December 2000

Factors Affecting Information Systems Volatility

Jon Heales
University of Queensland

FACTORS AFFECTING INFORMATION SYSTEM VOLATILITY

Jon Heales
University of Queensland
Australia

Abstract

The objective of this research is to investigate the effect that various factors have on an information system's life span by understanding how the factors affect an information system's stability. The research builds on a previously developed two-stage model of information system change whereby an information system is either in a stable state of evolution in which the information system's functionality is evolving, or in a state of revolution, in which the information system is being replaced because it is not providing the functionality expected by its users.

A case study surveyed a number of systems within one organization. The aim was to test whether a relationship existed between the base value of the volatility index and certain system characteristics. Data relating to some 3,000 user change requests covering 40 systems over a 10-year period were obtained. The following factors were hypothesized to have significant associations with the base value of the volatility index: semantic relativism (generation of language of construction), system size, system age, and the timing of changes applied to a system. Significant associations were found in the hypothesized directions except the timing of user changes was not associated with any change in the value of the volatility index.

1. INTRODUCTION

Organizations are likely to obtain the highest return on their investment in an information system during the period of operational stability, i.e., from implementation and acceptance through to obsolescence. Ways to improve the return include extending the period of operational stability or reducing the ongoing costs of maintenance and enhancement. This paper seeks to investigate factors that extend the period of operational stability.

1.1 General Problem Area

Users request many changes to an information system during the period of operational stability in an effort to improve the capabilities of the information system over time. Because the system improves its functional capabilities during the period of operational stability, this period is referred to as the *evolutionary phase* (Olle et al. 1991). Toward the end of the evolutionary phase, difficulties encountered with the implementation of user requests may trigger a new system development life cycle (SDLC): *revolution*. Provided the ongoing system changes satisfy users, the system remains stable and is unlikely to be replaced. Its volatility is low. Volatility is defined as the propensity of an information system to change its state from evolution to revolution. The more volatile an information system becomes, the more likely it will be replaced.

This paper examines a variety of factors that facilitate user defined system changes. Some factors may be under management control, e.g., the language of construction, while others are inherently systems characteristics, e.g., size. Nevertheless, knowledge of these factors can influence how management decides to organize and construct their information systems, and to a certain extent allow management to control systems longevity. Knowledge of these factors can also provide management with insights as to the consequences that these factors have on maintenance, enabling them to plan resources accordingly. This paper utilizes a theoretically-grounded model of information systems change to test the effects that various factors have on systems volatility.

1.2 Prior Research

Dekleva (1992) sought to understand the factors that affect the time and cost of evolutionary change. A number of activities taking place during the evolutionary process in a number of categories were examined, including user requests, support activities, and bug fixing. System size, system age, and the number of users all increased the total time spent on system enhancements.

Other researchers have examined various aspects of information system maintenance and have used a variety of change classification schemes to anchor their work (Belady and Lehman 1976; Dekleva 1992; Lederer and Prasad 1992; Marche 1993; Willcocks 1992). These findings are based on observation and have provided only limited theoretical bases to explain or predict an information system's behavior based on classifications of observed user change requests.

2. RESEARCH APPROACH

This research focuses on adaptive maintenance activities (Swanson 1976; Swanson and Beath 1989) rather than corrective or perfective maintenance activities because information system longevity is dependent on its ability to change in response to new requirements.

This study begins by briefly describing a theoretically-based model of information system change covering the operational life of the information system (Heales 1995, 1998). The model extends the work of Olle et al. (1991), Lederer and Prasad (1992), and Marche (1993) by providing a ontologically-based method for classifying information system changes. The model is useful because it introduces the concept of the volatility of an information system to change (replacement).

Work done by Wand and Weber (1995) forms the basis for classifying adaptive information system change activities. The basis for the classification is grounded in Bunge's (1977, 1979) ontological formalism, which Wand and Weber have applied to the information systems domain. They identify three sets of features that characterize information systems using the linguistic traditions of Chomsky (1965) to describe them. First, the *deep structure* of an information system conveys its semantic component. It is the manifestation of the semantics of the real-world system that the information system is modeling. Second, the *surface structure* provides the interface between the information system and its users. It reveals the semantic component of the information system to its users. Third, the *physical structure* delivers the semantic component. It manifests the nature and form of the technology used to implement the system. An instrument has been developed to classify user change requests into one of these three types of adaptive information system changes.

The volatility-index value of an information system through its evolutionary phase will follow a bathtub-shaped curve. Wand and Weber established that for an information system to be stable, its deep structure must represent a "good" mapping between the real world and the information system. Thus, when there is a requirement for deep structure changes to be made to an information system, and these are not being made, the mapping between the real world and the information system is not "good" and the information system becomes unstable. Heales (1995, 1998) extends this notion to show that, proportionally, the effort spent on deep structure changes will be higher during periods of system instability in an effort to improve the mapping. As Tyre and Orlikowski (1993) point out, cycles of intensive change can occur and, provided changes are made to the user's satisfaction, the mapping will remain good, volatility will not rise, and the system will remain stable.

The proportion of effort spent on deep structure changes, vis a vis all changes, is known as the volatility index. Deep structure changes will be more extensive at the beginning and the end of a system's useful life (see Figure 1).

An information system is often unstable at the beginning of its useful life (i.e., immediately after implementation I to A in Figure 1), because business requirements overlooked during system design and/or those that have arisen subsequently during the construction phase must be addressed. If they are not addressed, the implementation may be reversed and, in the extreme, the new system abandoned.

Toward the end of its life, a system can also become unstable because it cannot be adapted effectively and efficiently to meet business requirements. If a system's stability is not restored, it becomes obsolete and is replaced (y to B in Figure 1).

It is further hypothesized that there is a threshold x for each system such that when the volatility index exceeds x the system changes state to revolution. As long as the volatility index remains below x, the system will continue to evolve. Thus, ways to increase systems longevity are either to lengthen the curve, i.e., increasing the longevity of the system, or change the height of the curve (volatility index) on the y axis, i.e., change the mix of adaptive-change work performed by IS departments.

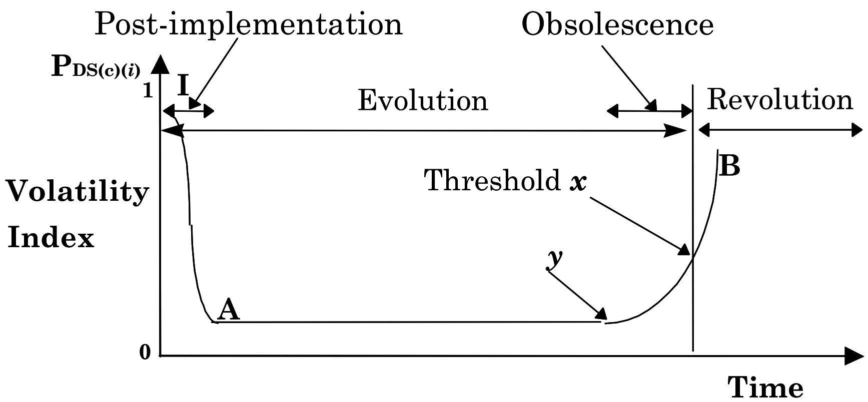

Figure 1. Volatility Index (Proportion of Effort Spent on Deep Structure Changes Over Time)

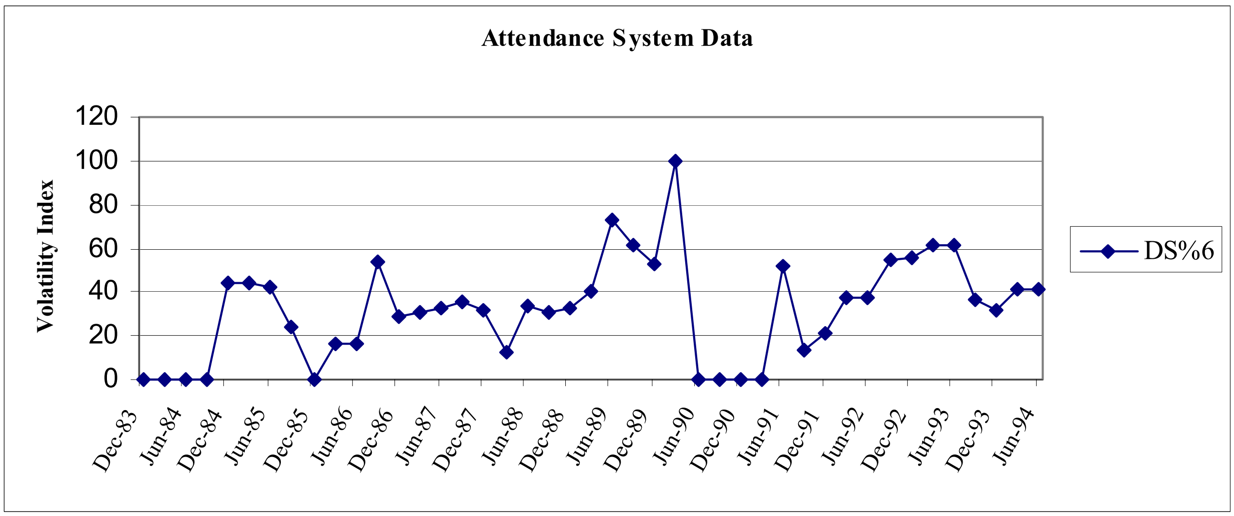

Figure 2. Volatility Index for Attendance System Data

To illustrate, Figure 2 shows the volatility index for one of the systems that was used in the analysis. The system was replaced in June 1990. The graph clearly shows the rise in volatility before replacement (June 1989) and the initial high volatility following implementation (June 1991).

This research focuses on system characteristics that are likely to affect stability. The system characteristics of interest were identified from work done by Dekleva (1992), Willcocks (1992), and Willcocks and Lester (1992). The following system characteristics were examined: the size of the system (in lines of code and number of users), the age of the system, the degree of semantic relativism[1] (e.g., RDBMS systems), and the timing of the change.

[1]Semantic relativism manifests the degree to which a data model can help in representing not only one, but many, diverse conceptions of the same real world, and at the same time allow them to cooperate (Saltor and Garcia-Solaco 1993).

3. THE MODEL

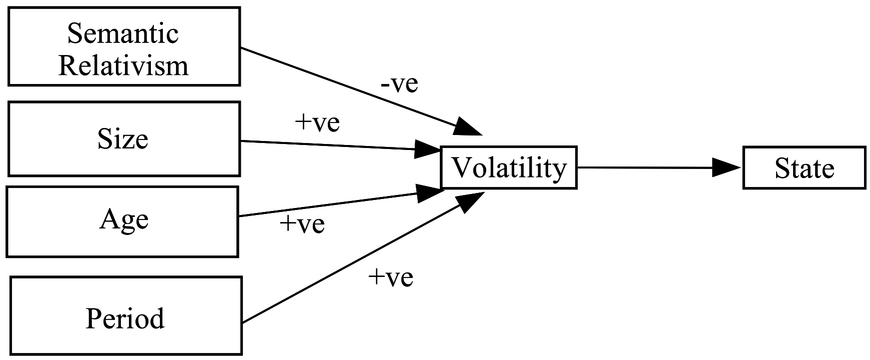

Figure 3. A Model of Factors Affecting an Information System's Chang in State from Evolution to Revolution

3.1 State

The dependent variable, *State,* is a dichotomous variable that reflects the status of an information system (see Figure 3). Immediately following implementation, an information system enters its evolutionary phase, and *State* takes on the value "Evolution." When steering committee (or managerial) approval is given to commence a new SDLC, *State* changes its value to "Revolution." In normal circumstances, information systems progress from revolution to evolution to revolution. Following a system's implementation, the value adopted by the variable *State,* in the new information system, is "Evolution." The cycle repeats itself.

Information system change requests are the accepted method for initiating and controlling information system changes (Arthur 1988; Martin and McClure 1983). User change requests that address the deep structure of information systems are known as deep structure change requests. Similarly, physical structure changes and surface structure changes are those user change requests that address the physical structure and the surface structure respectively.

3.2 Factors Affecting Stability

3.2.1 Semantic Relativism and Stability

The semantic relativism of an information system is an important feature associated with its flexibility and ability to support different users' conceptions (Saltor and Garcia-Solaco 1993). Changes to a system that increase its flexibility and its ability to support different users' conceptions are likely to imply changes to ontological constructs that manifest an information system's deep structure. Changes will also be necessary to the surface structure so that such changes can be observed by users. Deep structure changes are likely to require changes to the design specification.

In a system with high semantic relativism (e.g., using a relational data base management system, RDBMS), both surface structure changes and deep structure changes require less effort to implement. For example, RDBMS technology facilitates the creation of new reports and inquiry screens. Data input screen formats can be changed easily, provided they do not include new data elements or delete existing ones. These surface structure changes are facilitated by the use of the data base management system (Date 1986). Many deep structure changes, such as adding attributes or new fields to a data base, can also be made more easily by using a data base management system (Date 1986). Systems with high semantic relativism will more easily be able to cater for different users' conceptions, thereby reducing the pressure for an information system revolution.

A reduction in deep structure effort would have the effect of reducing both the numerator and denominator of the volatility index equation by equal amounts, thereby reducing the value of the volatility index. Because the reduction in deep structure effort is likely to be greater than the reduction in surface structure effort, *certeris parabis*, the value of the volatility index will be reduced.

P1 The volatility-index value during evolution will be lower in information systems with high semantic relativism than in systems with low semantic relativism.

Systems with high semantic relativism provide easier ways of adding to and changing views of an organization's data, i.e., making deep structure changes. This improved ability to accommodate deep structure changes should extend the life of the system because the backlog of deep structure user requests will not build up so readily. The volatility-index curve for a system with high semantic relativism is shown as the RDBMS curve in Figure 3. Note that the threshold *x* has moved further along the horizontal axis of the graph, extending the life of the information system from t_1 to t_2.

P2 Information Systems with high semantic relativism will have longer lives than systems with low semantic relativism.

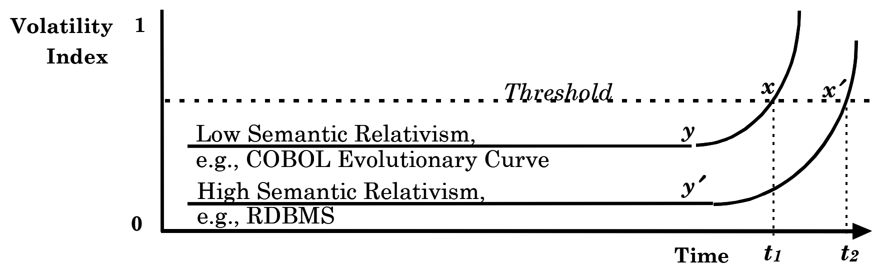

Figure 3. Effect of Semantic Relativism (RDBMS) on Volatility-Index Curve

3.2.2 Size and Stability

Measures for system size have been difficult to define. Executable lines of code (ELOC) has been found to be highly correlated with other measures of system size (Banker et al. 1991; Kemerer 1995; Wrigley and Dexter 1991). Dekleva (1992) found size contributed significantly to time spent on maintenance.

The number of users can also be considered a measure of size because of the impact that changes to a system have on an organization. Little has been done to investigate the relationship between the number of users, other systems characteristics, and error rates. Dekleva found a direct relationship, however, between the number of users and time spent on maintenance.

System size, therefore, is considered as a combination of lines of code and number of users.

The proportion of effort spent on deep structure changes is likely to increase for larger systems because they support greater functionality, their components are more interwoven, they have more extensive systems documentation, and greater numbers of users are required to agree to functionality and changes to functionality. Deep structure changes require more effort in larger systems because they involve changes to the systems design. On the other hand, there is nothing to suggest that surface structure and physical structure changes require more effort in larger systems because they do not substantially change the system's functionality.

P3 The volatility-index value during the period *A* to *y* (the period of stability) will be higher for larger systems.

3.2.3 Age and Stability

Maintenance time for information systems is likely to increase with age (Belady and Lehman 1976; Flatten et al. 1992; Lehman 1980). 1992). Dekleva's results support this proposition. Dekleva's measure of maintenance time includes all time spent on maintenance (corrective, perfective, and adaptive).

Assuming that the practice of developing and maintaining systems has improved over time because of improved tools and techniques, newer systems are likely to be better structured and documented. The effort involved in making changes to the design specification will be commensurately lower. This means that implementing deep structure changes should require less effort in newer systems than in older systems. Older systems will have a higher volatility-index value than newer systems because of the higher levels of effort required to implement deep structure changes in older systems.

P4 The volatility-index value will be higher in older systems.

3.2.4 Period and Stability

As systems age and are subject to ongoing change, they become more fragmented and intertwined. Documentation deteriorates because it is often not updated appropriately after changes are made. This makes subsequent changes more difficult to make and more time consuming. It follows that changes made later during the evolutionary period will take longer and require more effort than changes made earlier in the evolutionary period because of the erosion of code quality later during evolution.

P5 The volatility-index value will be positively correlated with the period in which changes are made.

4. RESEARCH METHOD

A cross-sectional analysis was performed by comparing the mean values of the volatility index for different systems where the systems being compared exhibited differences in those systems characteristics hypothesized to affect stability. The data required for the cross-sectional analysis were drawn from user change requests.

To eliminate bias arising from organizational differences, data used is from only one organization, a South African mining company, that maintained detailed records of all system changes. The mining company had 10 mines and employed approximately 750,000 mine workers throughout South Africa. The information systems department employed approximately 350 information systems personnel, of which approximately 130 were employed in the Systems Department. The cost of individual changes varies from less than \$500 to over \$300,000, with an average slightly in excess of \$4,000.

The proportion of effort spent on deep structure change, calculated for each time period, comes from the change history data base of the system in question. The change classification instrument categorized each change. The effort spent on each change is mapped to the appropriate time period using start and finish dates from the change data base to apportion the effort. Time periods used are quarters, giving, for example, 20 quarters extending over five years. Actual costs associated with changes (or worker-hour figures where cost figure figures are unavailable) were also derived from the change history database. The volatility index is then calculated according to the following formula:

$$P_{DS} = \frac{CDS}{CDS + CPS + CSS} \qquad \textbf{Eq. 1}$$

where CDS = Cost of deep structure changes
CPS = Cost of physical structure changes
CSS = Cost of surface structure changes

P_{DS} is a normalized variable ranging in value from 0 to 1. To smooth the curve, a cumulative period of *n*-6 to *n* is used. The $P_{DS(c)}$ for period *n* is given by the formula:

$$P_{DS(c)(n)} = \frac{\sum_{i=n-6+1}^{n} CDS_{(i)}}{\sum_{i=n-6+1}^{n} CDS_{(i)} + \sum_{i=n-6+1}^{n} CPS_{(i)} + \sum_{i=n-6+1}^{n} CSS_{(i)}} \qquad \textbf{Eq. 2}$$

5. MEASUREMENT OF VARIABLES

The independent variables in this research are all attributes of systems. Table 1 shows all independent variables and their associated operational measures.

Table 1. Summary of Research Variables and Operational Measures

Short Name	Variable	Operational Measure
DSPCT	Volatility Index	A number ranging from 0 to 1 and calculated using Eq.2. The volatility index represents a measure of an information system's stability.
LANG	Semantic Relativism	A number representing the generation of language used to develop the system. Determined by a panel of experts and measured on a seven-point scale.
PERIOD	Period	An integer for the period during which the change was requested.
AGE	Age	Dichotomous variable in response to questions about system demographics and supplied as part of a system's demographics by the IT manager (old is 1986 or before; new is after 1986).
LOC	Size	An integer representing executable lines of code, supplied as part of a system's demographics by the IT manager.
USERS	Size	An integer supplied as part of a system's demographics by the IT manager.

5.1 Semantic Relativism

One way to create diverse conceptions of the real world is to create different views of the data set. This outcome is normally achieved by programming output reports or displays with or without the aid of a database management system. Different representations of the same data set can be created more easily using high-level languages instead of lower-level languages. For this reason, the language (or languages) used for the construction of the system was considered to be a proxy for the level of semantic relativism provided by the system.

Transformation of the language (or languages) of construction to an ordinal measure was undertaken with the aid of a panel of four experts. The measure, LANG, is based on a seven-point scale ranging from 1, for a system written in assembler, to 7, for a fully 4GL-based system (see Table 2).

Table. 2. Scale for Ranking Information Systems Languages of Construction

System	Rank
Assembler	1
Conventional 3GL	2
3GL using advanced 3GL features (e.g., COBOL II)	3
3GL using 4GL query language	4
3.5GL (e.g., APL)	5
Advanced 3GL using 4GL query language	6
Fully 4GL	7

5.2 Size

System size comprises both executable lines of code and number of users. Because executable lines of code have been highly correlated with other measures of system size, they will be used to proxy as an element for system size.[2] The IT manager supplied the number of executable lines of code for each system.

[2]All comparisons are made between languages of the same generation (COBOL, RPG, and Basic). Two COBOL systems contain some assembler calls, which contribute to the line count. However, the assembler component is considered insignificant. Comparing different languages of the same generation may account for some error.

Information about the number of users of each system was requested from the IT manager who supplied the data in consultation with the systems analysts. An integer giving the number of users of each system was supplied by the IT manager.

5.3 Age

System age in months was requested of the IT manager. Unfortunately, records of system age were unavailable prior the introduction of a system used to maintain detailed records of all systems. System age was supplied by the IT manager for all systems as either "new" or "old." New systems are those that were implemented after 1986. System age (AGE) was measured, therefore, as a dichotomous variable.

5.4 Period

PERIOD is a time-dependent integer that corresponds to the period in which the user change request was raised. Each period corresponds to an elapsed time of three months. There are 45 periods.

5.5 Data Collection Procedures

The mining company user change requests were supplied on magnetic media. There were three files: one containing the user change requests; a second containing descriptions of the systems; and a third containing additional descriptive information about the company systems.

5.6 Classification of User Change Requests

The classification of the user change requests was undertaken by four independent coders. There were two reasons for using four coders. First, classifying the 2,962 user change requests was a formidable task for one person. Second, using a minimum of two coders per system enhanced the reliability of the coding (also see the reliability of instruments subsection below). Table 3 shows their profile.

Table 3. Profile of Coders Used to Code Changes in Mining Company Systems

	Position	Education	Industry experience
Coder 1	University Lecturer	B.Com MIS	15 + years
Coder 2	Student	B.Com MIS	0
Coder 3	Student	B. Comp Sc MIS	0
Coder 4	State Manager of IT Company	B.Com MBA	15 years

5.7 Coding of Language of Construction

The coding of languages of construction, LANG, was undertaken by a panel of experts. The panel consisted of a professor of Information Systems, a lecturer in Information Systems, who had 15-years industry and academic experience, an IBM systems specialist, who had more than 15-years experience in IBM large-scale systems, and the researcher, who has in excess of 15-years experience in computer audit and large-scale information systems development.

6. DATA COLLECTION

The mining company was able to supply user change request data for 2,962 user change requests covering 40 systems. This data covered the period from March 1983 through to June 1994. Table 4 details demographic information relating to the systems.

Table 4. Demographic Information for Mining Company Systems

Code	Description	Age	Lines of code	Users	Number Changes	Language
ENM	ENGINEERING INFORMATION SYSTEM	NEW	474230	300	92	COBOL, QMF, ISP
FBD	DIVISIONAL BUDGETING SYSTEM	OLD	79390	N/A	23	COBOL, Clist
FCB	COMPANY BILLING SYSTEM	NEW	376278	138	6	Cobol2,QMF,REX
FCI	CAPITAL PROJECT INFORMATION SYSTEM	OLD	224421	60	61	COBOL, RPF,Assm
FCS	LABOR COSTING	OLD	64350	N/A	35	COBOL2, RPF
FGL	FINANCIAL STORES SYSTEM	OLD	187601	N/A	61	COBOL2, RPF
FMP	GENERAL LEDGER - MMS AND MASTERPIECE	OLD	353920	100	7	COBOL, RPF, Assm
FPL	ACCOUNTS PAYABLE: MILLENNIUM SYSTEM	NEW	223063	60	18	COBOL, RPF
FSC	STORES PLANNING AND BUDGETING SYSTEM	OLD	N/A	N/A	1	APL
MIL	REGIONAL PRODUCTION PLANNING SYSTEM	NEW	268458	10	5	COBOL2
PB*	REGIONAL BONUS SYSTEMS	NEW	66239	50	29	COBOL, SAS
PEO	GDS OFFICIALS PAYROLLS	OLD	97711	8	190	COBOL
SIR	REGIONAL REPORTS	NEW	89119	200	N/A	COBOL2, QMF, Clist
STH	STORES CONTROL SYSTEM	OLD	N/A	N/A	8	COBOL2, APL
STO	MATERIALS CONTROL	OLD	409130	300	425	COBOL SAS, Clist
TCS	H/O TRAINING COST SYSTEM	NEW	N/A	3	N/A	SAS
TEC	ALL OTHER TECHNICAL SYSTEMS	OLD	N/A	N/A	N/A	SAS, Fortran
TPS?R	NUMEROUS MABRA SYSTEMS	OLD	N/A	10	39	Mabra ,APL
TPS	DATA CAPTURE/REPORTING SURVEY/PLANNING	OLD	N/A	100	50	APL
TSY	ORE RESERVES	OLD	N/A	50	37	SAS, PL1, Fortran
TSY	SURVEY TRAVERSE(PEGS)	OLD	N/A	50	N/A	COBOL, SAS
TWO	WORKSHOP ORDERING AND CONTROL SYSTEM	OLD	120185	40	19	RPF, COBOL2
VOF	VALUATION OVERFLOW SYSTEM	OLD	N/A	10	18	APL
XHA	ACCESS CONTROL	NEW	48199	60	7	Basic, MASM
XHB	MANPOWER PLANNING	OLD	92616	350	54	COBOL
XHB	EXTRACTS INFO FROM HURIS CALC STD LABOUR	OLD	N/A	N/A	N/A	APL
XHC	TIME AND ATTENDANCE MANAGEMENT REPORTING	OLD	189000	10	131	COBOL
XHC	TIME AND ATTENDANCE	OLD	346498	170	N/A	COBOL
XHE	DAILY PAID, CPS, MONTHLY PAID STAFF	OLD	485800	353	498	COBOL
XHF	MEAL ISSUE CONTROL	NEW	37149	10	5	Basic, MASM
XHJ	MEDICAL ABSENTEEISM RECORDING SYSTEM	OLD	36319	20	7	COBOL2
XHT	TIME AND ATTENDANCE	NEW	53000	59000	66	RPG/400
XHU	PERSONNEL MAINTENANCE	OLD	715314	1135	475	COBOL
XOP	OFFICIALS PAYROLL	NEW	337561	353	47	COBOL2, Assm, KTOOL
ZER	COMPANY EQUIPMENT REGISTER	NEW	63581	15	N/A	COBOL 2,QMF,Clist
ZIM	DB2 INFORMATION MANAGEMENT SYSTEM	OLD	17364	350	N/A	Assm
HPT	MEDIS	NEW	N/A	N/A	136	N/A
MIN	MINING GRAPHICS	N/A	N/A	N/A	13	N/A
PEU	UNION WORKERS PAYROLL	N/A	N/A	N/A	173	N/A

N/A: Data not available

6.1 Preliminary Data Analysis

Table 5 shows summary data for the raw variables. Recoding of variables due to non-normality of the data was undertaken for a hierarchical regression.

Table 5. Summary Statistics for the Variables in the Second Study

Variable	Min	Max	Mean	S.D.	Skew	Kurt
AGE*	1	3				
DSPCT (DV)	0	100	49.53	16.93	-0.972	0.924
LANG**	2	5	2.756	1.035	0.85	-0.832
LOC	17027	715314	440044	205543	-0.367	-0.549
USERS	8	1135	470.5	387.07	0.958	-0.651
PERIOD	5	40	22.93	8.63	0.261	-1.146

*Binary variable. **LANG is continuous, but computed from an ordinal scale.

6.2 Data Screening

Data from 1,071 user change requests were discarded from the analysis because either (1) too few system changes existed for coding to be undertaken, or (2) the changes had been coded U or N (unable to be coded or not applicable). Coding undertaken by Coder 1 was chosen for the calculation of DSPCT because she had coded all systems considered appropriate for analysis. Systems with fewer than 20 user change requests were discarded from the analysis because of statistical concerns. A number of change requests could not be used for a variety of reasons, for example, some user change requests were used to cover a number of minor unspecified changes over a given time period. Other changes, such as "Provide a report as per attached," also could not be coded.

The stores control system, STO, was coded by all coders to validate the coding instrument and to determine the degree of inter-coder agreement. Note that Coder 1 had the highest level of agreement with all other coders for the coding of the STO system.

Table 6 shows the Pearson correlation matrix for the variables. The Spearman correlation matrix was also examined because some of the independent variables are not strictly continuous in nature. The signs of all correlations in the Pearson and Spearman matrices are identical, and the values of the correlation coefficients have similar magnitudes.

Table 6. Pearson Correlation Matrix for the Variables in the Main Study

	DSPCT	AGE	LANG	LOC	PERIOD
AGE	0.463 (0.000)				
LANG	-0.485 (0.000)	-0.400 (0.000)			
LOC	0.298 (0.000)	0.180 (0.000)	-0.329 (0.000)		
PERIOD	-0.076 (0.002)	-0.218 (0.000)	0.114 (0.000)	0.003 (0.908)	
USERS	0.322 (0.000)	0.138 (0.000)	-0.362 (0.000)	0.876 (0.000)	0.032 (0.199)

Some correlation coefficients in Table 6 are relatively high, raising the possibility of a multi-collinearity problem. Nevertheless, the collinearity diagnostic output from SPSS was examined and found to be within the criteria recommended by Tabachnick and Fidell (1996). Dimension 5 had the highest condition index of 22.11, below the recommended value of 30.

7. PRIMARY ANALYSIS

Standard multiple regression using SPSS was used for the analysis. The regression equation is as follows:

$$DSPCT = \alpha + B_1 LANG + B_2 AGE + B_3 PERIOD + B_4 SIZE + \varepsilon$$

where α is the intercept

B_1 to B_4 are the regression coefficients
ε is the error.

Table 7. Standard Multiple Regression of Language, Age, Period, and Size on Volatility Index (N = 1642)

					Collinearity Statistics	
	B	**Beta**	**t**	**p-value**	**Tolerance**	**VIF**
LANG	-5.818	-0.305	-12.947	0.000	0.729	1.372
AGE	10.958	0.323	14.477	0.000	0.809	1.236
PERIOD	0.051	0.025	1.2	0.23	0.945	1.058
SIZE	0	0.139	6.345	0.000	0.0841	1.189
(Constant)	30.696		9.74	0.000		

Adjusted R Square = 0.337

Table 7 shows the results of the multiple regression. All variables except PERIOD contributed significantly to the regression. Systems constructed from higher generation languages were associated with lower values of the volatility-index value (LANG, β = -.305, $p < 0.001$) relative to systems constructed with lower generation languages. This result supports **P1**.

A significant positive relationship between the size of a system and its volatility index was found (SIZE, β = .139, $p < 0.001$). This result supports **P3**—namely, that the volatility index will be higher for larger systems.

A significant positive relationship between system age and the volatility index was also found (AGE, β = .323, $p < 0.001$). This result supports **P4**—namely, that the volatility index increases with age.

No association was found between the time period (PERIOD) in which changes are raised and the volatility index. It seems that either the erosion of code quality has no significant effect on the effort involved in implementing user change requests or that the erosion of code quality is not happening to any significant extent.

8. DISCUSSION

Table 8 summarizes the results of hypothesis testing. These results are discussed below.

Support for **P1** indicates that the effort spent on addressing deep structure changes decreases in relation to total effort spent on all changes as the language of construction increases in sophistication. This result implies that the use of high-level languages increases the efficiency of deep structure changes more than surface structure changes and is an avenue for further research.

Table 8. Support for Hypothesized Relationships

Hypothesis	Variables	Expected Direction of Relationship	Support
P1	LANG	Negative	Yes
P2	Not tested		
P3	SIZE	Positive	Yes
P4	AGE	Positive	Yes
P5	PERIOD	Positive	No

Support for **P3** indicates this outcome is consistent with the notion that an increasing number of users is being associated with an increasing proportion of deep structure changes. Also, as lines of code is a good proxy for complexity (Kemerer 1995), the results suggest the average volatility-index value during evolution is higher for more complex systems.

Support for **P4** also supports the notion that systems construction methods and techniques are improving over time.

Lack of support for **P5** indicates that erosion of code quality is either not being manifest, or that improved methods of performing maintenance are compensating for this issue. Either way, this represents an avenue for further research.

This study is also limited to a single organization. In this light, its generalizability to other organizations, other sectors of the economy, or other countries should be treated with caution.

9. IMPLICATIONS OF THE RESEARCH FOR THEORY

This research results from extending the body of ontological research into information systems to the area of information systems maintenance. It illustrates that ontologically-grounded theory can be extended to the development and empirical testing of real world phenomena. An information systems volatility index has been developed that can be further used to extend the boundary of information systems theory to examine the behavior of information systems that are subject to a variety of factors and environments.

A number of areas for further research were uncovered during the empirical testing, and a number of associations that came to light in the correlation need to be further investigated (e.g., the relationship between the number of lines of code and the number of users $r = 0.876$). Work needs to be done to further refine how the volatility index can be used to assist in resource management. The issue of code quality erosion and how to compensate for it is another avenue for research.

10. IMPLICATIONS OF THE RESEARCH FOR PRACTICE

10.1 Management of Staff Functions Within Organizations

Given that management wishes to maximize efficiency and effectiveness, they should seek to motivate IS departmental staff to devote most effort toward tasks that address deep structure changes. First, IS departmental staff are usually the only personnel with the necessary skills and access to an information system's functional schema. Second, changes made by user departments are often specifically tailored to those departments and are not useable by other organizational groups. Third, because surface structure changes do not require changes to the functional schema, they can often be performed by non-IS departmental staff who have sufficient technical ability.

If IS staff focus on deep structure changes, a reduction in the volatility-index value should occur. For example, the use of fourth-generation languages should lower the volatility index by reducing the effort involved in making deep structure changes. This approach will free up IS staff for other tasks.

10.2 Selection of Tools and Techniques

IS management should identify tools and techniques that lower the value of a system's volatility index to allow IS staff to reduce the effort involved in addressing individual deep structure changes. This will extend the system's life span by being able to cater for changes that otherwise would not be addressed.

During the design and construction phases of the systems development life cycle, IS management need to consider incorporating tools, techniques, and languages that result in a lower base value of the system's volatility index during its evolutionary phase. This action will help in extending the life of the information system as well as assisting IS staff to maximize their productivity by focusing on additional deep structure and other changes.

10.3 Means for Testing Tools and Techniques

Most tools and techniques are directed toward improving the efficiency and effectiveness of IS personnel. If the introduction of new tools and techniques improves the efficiency and effectiveness of IS personnel, a reduction in the value of the volatility index should be observed. A reduction in the volatility index value following the introduction of new tools and techniques would indicate that the new tools and techniques have been effective. IS management should seek to adopt those tools and techniques that lower the volatility index the most. The utility of the volatility index lies in its application to improve productivity by the appropriate allocation of resources and tools. Their effects on the index can be assessed and monitored.

11. CONTRIBUTION TO KNOWLEDGE

This research has contributed to knowledge in three ways. First, it has extended the model of information system change developed by Heales (1998) and based on Wand and Weber's (1995) model of an information system. Second, it has provided further confirmation that different types of user change requests occur and that they can be classified validly and reliably using an instrument developed in the research. Third, it has shown how the effects of various system characteristics change the volatility-index value.

Finally, the research has provided a contribution in an area of information systems activity that is often overlooked – namely, information systems maintenance. Furthermore, the research seeks to improve return on investment by providing the basis for determining ways to increase the life span of an information system.

References

Arthur, L. J. *Software Evolution: The Software Maintenance Challenge,* New York: John Wiley and Sons, Inc., 1988.

Banker, R. D., Datar, S. M., and Kemerer, C. F. "A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects," *Management Science* (37:1), 1991, pp. 1-18.

Belady, L. A., and Lehman, M. M. "A Model of Large Program Development," *IBM Systems Journal* (3), 1976, pp. 225-252.

Bunge, M. *Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World*, Boston: Reidel, 1977.

Bunge, M. *Treatise on Basic Philosophy: Volume 4: Ontology II: A World of Systems*, Boston: Reidel, 1979.

Chomsky, N. *Aspects of the Theory of Syntax*, Cambridge, MA: MIT Press, 1965.

Date, C. J. *An Introduction to Database Systems* (4th ed.), Reading, MA: Addison-Wesley, 1986.

Dekleva, S. M. "The Influence of the Information Systems Development Approach on Maintenance," *MIS Quarterly* (16:3), September 1992, pp. 355-372.

Flaatten, P. O., McCubbrey, D. J., O'Riordan, D. J., and Burgess, K. *Foundations of Business Systems* (2nd ed.), Orlando, FL: Dryden Press, 1992.

Heales, J. "An Empirical Analysis of Information Systems Change," in *Proceedings of Sixteenth International Conference on Information Systems*, J. I. DeGross, G. Ariav, C. Beath, R. Hoyer, and C. Kemerer (eds), Amsterdam, December 10-13, 1995, pp. 352-353

Heales, J. *Evolutionary and Revolutionary Maintenance of Information Systems*, Unpublished Ph.D. Dissertation, The University of Queensland, 1998.

Kemerer, C. F. "Software Complexity and Software Maintenance: A Survey of Empirical Research," *Annals of Software Engineering* (1), 1995, pp. 1-22.

Lederer, A. L., and Prasad, J. "Nine Management Guidelines for Better Cost Estimating," *Communications of the ACM* (35:2), February 1992, pp. 50-59.

Lehman, M. M. "Programs, Life Cycles, and Laws of Software Evolution," *Proceedings of the IEEE* (6:9), 1980, pp. 1060-1076.

Lyytinen, K. "Different Perspectives on Information Systems: Problems and Solutions," *Computing Surveys*, March 1987, pp. 5-46.

Marche, S. "Measuring the Stability of Data Models," *European Journal of Information Systems* (2:1), 1993, pp. 37-47.

Martin, J., and McClure, C. *Software Maintenance: The Problem and Its Solutions*, Englewood Cliffs, NJ: Prentice Hall, 1983.

Olle T. W., Hagelstein, J., I. G. Macdonald, C. Rolland, Sol, H. G., Van Assche, F. J. M., and Verrijn-Stuart, A. A. *Information Systems Methodologies: A Framework For Understanding* (2nd ed.), Wokingham, England: Addison Wesley, 1991.

Saltor, F., and Garcia-Solaco, M. "Diversity with Cooperation in Database Schemata: Semantic Relativism," in *Proceedings of the Fourteenth International Conference on Information Systems*, J. I. DeGross, R. P. Bostrom, and D. Robey (eds.), Orlando, FL, 1993, pp. 247-270..

Swanson, E. B. "The Dimensions of Maintenance," *Proceedings of the Second International Conference on Software Engineering*, 1976, pp. 492-497.

Swanson, E. B., and Beath, C. M. *Maintaining Information Systems in Organizations*, New York: Halsted Press, 1989.

Tabachnick, B. G., and Fidell, L. S. *Using Multivariate Statistics* (3rd ed.), New York: Harper Collins, 1996.

Turner, J. A. "Observations on the Use of Behavioral Models in Information Systems Research and Practice," *Information Management* (5:3), 1982, pp. 207-213.

Tyre, M. J., and Orlikowski, W. J. "Exploiting Opportunities for Technological Improvement," *Sloan Management Review* (35:1), 1993, pp. 13-26.

Wand, Y., and Weber, R. "On the Deep Structure of Information Systems," *Information Systems Journal,* July 1995, pp. 203-223.

Willcocks, L. "IT Evaluation: Managing the Catch 22," *European Management Journal* (10:2), June 1992, pp. 220-229.

Willcocks, L., and Lester, S. *Of Capital Importance: Evaluation of IS Investments*, London: Chapman and Hall, 1992.

Wrigley, C. D., and Dexter, A. S. "A Model for Measuring Information System Size," *MIS Quarterly* (15:2), June 1991, pp. 245-257.