Intranet Development Processes in Large Organizations: Characteristics and Management Implications

Rens Scheepers
School of Information Technology, Swinburne University of Technology, Australia
Tel: +61 3 9214 5505, Fax: +61 3 9819 0823, rens@it.swin.edu.au

Volker Gruhn
Department of Computer Science, Dortmund University, Germany
Tel: +49 231 755 2782, Fax: +49 231 755 2061, Volker.Gruhn@uni-dortmund.de

Yun Yang
School of Information Technology, Swinburne University of Technology, Australia
Tel: +61 3 9214 5505, Fax: +61 3 9819 0823, yun@it.swin.edu.au

ABSTRACT

This paper highlights characteristics pertaining to intranet development processes in large, complex organizations. Some key differences between traditional development processes and intranet developments are identified. An empirical case of corporate intranet development is used to illustrate these differences. In the light of this, a number of managerial considerations in terms of intranet design, implementation, costing and control are outlined.

INTRODUCTION

We continue to witness the explosive growth in worldwide Internet use and many organizations are in process of establishing or expanding their Internet presence to harness the Web's electronic business potential. In terms of global connectivity the Web is a breakthrough, because it allows organizations to locate and disseminate information in a standard, user-friendly way across a variety of incompatible technical platforms and across geographical boundaries (Castells, 1996). At the same time we see many intranet developments based on the same technology within organizations.

An intranet can be defined as the application of Internet technology (and specifically the World Wide Web service) for prescribed communities of users. Well-understood and widely available Internet technology and standards (Web servers, browsers, protocols) are employed, but access is usually restricted exclusively to specified organizational members, often by means of firewalls, or even physically separating the intranet from external networks (firebreaks) (Riggins & Rhee, 1998; Damsgaard & Scheepers, 2000). By utilizing their intranet organizations have the ability to share information, collaborate and transact across various incompatible technical platforms and information systems, and across functional, structural and geographical boundaries within the organization (Bernard, 1996). Intranets are now also seen as platforms for organizational knowledge management (Davenport & Prusak, 1998).

Since the early days of computerization, much has become known about the development and management of computer-based systems development processes. Some well-known traditional software development processes are the waterfall model (Royce, 1970), evolutionary development such as prototyping (Brooks, 1975), the spiral model (Boehm, 1988) and so forth. However, the organizational application of Web-based technologies (such as intranets) is a more recent phenomenon and such well-defined process models for Web-based systems development are only evolving now.

We concur with others who argue that Web-based information systems differ from "traditional" information systems and that these differences raise new managerial challenges (Jarvenpaa & Ives, 1996; Isakowitz et al., 1998; Lyytinen et al., 1998; Eriksen, 2000; Strauch & Winter, 2001). We argue further that, due to the ubiquitous nature of the underlying technology and the collaborative nature of intranet-based development, new complexities emerge in terms of understanding users' requirements and knowledge of users' behavior. This is especially true in large and complex organizational settings.

The paper is structured as follows. We contrast some characteristics of traditional software development processes with that of Web-based and intranet developments in particular. We then illustrate how these characteristics manifest in a case of intranet-based development that we studied in a large telecommunication provider in South Africa. We discuss our findings and highlight some key managerial implications.

COMPARING TRADITIONAL AND WEB-BASED DEVELOPMENT PROCESSES

A number of research studies have contrasted differences of Web-based development processes when compared to more "traditional" systems development processes (e.g. Isakowitz et al., 1998; Lyytinen et al., 1998; Romm & Wong, 1998; Eriksen, 2000; Holck & Clemmensen, 2001; Murugesan et al., 2001; Strauch & Winter, 2001; Vogelsang & Carstensen, 2001). The differences mainly emanate from the malleable and ubiquitous nature of Web technologies, the application scope of such systems and the nature of the user/developer relationship in such developments. An exhaustive review of this literature is beyond the scope of this paper, but we highlight some key differences that pertain to intranet development processes in large and complex organizational settings.

User Community

In traditional systems development, developers and users are often typecast as separate communities and the relationships between users and developers are typically captured contractually (formally or informally) (Davis, 1993). Where the developers know the user community, user groups or individuals can usually be contacted directly, or when working with large numbers of users, developers may liaise with user representatives or conduct focus groups with selected users. In addition the user community may be delineated more formally and such arrangements typically manifest in the granting of user identifiers, passwords and profile assignments for specific systems/functionality.

In contrast, in Web-based development, the developer may find it quite difficult to learn about the exact needs and composition of the target user base (Oinas-Kukkonen et al., 2001). Due to the ubiquitous nature of the technology, essentially anyone with a browser and the appropriate access becomes a potential user and is able to "surf" to a particular website or Web-based application (Lyytinen et al., 1998). In large organizations this means that the user base may potentially be vast and very heterogeneous in terms of needs (Vogelsang & Carstensen, 2001). In intranet-based developments, many different "users", departments and functions within the organizations may also actively engage in their "own" decentralized developments (Bhattacherjee, 1998). This results in a situation where the clear distinction between "user" and "developer" roles becomes blurred.

Knowledge of User Needs

In traditional systems development, a number of development processes have been developed that seek to formalize the elicitation and analysis of user requirements and the identification of corresponding systems functionality. These often manifest in formal requirements specifications that are typically documented by means of established rules and notation (Heninger, 1980; Prieto-Diaz & G., 1991). The aim is usually to capture the exact user needs in the early stages of development, since the costs of addressing requirements at later stages can be very high (Sommerville, 1996). Traditional systems development process models seek to derive user requirements as accurately as possible, but requirements do change over time and systems need to evolve in response (Harker et al., 1993). While systems may already be in operation additional requirements may surface that need to be incorporated, typically during maintenance stages (Lientz & Swanson, 1980). In traditional systems development, vague user requirements may be addressed for example by means of prototyping or iterative development process models. Such approaches seek to elicit user feedback early during the process when it is relatively cheaper to incorporate systems changes.

Given the potential of a highly heterogeneous users base in Web-based developments, precise user requirements may be quite more complex to pinpoint (Oinas-Kukkonen et al., 2001). In this scenario it may be necessary to make assumptions about generic needs of users. In the case of intranets, precious little may be known about the user beyond that the person is a member of the organization with valid access.

User Feedback

One of the major goals in traditional systems development is to gain user feedback (even on design) as early as possible in the development process. In doing so, the risk of costly reworks in later stages can be reduced (Pressman, 1997).

In Web-based development, it becomes difficult to obtain design feedback prior to the actual launch of the system (Holck & Clemmensen, 2001). The highly interactive and content-rich nature of Web-based systems means that feedback is often cumbersome to obtain pre-implementation. Although website "mock-ups" may be used, much of the user feedback is only available post-implementation (after users have experienced the first release of the application). Usage statistics and user surveys are (post-implementation) mechanisms for obtaining user feedback in Web development processes (Huizingh, 2000).

Primary Responsibility for Functionality, Technical Infrastructure and Information

In traditional systems development processes the primary responsibility for the development of functionality and the underlying technical infrastructure of organizational systems often rests with the central IT section (see for example Lucas et al., 1990; Pressman, 1997). On the other hand, responsibility for the information captured in systems usually rests with the end users of the system (at least in terms of information quality). Thus, we see a separation of responsi-

bility in terms of use and development in traditional systems development. The exception to the rule is end-user computing where users assume some responsibility for development.

In the ubiquitous world of Web-based systems development, the clear distinction between "user" and "developer" roles becomes blurred (Lyytinen et al., 1998; Holck & Clemmensen, 2001; Vogelsang & Carstensen, 2001). Content development activities usually rest with users; however some "users" not only develop content, but also design websites and embedded functionality. Using advanced Web authoring tools, "users" may be able to create Web pages with embedded scripts, search functionality, interactive forms and even build applications that perform complex database transactions (Damsgaard & Scheepers, 2000).

Development Paradigm

The formal IT function most often assumes the responsibility for the development processes in traditional systems development (Humphrey, 1989; Pressman, 1997). Developers interact with users who in turn provide valuable requirements and feedback. Although some users may actively participate beyond merely supplying requirements (as in the case of end user computing), many of the traditional systems development process models are based on a central node of development responsibility.

The environment in which Web-based development takes place suggests a departure from this "centralized" paradigm. The ubiquitous development environment allows different user/developers to collaborate (formally, informally or even quite independent from each other) on various website developments. In intranet-based development the responsibility for developing infrastructure (e.g. search engines, building links with legacy systems, security mechanisms) often rests with the formal IT function whereas the responsibility for content and development activities are highly dispersed (Scheepers & Rose, 2001). This highly dispersed mode of systems development resembles a "federal" or collaborative development paradigm.

AN ILLUSTRATIVE CASE OF INTRANET DEVELOPMENT

In the light of the characteristics of Web-based development in general and intranet developments in particular, we reflect on the development process of the intranet at PhoneCo (a pseudonym). PhoneCo is a large telecommunications service provider in South Africa. The organization consists of a head office, service groups and regional offices in all the main centers of southern Africa. PhoneCo has about 70 000 employees. The organization had an extensive computing network infrastructure in place prior to the intranet development with 35 000 PCs connected to the company network.

Data Collection

We conducted a longitudinal field study using the case research method (Benbasat et al., 1987; Eisenhardt, 1989). Data were collected over a period of five years from 1997 to 2001. Formal and intensive rounds of data collection were interspersed with periods of informal data collection (Yin, 1989). An initial formal baseline study was conducted between December 1997 and January 1998. Formal follow-up interviews (mostly with the same interviewees) were held in November 1998 and again in January-February 2001. Data were regularly collected on a more informal basis by means of e-mail and phone contact, lunch meetings, discussions and by attending presentations made by the case organization (Darke et al., 1998). Almost 50 formal and informal interviews with various intranet actors were conducted over the period of the study.

Overview of the Development Process

Like many large organizations worldwide, PhoneCo started to develop an intranet during the mid 1990s. The PhoneCo corporate intranet could be seen as a formal response to numerous, spontaneous

and independent intranet efforts throughout the organization. Literally dozens of "island" intranets were created as employees throughout the various units and departments of the organization began experimenting with their "own" intranets during 1996/97. The PhoneCo corporate intranet (initially called "Content Book") was initiated in 1997 as an attempt by the central IT function to integrate and formalize these decentralized development efforts.

An intranet coordinator was appointed to oversee the ongoing development of Content Book. A senior executive actively resourced and supported the corporate intranet development and the intranet coordinator employed various tactics (such as posters, brochures to staff, an intranet café) to promote corporate intranet use in the organization. Content Book became the primary mechanism for locating information within the organization intranet (no central intranet search facility existed at the time).

The main feature of the corporate intranet was a structured list of information "topics" with hyperlinks to intranet pages elsewhere on the corporate intranet and at unit level, hence the intranet's original name. The intranet coordinator structured the topics and maintained the hyperlinks on an ongoing basis. Users began to rely on this manually maintained list of topics to locate information on the PhoneCo intranet.

Between 1997 and 2000, Content Book was "reincarnated" a number of times. For example, in April 1998 the home page was completely revamped and renamed to "The Station". During this time, the notion of the corporate intranet as the starting point from which to locate information throughout the organization was further embedded and search functionality was subsequently added. The corporate intranet evolved into a hierarchical design structure as opposed to an alphabetical listing. Between 1999 and 2000 these ideas were further developed and the Public Relations (PR) section of the organization became actively involved with the intranet. The PR section solicited inputs from employees as to the most popular intranet sites and in response the corporate intranet was again redesigned and re-launched, this time as the "Intranet Portal". The "Intranet Portal" was viewed by staff as the "corporate" part of the intranet and formed the apex of the larger organizational intranet. The corporate home page featured company news announcements, hierarchical menus/submenus to unit level intranets, popular "hotlinks" and organization-wide intranet search functionality.

Between August 2000 and February 2001 the PhoneCo intranet home page registered more than a million "hits" in total.

Current Architecture

The current architecture of the PhoneCo intranet is depicted in Figure 1. The "corporate" intranet is owned and maintained by the central IT and PR sections. Units (such as departments or other organizational functions) can either create their own decentralized child intranets, which may be linked to the corporate intranet. Alternatively, units can rent space on the corporate intranet itself in the "server farm". The server farm is a virtual Web server environment, where units can store and maintain their intranet content as if they physically maintained their own Web servers. The server farm provides security features, standard templates for units to use, backup facilities, tracking of usage statistics, etc. Intranet pages located here are automatically indexed for subsequent searching by the organization-wide search engine. For the more autonomous unit intranet configurations, manual intervention is currently necessary to index these locations for organization-wide searches.

The majority of the links emanating from the intranet portal are to mainly static electronic documents (e.g. the employee manual, products and services information, a guide to manage customer complaints). A few simple intranet applications are available via the corporate intranet including a tariff calculator to determine call costs, a time difference calculator and an application to log desktop problems.

User Community

The PhoneCo user community is basically everyone in the organization with access to the intranet. As such, the user base is large and

Figure 1: PhoneCo intranet system architecture
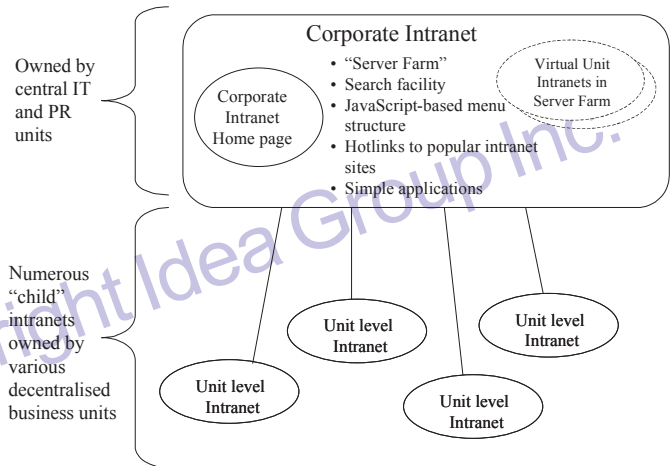
very heterogeneous. Users are located in different units and functional areas and are geographically dispersed. Some intranet "users" are also "developers" in their own right; hence the large number of decentralized unit level intranet efforts.

Knowledge of User Needs

For designers of the intranet (especially those in the central IT section responsible for the corporate intranet), the essence of the problem is that they need to design features "for the common good" of everyone in the organization. It becomes a problem who to ask and how to prioritize developments given the size and diversity of the user base. Unit level intranet developers complain of a similar problem; despite having a narrower target user base in mind for their child intranet site, the ubiquitous nature and organization-wide search facilities mean that diverse users from any part of the organization (outside their intended target market) may be using their site. At present PhoneCo intranet users are not required to log in. Hence little is known of intranet users and needs, apart from that they are PhoneCo employees.

User Feedback

The PhoneCo corporate development process reflects a "trial and error" approach where an initial attempt ("Content Book") was made which was then revamped over time in a number of quick successions ("The Station", etc.) until a more stable structure emerged that the users eventually got used to ("The Intranet Portal"). Although more content and applications were added to the intranet over time, the series of revamps largely dealt with the structure, functionality and user interface of the corporate intranet.

To overcome problem of vague user requirements, the PR section (since 2000) started to survey employees to determine general needs for intranet-based content. Employees are also regularly surveyed to establish their intranet use patterns and these inputs are used to help determine the menu structure, "hotlinks" and content on the corporate intranet home pages. It is currently technically cumbersome for PhoneCo to automatically track and index all intranet accesses (given the numerous "child" intranet sites which possibly use different Web-server platforms and software).

Primary Responsibility for Functionality, Technical Infrastructure and Information

Responsibility for both the technology development and content is shared between the corporate and unit levels. The central IT section takes most of the responsibility for the technical infrastructure (e.g. search engine, server farm, etc.). The PR section is primarily responsible for the corporate intranet content. For the larger intranet, the

user and developer activities are highly dispersed and user and developer roles are blurred with some intranet "users" not only developing content, but also sophisticated applications at unit level.

Development Paradigm

The development process for the larger PhoneCo intranet can be regarded as Collaborative/Federal. Responsibility is shared between the central IT and the large number of unit level intranet developers. PhoneCo has an intranet policy document in place which specifies usage and content standards, the general look and feel of intranet sites, content quality standards, etc. However, within the scope of the policy, each unit has a high degree of autonomy in determining their own intranet server platforms, server and authoring software, and intranet content and structure.

IMPLICATIONS FOR MANAGING INTRANET-BASED SYSTEMS DEVELOPMENT

The case illustrates some of the challenges faced by many intranet managers especially those in central IT sections of large organizations. These challenges can be attributed to the highly distributed nature of intranet-based system developments and the large, and often heterogeneous user communities that such systems tend to serve (an argument supported by Lyytinen et al., 1998). As such the knowledge about specific and individual end user needs is vague and it becomes necessary to make assumptions about generic requirements. The responsibility for development is distributed among a number of "child" intranet user-developers that not only contribute information content, but also develop functionality. As opposed to the centralized paradigm of traditional systems development, intranet-based development processes correspond more to a federal and highly collaborative paradigm. In these respects we highlight some of the implications for the management of Web-based development processes.

Implementation and Costs

We argue that it is very likely that intranets (and especially the corporate part of the intranet) will be developed and rewritten in a number of "trial and error" loops before it stabilizes. We further argue this is more so that with traditional evolutionary software development processes, because precious little may be known about the user community initially and feedback on design becomes only available post-implementation. As such it is difficult to develop comprehensive estimates until some initial prototyping has been completed done (Larsen & Bloniarz, 2000).

Accounting for the true development cost becomes complex, because it may accumulate in a variety of units in the organization. Apart from the development process, managers need also to be wary of unnecessary duplication of technical infrastructure and examine ways to reduce costs (e.g. centralization). Other components of the total cost that need to be monitored include development staff, ongoing content development and maintenance and costs associated with hosting the site (Larsen & Bloniarz, 2000). In terms of development staff, besides traditional software development skills, graphical and multimedia design skills also become important in Web-development processes and these costs need to be factored in (Lyytinen et al., 1998; Huizingh, 2000).

We thus caution managers against the belief that Web-based development is less costly than traditional system development. Subsequent development iterations can also cause costs to escalate.

Design Guidelines

Today the dominating architecture for distributed environments (including Web-based systems) is based on the client-server model. Based on a 3-tier-architecture, the front-end user interface, back-end servers and data repository can be separated (Lewandowski, 1998). Intranet managers should encourage developers to exploit such flex-

ible configurations to leverage their design (and cost) benefits. It is often the front-end that needs to be "reincarnated", while the back-end remains more stable. Data repository accesses should be highly generic in order to accommodate changes to front-end processes more easily. One approach to achieve this is to mandate component-based development to improve reusability and hence flexibility. In the same manner the interaction between the front-end (e.g. the Web browser) and the server can be made more flexible by mandating the use of generic components on the server-side that can be relatively easily tailored to accommodate changes.

Standards and Control

Developing Web-based systems in the Collaborative/Federal paradigm raises the issue of control of such efforts. When an intranet is left to "grow wild", poor systems quality and information chaos will be the likely results (Phillips, 1998). Standards and control are however more complex to address in intranet-based development processes. The managerial challenge is to balance the creativity and shared ownership that is associated with distributed development, with control and standardization to ensure overall quality. The nature of the challenge changes over time and intranet managers need to "know when to control and when to let go" (Scheepers & Rose, 2001).

Monitoring the quality of development and information content is more complex in the intranet-based development process where numerous decentralized developers and content providers may be involved. This raises the need for revisiting existing development and use policies and establishing "soft-touch" standards.

CONCLUSIONS

We highlighted some characteristics pertaining to intranet-based development efforts in large organizational settings. Intranet development differs from more traditional systems development and this requires fresh new approaches in terms of its management. The intranet managers' role should combine traditional development and implementation project skills, with facilitation and support skills to accommodate decentralized intranet efforts and to encourage the various intranet contributors to collaborate in a productive way.

Our observations point to a number of implications for the management of intranet-based development processes. Due attention needs to be paid to the costing of intranet developments. The iterative "trial and error" pattern associated with such developments can easily result in cost escalation. To avoid this, more flexible design alternatives such as the separation of front-ends and back-ends and the use of generic components need to be investigated. Since many organizational role players may be involved in a collaborative/federal paradigm, due attention should be paid to issues such as the unnecessary duplication of infrastructure, the control of quality of information content and standards governing decentralized intranet developments.

REFERENCES

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly, 11* (3), 369-386.

Bernard, R. (1996). *The corporate intranet*. New York: John Wiley & Sons.

Bhattacherjee, A. (1998). Management of emerging technologies: experiences and lessons learned at US West. *Information & Management, 33*, 263-272.

Boehm, B. (1988). A spiral model for software development and enhancement. *Computer, 21* (5), 61-72.

Brooks, F. (1975). *The mythical man-month*: Addison-Wesley.

Castells, M. (1996). *The rise of the network society*. Oxford, UK: Blackwell Publishers.

Damsgaard, J., & Scheepers, R. (2000). Managing the crises in intranet implementation: a stage model. *Information Systems Journal, 10* (2), 131-149.

Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal, 8*, 273-289.

Davenport, T. H., & Prusak, L. (1998). *Working knowledge: how organizations manage what they know*. Boston, Massachusetts: Harvard Business School Press.

Davis, A. M. (1993). *Software requirements: objects, functions and states*. Englewood Cliffs NJ: Prentice-Hall.

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Academy of Management Review, 14* (4), 532-550.

Eriksen, L. B. (2000). Limitations and opportunities of systems development methods in web information systems design. In Baskerville, R., Stage, J. & DeGross, J. I. (Eds.), *Proceedings of the IFIP 8.2 Working Conference, Organizational and social perspectives on information technology* (pp. 473-486): Kluwer Academic Publishers.

Harker, S. D. P., Easton, K. D., & Dobson, J. E. (1993). The change and evolution of requirements as a challenge to the practice of software engineering, *Proceedings of Requirements Engineering (RE93)*, San Diego, 266-272.

Heninger, K. L. (1980). Specifying software requirements for complex system. New techniques and their applications. *IEEE Trans. on Software Engineering, 6* (1), 2-13.

Holck, J., & Clemmensen, T. (2001). What makes web-development different, *Proceedings of The 24th Information Systems Research Seminar in Scandinavia*, Ulvik in Hardanger, Norway, 527-539.

Huizingh, E. K. R. E. (2000). The content and design of web sites: an empirical study. *Information & Management, 37*, 123-134.

Humphrey, W. S. (1989). *Managing the software process*: Addison-Wesley Publishing Company.

Isakowitz, T., Bieber, M., & Vitali, F. (1998). Web information systems. *Communications of the ACM, 41* (7), 78-80.

Jarvenpaa, S. L., & Ives, B. (1996). Introducing transformational information technologies: the case of the World Wide Web technology. *International Journal of Electronic Commerce, 1* (1), 95-126.

Larsen, K. R. T., & Bloniarz, P. A. (2000). A cost and performance model for Web investment. *Communications of the ACM, 43* (2), 109-116.

Lewandowski, S. (1998). Frameworks for Component-Based Client/Server Computing. *ACM Computing Surveys, 30* (1), 3-27.

Lientz, B. P., & Swanson, E. B. (1980). *Software maintenance management*: Addison-Wesley.

Lucas, H. C., Ginzberg, M. J., & Schultz, R. L. (1990). *Information systems implementation: testing a structural model*. Norwood, New Jersey: Ablex Publishing Corporation.

Lyytinen, K., Rose, G., & Welke, R. (1998). The brave new world of development in the internetwork computing architecture (InterNCA): or how distributed computing platforms will change systems development. *Information Systems Journal, 8*, 241-253.

Murugesan, S., Deshpande, Y., Hansen, S., & Ginige, A. (2001). Web engineering: a new discipline for development of Web-based systems. In Murugesan, S. & Deshpande, Y. (Eds.), *Web engineering, Lecture Notes in Computer Science 2016* (pp. 3-13). Berlin: Springer-Verlag.

Oinas-Kukkonen, H., Alatalo, T., Kaasila, J., Kivelä, H., & Sivunen, S. (2001). Requirements for Web engineering methodologies. In Rossi, M. & Siau, K. (Eds.), *Information modeling in the new millennium* (pp. 360-382). Hersey, PA: Idea Group Publishing.

Phillips, T. (1998). Members only Web. *Management Today*, 70-74.

Pressman, R. S. (1997). *Software engineering: a practitioner's approach* (International ed.). New York, NY, USA: McGraw-Hill.

Prieto-Diaz, R., & G., A. (1991). *Domain analysis and software system modeling*: IEEE Press.

Riggins, F. J., & Rhee, H.-S. (1998). Toward a unified view of electronic commerce. *Communications of the ACM, 41* (10), 88-95.

Romm, C. T., & Wong, J. (1998). The dynamics of establishing organizational web sites: some puzzling findings. *Australian Journal of Information Systems, 5* (2), 60-68.

Royce, W. W. (1970). Managing the development of large software systems: concepts and techniques, *Proceedings of IEEE WESTCON, Los Angeles*, 1-9.

Scheepers, R., & Rose, J. (2001). Organizational intranets: cultivating information technology for the people by the people. In Dasgupta, S. (Ed.), *Managing Internet and Intranet technologies in organizations: challenges & opportunities* (pp. 1-20). Hershey, PA: Idea Group Publishing.

Sommerville, I. (1996). *Software Engineering* (5th ed.): Addison-Wesley.

Strauch, B., & Winter, R. (2001). Conceptual web site modeling. In Rossi, M. & Siau, K. (Eds.), *Information modeling in the new millennium* (pp. 427-441). Hersey, PA: Idea Group Publishing.

Vogelsang, L., & Carstensen, P. (2001). Some central characteristics in Web-application development, *Proceedings of The 24th Information Systems Research Seminar in Scandinavia*, Ulvik in Hardanger, Norway, 331-344.

Yin, R. K. (1989). *Case study research: design and methods*. Newbury Park, CA: Sage Publications.

Related Content

Towards an Understanding of Performance, Reliability, and Security
Ye Wang, Bo Jiangand Weifeng Pan (2018). *Encyclopedia of Information Science and Technology, Fourth Edition (pp. 7588-7598).*
www.irma-international.org/chapter/towards-an-understanding-of-performance-reliability-and-security/184454

Fuzzy Rough Set Based Technique for User Specific Information Retrieval: A Case Study on Wikipedia Data
Nidhika Yadavand Niladri Chatterjee (2018). *International Journal of Rough Sets and Data Analysis (pp. 32-47).*
www.irma-international.org/article/fuzzy-rough-set-based-technique-for-user-specific-information-retrieval/214967

Assessing the Potential Improvement an Open Systems Development Perspective Could Offer to the Software Evolution Paradigm
James Austin Cowlingand Wendy K. Ivins (2016). *International Journal of Information Technologies and Systems Approach (pp. 68-87).*
www.irma-international.org/article/assessing-the-potential-improvement-an-open-systems-development-perspective-could-offer-to-the-software-evolution-paradigm/152886

Towards a General Theory of Information
Laura L. Pan (2018). *Encyclopedia of Information Science and Technology, Fourth Edition (pp. 4459-4469).*
www.irma-international.org/chapter/towards-a-general-theory-of-information/184153

An Efficient Intra-Server and Inter-Server Load Balancing Algorithm for Internet Distributed Systems
Sanjaya Kumar Panda, Swati Mishraand Satyabrata Das (2017). *International Journal of Rough Sets and Data Analysis (pp. 1-18).*
www.irma-international.org/article/an-efficient-intra-server-and-inter-server-load-balancing-algorithm-for-internet-distributed-systems/169171