A study of flexibility and change in a Web-based Information System

Jason Peters, Guy Fitzgerald and Howard Harris

Department of Information Systems & Computing, Brunel University, Uxbridge, UB8 3PH.
Tel No: +44 (0) 1895 274000 Ext. 4862
Fax: +44 (0) 1895 251686
E-mail: Jason.Peters@brunel.ac.uk
E-mail: Guy.Fitzgerald@brunel.ac.uk
E-mail: Howard.Harris@brunel.ac.uk

Stream: Co-ordination
Topic: Systems development

Keywords: Data stability, Flexibility (Flexibility Analysis), Maintenance, Web development

Abstract:

Organisations today are inundated with pressures of change and transformation induced by environmental and external factors. In order to remain viable business processes and systems within these organisations have to continually change or be adapted, in order for them to remain competitive. Failure to adapt and change can lead to failure of both information systems and organisations as a whole. Within this context of dynamic change a study of change requests for a web-based system developed by a web-development company is analysed. Web based information systems are now an important and integral part of most corporate and small businesses and web-applications are often critical business systems. The findings of the study suggest that particular categories of change can be identified and that these can help to make ongoing and future systems more flexible. This study is part of an on-going research project that seeks to understand and enhance the flexibility of information systems such that they will be more adaptable and easy to change as systems evolve and change due to external influences. The paper discusses the implications of the findings with some conclusions relating to web-based and traditional development, and data and process stability.

1 Introduction

Organisations today are inundated with pressures of change and transformation induced by the environment and external factors (Nelson, Nelson et al. 1997). The implications of which mean that business processes and Information Systems (IS) within these organisations have to continually change or be adapted, to remain competitive. Unfortunately, IS are often asynchronous with the business, and is frequently described as being static in nature. Nevertheless failure to adapt and change will lead to major failings of the organisation or information system in question (Ewusi-Mensah, 1997).

In recent years, there has been a growing interest among practitioners and academics in new ways of introducing flexibility into the Information Systems Development (ISD) lifecycle. The holy grail of ISD is to build software without any imperfections, (fully tested, conforms exactly to its original specification, etc.). But even such a perfect piece of software needs to be flexible, as many of the real world needs lie unknowable in the future at the time the system is designed. The traditional sense of systems requirements do not consider the future requirements of the business within the system and are thus susceptible to changes and modifications.

The contention of this paper is to report on a study of change requests in a case study on a Web development company. The change requests were analysed according to the types of objects that they affected as follows: data, processes, interface, new and upgraded and other. This study seeks to leverage further understanding of maintenance and change and categorise ‘what type’ of changes occur and what effects they have. Beyond this the paper endorses the need for flexibility and the development of more flexible systems, i.e. more easily able to accommodate change. As part of this we need to know what is most likely to change in order to have the ability to specifically accommodate the change. Having knowledge of the future changes can be addressed within the design of a software system. Otherwise flexibility is too broad and becomes impractical as it requires the accommodation of all potential change.

The company and project presented in this paper works as an external supplier in a close long-term relationship with a client. The traditional literature would describe the relationship as in an outsourced IT/IS contract. However, as the activities centre around web based development, the description of the company/client relationship and approach is perhaps better described in web/E-Commerce terms as a part of a ‘value chain service provider’, working in a ‘value network’ in a ‘dynamic market relationship’ (Timmers 1999 p. 183).

The paper is organised such that a review of maintenance in ISD is presented and the position of the study placed. An overview of the case study sets the context under which the study was conducted (Section 3) and the research method and project specifics discussed in Section 4. Section 5 reports on the findings from the data collected that feeds into the implications and discussions formulated from the analysis phase (Section 6). Lastly a conclusion summarises the paper and highlights further research that is required on this initial study and findings.

2 Maintenance

Maintenance, the adding, modifying or changing of existing software, has long been a significant issue of information systems development. The cost of maintaining systems has usually been a larger proportion of overall development costs than estimated and tends to extend over the life of the project. Over the years figures for maintenance have been consistently high and remain high despite huge changes in technology and development practices. In terms of time and effort, Yourdon (1997), estimated that it can consume as much as 80% of development costs. A survey conducted by Guimaraes (1983) found that expenditure, including application programming enhancements, accounted for 53% of development costs. More recently a study by Palmer (1999) stated that operations, maintenance, and version upgrades accounted for 50% of the total life-cycle cost of phases, and Stephen (2000) reported a similar figure of 67%, and Dishaw and Strong (1998) found maintenance to be 70% of software development budgets.

Maintenance has been defined as the 'modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment' (Kemerer 1997). However, the maintenance phase is not necessarily a clear and concisely defined separate stage, as the traditional lifecycle models infers, i.e. a separate post development stage. Maintenance is not always part of a system development methodology (Avison and Fitzgerald 2002).

The role of maintenance conventionally confines itself to changes of an existing system. A finer grained form of categorisation is required about the subjects or objects of maintenance such as the data and process, the actual artefacts that need to be modified within a system.

Maintenance has not always been well defined but often it is categorised into three types/roles: *corrective*, *perfective*, and *adaptive*. (Lientz and Swanson 1978), (Lientz and Swanson 1978) (Pressman 2000). These are broad categories responding to traditional software development projects. Perfective maintenance is enhancing and modifying a system to respond to changing user requirements and organisational needs (Lientz and Swanson 1978) which has been estimated to consume up to 60% of effort in software development (Lientz and Swanson 1978). We are proposing that with the increasing functionality of Web development and the ever increasing demand for change, the conventional role of maintenance, that of corrective, perfective and adaptive do not encompass the capacity and need for flexibility. To overcome this weakness we have adapted the meaning of preventative maintenance (Chikofski and Cross 1990) as an additional fourth role. This new role aims to forward project the types of change required in the form of constraints and enablers either for new ISD or for additional requirements analysis on existing systems. Such a process thus recognises and prepares for the contingencies of change to happen, to facilitate in-built flexibility.

3 Case Study

The findings presented in this paper are a small part of an ongoing research study based on a case study approach. The case of concern here is based on the IT-development division of a small to medium sized enterprise (SME), referred to as B&C. They provide "effective communication solutions to some of the most highly regarded corporate and financial institutions". Its core business operations is to act as a design consultant company, providing 'solutions' to organisations seeking professional design to corporate brochures, annual reports, identity products and corporate literature, interim reports and new issues. B&C has an extended and established client base that is growing in size partly due to a market push into web based resources.

Due to customer needs/demands and government legislation relating to changes in the medium through which annual reports can be disclosed, B&C ventured into on-line market. Having employed IT professionals they now offer design, development and implementation of web sites, intranets and extranets, CD ROMS, interactive presentations and bespoke applications. They also offer consultancy on site hosting, support, maintenance and site marketing. As a multi-disciplined team working together they offer solutions across all platforms and media. B&C's role can be described as that of a supplier, providing design and IT solutions to a wider community (Clients) seeking specialised expertise in the development of web-based business systems.

The IT department within the company is a small team with roles that are solely dedicated to the development of web-sites (technical areas) and others that span the whole company needs (e.g. designers and contract specialists). The small, specific team (six) has remained stable, varying when extra help is required through the use of contract staff.

Initially B&C developed web-sites that could be considered uncomplicated, or as Curry (2000) identified 'brochure' development. Recently the company responded to increasing Web functionality demands from the clients, demanding more complex and critical to the process of the business, hence, the project size and complexity has risen dramatically. The increases in complexity of these new projects require complex back-end functionality. In addition this has inevitably led to the need to address the issues of 'change', illustrated by the following comment of a developer when interviewed:

> "The nature of web sites never remain the same, they are forever changing".

B&C traditionally used a waterfall model based methodology customised to their particular needs and the needs of the project. Different tools, techniques, models and methods were utilised in different parts of the application to be developed. This methodology however has proven difficult and inappropriate to use for more complex web application development as the rate of change experienced within the requirements of each project proved that such an approach did not fulfil the dynamic nature of web applications. Therefore the existing methodology has now progressed to a more prototyping approach and seems, thus far, to be proving successful. This

success was based on improvements on shorter development lifecycles that met original requirements more closely.

4 Research Method

Yin (1994) is generally acknowledged for setting the reasoning for the single case study, which may be embedded in an organisation. The Case study presented here is a part of the ongoing research project in B&C but addresses the particular element of maintenance and change in web development projects. The research is of an interpretive understanding of the maintenance practices seeking a detailed description and comparison of categories based on a previous paper (Fitzgerald, Philippides et al. 1999), as such it follows Yin's (1994) approach of 'Theory building' of confirming theory, challenge, or extend the theory, or even 'to propose alternative explanation' (Yin, 1994). Secondly, the case study is 'an empirical inquiry' and the boundaries between phenomenon and context are not clearly evident (Yin, 1994).

The data collection methods used is based on an examination of paper-based documents such as, notices, memo and e-mail and formal client change request documentation.

Three semi-structured interviews (twenty to forty five minutes each) were conducted with the developer, designer and project manager. Validation was conducted through triangulation of documental records, field notes and some supplementary interviews conducted post data collection stage via e-mail, telephone calls and site visits.

This paper focuses on one particular project (referred to as WF). WF was constructed for a large software development company catering for human resource specific applications, e.g. payroll systems. The web-site entailed a back-end database system, a content management system and an open forum (community area) as well as the standard brochure (See Figure 1 for relationships). B&C's feasibility study uncovered issues pertaining to the problematic nature of incorporating change requests into the developed system. It was recognised that a large site should contain the flexibility, efficiency and usability to its clients as well as a coherent (robust) design that aids in the systems maintainability.

The project was conducted on several broad levels that can be further subdivided. The first stage was that of the "Pitching" phase that involved several key management persons from B&C bidding for the contract. Once the contract was confirmed the following stage was the conceptual layout of the proposed site. This involved the project managers and sometimes the designer entailing in several meetings to ensure the correctness of the client requirements. The outcome of this stage was a list of requirements for WF (project brief). The next phase saw the beginnings of the actual site development; this involved the design of several web-site templates. Once a design was agreed upon, the final stage was the actual implementation by the developer of the site.

When the WF project was completed, B&C handed over the system to the clients, who then embarked upon a lengthy process of testing and system checking (in-house) to ensure that it met their requirements, this additionally led to a new list of 'requirements' in the form of changes. Over a period of several months (after

completion of the project), five incremental documents were produced from the client entitled 'Software Changes Outstanding'. The documents contained a list of change requirements broken into the relevant sections of the application to which it applied; for example products/special editions/education. The staff members at B&C colour coordinated each change request in order to clarify which member of staff it was applicable to. The software change document was considered as a long list of change requests and were not categorised under any specific 'change types' (see below).

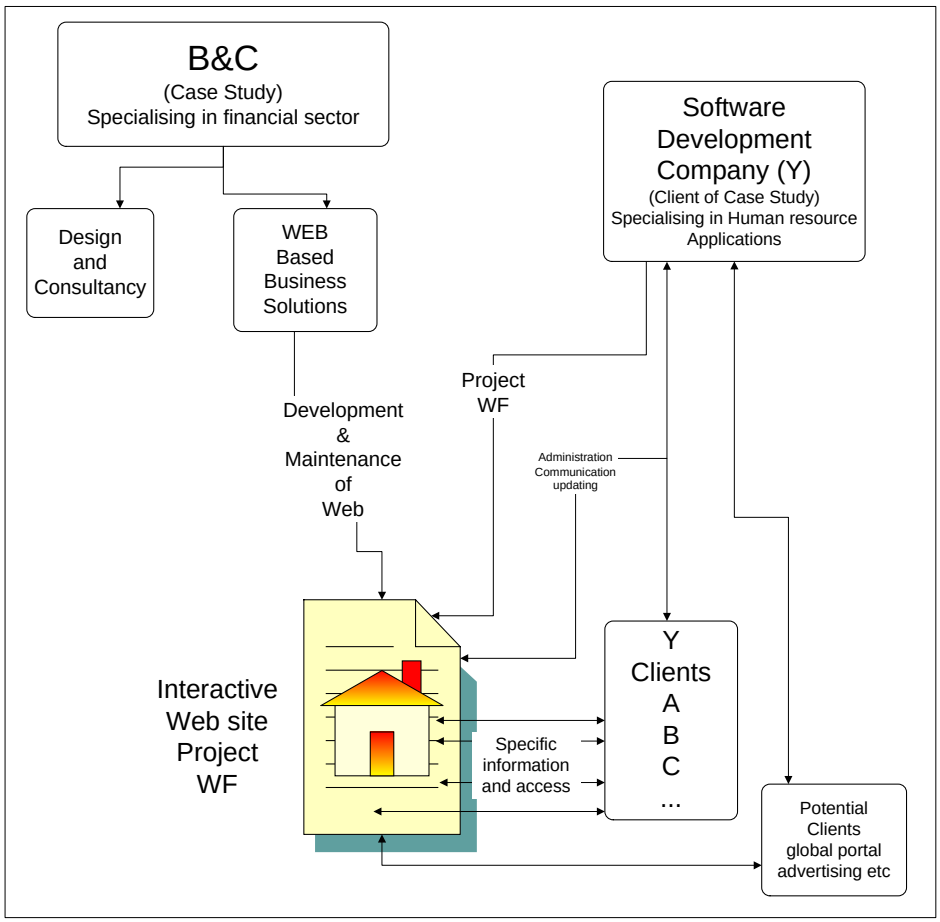

Figure 1: Relationship of the Case Study B&C to Client

This copy of the 'software changes' represented the main input document for this part of the study. The list consisted of over 210 change requests from the client and provided the basis upon which the researchers first categorised the changes and subsequently matched them to the category list of Fitzgerald, et al. (1999). The main areas requiring change were categorised as *Data* (e.g. new/amended fields in files or databases), *Processes* (e.g. logic/code changes, new procedures), *Interface* (e.g. user/systems interfaces), *New and upgraded* packages and *Other*.

The categorisation of change requests was spilt into two phases. The first phase involved the authors placing each change request into the categories (where a change was difficult to categorise, it was left for discussion and approval of the developer/project manager) that established initial results. In order to validate the initial results, the second phase involved the categorisation phase being repeated by the developer and project manger and it was found that there were discrepancies

between our findings and the results obtained by the developer/project manager results. The changes to the results were obviously cross-checked and modified where necessary until a final categorisation was achieved. Table 1 below presents a sample of five client 'change requests' and the categories to which they were attributed.

Change Request	Category
When viewed via the admin area, the add new forum section of the 'All forums' page now includes an 'active' checkbox to suspend a forum. However, this checkbox actually belongs on the 'Topic Summary' page for each forum so you can suspend, or re-state the Forum as required. There is no need for the option to suspend a new forum you are just creating.	**Data**
Any chance of including a date in the story title (in exactly the same way as helpdesk/FAQ's)? As with FAQ's it should be possible to edit the defaulted 'today's' date, or even remove the date.	**Processes**
Right hand margins in pop-up pages still too tight to the edge of the page. Good example is products/applications/human resource manager/absence and holiday manager link.	**Interface**
Several clients have reported 'problems' with opening PDFs on the site (e.g. the PDF won't open) which we believe are really due to problems with the speed of their own PCs, their Internet connection etc. One way to mitigate this perceived problem would be to display a 'progress' message, or preferably a small and unobtrusive icon, whilst PDFs are loading. Would this be possible?	**New and upgraded**
The browser 'back' button must be pressed twice in my IE5. This does not seem to be the case in later IE versions. B&C were to investigate this.	**Other**

Table 1 – Showing examples of change requests and their relevant category

Several issues emerged during the classifying phase; the first related to the complexity involved with each change request. Some changes required changes to be made to several elements of WF, for example a 'process' change may also entail an 'interface' change. This complex type of change was addressed by placing the change request under two (or more) relevant categories. Secondly, it was also identified that many of the change requests did not nicely fit into the change categories used as the initial framework for the study (i.e. those identified by Fitzgerald, et al., 1999) and thus had to be categorised under 'Other'. Thirdly it was difficult to calculate the exact effort required for each category due to the varying nature of individual change requests.

5 Findings

The following table (Table 2) summarises the results achieved from the categorisation phase of the study.

Main change Areas	No. of changes	(%)
		(Rounded-up)
Data	4	2
Processes	76	34
Interface	22	10
New and upgraded	8	4
Other	110	50
Total	220	100

Table 2 – Categorisation of changes made to the web site (WF)

From the above table it can be shown that *Processes* and the *Other* categories together were the highest request changes from the client, (84%) whilst Data, Interfaces and New and upgraded packages collectively came to only 16% in total.

This differential we now further examine by refining the 'Other' category (50% of the changes) into sub-categories that were established during the study (see Table 3).

Main change Areas	No. of changes	(%)
		(Rounded-up)
Graphical Interface	13	12
Content	81	74
Advise	10	9
Specification Errors	6	5
Total	110	100

Table 3 – Showing breakdown of Other category

It may be relatively contentious, (as there is no accepted defined list) but we feel that the categories shown in Table 3 fall mainly into the scope of 'non-functional' requirements of ISD. The breakdown of the 'Other' category is as follows:

- *Graphical interfaces* refer to the editing aspects of the GUI and was further refined to manipulation, major and minor.
- *Content* changes refer to the need to change information required on the web site itself. If the content is based on a content management system (CMS) then these changes are more easily accommodated, however if they are not then a long process of HTML coding is required. Content represented 74% of changes, inferring that the information within the website itself is most likely to change and will need to be adapted.
- *Advise* relates to changes that the user thought would be required but in discussion and with explanation are found not to be required, these might be misunderstandings and misinterpretations. Nevertheless they took time to be resolved although did not result in software changes.
- *Specification Errors* refers to those changes required because either the brief was misunderstood or not followed correctly or errors missed in testing.

There are key findings that emerge from these results that need further examination and analysis work, which we presented in the next section. In summary we can encapsulate our discussion by the following two findings. The category of Content (74%) found in 'Other' (50%) and Processes (34%) in relation to data (2%) category found in the main category.

6 Implications and Discussion

Comparison of case study findings

The analysis revealed several topics of interest. Firstly, that there was some confirmation and similarity with the findings of Fitzgerald's et al's. (1999) survey in respect of the category headings and the broad ordering of the categories. Table 4 below reflects these orders.

Main change Areas	Fitzgerald et al Study (%)	This Study (%)
		(Rounded-up)
Data	15	2
Processes	24	34
Interface	18	10
New and upgraded	16	4
Other	27	50
Total	100	100

Table 4 – Comparison of Data

Secondly, our initial analysis suggests that the smallest object that required changes was Data; although not large in either case it was particularly small in this study. This certainly needs further investigation nevertheless this may be due to the difference between traditional development (Fitzgerald, et al's study) and web development practice. Thirdly, 'Other' contains the sub category of 'content' (74% of 50%). In websites much of the data is contained in the content of the page itself whereas in traditional systems the data is usually all contained in files or databases. This might serve to explain the higher level of changes to data in the Fitzgerald study, i.e. they were all database changes whereas in the website some of the data changes are defined as page content changes. So possibly the difference is just one of semantics. It may imply that web designers should ensure that page content be in databases (e.g. CMS) or some other more flexible/changeable form.

Interface changes constituted only 10% of overall changes, although considerably small it was recognised that within Web development it is essential that the design GUI gets signed off and does not change as the cost and time associated with this amendment deters both parties. This is not to say that the GUI does not change especially within Web media, on the contrary, it was discovered that within the Other category Graphical Interface changes represented 12% (second highest form of change). Whereas Interface changes entail new design issues spanning the whole web site, Graphical changes relate to the altering of pre-existing designs. This was further sub-divided into manipulation (simple edit of graphic or size etc), major (requires designers involvement and coding) and lastly minor (requires designers involvement with no coding requirements). Fitzgerald et al (1999) study showed that 18% of changes were attributed to Interface changes.

Data and process stability

There has for some time been a debate about the relative stability of data and processes (Avison and Fitzgerald, 2002). In the main data (data structure rather than content) has been considered to be more stable, i.e. less likely to require changing over time, than business processes. So for example, in a university student registration system the entities of student, degree, modules, etc. is less susceptible to change than the processes of registering students and allocating courses and modules, etc. These processes, it is argued, change all the time. Thus, in systems development data is a better basis for building a robust system rather than basing it on processes. Indeed this is the argument made for the superiority of data based and data modelling approaches, such as Information Engineering (IE), etc. Indeed in the early database days the concepts of data based conceptual modelling based upon the ANSI-SPARC three level architecture was the norm. The conceptual schema was meant to be "a complete

view of the data requirements of the organisation independent of any storage consideration. That is the entities the attributes and their relationships" and it should be "a complete and accurate representation of the data requirements of the enterprise (Connolly, 1996). Of course in practice you need both data and processes but it is a question of which is the initial basis for design and in ISD there has long been a friction between data analysis and business process analysis, often considered as separate elements (Marche, 1993).

Although the debate concerning the relative stability of data and processes has often been invoked there has been relatively little data to support either contention. However, both the Fitzgerald, et al study (data 15%, process 24%) and this one (data 2%, processes 34%) seem to indicate that the changes are indeed more common in processes than data. (Although see above for the caveat concerning the very low data finding of 2%).

Flexibility

Maintenance and change are clearly issues that are of great importance in ISD. Although many different system development techniques and methods have evolved to support the management of complexity in the design process (Avison and Fitzgerald, 2002) the issue of change and the differential rates at which the technological progress are often displaced has not been widely addressed.

However, the notion of addressing flexibility, i.e. the ability to embrace change easily, is becoming increasingly important, especially in the business world. Ciborra (1993) suggests that; "Within a firm … what seems to matter is its flexibility and adaptive capability in the face of environmental discontinuities". The need for flexibility is frequently mentioned as a response to uncertainty (Land 1982; Fitzgerald 1990; Avison, D. E., P. L. Powell, et al. 1995; Kanellis, Paul et al 1996) and refers to the capability to adapt to changing conditions in order to remain viable. The flexibility concern is addressed across many disciplines ranging from the management literature, manufacturing, architecture, and organisational strategy and the design of flexible IS in business (Frazelle 1986; Veryard 2001; Fitzgerald and Siddiqui 2002).

The difficulty of flexibility is that it is not easy to achieve and secondly, information systems and information technologies are anything but flexible and adaptable (Fitzgerald, Philippides et al. 1999). Indeed information technology is often viewed as a barrier to change rather than an enabler of change (Fitzgerald and Siddiqui 2002). In 1990 the Economist suggested that "businessmen have discovered a … disconcerting problem: markets change, but computer systems do not" (The Economist, 1990), however Allen and Boynton (1991) considered this as the new challenge for business today of the dynamic nature of changing information requirements.

Fitzgerald has defined flexibility as "the ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed" (Fitzgerald 1990). However, flexibility is not amenable to a simple definition (Golden and Powell 2000) and therefore different perspectives exist. According to Rochester (1989), there are three broad views of flexibility in IS/IT. First, the business view represents the capability to build or adapt information systems in response too corporate changes. Rochester (1989) states "… we see system adaptability to changing business strategies as the most important definition of system

flexibility". Second, the user view of flexibility essentially means intuitive and adaptable interfaces. Application software should be intuitive to the user and thirdly, the ISD perspectives of portability, connectivity and maintainability. This study concurs with the latter element of Rochester's third perspective (maintainability). Although understated he describes the view of maintenance as a re-engineering phase, where IS are valuable assets to organisations.

Fitzgerald (1990) proposed a new lifecycle stage called Flexibility Analysis (FA). Although FA particularly addresses the initial stages of ISD it calls for a change in the conceptual thinking about the 'maintenance phase' of software development. A stage that attempts to identify and ameliorate the likely future changes that could impact a system (Fitzgerald and Siddiqui 2002). FA attempts to address the current 'snapshot' view of requirements in ISD and additionally identify requirements for future flexibility (Figure 2). The findings from this study would seem to support this notion and increases our understanding of 'what' changes in the maintenance/adaptation phase.

The current classification of maintenance (e.g. corrective, perfective and adaptive) does not address the 'what' elements of a system that changes at the initial conception analysis stage. The categories Data, Processes, Interface, New and Upgraded and Other (such as non-functional) potentially provide the constraints and enablers for new projects, or even the same project, that are under continual development, such as Web based projects.

This implies that more detailed attention is required to identify the type and nature of the changes that impinge on a system. This knowledge would also form a new kind of requirement constraint in the design at the design stage. First, to accommodate the change into the system and, if this is not possible, secondly to design the system in such a way as to make the addressing of the specific potential change relatively easy, i.e. by designing for flexibility.

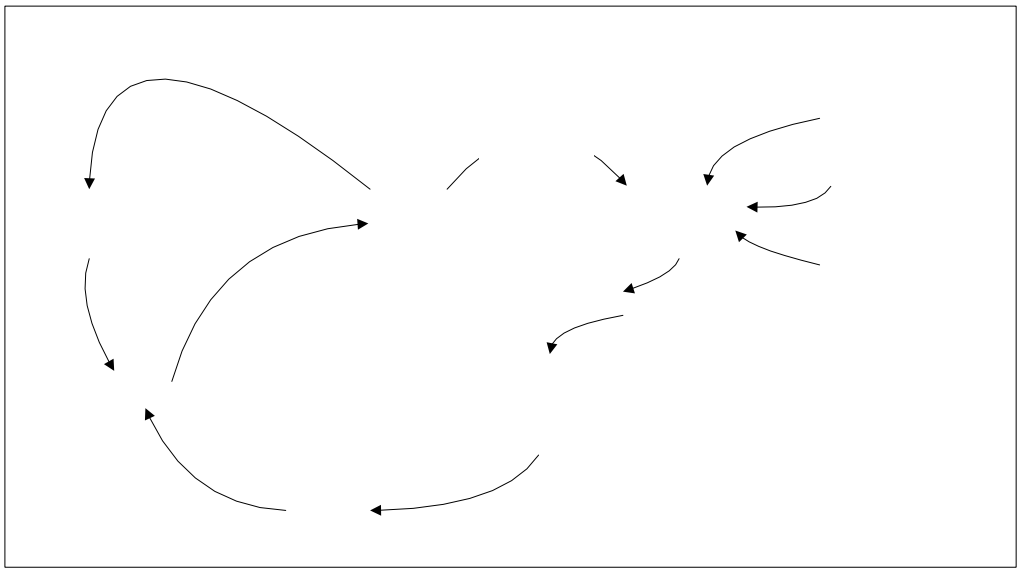

Figure 2: Feeding into Flexibility Analysis

For example, if a category is 100% efficient, that is no change is undertaken in maintenance (i.e. no identified constraints) it means that that category does not require flexibility. However if a category has maintenance changes of 34% as found in the process, then this implies and suggests that at the 'outset' there is a need in or before the requirements stage to accommodate change (flexibility) in the future.

Figure Two illustrates the positioning of preventative maintenance, the bridge to flexibility analysis and that of maintenance. This is an extension of a project-forward into the next iterative lifecycle; it feeds forward into new projects as in the form of constraints and enablers. This paper has examined the objects of maintenance under the current role of adaptive maintenance, finding that the conventional viewpoint only caters for a feed-back into the implementation stage, however we are suggesting that using preventative maintenance we would be able to ascertain which categories are subject to change, which need inbuilt flexibility. This would also have the additional potential benefit for future cost reductions of what is the most expensive part maintenance.

A third implication relates to the nature of web-based information systems compared to traditional systems. Web based systems seem to require a very high degree of change and appear to be more evolutionary than traditional systems. This might be because they are innovative and new and that in time they will settle down or it might be implicit in their nature. Either way it seems that maintenance and flexibility are even more important in this environment than ever.

7 Conclusions

This paper reports on a sub-part of an ongoing research project involving the web development company B&C. Specifically it reports on the investigation and analysis of changes that were requested by the client company in an ongoing web development project (WF). The development methodology adopted is essentially a prototyping approach with a special emphasis upon the provision of adaptive/enhancement maintenance in the post development stage. The findings from the analysis confirmed that the maintenance phase is not necessarily a concisely defined separate stage.

The traditional life cycle model and much of conventional IS practice has a separate post development stage, the conventional roles of maintenance (i.e. corrective, perfective and adaptive) only feeds back into the existing implemented product. By capturing and identifying what gets changed future project constraints and enablers can be identified and incorporated into flexibility analysis.

The findings from the study indicate that maintenance is alive and well and an important issue not just in traditional ISD but also specifically an issue in web-based systems. The study categorises the changes in the project into types taken from the literature and initially seems to support findings from previous studies. However, in this study the category of 'Other' was found to account for about 50% of the changes, i.e. perhaps the literature categorisations were not sufficient. The category of 'Other' was then sub-divided into four sub-categories, which was more illuminating. The identification of the categories provides input and ammunition for better design and the potential to make systems more flexible.

A number of implications of these findings are identified, including some reflections on the data vs. processes debate and which is most stable and the better basis for design. The study appears to support the case for the greater stability of data rather than processes. However, it might be that data needs to be thought about in a different way when used in a Web environment. Perhaps a new type of 'Data' description is needed indicative of a more dynamic move away from thinking about 'Data' as a series of entities made up of fixed conceptual relationships, contained in a Database, towards thinking more about 'Content' in a hyperlinked relationship.
The research study reported on here is at an early stage and further research is required, however, it is hoped that this study provides some evidence from a real and mature web development environment relating to the nature of required changes and the types of elements of a system affected. Clearly the study has limitations, for example it is just one project in one company. Nevertheless, within that context it provides what are rich but very rare (in that other studies of this kind are very few and far between) details and findings with some significant potential implications, even for the one company.

Acknowledgements

This work could not be conducted without the invaluable access provided by B&C and the numerous people that have participated in the study (especial thanks are due to Bruno Deivendranbose).

References:

Allen, B. R. and A. C. Boynton (1991). "Information Architecture: In Search of Efficient Flexibility." MIS Quarterly: 435-445.

Avison, D. E. and G. Fitzgerald (2002). Information Systems Development: Methodologies, Techniques and Tools. The McGraw-Hill Companies. London, 2nd Edition.

Avison, D. E., P. L. Powell, et al. (1995). "Addressing the need for flexibility in information systems." Journal of Management Systems 7(2): 43-60.

Chikofski, E. J. and J. H. Cross II (1990). "Reverse Engineering and Design Recovery: A Taxonomy." IEEE Software 7(1): 13-17.

Ciborra, C. U. (1993). Teams, markets and systems. Cambridge: Cambridge University Press.

Connolly, T., C. Begg, et al. (1996). Database Systems A Practical Approach to Design, Implementation and Management. Harlow, Addison-Wesley.

Curry, W (2000). The Global Information Society. Chichester. John Wiley & Sons.

Dishaw, M. T. and D. M. Strong (1998). "Assessing Software Maintenance Tool Utilization using Task-technology Fit and Fitness-for-use Models." Journal Of Software Maintenance: Research And Practice(10): 151-179.

The Economist (1990). How computers can choke companies. 315, June 9: 65-67.

Ewusi-Mensah, K. (1997). "Critical Issues in Abandoned Information Systems Development Projects." Communications of the ACM 40(9): 74-80.

Fitzgerald, G. (1990). "Achieving flexible information systems: the case for improved analysis." Journal of Information Technology 5: 5-11.

Fitzgerald, G., A. Philippides, and Probert, S. (1999). "Information system development, maintenance and enhancement: findings from a UK study." International Journal of Information Management 19: 319-328.

Fitzgerald, G. and F. A. Siddiqui (2002). "Business Process Reengineering and Flexibility: A Case for Unification." The International Journal of Flexible Manufacturing Systems 14: 73-86.

Frazelle, E. H. (1986). “Flexibility: A Strategic Response in Changing Times.” Industrial Engineering 18(3): 17-20.

Golden, W. and P. Powell (2000). "Towards a definition of flexibility: in search of the Holy Grail?" Omega The International Journal of Management Science 28(4): 373-384.

Guimaraes, T. (1983). "Managing Application Program Maintenance Expenditures." Communications of the ACM 26(10): 739-746.

Kanellis, P., R. J. Paul, et al. (1996). “An Archetype for Researching Information Systems Flexibility”. Proceeding of The Fifth International Conference on Information Systems Development, Gdansk - Poland.

Kemerer, C. F. (1997). Software project management : readings and cases. McGraw-Hill Co,

Land, F. (1982). "Adapting to Changing Requirements." Information & Management 5: 59-75.

Lientz, B. P. and E. B. Swanson (1978). "Discovering Issues in Software Maintenance." Data Management 16(10): 15-18.

Marche, S. (1993). "Measuring the stability of data models." European Journal of Information Systems 2(1): 37-47.

Nelson, K. M., H. J. Nelson, et al. (1997). Technology Flexibility: Conceptualization, Validation, and Measurement. Proceedings of the Hawaii international conference on system sciences, Hawaii, USA : IEEE Institute of Electrical and Electronics.

Palmer, J. D. (1999). Systems Integration. Handbook of Systems Engineering and Management. A. P. Sage and W. B. Rouse. New York, John Wiley & Sons Inc**:** 483-518.

Pressman, R. S. (2000). Software Engineering. McGraw-Hill. London, UK, 5th Edition.

Roschester, J. B. (1989). "Building More Flexible Systems." I/S Analyzer 27(10): 1-12.

Schach, S. R. and A. Tomer (2000). "A maintenance-oriented approach to software construction." Journal Of Software Maintenance: Research and Practice 12: 25-45.

Timmers, P. (1999). Electronic Commerce: Strategies and models for Business-to-Business Trading. John Wiley & Sons Ltd. Chichester,

Veryard, R. (June 2001). “Business Flexibility - A Framework for Understanding, Planning and Achieving Flexibility”. CBDi Forum.

Yin, R. K. (1994). Case Study Research: design and Methods. Sage Publications. Thousand Oaks California

Yourdon, E. (1997). Rise and Resurrection of the American Programmer. Prentice-Hall. Upper Saddle River New Jersey,