have been placed on efforts for such upstream activities. No “technology” took root and only few people cares for that situation in this nation.

The 2003 White Paper on Information and Communications in Japan published by the Ministry of Public Management, Home Affairs, Posts and Telecommunications indicated that Japanese organizations often failed to discuss even the fundamental questions that should be answered prior to system construction, such as what are the possible effects of constructing the information system[1]. That is to say, organizations are not fully prepared before actually starting to build an information system, with respect to which aspects to be stressed in building the system and what potential returns to be expected.

This weakness of underestimating or neglecting the upstream part of system construction can be attributed to several and complex reasons, including institutional, traditional, technological, and social ones. For example, in terms of institution, software systems are often compared with architectural structures because of their common nature of serving as a social infrastructure as a whole while being privately owned. It has been pointed out that while there are mechanisms, such as construction authorization, instituted for architectural construction to conduct reviews during the design stage, there exists no equivalent scheme in the world of information systems construction. Furthermore, unlike architectural construction, design and installation are performed by the same company in systems construction. There has been a debate about whether these problems in the systems construction arena are something traditional or social. Some people declare that Japanese tradition does not ask any detailed specification before engagement. This article addresses these issues primarily from the viewpoints of human resources development and science and technology. More specifically, it discusses the significance of the upstream process, reviews technologies available for improving quality and reliability in systems construction, and proposes methods to foster human resources who have expertise in such technologies.

2 Definition of the upstream process in the system life cycle

There is no clear, universally agreed on definition of the upstream process. In an effort to define the upstream process, this article presents the system life cycle standard, which stipulates the overall construction and operation of systems. An international standard on system life cycle processes was established in 2002 as ISO/IEC 15288:2002, based on which the Japanese Standards Association is now working on the drafting of the JIS X0170 Japanese standard. According to ISO/IEC 15288, the system life cycle from creation to termination/disposal consists of the following six stages.

(a) Concept
(b) Development
(c) Production
(d) Utilization
(e) Support
(f) Retirement

These six stages can be divided into three phases as shown in Figure 1; upstream, midstream, and downstream.

ISO/IEC 15288 further defines 11 detailed technical processes (Table 1).

As Figure 1 indicates, the concept stage aims at laying the groundwork for systems construction. During this stage, the systems environment and the stakeholders must be identified. The concept stage includes the requirements definition process in which requirements of

Figure 1:Definition and elements of the upstream process

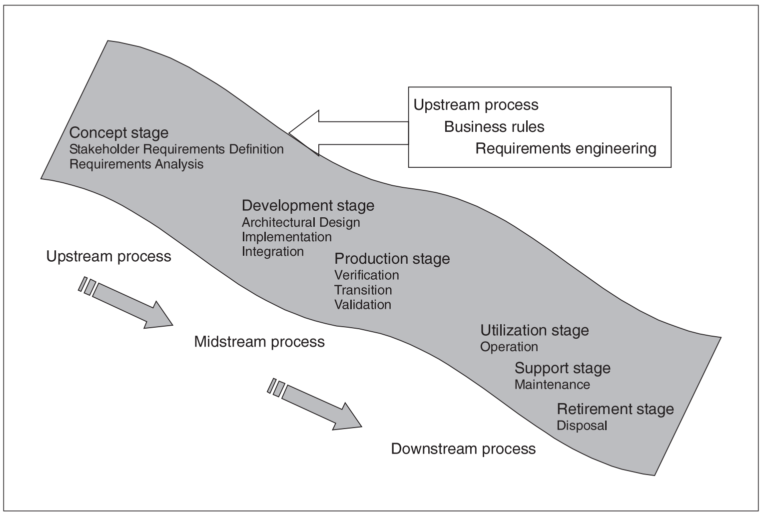

Table 1:Eleven technical processes

(a) Concept stage	(1) Stakeholder Requirements Definition process (2) Requirements Analysis process
(b) Development stage	(3) Architectural Design process (4) Implementation process (5) Integration process
(c) Production stage	(6) Verification process (7) Transition process (8) Validation process
(d) Utilization stage	(9) Operation process
(e) Support stage	(10) Maintenance process
(f) Retirement stage	(11) Disposal process

the stakeholders are elicited and defined, and the requirements analysis process in which relations/conflicts over these requirements are analyzed.

3 Problems in systems construction – The significance of the upstream process

There have been a few reports on specific problems caused by imperfections in the upstream process. A famous example among reported is a fault in the London ambulance dispatch system that occurred in October 1992 in the U.K. The project, which was ambitious for that time, intended to build a computer system, in place of traditional manual operations, that could automate emergency call taking and ambulance dispatch. In reality, however, the system failed to properly dispatch ambulances, causing chaos. The report about this case[19] shows that one of the major causes was that the system had been designed in the first place without asking or considering requests from ambulance crews who would use the terminals in the ambulances. Although the system required ambulance crews on board to report their vehicle's status and location to the dispatch center so as to be given their next destination, there had been no consideration during system development about whether the crews were capable of using such a system, an essential element of stakeholder requirements. Consequently, ambulance crews had difficulty in operating the terminals, disrupting overall ambulance dispatches.

With respect to the entire life cycle as shown in Figure 1, there is an interesting analysis[21] on how the cost to develop a system is allocated to individual processes. It shows that, in the U.S., the maintenance cost accounted for approximately 30% of the entire system development cost until the 1970s, while the figure increased to almost 80% by the 1990s.

It can be easily imagined that a tremendous amount of correction and modification work arises if a failure is found in a system after implementation, as was the case with the London ambulance dispatch system. However, one should also be aware that even an ordinary system cannot be operated properly without an enormous cost spent in the downstream process.

The fact that the downstream process cost accounts for a considerable part of the total expenses is, in general, well recognized even in the manufacturing industry. For instance, manufacturers are facing the need to incorporate recyclability into their upstream process such as design for minimizing overall cost, because now recycling after disposal has become an inevitable process for manufacturers.

The most effective approach to minimizing the maintenance cost is to make thorough efforts in the upstream process to ensure quality, in an attempt to eliminate any need for corrections in later processes. "Ensuring quality in the upstream process" is a well-know principle across industries including manufacturing. If a defect is ever discovered in the product after shipment, its financial impact can be so great that even the manufacturer's existence may be threatened.

Another fact to be noted is that the cost to maintain information systems is eventually borne by all the members of the society. Increased cost for the system is eventually passed on, for example, to bank customers in the case of a banking system in the forms of higher fees and lower interest rates.

However, little emphasis has been placed on the upstream process, which is also known as the "concept stage" in system life cycle processes, because it has been believed to be a preliminary step to the development stage in the midstream process and, therefore, reduced to be a mere process of collecting and organizing information before moving up to the development stage. Regrettably enough, there are still some cases where a system developed through considerable effort turns out to be unusable just because of the negligence of the upstream process.

A factor behind such poor practices is that only few organizations in Japan, whether a business enterprise or a national or local government office, have documentations that explicitly describe, in a format unified across the organization, individual job procedures and liaison with associated internal sections. This is a great disadvantage in systems construction, because such prerequisites are not readily available to initiating the identification of the people concerned and the collection and analysis of their requirements.

Some system developers skip the upstream process, despite their understanding of its necessity, based on the fear that the additional workload required for completing such documentation cannot be justified or can be too time consuming,. In addition, because of the lack of the formal documentation, system developers continue to receive additional modification requests from customers or users even after the specifications have been defined; a convincing reason for system developers to feel that any effort in the upstream process would be in vain.

One thing that makes the upstream process issue more complex is that failure to meet the prerequisite, such as the inexistence of formalized documentation of operations, does not always bring problems in the system. In fact, there are some successful examples of system development in which the explicit upstream operations were avoided by letting the staff who would use the system participate in construction of the system. This approach has proven effective, in particular, in the development of systems for manufacturers' production sites, a sector in which Japan boasts competitive strength. From another point of view, however, you can say these organizations did not have to conduct additional work particularly intended for the upstream process, because the equivalent operations are continually performed on a daily basis.

The thorough implementation of upstream activities is indispensable for improving the quality in system construction while minimizing the overall cost, including maintenance, even though whether the upstream activities may be started anew or not is a decision depending on the situation.

4 Technical elements of the upstream process

The upstream process should involve the establishment of "business rules" that, as described below, encompass the entire business operations in addition to the stakeholder requirements definition process and the requirements analysis process as technical processes,. Without business rules, requirements submitted by diverse stakeholders from different perspectives cannot be aligned consistently and will thus become dispersed.

The requirements definition and requirements analysis processes fall into the technological field known as requirements engineering. The significance of requirements engineering came to be acknowledged worldwide in the 1980s, as described later in details. Behind this is the perception that no matter how much productivity and quality are improved in the design and production, system developers eventually fail to satisfy user needs as long as an error or poor quality is inherited from the requirements processing. To make matters worse, developers are recently expected to assume that requirements are subject to change not only in the development and production stages but even in the utilization stage, thereby continuously facing the need to monitor and modify requirements as well as to verify the system against the ever-changing requirements. Requirements engineering provides techniques to process such "requirements" in almost all processes across the system life cycle.

In the 1990s, it became clear that collecting the existing user requirements alone was insufficient in building a completely new type of system or integrating existing systems into a new system. Also recognized was that, when constructing a large-scale system by combining multiple sub-systems developed in parallel, overall system efficiency could not be achieved if attention was paid only to individual sub-systems.

To cope with these issues, new initiatives have been launched with the view of defining "business rules" with which the structure of the system's operating environment is identified and the organization's policy and procedures to conduct operations are specified. These "business rules" are unlike protocols defined in schemes such as EDI for business-to-business commerce. Instead, they describe procedures through which individual jobs are conducted within the framework of a company or a group of companies.

5 Requirements engineering

Requirements engineering, as mentioned before, deals with technologies used for processing “requirements” for a system across almost all processes of the system life cycle. These technologies are intended for requirements acquisition, requirements analysis, requirements evolution, requirements management and so forth, as part of system/software engineering, an academic discipline that aims to increase productivity in the overall system.

The IEEE has been holding an international conference[9] and a symposium[10] regarding requirements engineering annually since 1993. In Europe, the REFSQ (International Workshop on Requirements Engineering Foundation for Software Quality) conference has been held annually in connection with the CAiSE conference on system engineering, while Australia has been hosting workshops and symposia since 1993 in this field. In addition, Europe has launched projects on requirements engineering through its schemes, such as with IST and ESPRIT, and the International Federation for Information Processing (IFIP) has established Working Group 2.9, which is dedicated to software requirements engineering. The significance of requirements engineering as a field of engineering is fully recognized, although requirements engineering has not become a term that is used as commonly as software engineering and system engineering and has yet to be unquestionably established as an academic discipline.

Professor Motoshi Saeki of the Tokyo Institute of Technology claims that today’s international conferences on requirements engineering have their origins in Japan. When Japan hosted the International Conference on Software Engineering for the first time in 1982, Yutaka Ohno, then the professor of Kyoto University, organized an off-the-record workshop in Kyoto, an attempt that inspired participants to hold regular international conferences on requirements engineering.

This occurred at a time when there was an growing expectation for potential effectiveness of logical specification methods in software development, while Japan was undertaking research into logical methods through its Fifth Generation Computer Systems project and other initiatives. The formal specification methods at the time were applicable only to a limited range of problems, and could not provide effective solutions to practical problems.

Some universities in the U.S., Europe and Australia offer courses dedicated to requirements engineering. In Europe, research and development projects in the area of requirements engineering are ongoing as part of industry-academia collaborative programs such as with ESPRIT and IST. Recognizing the essential role of requirements engineering in national projects, particularly in areas such as space development and military and defense, both the U.S. and Europe are now even conducting field tests for requirements engineering techniques. Requirements engineering is also studied as part of software engineering and system engineering.

The following sub-sections describe findings of an analysis of presentations at IEEE international conferences and symposia since 1993, as well as the current status of requirements engineering including available tools.

5.1 IEEE conference presentations and presenters

The total number of presentations at IEEE conferences between 1993 and 2002 was counted and broken down by presenter types; namely, universities, national research institutes, and private companies and consultants. The total number for these 10 years counted 376, which consisted of 220 universities, 41 national research institutes, and 94 companies and consultants (figures broken down do not include presenters whose affiliation was unknown). Recently, there has been an increase in the number of presentations from companies, indicating a trend that businesses as well as national research institutes are more actively embarking on development and field trials.

On the whole, nearly two-thirds of the total presentations were derived from universities, underscoring the academic sector’s dominant presence as the research body.

Only nine presentations were delivered by Japanese organizations in the 10 years. The presenters were the Tokyo Institute of Technology, Osaka University, Kyoto University, Hiroshima City University, and Ritsumeikan University from academia, and NTT, NEC, and Anritsu (note that NEC and Anritsu presentations were delivered by their U.S. subsidiaries) from industry.

5.2 Subjects of IEEE conference presentations

To identify trends in presentation contents, presentations were divided into 10 fields: models, specification methods, requirements acquisition/definition, requirements evaluation, requirements verification, requirements evolution (change), requirements reuse, engineers, techniques and tools, and others.

The mainstream presentation themes remained consistent: requirements acquisition and requirements definition. In other words, since 10 years ago the requirements engineering community has been tackling the same problem of how to elicit, compile, and define requirements, and no one has arrived at a satisfactory solution. This is an issue that is also associated with requirements evolution (change), the eternal challenge in systems construction.

Subjects that were addressed in parallel with requirements acquisition and definition were requirements evaluation and verification. Issues with them included system security analysis, risk prediction, and risk assessment. Some presentations addressed the question of what are the system requirements that are tolerant to human errors.

With respect to requirements acquisition, sociological techniques have been attracting attention since the early days and have been actually tested. Recently, requirements acquisition has become related to business administration issues such as knowledge management and knowledge sharing.

In requirements specification, models serve as tools as well as techniques. A variety of notation and logic schemes are used for specification. How to derive formal specifications from informal, natural language-based expressions is also discussed, which is a longtime issue in this area.

One of the major issues in software engineering is reuse. There has been progress in code reuse as well as in design reuse. Recognizing the persistent argument that reuse is most effective in requirements in the upstream process, reuse from this perspective recently became a topic discussed in these conferences on requirements engineering.

Some presentations focused on the issues of human resources development for requirements engineering and of the fundamental principles for requirements engineers.

Other noteworthy themes that were not selected for the above categorization include traceability, cost analysis, technology transfer, international/industrial standardization for requirements engineering, and requirements evaluation for commercial software. In particular, the traceability issue regarding how the requirements are reflected into design and incorporated into the system is likely to become an important factor, especially in relation to the future environments for systems development. One presentation on traceability proposed that even concerned parties be traced from the viewpoint of sociological organizational theory.

A total of nine presentations were delivered by Japanese organizations, of which two were on models, two on requirements acquisition, one on requirements analysis, one on requirements evolution, one on reuse, and two on techniques and tools. Of them, however, two presentations were based on joint research between U.S. subsidiaries of Japanese firms and universities in the U.S. and Europe.

5.3 Evaluation of the current status of requirements engineering

Requirements engineering is an academic discipline that has developed as a part of system engineering or software engineering. Some people still question, however, how much requirements engineering is matured as an establishedacademic discipline. And some argue how effective it has become in practical environments of the systems or software industry.

It is not an easy task in requirements engineering to yield results with good numbers. Since requirements engineering, among other fields in system or software engineering, is an exploratory and empirical field and where the human factors have so much significance. An honest evaluation from an external point of view would be that requirements engineering, aside from its principles, has yet to provide any techniques or tools that can instantly benefit field engineers.

The good news for the requirements engineering community is that background material for requirements acquisition is becoming machine processable because businesses and government organizations have deployed computer systems to process and accumulate their administrative data. In some cases, even corporate management policies and goals are presented in a machine processable way, as seen in the efforts toward "business rules" discussed in the next section.

Under technological circumstances where mechanization has become possible for the processes from modeling and design to implementation of a system, the next challenge for requirements engineering would be to allow machine-based management of requirements changes and provide traceability for a failure to be trace back to the requirements concerned.

Examples of requirements engineering tools are shown in Table 2.

6 "Business rules"

6.1 Background and brief history

From the viewpoint of system development engineers, the historical development of "business rules" can be outlined as follows (Table 3).

The concept of business rules was first introduced through a report titled "Defining Business Rules[20]," which was published in 1995 by the GUIDE Business Rules Project, an initiative that was organized in 1993 by GUIDE International Corporation, a U.S. association of IBM mainframe users founded in 1955 (an update to the report, containing models in Unified Modeling Language (UML) notation, was issued in July 2000). In 1997, the Business Rules Group[3] was formed and they introduced the Business Rule Motivation Model, which is described in 6-4.

The Object Management Group (OMG)[18], an international industry consortium that focuses on the standardization of object-oriented

Table 2:Examples of requirements engineering tools

Tool	Developer/Vendor	Brief description
REVEAL	Praxis Critical Systems[11]	A requirements engineering methodology. Enables the verification of requirements traceability in low-level systems by focusing on system integration and using Jackson's "World and the Machine" model. Uses Telelogic's DOORS as a tool.
Ask Pete Support Web[12]	NASA Glenn Research Center	A free tool for cost prediction and project planning. Can work with ARRT, listed below.
DDP/ARRT (Defect Detection Prevention[13] /Advanced Risk Reduction Tool[14])	JPL	DDP is a tool for risk prediction and prevention. It is available in the forms of RBP (Risk Balance Profile), a simplified version, and ARRT, a software-specific version of DDP. Available for free. The Java version is under development.
ISAT: Interactive Specification Acquisition Tools Project[15]	AT&T Lab. Research	A research project aiming at automated specification and validation, centering on reactive systems such as communications systems. Performances on several prototypes have been reported. The basic model is the state machine.
SCR (Software Cost Reduction) method[16]	U.S. Naval Research Laboratory	A method created by integrating principles and tools that have been developed since the 1970s at the Naval Research Laboratory to practice software engineering. The core is the Rational Design Process for software. It also uses techniques such as David Parnas's Four Variable Model (monitored, controlled, input, and output variables; NAT (assumption), REQ, IN, and OUT relations), the SCR requirements model (defining the system state), and the SCR table.
i-COST method[17]	Equity Research (Japan)	A method for cost evaluation and quotation for software systems. It can analyze operational cost, in addition to initial cost. For initial cost analysis, it uses the function point method for quantitative evaluation of system functions, and the COCOMO method, that is used in the U.S. and Europe.

technologies, embarked on a campaign in 2000 to promote the Model Driven Architecture (MDA) under the leadership of Chairman R. Soley. This is an approach that intends to automatically generate programs from formal models written mainly in UML, OMG's standard object-oriented modeling language. In the natural course of development, a debate took place over how to ensure the correctness of the initial system model, resulting in the formation of the Business Rules Working Group within the framework of OMG in 2002.

Table 3:Historical development of "business rules"

1960s	Programming is everything	Before engineering
1970s	Structured programming, structured design, structured analysis	Testing became possible
1980s	Information engineering and object orientation	Rediscovery of data
1990s	Business rules	How should a system work?

Source:David C. Hay, "Managing Business by the Rules," 1999; Table compiled by the author.

6.2 Definition and objectives of business rules

"Business rules" are intended to describe the business structure pertaining to the system to be constructed and to identify the expected functions of the system within the structure. The Business Rules Group[3], the Business Rules Community[2], and the OMG Business Rules Working Group are the leaders in this domain. The definition and objectives of business rules slightly vary from group to group. The variation probably reflects either chronological development in the form of business rules or subtle differences in interpretation on different occasions. The definitions are shown in Table 4.

These three definitions share the same basic concept of describing the business structure and consequently identifying the functions of the system within that structure, while slightly deferring in the focus of attention, depending on the assumed entity, objective, and situation.

6.3 Ongoing efforts into business rules

In the U.S. and Europe, a range of entities including universities, leading companies, venture businesses, and consultants are involved in research and development on business rules, and even human resources development services are available in this field.

In general, a typical organization is said to contain a few tens of thousands to a few million business rules (an argument in the Business Rules Group). Japan lags behind Western countries in the business rules arena. The country has made little effort to launch such R&D schemes that involve researchers in human sociology, whose contributions are important in studies of business rules as the basis for systematizing business procedures. In addition, while the viewpoint of system users is more critical than that of system developers in addressing business rules, there have been few approaches taken from the system users' perspective in Japan.

6.4 Examples of tools and methodologies for "business rules"

One major example is the Business Rule Motivation Model[4]. This aims to identify the business through the model described below. The OMG Business Rules Working Group basically adopts the same model.

In the Business Rule Motivation Model, the functions of an Organization Unit are roughly

Table 4:Activities in groups on business rules

Group & year	Definition of a "business rule"	Explanation
Business Rules Group (1995)	A statement that defines or constrains some aspect of the business	Also intended for reverse engineering from existing systems.
Business Rules Group (1997)	To identify overall business activities through ends and means	Assumes that business rules aim to allow people in the business to describe, analyze, and explain to system engineers their business procedures in their own language (instead of IT language).
OMG (2003)	A directive intended to influence or guide business behavior, in support of business policy that has been formulated through analysis of strengths, weaknesses, opportunities, and threats (SWOT analysis)	OMG is working toward the establishment of a standard on these business rules and business models.

Table 5:Available tools and methodologies

Tool	Developer/Vendor	Brief description
MooD2003 Web Publisher[5]	The Morphix Company (U.K.)	Provides the Business Object Repository and uses the Business Context Models. Defines scenarios, processes, the process hierarchy, the process index, the object index, users, etc., and provides zoom-in capability to review, modify, or add elements to a greater detail in each model.
DEMO - Demo Engineering Methodology for Organizations[6]	Delft University of Technology (Netherlands)	Advocates organizational engineering (OE), as opposed to traditional organizational science (OS), which emphasizes teleonomic definitions on system functionality and behavior. Uses the communicative act, a model in cognitive science, as the fundamental theory.
MEGA Suite 6.0[7]	MEGA International Inc. (France)	Consists of MEGA Process, MEGA Architecture, MEGA Integration, MEGA Development, MEGA Database, and MEGA Repository. Enables cost prediction and risk management for processes. Also provides workflow functions through MEGA Integration, which involves EAI (Enterprise Application Integration).
Proteus, Rule Track[8]	Business Rule Solutions (U.S.)	Provides Proteus, a business rules development methodology, and Rule Track, a development tool.

identified in the framework of Means-Ends analysis. In addition to Means and Ends, which are internal elements of the Organization Unit, Influences and Assessments (SWOT assessment) are used to evaluate the environment. The evaluation results are measured in terms of Risk and Potential Reward.

In this model, the Vision, a statement about the Organization Unit's aspiration, is correlated with the Mission, which works as the Means. In a similar manner, the Strategy is defined for the Goal, which gives a concrete form to the Vision, and so are Tactics for the Objective, which quantifies the Goal.

Business rules act as Elements of Guidance in implementing the defined Strategies and Tactics (a Course of Action). A Business Rule forms a counterpart to a Business Policy. Business Policies are formulated in compliance with external Regulations, on the basis of an analysis of the Organization Unit's strengths, weaknesses, opportunities, and threats (SWOT analysis).

This is how business rules allow people in the business to describe, analyze, and explain to system engineers their business procedures in their own language (instead of IT language). As a result, system engineers are enabled to define the role of the system pursuant to the given business rules.

Other tools and methodologies currently available are listed in Table 5.

7 Conclusion

Japan, including its universities, national research institutes, and industries, lags behind in research and development on the upstream process. This can partly be attributed to poor awareness of the upstream process among the purchasers of systems, such as companies and government organizations.

In other words, Japanese customer organizations, as shown in the White Paper on Information and Communications in Japan, tend to neglect efforts to establish their "business rules" and define their requirements, which are the processes that should be completed before they contract out the development of any system. In an extreme case, an organization would oblige the potential contractor to prepare a quote free of charge. Worse yet, there exist a considerable number of organizations that embark on system construction projects with no attention paid to the upstream process and let the resulting systems grind to a halt before finally starting to discuss who is responsible.

Here are some proposals to solve these problems:

(1) Create environments that make "business rules" and requirements definitions to be identified during system development

More specifically, this refers to considering the enactment of legislation that enforces a process equivalent to construction authorization in housing construction, or making the third-party auditing of the customer's purchase specification mandatory at least for projects concerning public systems.

(2) Foster human resources who can develop "business rules" and requirements definitions

It is essential to foster human resources who can help the purchaser, or the user of the system, formulate correct requirements specifications. A possible solution is to make courses on "business rules" and requirements definition for information systems requisite for students who pursue a master's degree in business administration (MBA). Another important approach is that companies and government organizations provide education on business rules and requirements engineering for personnel not only in their systems departments but also in the departments that will purchase and use the systems.

A serious problem to Japan, when it comes to comparison with the U.S. and Europe, is its particular weakness in the area of university research and education. One of the reasons for this is Japanese universities' slow moves into the fields such as requirements engineering that require knowledge on both science and humanity.

(3) Encourage research and development in the upstream process

Major elements of research and development are listed below.

(a) Formal methods and mechanical verification

Given the advance in automation in the midstream process, important elements in the upstream process are technological development in the area of formal methods that use mathematical specifications and mechanical verification based on these methods. For example, efforts in this direction will allow system engineers to automatically detect contradictions between "business rules" and conflicts between requirements, thereby contributing to integrity checking or comparisons of requirements with the functions provided by commercially available packages. In addition, they will further facilitate the automation of the processes from design to coding, which follow requirements definition.

(b) Technologies for requirements acquisition from stakeholders

To achieve such technologies, not only superficial communication techniques but also techniques to understand what is not said are required. The key to this is the fusion of knowledge and technology between the science and non-science fields, that is to say, the cultural aspect, including humans, and the administrative aspect, as well as the scientific aspect, need to be taken into consideration in an integrated manner. This indeed requires not only systems and software, but also expertise in the concerned field and common sense.

(c) Quantitative evaluation

Because of insufficient quantitative evaluations conducted on technologies and methods for the upstream process, the benefits of upstream activities are underestimated. A possible first step in response to this is to collect data on system construction so that, for example, in the case of a public system, they can be offered for public use. If persistent efforts are made to perform quantitative evaluations using such a database, with a view to extensively analyzing correlations between upstream activities and downstream quality, cost and so forth, effective outcomes will result.

Acknowledgements

The author would like to express sincere thanks to Professor Kokichi Futatsugi of Japan Advanced Institute of Science and Technology, and Professor Motoshi Saeki of the Tokyo Institute of Technology, both of whom provided invaluable advice since the draft stage. The author also deeply appreciates the precious opinions from Visiting Researcher, Masao Watari and Senior Research Fellow, Kumi Okuwada, and other researchers of the Science and Technology Foresight Center.

References

[1] 2003 WHITE PAPER Information and Communications in Japan, in Chapter 1, Figure 2 "Essential elements for optimizing the effectiveness of IT investment" (extracted from successful U.S. and Japanese companies) of Section 2 "Strengthening Competitiveness of Companies and Developing Industry" (in Japanese). The following seven elements are listed as the areas where Japanese companies' results were lower than those of U.S. counterparts by more than 10%.

- "Selection of Core Competence" of business/operations (Identification of core operations, etc.).
- "Verification of cost-effectiveness" before systems construction.
- Reengineering of operations, organization, and systems to comply with IT system operations.
- "Selection and concentration" of investments in information systems.
- "Verification of cost-effectiveness" after the introduction of systems.
- Reutilization of business effects (e.g., Reallocation of reduced cost to new information systems).
- Review of investments in information systems and reinvestment.

[2] Business Rules Community: http://www.brcommunity.com

[3] Business Rules Group: http://www.BusinessRulesGroup.org

[4] "Organizing Business Plans - The Standard Model for Business Rule Motivation," The Business Rules Group, Published November 15, 2000 Rev. 1.0: http://www.businessrulesgroup.org/second_paper/BRG-BRMM.pdf

[5] The Morphix Company: http://www.morphix.com/

[6] Demo Engineering Methodology for Organizations: http://www.demo.nl/start.php

[7] MEGA International Inc.: http://www.mega.com/us/home/index.asp

[8] Business Rule Solutions LLC.: http://www.brsolutions.com/

[9] IEEE International Conference on Requirements Engineering

[10] IEEE International Symposium on Requirements Engineering

[11] Praxis Critical Systems: http://www.praxis-cs.co.uk/

[12] Ask Pete Support Web: http://osat-ext.grc.nasa.gov/rmo/pete/

[13] Defect Detection and Prevention (DDP): http://ddptool.jpl.nasa.gov/

[14] Advanced Risk Reduction Tool (ARRT): http://eis.jpl.nasa.gov/%7Emfeather/Requirements-etc.html

[15] ISAT: Interactive Specification Acquisition Tools Project: http://www.research.att.com/~hall/isat-project.html

[16] Constance Heitmeyer, "Software Cost Reduction," Encyclopedia of Software Engineering, Two Volumes, John J. Marciniak, editor, January 2002.

[17] Nikkei Computer, Aug. 12, 2002 issue, pp. 138-146: http://www.atmarkit.co.jp/fbiz/feature/0305itroi/01/01.html

[18] Object Management Group website: www.omg.org

[19] Finkelstein, A. & Dowell, J. "A Comedy of Errors: the London Ambulance Service case study" in Proc. 8th International Workshop on Software Specification & Design IWSSD-8, (IEEE CS Press), 1996, pp. 2-4. also from: http://www.cs.ucl.ac.uk/staff/A.Finkelstein/las.html

[20] "Defining Business Rules ~ What Are They Really?," Business Rules Project, GUIDE, 1995.

[21] Carma McClure, Japanese translation of "The Three R's of Software Automation: Re-engineering, Repository, Reusability," translated by Best CASE Research Group, Kyoritsu Shuppan, 1993.

(Original Japanese version: published in November 2003)