Exception-Based Approach for Information Systems Evaluation: The Method and its Benefits to Information Systems Management

Heikki Saastamoinen
University of Jyväskylä, Finland
Jyväskylä Polytechnic, School of Information Technology, Finland
heikki.saastamoinen@jypoly.fi

Exceptions are events that cannot be handled by an information system by following normal processing rules. Exceptions arise for two main reasons: flaws in system design and post implementation changes in the system domain. Only few exceptions should arise in an information system serving its user community well. In practice, this is rarely the case and exceptions are sometimes rather common even with routine processes. In this paper, an exception-based approach to evaluate information systems is presented together with practical examples of its use. The benefits of the analysis to information system management are elaborated on.

Keywords: Information Systems Evaluation, Exception Handling, Information Systems Management

1. Introduction

Information systems are implemented with an inherent assumption that they – or the users using them – can handle the associated events the organization faces. This assumption, however, is not completely relevant. The implementation process inherently includes two factors that are hard or impossible to control: system's design and system's environment (Berki et al 2004). In a large number of cases, the system covers its domain area only partially (cf. Wand and Weber, 1995), that is, all requirements are not incorporated in the system, the requirements are conflicting, they are misinterpreted, or will become badly implemented. The just finished system or software is not what was expected.

Even in ideal cases where all requirements are met by perfect design, the system starts to outdate from the very day of its completion, due to the changes in its use environment, that is, the domain area the system was designed to cover changes after the system was implemented. Regardless of the technical or organizational environment, the system is supposed to serve, the organization and the world around it are in constant change, eventually causing changes in the technical environment as well.

The extent the above mentioned two factors influence the usability of the systems varies greatly. When an information system is not able to handle all events of its domain, exceptions arise.

Exceptions can formally be defined as cases for the handling of which no applicable rules exist (Auramäki and Leppänen 1989). For most standardized organizational processes, information systems supporting them form the majority of these rules. Thus, exceptions are in every-day life observed as events that cannot be handled by a system. Incoming invoices not matching with the purchase and stock data, and engineering orders with insufficient data are just a couple of examples of such exceptions.

The approach presented in this paper uses the number and kind of exceptions to analyze the operational usability of information systems. It is claimed that a system associated with a high ratio of exceptions versus normally handled events is not serving the organization. Various characteristics of exceptions are briefly discussed to provide means for more thorough analysis of the system and the process in order to find the most crucial points to be improved. Four main benefits of such analysis to information systems management are suggested. Case examples are provided to illustrate how the evaluation can be done and to demonstrate the value of such exception-based analysis.

2. Rules, exceptions and information systems

Before we can formally analyze the concept of exception, the concept of rule must be discussed. The simplest method of coordinating interdependent sub-tasks is to specify their behavior before their execution in the form of rules or programs (March and Simon 1958). Rules can be viewed as instruments of policies aiming to solve problems (Twining and Miers 1976). The primary virtue of rules is that they eliminate the need for further communication among

Reference this paper as:
Saastamoinen, Heikki (2005) "Exception-Based Approach for Information Systems Evaluation: The Method and its Benefits to Information Systems Management" *The Electronic Journal of Information Systems Evaluation*, Vol. 8, Iss. 1, pp 51-60, available online at www.ejise.com

organizational units (Galbraith 1973). It is sometimes said that the main function of rules is to guide behavior (Twining and Miers 1976). According to Galbraith (1973), rules thus perform the same functions for organizations that habits perform for individuals - they eliminate the need for treating each situation as new. In addition, rules provide stability to the operations of an organization. When people come and go through an organization, the rules provide a constant for handling routine situations. Thus rules not only transfer past learning, they also control behavior within the organization. These two roles permit the transfer of past learning, and provide a unique solution when a task itself does not provide it (Cyert and March 1963).

We here use the term of "rule" as a generic concept that refers to various types of norms, prescriptions and directives. In a broad sense, a rule can be defined as a general term that includes precepts, regulations, rules of thumb, conventions, principles, guiding standards and even maxims (Twining and Miers 1976). Good business practice, standard industry practice, and ethical business practice have been seen as rules as well (Cyert and March 1963). In addition, habits and other structures that guide actors' actions are kinds of rules (Williams and Lochovsky 1989). These, however, are usually not precise enough to be used as the basis of event handling or exception handling.

In general language, the term "exception" refers to an abnormal event. That is also the case with information system exceptions. Before the concept of exception or any other concept related to it can be defined, the concept of a "normal event" has to be clarified. A *normal event* can be defined as an event with the event handling rules necessary for identifying as well as for handling it (cf. Auramäki and Leppänen 1989, Saastamoinen 1993, Saastamoinen and Savolainen 1992). The term "event" here refers to both internal and external events (cf. Wand and Weber 1995). An organization might, for example, have a domain area of order fulfillment for which it has implemented an ERP system including manufacturing, financials, planning and other such modules, to handle events associated with that domain. Examples of events triggering system processing include 'customer orders', 'incoming invoices', and other such external events but also internal events such as 'item stock too low', 'product ready for shipping', etc.

It is these normal events, the information systems are built to process. Kunin (1982) talks about *main line* as a procedure for the most predictable normal events of a certain type. However, when one is working with information system modeling, one eventually must deal with the problematic details caused by flaws in the main line. These details can become variations and exceptions. Kunin (1982) gives the following definition of the concept of variation: A *variation* is work that is added to the main line, i.e., a variation is a procedure for less predictable but still known events of a certain type. An *exception* is an event for the handling of which no applicable rule exists (Saastamoinen, 1995a).

Information systems are formal representations of rules for processing certain events. The systems do not exist alone including only software and hardware and other such "firm" things, but they are always associated with the context they are used in the domain. In any given case, certain processing rules within the domain area are formally implemented in the form of (hardware and) software; the rest is to be handled manually by the users of the system. Thus, from this perspective, an information system is able to process certain events belonging to the categories of *main line* and *variations,* as discussed above, but is not able to handle exceptions. The larger the portion of events that can be handled by the system, the better the match between the system and its domain. The higher the number of exceptions to be handled fully or partially manually, the weaker the match is. Thus, the number and kind of exceptions can be used as an approach to evaluate information systems.

3. Characteristics of exceptions

The nature of exceptions is negative – even though they are sometimes claimed to have positive impacts similarly (cf. Auramäki and Leppänen, 1989). Most organizations are built to perform in a planned and ordered manner around their core processes and main functions. Exceptions are not a part of those plans and require additional attention and work causing processing delays and additional costs. The costs can be significant even with presumably routine processes (Saastamoinen 1995b). Even though all exceptions share this negative virtue, they are different in many other ways.

Auramäki and Leppänen (1989) discern three elements of exceptionality: acceptability, frequency and degree of difference. Inspired by their initial work, a more comprehensive taxonomy of exceptionality was developed.

The dimensions of the taxonomy are (Saastamoinen 1995a):

- *Exceptionality:* The difference between an exception and a normal event based on rules.
- *Handling delay:* The time between the appearance of an exception and when it can be handled.
- *Amount of work:* The amount of extra work caused by an exception when compared to a normal event.
- *Organizational influence :* The number of people the exception involves.
- *Cause:* The reason for an exception.
- *Rule impact:* The change an exception causes to an organization's rules.

In addition to the above dimensions, *frequency* is a noteworthy characteristic of exceptionality from the viewpoint of system evaluation. Even though almost all exceptions occur infrequently (Saastamoinen et al. 1994), there are some kinds of exceptions that happen more often than others when their frequencies are observed over a period of time.

There are several studies (e.g., Auramäki and Leppänen 1989, Saastamoinen et al. 1994, Saastamoinen 1995a) focusing on classifying exceptions and offering detailed analysis of different dimensions of exceptionality. From the viewpoint of information system evaluation its necessary to master those classifications only in the level of details to understand which kinds of exceptions are more harmful than others and should be addressed with higher priority when the system or process are further developed. The longer the delay, the higher the amount of work, and the more people influenced by the handling, the higher the frequency, the more severe an exception is. For the purposes of information system evaluation, the classification is discussed in more detail in Saastamoinen (2004).

The above discussion regarding the concepts of exception and its characteristics is merely an introduction to the phenomenon. There are few key papers addressing the issue in a general level. In addition to the work already referred to in the above, Suchman (1983), Ellis (1979 and 1983), and Strong and Miller (1989 and 1995) report the first studies concentrating on the real nature of exceptions.

4. Exception handling as an approach for system evaluation

As exceptions are undesired events indicating a mismatch between an information system and its domain, analyzing the number and kind of exceptions provides valuable information not only about exceptional events themselves, but also about the entire system. For example, chances in the number and kind of certain exceptions are a good measure when the value and benefits of a new system in place are evaluated and the new and old systems are compared. Furthermore, the number and kind of exceptions can also be used to evaluate system adaptation, as the number should decrease and severity diminish when the users learn the new process and system. The same analysis can be used even when the successfulness of various rollouts of the same packaged software are compared or evaluated.

4.1 Overview of the approach

Using the number and kind of exceptions for the above mentioned or other purposes is rather an approach than a method. However, practice has proven that certain steps are to be taken for the evaluation to be reliable and to form a solid basis for further development activities. The most important steps to be taken – or issues to be considered – are listed in Table 1:

Table 1: The steps of the approach

Main steps	Steps
Preparatory tasks	Selection of the process and information systems to be evaluated
	Initial analysis of causes
	Constructing a research form or data collection system
	Selection of the evaluation period
Evaluation	Briefing of the employees
	Controlling the data collection
	Analyzing the results
Developing the system and organization	Publishing the evaluation results
	Development actions

The system to be evaluated with this approach needs to be selected with caution. This is, however, in many cases a step that has already implicitly been taken – there is an information system with presumably high number of exceptions. The approach is not inherently limited to certain kinds or types of systems only, however, for optimal results certain precautions have to be made. Quite many information systems are much too complicated, involve huge amounts of users and process so large a number of different kinds of events, that it is not feasible to evaluate the entire system. One can also assume, that it is not the whole system, e.g.,

an entire ERP-package, that would need to be evaluated, but rather a part of it, for example, pay-roll processing, order entry and confirmation, or invoice processing.

Before the evaluation starts, one should carefully study the systems history: are there some factors external to the system itself that are likely to cause the high number of exceptions? For example, a system might have been implemented as a part of the corporate policy even though it was not really meant for this specific line of business, the system is already a couple of decades old, the users did not receive proper training for the system or were not able to participate in its design, recent changes in the organizational structure or processes - just to mention a few of such factors. These issues need to be noted in advance to be able to explain the results.

One should be able to clearly define what are normal cases are and how they are processed. Knowing more about exceptions has value of its own, but for the purposes of developing the system, process, or organization, the information is much more usable when it can be reliably compared to respect normal events. Furthermore, if one cannot determine what a normal case is and how it is processed, there is hardly a way to analyze the exceptions. This is not necessarily a problem with an organization's ability to define its processes, but can also be an indication of the system's nature. In the beginning of their famous and widely used textbook "information system management in practice", McNurlin and Sprague (2001) distinguish two main types of information work; *procedure based* and *knowledge based.* It is of capital importance to note, that analysis of exceptions is likely to be more beneficial with systems supporting procedure based work. Knowledge based work is often based on ill-structured procedures and its output measures are less defined. With such work one can hardly define a normal event or the task might even have been set to create new ideas, to make decisions, or to create something new.

One of the main benefits of the evaluation is the information of the real causes of the exceptions. Of course, during the course of an analysis one can ask with every exception what caused it. However, in many cases, a majority of the causes are already known, only their real number, frequencies, and relative portions are not.

When a data collection form or a system is designed, it might be useful to utilize the list of known causes as the basis of the form. When certain parts of a system and process are analyzed, the exceptions are likely to be very much alike. For example, for some reason an incoming invoice does not match with the system's data about the corresponding order or goods received. This kind of exceptions - as the cases described at the end of this paper demonstrate - can occur quite frequently. The idea of the evaluation is then not to collect information about exceptions in order to be able to classify them by using taxonomy but rather to gain understanding on why the exceptions take place and how the organization handles them.

When an evaluation period is selected, extraordinary periods of time, such as holiday season, ends of reporting periods, and short peak seasons, should be avoided as the information gathered from such periods is hard to generalize and the results of the study would be too vulnerable for critique. Experience has also shown, that the period should be fairly long to be able to form a picture of how the organization normally handles exceptions. If the evaluation period is short, e.g., one week only, and people know their actions are monitored, they are likely to try to work more efficiently and more precisely to prove their own skills and value. However, a bit longer period, e.g., a few weeks or a month, seems to eliminate that problem by being already too long for the employees to work with other than their normal pace and accuracy. One more factor influencing the period to be selected is the desired sample size, which also speaks for a longer period.

The employees participating in the evaluation have to be well informed. Unless the information system to be evaluated can by itself be used to collect the data, or there is another system associated with it that can be used for the purpose, the collection of the data relies completely on employees. If the evaluation period is of sufficient length and the number of exceptions is high, this can result to a significant amount of additional work. The importance of filling in every form completely and accurately – either on screen or in a sheet of paper – has to be stressed out. One cannot overlook the fact that many exceptions are internally caused by staff inexperience or staff carelessness, and some of those people causing the exceptions initially might be the people also partially handling them on the later stage. With that in mind, it has to be

emphasized that one is evaluating the information system and the process it supports, not the people using it.

As the above is unlikely to be fully absorbed by all participating employees, it is vital for the researcher to be active especially in the beginning of the evaluation period. Should there be forms incompletely filled or with inaccurate data, the problem needs to be addressed immediately to avoid further such problems and to correct the data perceived inaccurate.

After the data is collected, it can be analyzed with normal statistical methods. It still has to be noted, that part of the people participating might have been the people causing the problem initially and that others might be the ones the work of whom is slowed down or complicated by the problems. The first group might have had an objective to diminish the problem while the latter group might want to emphasize it.

4.2 Cautions regarding the use of the approach

There are a few general cautions that have to be made about the use of this approach. First of all, this approach does not provide its applier with a holistic view to the information system evaluated; it only focuses on the part of the system not performing as planned. One should not judge the entire system or the organization based on the results of this analysis. Even though the number of exceptions observed might be relatively high, a conclusion that either the system is wrongly implemented or totally outdated, or that the organization is incapable of using the system, might be premature.

Secondly, this approach does not vary from any other approaches: the researcher using the approach needs to understand well what he or she is doing, and the one implementing the approach should know the organization well enough to be able to interpret the results correctly.

If an overly high number of exceptions are observed, definite conclusions need to be drawn and corrective actions have to be taken, however, a more throughout analysis is just required before them. A large number of exceptions observed can be an evidence of fatal mismatch between the information system, its users, and the organization. It can also be an indication that the process underlying the system and the one assumed by the organization do not match - or it can be due to the fact that the target system of analysis or the period of the analysis were not well selected.

4.3 Using the approach for information systems management

An analysis based on exceptions can be very valuable from the viewpoint of information systems management. The analysis provides an information system management team with the four main benefits:

- Structures prioritization of system development and maintenance activities
- Shifts the focus from technology to processes
- Increases communication with the user community
- Provides feedback about the performance of information system operations

In the following, each of these benefits is briefly discussed with some emphasis on the first one of them.

4.3.1 Prioritization of system development and maintenance activities

Information systems are to serve the organization. In terms of the *procedure-based* systems (cf. McNurlin and Sprague, 2001), exceptions should not exist in large numbers. If that is the case, an organization has a system in place that fails to fulfill its purpose. The flaws revealed by using the approach should be corrected – on the system or on the organization that is using it – for the organization to perform well.

For fulfilling the ultimate goal of any given organization, information systems are unlikely to be equally important. Some systems have a more crucial role while the others are merely supporting some secondary activities. We can call this relative importance of the systems to their *impact* on organizations' performance. The impact depends on a number of issues varying from organization to another: relative amount of revenue flowing through the system, number of users using the system, system's visibility to external customers, the system being or not being a part of e-commerce solutions or portals – just to name a few.

By combining the two, the number of exceptions observed and the impact of the systems, an organization can outline a graph to pinpoint the systems that most urgently need to be either further developed or replaced

(see figure 1). It needs to be noted here, that this prioritization only covers systems already existing; new systems possibly required by the business are not included.

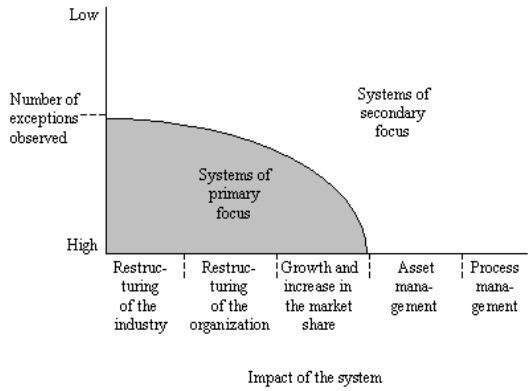

Figure 1: Prioritization of IS development activities

As the impact of the system is a fairly vague concept, it is strongly recommended here that a formal approach to classify systems by their impact should be used. For example, a split to operational control (including process management and asset management) and to organizational effectiveness (including growth and increase in market share, restructuring of the organization, and restructuring of the industry) presented by Primozic, Primozic, and Leben (1991) can serve as a neutral framework over multiple personal opinions regarding the importance and impact of organization's information systems.

4.3.2 Focusing on processes

Exceptions arise as a result of a lack of applicable rules needed to handle events. As discussed earlier, those rules can be incorporated into information systems, or they can exist in the minds of the user community. When exceptions are analyzed, it is found out that they are virtually never caused by technical malfunctions (Saastamoinen et all, 1994). On the contrary, the majority of them seem to be caused by people; either by the users of the system or closely connected parties such as suppliers (Saastamoinen, 1995b).

Keeping the above in the mind, the activities taken to decrease the number of exceptions are unlikely to be only software development projects or other such technical undertakings. The focus naturally shifts on the issue how the system is used. The process the system serves and how the process and the system match will be analyzed first, hardware or software implementations are to follow on the later stage.

4.3.3 Increased communication with the user community

As is apparent in the above, exceptions cannot be avoided by the actions of information system professionals only. The issue of exceptions has to be thoroughly addressed by the users and IS professionals together. This has few obvious bur remarkable benefits: it helps IS professionals to understands the business, fosters relationships between the IS professionals and the users, and assists in creating a common vision with the users.

Furthermore, as briefly discussed earlier, exceptions have two root causes: flaws in systems design and changes in the systems environment. The first of the two is highly related to the communication. It is claimed here, that the systems are not necessarily overly difficult to design to match the reality more closely. In large number of cases the requirements of the users are simply not expressed well enough and are understood even more poorly resulting to the requirements not being implemented at all. To avoid the same recurring while systems are updated, all communication assisting in real exchange of information and contributing on mutual understanding on the system and its requirements is of essence for the entire organization.

4.3.4 Feedback about the performance of information system operations

Using the approach to evaluate information systems is not only valuable as such, it furthermore can server as a way to evaluate the performance of information system operations. A multitude of models and scoring systems exist to evaluate the performance of information systems and technology. Some of the models are widely used in practice, some can even be found in the literature. What appears to be the most difficult factor when IS/IT operations are to be evaluated objectively, is the performance of the information systems, that is, the match between the systems and the processes it serves.

This evaluation can be greatly facilitated by systematically using the kind of an approach reported in this paper. It is suggested here, that the results of the same analysis that is used to prioritize the development and maintenance activities, as suggested earlier in

this chapter, could be used also to evaluate the performance of the information system operations of the organization. Evaluating IS/IT operations of an organization without proper emphasis on the support the systems give to crucial processes would clearly be less than adequate.

5. Case example: A large engineering shop

This approach has been used extensively in a multitude of cases, however, a set of two case studies performed in a large engineering shop in 1993 (Saastamoinen, 1995b) and again in 2001 are the best examples for the purposes of this paper. Even though the study carried out in 1993 involved also other systems, both of the studies focused on unmatched incoming invoices.

An invoice does not match if the data stored from the invoice does not match the data of the corresponding order or the data of the corresponding deliveries. For example, if the unit price in the order database does not match the unit price in the invoice database, the corresponding invoice is unmatched. Likewise, if an item invoiced cannot be found in the storage database or the number of delivered items does not match the number of invoiced units, the invoice does not match.

These unmatched invoices are exceptions for transaction verifiers. When they verify an invoice, they are supposed to be able to decide whether it is correct or not. If it is unmatched, they may make inquiries as to whether it is really incorrect or not. For example, they can make a query to the storage personnel to find out whether items that seem to be missing have been delivered but are not yet in the storage database. Or they can compare an unmatched invoice to other orders from the same supplier in order to find out whether an invoice contains items from other orders not mentioned in the invoice. However, even though these methods are sometimes applicable, there are no rules that indicate which method might produce a solution. Thus, unmatched invoices can often be classified as established exceptions. Sometimes there seems to be no method for verifying an invoice at all, in which case an unmatched invoice is an otherwise exception for the transaction verifiers.

This problem was well recognized, however, its real extent was not known and purchasers tended to diminish the problem and even blamed transaction verifiers about making inquiries for no or minor reasons.

The department of finance had initially analyzed and listed the most typical causes for the invoices not to match. This listing was used as a basis of a research form to be used in the detailed study. The form was constructed to gain more profound understanding of the problem: what are the real frequencies of the kinds of exceptions, what causes them, what does their handling cost, and how much delay do they cause before a proper payment can take place.

A period of four weeks was carefully selected – there were no major holidays, no closing of books, no special reporting, and the business was generally assumed to run as usual. Throughout the period, every unmatched incoming invoice was inspected individually by physically attaching the research form with each of them and by making sure that the information requested in the form was carefully filled in at each stage of the process.

Before the actual study started, all the employees that would have to fill in the form were briefed in departmental briefings where also their managers were present to stretch the importance of the study. Individuals missing from the briefings were informed individually. The first two weeks included a lot of individual discussions with the employees who failed to correctly fill in the form. They were all purchasers, who often found themselves as the causers of the exceptions and did not want to report that they had ordered the goods with outdated prices or had not specified the terms of delivery, just to name a few of the problems they had caused.

The study covered a total of 2687 invoices of which 902 unmatched and were thoroughly analyzed. There were a total of 1367 causes for the invoices not to match. These causes where classified to 21 categories. Also resulting delays in the handling of the invoices were calculated. Possible unnecessary money transfers caused by various exceptions were evaluated as well. Various management actions taken as a consequence of the study have been reported in (Saastamoinen 1995b) and (Saastamoinen 1995a).

The same study was repeated eight years later. The engineering shop had implemented a new ERP-system replacing its old legacy system still in use on the time of the previous study. From the viewpoint of transaction

verification, the new system had certain major advantages and disadvantages that were often brought up to general discussion inside the engineering shop - even though their factual impact to the work of the people had not been studied. The core problem itself – unmatched incoming invoices – again appeared as a major problem in the order fulfillment process.

The study in 2001 also covered all the invoices during a period of four weeks and utilized a very same kind of a research form. This study not reported in detail in public provided similar information as the previous study, but also provided the organization with a longitudinal view into the system and process, resulting in a more comprehensive understanding of the situation.

For example, as a great majority of the exceptions were caused internally as a result of staff carelessness or incompetence, the kind of exceptions had increased dramatically as a result of decentralizing the Department of Purchase. Many exceptions caused by the suppliers had not received the attention of the purchasers as the same supplier now received orders from dozens of part-time purchasers instead of a few full-time professionals. To reduce the number of invoices that had to be sent to purchasers prior to payments, the engineering shop had also given the transaction verifiers an authorization to approve exceptional invoices if the monetary error was within a certain limit. Accompanied with the decentralization, this had caused the company to loose a view of supplier performance outside of the real hard performance factors, such as on-time delivery and quality issues.

The study also provided factual information about how the new system served transaction verifiers when compared to the old system. This information was also, like in the previous study, converted to salary related costs as a result of certain tasks taking more or less time.

The exception based approach was in both cases found to be a proper approach to evaluate the system. If the system would have been evaluated just by external measures, one could have claimed that there was hardly anything to improve – goods were purchased, they were received in time and invoices were paid. Neither the new nor the old software collapsed, the data was not corrupted, the systems produced desired reports, etc. However, a more internally focused exception-based analysis could, without doubt, point out major flaws in the system and its use.

6. Summary

Exceptions are events for the handling of which no rules applicable as such exist. Exceptions are inevitable and common and they are often associated even with routine tasks and processes. Even though all exceptions cannot be avoided, most of them could be handled normally by information systems, if the systems and processes and the people using the systems as a part of the processes were aware of the exceptions they cause and the match between the system and its domain would be tighter.

Information systems can be evaluated by focusing in exceptions. The approach field tested in various industrial organizations over the past eight years and presented in this paper can be used to analyze systems functionality in the user's level, to compare systems, and to provide factual information on the system and process flaws.

Focusing on exceptions can be very beneficial for information systems management in practice. Such analysis formalizes prioritization of system development and maintenance activities, shifts focus from technology to processes, increases communication with the user community, and provides valuable feedback about the performance of information system operations.

References

Auramäki, E., Leppänen, M. (1989) "Exceptions and office information systems" in Pernici P., Verrijn-Stuart, A. (Eds): *Office Information Systems: The Design Process,* Elsevier Science Publishers B. V. (North-Holland), IFIP, pp167-182.

Berki, E., Saastamoinen, H.T., Zhang, Z., Georgiadou, E., Holcombe, M., Ross, M., Staples, G. (2004) "The Academic, Industrial and Organisational Challenges of Software and Requirements Engineering to Accommodate Total Quality Management" in *The Proceedings of Software Quality Management 2004*.

Cyert, R.M., March, J.G. (1963) *A behavioral theory of the firm,* Prentice-Hall, Englewood Cliffs, NJ, p104.

Ellis, C.A. (1983) Formal and informal models of office activity. In *Information Processing 83,* Mason, R. E. A (ed):

Elsevier Science Publishers B. V. (North-Holland), IFIP, pp11-22.

Ellis C.A. (1979) Information control nets: a mathematical model of office information flow. In *Proceedings of ACM conference on Simulation, Measurement and Modeling of Computer Systems,* pp225-239.

Galbraith, J.R. (1973) *Designing Complex Organizations,* Addison-Wesley, Reading, MA, pp10-11.

Kunin, J. (1982) *Analysis and Specification of Office Procedures,* Massachusetts Institute of Technology, Laboratory for Computer Science, MIT/LCS/TR-275, p56.

March, J., Simon, H. (1958) *Organizations*, John Wiley, New York, pp142-150.

McNurling, B.C., Sprague, R.H.Jr. (2001) *Information Systems Management in Practice,* Prentice-Hall, 5th edition, New York, pp17-18.

Primozic, K., Primozic, E., Leben, J. (1991) *Strategic Choices: Supremacy, Survival, or Sayonara,* McGraw-Hill, New York.

Saastamoinen, H.T. (2004) "An Exception Based Approach for Information Systems Evaluation", in Remenyi, D. (Ed), *Proceedings of the IX European Conference on Information Systems Evaluation ECISE 2004,* Amsterdam, November 11-12, pp371-378.

Saastamoinen, H.T. (1995a) *Exception Handling in Information Systems,* Ph.D. Thesis, University of Jyväskylä Press, 1995, pp57-120.

Saastamoinen, H.T. (1995b) "A Case Study on Exceptions", *Information Technology and People,* Vol 8, No 4, pp48-78.

Saastamoinen, H.T, Markkanen, M., Savolainen, V.V. (1994) *Survey on Exceptions in Office Information Systems*, University of Colorado at Boulder Technical Report CU-CS-712-94.

Saastamoinen, H.T. (1993) "Rules and exceptions" in Kangassalo, H., Jaakkola, H., Hori, K., Kitashi, T. (Eds), *Information Modeling and Knowledge Bases IV: Concepts, Methods and Systems,* IOS Press, Amsterdam, pp271-286.

Saastamoinen, H.T, Savolainen, V.V. (1992) "Exception handling in office information systems" in *Proceedings of the Third International Conference on Dynamic Modeling of Information Systems,* Noordwijkerhout, The Netherlands, pp345-363.

Strong, D.A, Miller, S.M. (1995) *Exceptions and Exception Handling in Computerized Information Processes.* ACM Transactions on Information Systems, Vol 13, No 2, 1994, pp206-233.

Strong, D.A., Miller, S.M. (1989) Exception handling and quality control in office operations, *Boston University School of Management Working Paper Number 89-16,* Boston, MA.

Suchman, L.A. (1983) Office procedure as practical action: models of work and system design. *ACM Transactions on Office Information Systems* Vol. 1, No. 4., pp320-328.

Twining, W., Miers, D. (1976) *How to Do Things with Rules, A Primer of Interpretation,* Weidenfeld and Nicolson, London, pp49-69.

Wand, Y., Weber, R. (1995) "On the deep structure of information systems" in *Information Systems Journal,* Vol 5, pp185-202.

Williams, L.J., Lochovsky, F.H. (1989) "Supporting knowledge migration in organizations" in Ritter, G. (Ed.): *Information Processing 89,* Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1989, pp259-264.

 ISSN: 1566-6379