

Running Head: DCIS810-DOYLET

Web-based Decision Support model  
for Software Developers accountability

<http://decision.blogspot.com/>

Final Report - January 25, 2004

DCIS 810 – Project in Decision Support Systems

Fernando E. Doylet

Graduate School of Computer and Information Sciences

Nova Southeastern University

Web-based Decision Support model  
for Software Developers accountability

Surfing the now frequent waves of technology changes, could be the most reliable way to stay afloat, providing more value to our customers, distancing from our competitors; but, how could we justify the time and effort required to do it?; to answer that question, this model is focused on the ethno-methodological view of accountability, which agrees with the phrase: decision-making speed can be increased with less vigilant information processing (Kennedy, 1992).

Following the lead of Customer Decision Support Systems (CDSS) that must be placed within the context of the other Web and Internet facilities available to a customer, and must also link to and employ generic Web facilities as necessary; like newsgroups, where customers can meet and communicate with others. (O'Keefe & McEachern, 1998, pp. 72, 74); this prototype is dedicated to Software Developers, so they can also benefit from the web.

The fact that all individuals have limited information processing capacity (Kennedy, 1992), is the source of the problems that may be solved by the Software Developers who creates or improves his own web-based Decision Support System (DSS), while experimenting the latest free tools available on the Internet, on his own or using this prototype. This model provides a learning environment, aimed to dynamically update the skills of Software Developers, while framing those skills with the ideals of continuous exploration of new tools, community support, and responsible accountability.

### *Problem Identification*

To clearly identify the problems that this prototype is trying to solve, this section presents the reasons for the traditional attitude of Application Developers, the ongoing adoption of management solutions that may be contra-productive, and the increasing sophistication and expectations of actual customers.

The known attitude of Applications Developers who want to learn only as they need the information (Nykaza et al., 2002, p. 138) is related to the traditionally slow and little effective means of creation and transfer of knowledge, which needs to be supplemented with more efficient electronic means (Alavi & Leidner, 1999, p. 7), like the popular tools found on the Internet.

Recognizing that the human brain is subject to two limitations: limited information-storing capacity, and limited retention capacity, Decision Support Systems (DSS) attempt to broaden managers' limited rationality by adding the computer's capacity to that of the limited human brain (Turban & Meredith, 1981, p. 605). Software Developers use their own special development environment tools, as their DSS, complemented by the internal and external documentation produced for the software they develop.

Like other business decision-makers, Software Developers suffer from information overload and obvious under-utilization of large quantities of relevant information (Tegarden, 1999, p. 8), which is causing Knowledge Management Systems (KMS) to begin to appear in organizations (Alavi & Leidner, 1999, p. 2), or DSS that may be constraining productivity (Tegarden, 1999, p. 8).

Meanwhile, Companies are exposed to "an increasingly sophisticated customer base that demands a higher level of immediate service across multiple access channels." (Pan & Lee,

2003, p. 95). With the new channels found on the Internet, "customers are better informed, more demanding, and likely to be less loyal as their expectations are increasing faster than businesses can deliver" (Pan & Lee, 2003, p. 96).

### *System Justification*

The usually isolated development of Software, is increasing the communication gap between traditional Software Developers and most of their Clients now browsing the Internet. Helping or motivating each Developer to build his or her own DSS, using free and popular web tools, could improve their relationship with their clients, and will provide them with more resources for better accountability.

Promoting an open and customizable DSS model on the Internet, Software Developers will be conveniently exposed to many contemporary web tools and their unique characteristics; that should continuously enhance their skills, productivity and accountability; moreover, presenting them with the opportunity to be part of a virtual community with other Developers, could motivate them to communicate with [or to help] each other.

Gaining confidence on the Internet is another purpose of building this model. The Internet contains just about anything you need to know ...if you know where to look, while business insiders offer a closed box type of help (Stapleton, 2003, pp. 35,37). Establishing a personal DSS, each Software Developer will be able to maintain and improve a customized environment, with documents or links directly related to the knowledge and skills he requires, helping him in the continuous discovery of the evolving information super-highway.

A better understanding of the emerging Web Information Systems (WIS) what is the Information Systems constructed using Web Technologies (Rossi et al., n.d.) will also result

from experimenting with this model. A clear difference between a set of Web pages and a WIS is that WIS supports work, which pedagogically have a big impact on what we teach and how we teach it, creating fundamental questions that should occupy academics (Bieber et al., 1998, p. 415, 417-418).

The centralized main page of this prototype is a web-log (blog), and two other types of blogs are included as practice tools, because of their valuable auto-archiving feature; more than a million blogs populate the web, with thousands of new blogs added every day; this new web format has already had an important impact on online journalism as well as education (Schroeder, n.d.); moreover, blogging software doesn't require an extensive infrastructure, and it is expected that in the near future mobile blogs won't just save time and effort, they may save lives (Cohen, 2003).

Last but not least, seeding a continuous work relationship with the Internet, will motivate Software Developers to stay in tune with popular de-facto standards, to better satisfy their Customers while maintaining consistency with other interaction channels (Pan & Lee, 2003, p.95); those are the reasons why this prototype is conceptualized as a practice environment or learning guide, that sharpens Software Developers' skills, to exploit fairly new web systems, enhancing the way Developers do their job.

### *Users Give and Take*

The potential users of this prototype are Software Developers at any level, like professionals, students, amateurs, or any other computer system troubleshooter; that is how the target user of this prototype is defined, its content selected, and the way it is promoted.

Any user of this prototype may provide:

- a brief or detailed explanation of a problem or situation;
- a document or address to obtain feedback;
- voluntary contributions of practice content, time or tools.

In return, those users may obtain:

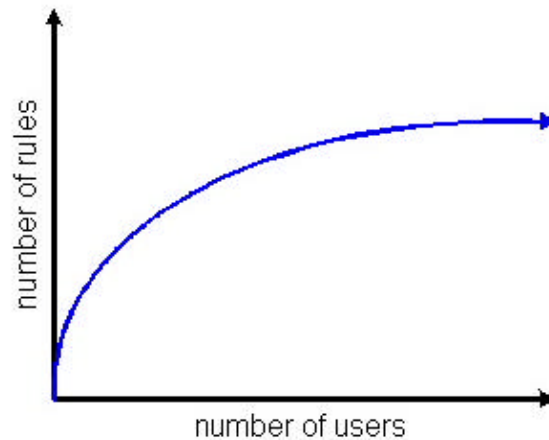
- practice with any tool available, for familiarization or discovery;
- reference to more appropriate resources;
- feedback from other user(s) of the system.

Process improvements involving shortening the proposal time for client engagements, saving time, improving project management, increasing staff participation, enhancing communication, making the opinion of plant staff more visible, reducing problem-solving time, better serving the clients, and providing better measurement and accountability (Alavi & Leidner, 1999, p. 19) are benefits that easily outweigh the effort of considering a KMS. Similar benefits could be obtained from experimenting with this prototype; including staff participation and visibility and measurements, if a private virtual community is built for those purposes, between Developers and co-Workers.

#### *Decision Criteria and Alternatives*

This prototype offers immediate exposure to the user that decides to start experimenting with the tools available, giving access to temporary account identities and passwords; since almost every testing becomes an instant contribution to the system, visible to other visitors, the calculated risk of abuse is compensated by the opportunity of demonstrating the benefits that those tools may produce.

If the user likes the experience, he will come back for more; or may decide to join the community or list-group, to receive communications or updates. As the number of participants and community members grow, administrative decisions are needed to delete inappropriate contributions; those decisions will become rules (see Figure 1).



**Figure 1. Administrative Controls**

For example, and to apply common sense, the following rules are followed...

#### Startup Rules

- to avoid waste of time:
  - ❖ any unclear request will be rejected.
- to ensure a way to send feedback:
  - ❖ all questions or requests must contain email or website address.
- to comply with the laws of the land:
  - ❖ personal or institutional attacks are deleted;
  - ❖ communications about (or links to) illegal activities, are deleted.

Likewise, the following alternatives are recommended to improve user's satisfaction...

- ✚ time sensitive problems could be referenced to more appropriate resources or Experts;
- ✚ unclear problems could find clarification from other user(s) in the community;
- ✚ a recommended solution may require the experimentation with an available tool;
- ✚ most problems need to be defined with focus and detail, to understand how they could be solved;
- ✚ for some problems, finding the appropriate resource, source of information, or subject matter expert, is essential;

Technology allows us to interact across time and space; the effectiveness of those interactions depends on our ability to understand each other's contexts, because the de-contextualization of information is a major cause of the difficulties often associated with electronic media (Friedman et al., 1999); anticipating those misunderstandings, the administration of this model focus on accountability, to narrow its scope, and purposely shift its views from limits to possibilities (Kappel, 2000).

### *Focus on Accountability*

As distinct from many other inanimate objects, computers perform tasks previously performed by humans in positions of responsibility; that make humans responsible for the computer actions; additionally, the property rights claimed over computer software by declaring ownership, produces accountability, considered conveniently distinct from the liability to compensate. If we consistently respond to complains by not pursuing blame and responsibility, we are effectively accepting a general erosion of accountability; on the contrary, a community that insists on accountability, in which agents are expected to answer for their work, signals esteem for high-quality work (Nissenbaum, 1994).

Each and every one of us needs to take on the challenge of personal accountability, in a broad, communal sense, for our actions; consider that accuracy in an account may be achieved by relaxing precision, to better understand context. In ethno-methodological studies, accountability is a central concept for understanding how people organize their everyday actions and interactions (Eriksen, 2002).

According to ethno-methodology (the study of the ways in which people make sense of their social world), we perceive our social world through a series of patterns we have built up for making sense of, and coping with, the variety of situations that we encounter everyday; those taken-for-granted patterns are to some extent necessary, in order to avoid confusion; suggesting that all meanings are subjective, ethno-methodology have the concept of Indexicality: the way people make sense of a remark, sign or particular action by reference to the context in which it occurs (Poore, 2000).

In simpler words, explaining the context of every situation, with relaxed precision, produces more accurate accountability. We are accountable to all with whom we exchange promises or expectations (Goethel & Gabler, 1999), which causes many of our problems and frictions with others; accountability provides one significant means through which damage, caused by awry interactions, can be repaired and trust reestablished (Friedman & Grudin, 1998).

Accountability, defined as the requirement to justify one's judgments to others...

- is an important element of the auditing environment;
- motivates cognitive effort, which translates into better performance, improving self-insight;
- sensitizes people to shortcomings in the views they hold, and to alternative ways of viewing behaviors or events;
- mitigates over-attribution;
- may affect both the process and the outcome - how and what people think;
- improves judgment consistency and consensus (Kennedy, 1992).

This ethno-methodological view of accountability is the focus of the prototype here presented.

### *Problem Solving Methodology*

Using free Internet tools, an interactive interface was built, to guide the user efforts with efficiency; the resulting interface is considered a DSS for Software Developers, to be used to reduce the complexity associated with finding information, the tools, or the support Developers may need (Grenci & Todd, 2002, p. 66, 71)

Simplicity is extremely important to obtain an acceptable download time, avoiding multiple scripts or graphics, while focusing on the decisions to be made by the Software Developer (O'Keefe & McEachern, 1998, p. 78, 77); it is also important to maintain the awareness that different people learn, think, and solve problems in different ways (Tegarden, 1999, p. 12).

Setting the components of the main page in a reasonable structure: from reliable fixed code to unreliable scripts [in loading order] from top to bottom and from left to right (plus a soft combination of fonts and colors to help their visualization) making the parts perceptible to the mind or imagination, allows Software Developers find the information without unnecessary waste of time or overload (Tegarden, 1999, p. 6, 9).

Since ‘how you communicate the case for change’ matters, it's usually necessary to give the information in a number of different ways; that is the reason why some important links are repeated through the main page of this model; this prototype also engages visitors in coming up with strategies for making the vision real (Kappel, 2000), allowing three ways to send contributions to be seen on the main page, apart from the options to send content to [or interact with] any of the other tools available.

This model accounts for the detailed explanation and relationship between the tools available, the assumptions and limits of the model, and plan for easy maintenance (O'Keefe & McEachern, 1998, p. 78). Open communication methods are included, assuming that easy feedback will help us improve other assumptions; most limits of this model are imposed by each tool provider, and the scope focused in our target users and goal; maintenance will be shared and delegated to users moving towards higher levels of membership.

This solution proposes that Software Developers continuous learning and use of the latest web technologies, provide opportunity for redundant organization and storage of information and knowledge, new skills, and improvement of accountability. Focusing the model to deal with accountability, motivates Software Developers to explore the possible ways to support their multiple activities, with the skills they may obtain.

Like the International Computer Driving License (ICDL), this prototype establishes a model for computer skills education and training, to provide flexibility to the workforce that can adapt as their roles change and their employers develop, facilitating increased mobility across the workforce and wider community; it is an ideal way to improve and consolidate knowledge, to build confidence and to improve both productivity and decision making (ECDL, 2003).

It is expected that some visitors to the system will join the community, and others will experiment with the tools freely available; each visitor with a problem, will choose to look for [or experiment with] a possible solution, identify a more appropriate source of information, or share the problem to obtain feedback.

Problems can be shared in many ways:

- sending a message to the community or list-group;
- publishing the details on the body of the main page web-log;
- writing a short question or request on the chat area (Open Comments) of the web-log;
- sending a private message, to be shared with the community or list-group; anonymously if needed.

Experience shows that real commitment can only occur when individuals are given a rich and full opportunity to explore and understand the implications of the change for them (Kappel, 2000); therefore, to motivate users' retention, membership levels are structured and announced to each new member in the community list-group, including options, privileges and requirements; likewise, step-by-step instructions will be produced, to facilitate the progressive understanding of all the tools available, while obtaining rewards in the form of higher membership levels, from regular user, to power user, to administrator.

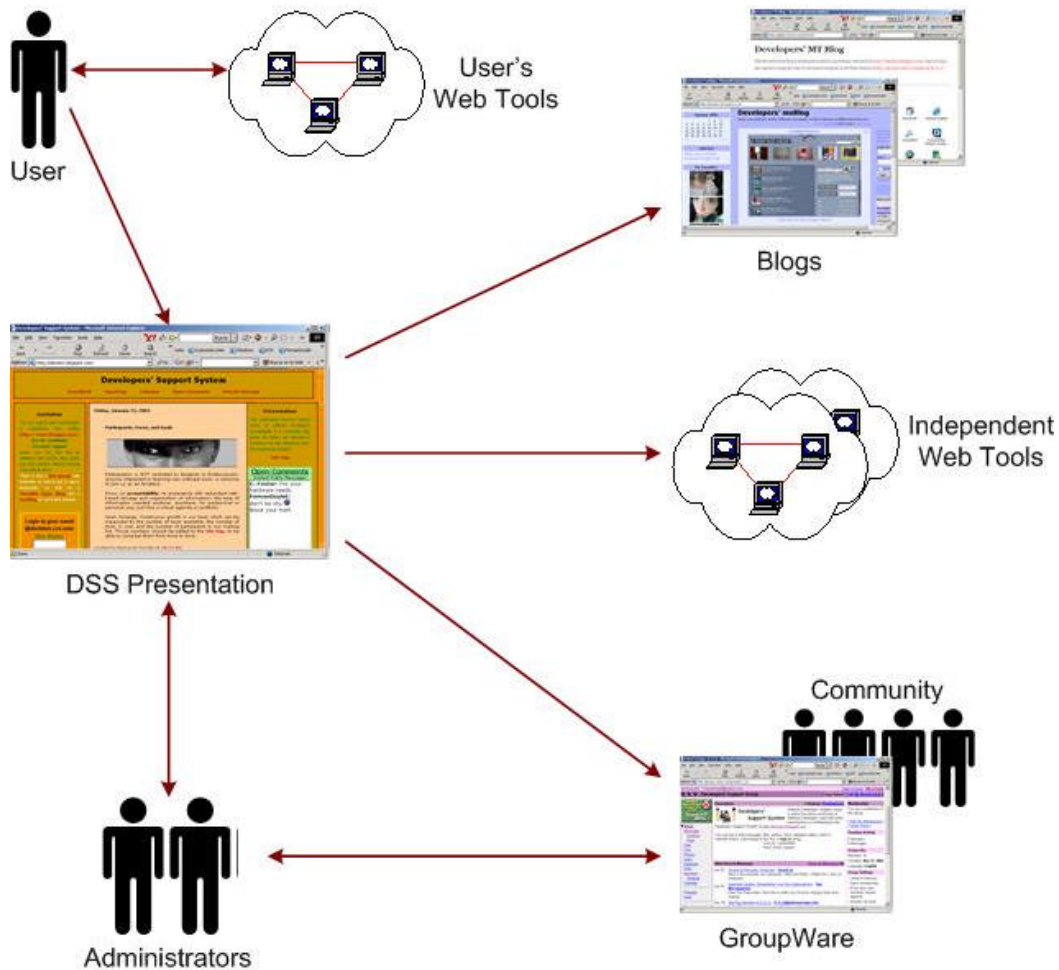
This model is presented in a way to demonstrate immediate value for Software Developers, to motivate them to experience the tools by themselves, with the temporary identities provided; the intention is to create genuine personal interest, instead of ‘pushing’ the valuable information to their desktop, to promote the conversion of the information provided into knowledge (Alavi & Leidner, 1999, p. 23-24). That is the reason why Software Developers were chosen as the target users, to demonstrate benefit with examples of their interest; with a little persistence, this model may change information sharing behavior and communication patterns (Alavi & Leidner, 1999, p. 7)

As a targeted Customer Decision Support Systems (CDSS), two modeling problems were encountered: the temptation to provide a very broad model requiring lots of data, and assumptions concerning the underlying assumptions in the model (O’Keefe & McEachern, 1998, p. 78). To counter those problems, the prototype was structured in a modular design, with a centralized portal or main page that summarizes the content of the whole system, with clear strategy and goals, allowing feedback, and giving room for organized growth.

While there are no universally agreed-upon method for representing an specifying DSS models, the repeated use by end users represents the final authority on the value of a DSS (Gregg & Goul, 1999, p. 93-94, 91); so the presence of this model on the web, and the tracking of its content and users, will provide the necessary feedback to obtain and maintain its synergetic goal of continuous awareness, to improve accountability.

### Design Specifications

In perspective, the prototype design is shown and explained in its centralized portal, which is a Main Page that routes each visitor to their preferred destination, while informs about the content, purpose and status of each of the many parts of the system (see Figure 2).



**Figure 2. Developers' Support System Design**

The second most important section of the prototype is the GroupWare, also called List-Group, Mailing List or eGroup, where each Visitor may decide to join, to become a Member of the Community with other previous Visitors; this GroupWare includes five storage tools and three interactive tools, to benefit the development of the Community. “Internet-based groupware

allows experts and end users to collaborate in managing and accessing knowledge and data resources” (Sedbrook, 1998).

Two specialized Blogs and five Independent Web Tools are also part of this prototype, and the list will increase as the Community grows and some of its Members reach the level of Administrators.

#### *Hardware Components Required*

Any desktop or laptop computer with an Internet Browser and connected to the Internet, should be able to access this prototype.

#### *Software Components Required*

Microsoft Internet Explorer version 6 is preferred, but any other Internet Browser will work with their own different interpretations of portions of the code; for example, using Netscape 7.1 the interface background looks different, but works the same ; to experience all the functions of the model, JavaScript, graphics and popups should be allowed.

#### *Databases and Expert Systems*

Most web-based tools integrated in this model work with their own Databases, to store and organize their content, users and attributes; their logical data models are usually hidden from the users.

Some of the web-based tools included in this model have Expert Systems functionality in their Help System; specially the GroupWare, also known as List-Group, Mailing List or eGroup. Like the databases, there are no internal details available to explain their knowledge base.

### Prototype Presentation

The prototype is at <http://decision.blogspot.com/> with links to all the tools available, as shown in the Site Map included in the presentation on its top right side (see Figure 3).

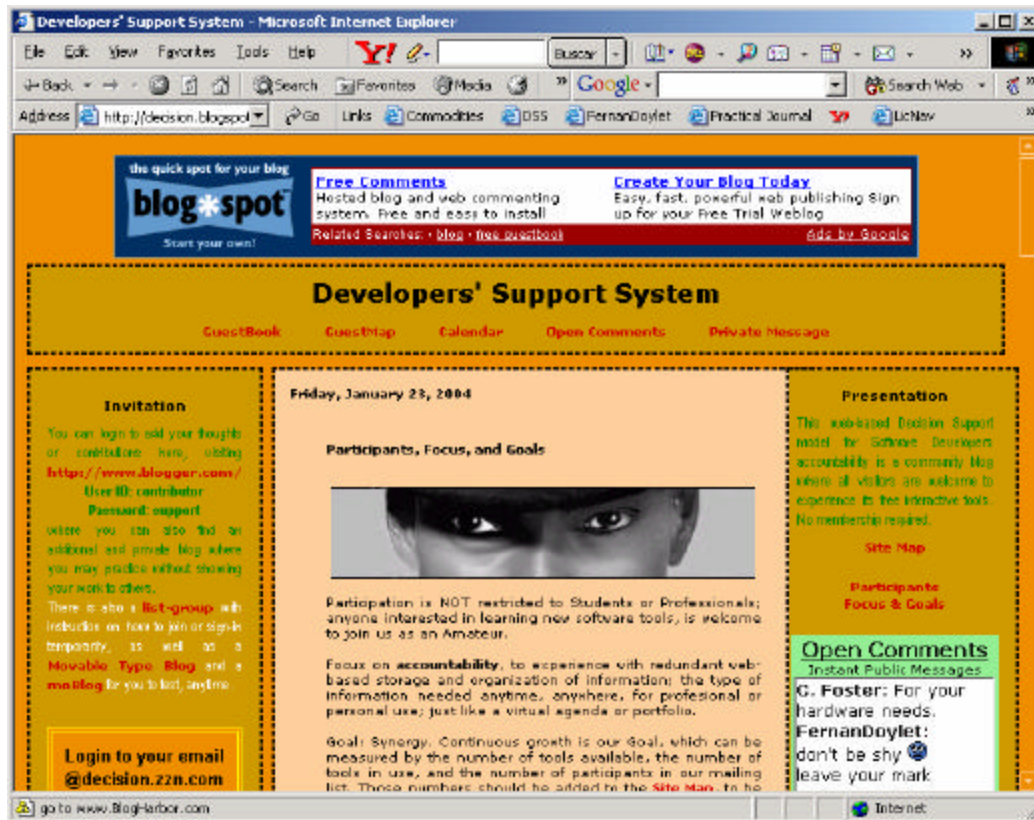


Figure 3. <http://decision.blogspot.com/>

If for some unexpected reason, the prototype is temporarily unavailable or unreachable, try <http://www.doylet.org/NSU/> to reach the Site Map, and a list of the Design Components with links to the other tools available.

### System Tests

Most tools have been tested with the Administrator identity, as well as the restricted Temporary User identity, sending initial content to make each Visitor comfortable with the system and aware of the expectations.

## References

- Alavi, M., & Leidner, D. (1999). *Knowledge Management Systems: Issues, Challenges, and Benefits*. Atlanta, GA: Communications of AIS Volume 1, 1999 Article 7
- Bieber, M., Conger, S., Ives, B., Janko, W., & O'Keefe, B. (1998). *What's so different about the World Wide Web anyway?*. Atlanta, GA: Association for Information Systems.
- Cohen A. (2003). *Blogging for Business - Case study: Western States Information Network*. PC Magazine. Retrieved January 8, 2004, from <http://www.pcmag.com/article2/0,4149,1401424,00.asp>
- ECDL Foundation. (2003). *About the ECDL Foundation & ECDL/ICDL Programme*. Retrieved January 22, 2004, from <http://www.ecdl.com/main/about.php>
- Eriksen, S. (2002). *Designing for Accountability*. Department of Human Work Science and Media Technology, Blekinge Institute of Technology. (ACM ISBN 1-1-58113-616-1/02/0010)
- Friedman, B., & Grudin, J. (1998). *Trust and Accountability: Preserving Human Values in Interactional Experience*. CHI 98 \* 18-23 April 1998 (AMC ISBN 1-58113-028-7)
- Friedman, B., Thomas, J. & Panelists (1999). *Trust Me, I'm Accountable: Trust and Accountability Online*. Chi 99 15-20 May 1999 (ACM ISBN: 1-58113-158-5)
- Goethel, J., & Gabler, C. (1999). *Accountability in a Multi-Faceted Program*. Retrieved January 3, 2004, from <http://www.gse.harvard.edu/~ncsall/fob/1999/goethel.htm>
- Gregg, D., & Goul, M. (1999). *A Proposal for an Open DSS Protocol*. New York, NY: ACM Press 0002-0782/99/1100

- Grenci, R., & Todd, P. (2002). *Solutions-Driven Marketing*. McIntire School of Commerce, University of Virginia: ACM 0002-0782/02/0300
- Kappel, M. (2000). *How great expectations become reality*. IIE Solutions 32 no3 43-6 Mr 2000.
- Kennedy, S. (1992). *Debiasing audit judgment with accountability: A framework and experimental results*. Duke University, 1992. Ann Arbor, MI: U.M.I
- Nykaza, J., Messinger, R., Boehme, F., Norman, Ch., Mace, M., & Gordon, M. (2002). *What Programmers Really Want: Results of a Needs Assessment for SDK Documentation*. Toronto, Ontario, Canada: ACM 1-58113-543-2/02/0010
- Nissenbaum, H. (1994). *Computing and Accountability*. January 1994/Vol 37, No 1  
Communications of the ACM (ACM 0002-0782/94/100)
- O'Keefe, R., & McEachern, Tim. (1998). *Web-based Customer Decision Support Systems*.  
England: ACM 0002-0782/98/0300
- Pan, Sh., & Lee, J. (2003). *Using e-CRM for a Unified View of the Customer*. New York, NY:  
ACM Press 0002-0782/03/0400
- Poore, S. (2000). *Ethnomethodology - An Introduction*. Retrieved January 22, 2004, from  
<http://www.hewett.norfolk.sch.uk/curric/soc/ethno/intro.htm>
- Rossi, G., Schwabe, D., & Lyardet, F. (n.d.). *Improving Web Information Systems with Navigational Patterns*. Retrieved October 15, from <http://www8.org/w8-papers/5b-hypertext-media/improving/improving.html>
- Sedbrook, T. (1998). *A Collaborative Fuzzy Expert System for the Web*. University of Northern Colorado. The DATA BASE for Advances in Information Systems (Vol. 29, No. 3)
- Schroeder, R. (n.d.). One Path to the Blog. *eLearn Magazine*. Retrieved January 15, 2004, from  
[http://www.elearnmag.org/subpage/sub\\_page.cfm?section=3&list\\_item=14&page=1](http://www.elearnmag.org/subpage/sub_page.cfm?section=3&list_item=14&page=1)

Stapleton, J. (2003). *Executive's Guide to Knowledge Management*. Hoboken, NJ: John Wiley & Sons, Inc.

Tegarden, D. (1999). *Business Information Visualization*. Atlanta, GA: Association for Information Systems.

Turban, E.. & Meredith, J. (1981). *Fundamentals of Management Science*. Plano, TX: Business Publications, Inc.

Table of Figures

Figure 1. Administrative Controls .....7

Figure 2. Developers' Support System Design .....14

Figure 3. <http://decision.blogspot.com/> .....16

Table of Content

Web-based Decision Support model for Software Developers accountability .....2

*Problem Identification* .....3

*System Justification* .....4

*Users Give and Take*.....5

*Decision Criteria and Alternatives*.....6

*Focus on Accountability* .....8

*Problem Solving Methodology* .....10

*Design Specifications*.....14

*Hardware Components Required* .....15

*Software Components Required* .....15

*Databases and Expert Systems*.....15

*Prototype Presentation* .....16

*System Tests*.....16

References.....17