End-Point Visualization as a Software Development Approach: Coaching vs. Managing Software and Analysis Teams

Christopher T. Street, Queen's School of Business, Queen's University, Kingston, ON 613 533-8157
cstreet@business.queensu.ca
George E. McMaster, Brandon University, Brandon MB 204 727-7460
mcmaster@brandonu.ca

ABSTRACT

The software industry has not attained the productivity results that they would like to have achieved. Yourdon et al [1991], comment that 95% of projects are never delivered as specified and on time, and more recently, this statistic has only improved to roughly 50%. Past researchers have suggested that a contributing cause of these poor results may lie in the use of structured analysis and design ("waterfall" method) as a methodological approach. In this paper, the authors discuss the concept of "End-Point Visualization" as an alternate IS design methodological approach. Two case studies of software analysis and design teams using this methodology are presented. The methodology was first tested in the development of an HR and accounting system for the franchisees of a major Canadian health care company, and then later in a series of upper level systems analysis and database undergraduate classes. Seemingly simple and cost effective ideas were found to make a large difference in the quality and timeliness of the software products delivered. End Point Visualization, a paradigm for management through coaching, is found to apply successfully to analysis and software development teams. The authors found that using this design approach resulted in improved software development outcomes such as on-time delivery, system quality, and customer satisfaction. Achieving these outcomes significantly reduced the stress normally associated with managing the development teams.

INTRODUCTION

A long-standing theme within IS research has been that of methodology development and practice. One must look no further than the statistics regarding IS project outcomes to understand why researchers and practitioners have spent so much effort on this topic. The low rate of IS project success has been with us for so long that it has become somewhat of an 'urban myth'; early reports indicated poor success rates more than a decade ago (Yourdon, et al. 1991), while contemporary results show the rate has not yet reached 50% (Russo & Wynekoop 1995). In response to the many authors who have called for more research into investigating alternate design methodologies as a way of improving these statistics (e.g. Russo & Wynekoop, 1995; Eriksson, 2001; Brugha, 2001; Iivari, Hirschheim, & Klein, 1998), this paper describes a methodological approach based on the concept of 'End-Point Visualization' (EPV).

We will describe what EPV is, as well as how it has been used as a foundation for business planning methodologies in the general management disciplines. We will then present two case studies illustrating the EPV-based IS design methodology in practice. Thirdly, a discussion of EPV principles in action will be presented, and from that a number of general principles developed that illustrate what the EPV software development process approach is, why we think it works, and how it differs from the traditional 'waterfall' versions of the software development life-cycle (SDLC). This paper will close with a brief conclusion outlining the future research opportunities regarding this design approach.

BACKGROUND

The literature review will begin by first detailing the distinction we make between Information Systems Development Methodologies (ISDM) versus Information Systems Development Approaches (ISDA). A description of EPV will follow. A brief explanation of the use of EPV as a business planning and development methodology will complete the review and provide the reader with sufficient background to illuminate the two case studies.

SDLC: Methodologies and Approaches

A methodology can be defined, in general, as "a codified set of goal-oriented procedures that guide the work and cooperation of various parties" (Iivari et al. 1998 p 165). In the realm of IS design, a systems development methodology is normally considered to be a collection of processes, principles, and standards intended to structure, simplify, standardize, and improve the practice of planning, designing and implementing information systems (Russo & Wynekoop, 1995; Iivari et al. 1998). IS practice is inundated with literally hundreds of different IS methodologies (Russo & Wynekoop, 1995) going by names such as "Method/1" (Anderson Consulting) or "Information Engineering" (Finkelstein, 1992) to name two more popular commercial methodologies. Despite the large number of 'versions' however, the majority differ only on the surface and are in fact grounded in the same general approach, structured analysis and design ('waterfall' methods) (Fitzgerald, 1994).

In contrast to a development method, Iivari et al. also discuss the more general development approach, which they call the Information Systems Development Approach (ISDA). An ISDA is defined as "a set of goals, guiding principles, fundamental concepts, and principles for the IS development process that drive interpretations and actions in IS development" (Iivari et al. 1998, p166). An ISDA is the name given to the class of methodologies that share the same fundamental concepts and principles. An ISDA is an abstraction of an ISDM. Others have also referred to the ISDA as a 'meta-methodology' within which different methods are operated; we interpret meta-methods as a design approach versus a methodology as we have defined them. For example, Method/1 is considered the 'methodology' while structured analysis and design (waterfall) is the 'approach'. The distinction is important because we believe the analysis of design methods is best studied at the approach level versus the specific methodology.

Iivari et al. (1998) also argue that the proper unit of analysis for software methodology research is the 'approach', because this is the level that captures the essential characteristics. Others echo this sentiment on the grounds that the structure or 'essential features' of a particular methodology exist at the approach level, and that this is the level where misunderstandings regarding how to best interpret or adapt a specific method are rooted (Brugha, 2001; Russo & Wynekoop, 1995). Following in the spirit of focusing on the development approach, we base the remainder of our analysis on the methodological approach of End Point Visualization and Backward Shaping.

End-Point Visualization & Backward Shaping

Coaching in the highest echelons of sport today extensively uses End Point Visualization. This is a natural creative process where you 'visualize' or look at the end result and then pace plans and activities all the way back to the present to see what action can be immediately taken and what action can be taken at stages along the way. One of the first high-level athletes to clearly describe the process was Jack Nicklaus, in his excellent book, "Golf My Way". He defines the method clearly using an example: " I never hit a shot, even in practice, without having a very sharp, in-focus picture in my head. It's like a

color movie. First I see the ball where I want it to finish, nice and white and sitting up high on the bright green grass. Then the scene changes, and I see the ball going there: its path, trajectory, and shape, even its behavior on landing. Then there's a sort of fade-out, and the next scene shows me making the kind of swing that will turn the previous images into reality. Only at the end of this short, private Hollywood spectacular do I select a club and step up to the ball."

We note the clarity of the Nicklaus visualization and the detail he uses in the construction of the images beginning at the end, as well as the holistic approach of defining activities in terms of the end-result. These are two very important points. We also found these points to be critical in managing software teams in addition to the visualization, reinforcing the great feelings that the completion of the project would give. This helps to lock the proposed performance and finishing of the project into the sub-conscious. We believe that other events "cloud" the true goal with many projects. For example, developing the individual components of a useful software system often becomes a goal that shapes the team's goals, sometimes putting the team off course and off the time deadlines.

The term 'EPV' was developed by Mr. Vic Lindal (see Lindal, Victor), former coach of the Canadian women's national volleyball team. He noticed that some sport psychologists used forward processing in teaching visualization, yet many of the athletes who had achieved great success used an end point system. The previous Jack Nicklaus excerpt is a great example. Lindal further noted that Nancy Greene, winner of gold medals in women's downhill skiing, used the EPV system extensively. He then read an article in a Science of Mind magazine on a research paper called the "Air Port Study" done at the U of Santa Barbara(see Lindal[2003]) where the author had his students read all the books on the visualization technique in other contexts. (He called it the "airport study" because so many of the books were found at airport bookstores) He noted that there were two types of processes: "forward" and "end". He then set up an experiment with his students. One group used forward processing, and the other group used the end result method. The end-point group had significantly better results in their tasks than did the forward-process teams. The more Lindal looked into this phenomenon, the more he believed that end-point visualization was a significant component of the natural creative process.

EPV has been discussed in business planning realms as well. For example, EPV was found to be a critical component in the investigation of success principles that are evident in successful entrepreneurs [McMaster & Wicks (2003)]. Also, major corporations have achieved sustainable success after adopting the EPV methodology ["McDonalds" 2001]. In a September, 2003 volleyball seminar held in Winnipeg, Canada, the invited speaker and former women's national volleyball coach, Mr. Vic Lindal, stressed to the delegates the importance of EPV as a complete coaching strategy. [In my work with Mr. Lindal, I note that he uses the methodology in almost every avenue of his life, even for example in seemingly mundane tasks such as arranging meetings]

After applying EPV principles in action on many problem domains and in different contexts we are optimistic regarding its use as an ISDA. The next section provides two case examples of these principles in action.

CASE METHODOLOGY

We will describe the E.P.V. "coaching" methodology and its application by considering two very diverse business and environmental situations that benefited from the application of this concept. The two case examples were chosen to highlight how the EPV approach worked in both a classroom and in a professional software analysis and programming setting.

Case Study 1: A Large Commercial Application

Case Context. The second author founded We Care Home Health Services in 1984, a Canadian franchise company that currently has grown to be one of the largest Home Health care companies in the country. A necessary ingredient in a franchise situation is to create a business system that can be closely followed and optimizes the chances of success (Gerber, 1996). Software that can allow other people to duplicate a significant portion of the success found in the original prototype business is a desirable ingredient. Franchisees typically pay a percentage of their gross revenue to franchise head office, whether they are making a profit or not. In return, they justifiably demand high quality service and a timely delivery of any promised products. As we developed the software for the corporation, it was evident that the franchisees would continually monitor our progress as well as competitor products to ensure that they were getting value for their dollar. Threats of having franchise fees withheld pending delivery of a working IS were suggested. This threat was taken seriously; franchisees following through on the threat would effectively result in funds for the project being cut. As a result, an environment was created where the authors were placed under considerable stress to meet deadlines and to provide the best possible product. Ultimately, our software was delivered on schedule and to the satisfaction of the franchisees. Many coaching tools were adapted from other contexts during the software development activities, which we found made the management of the project less stressful and much more enjoyable.

Methodology In Use. What we did was to take a look as clearly as possible at the required end-result, a fully-functional and integrated human resource (HR), scheduling and accounting system for new franchisees. To place this idea in concrete terms, we found that this activity began with clear pictures of what the team would be doing when they make the presentation to the client at the time that the software or analysis result is delivered. The end-point chosen was the final delivery meetings with franchisees to present their new systems to them. The team was sure to see all the aspects of the delivery, the roles that each person will play in the presentation, the skills that will be required and used to achieve a great result: a high-quality IS that results in a satisfied client.

Some of the large franchises had been using manual systems very effectively in combination with off-the-shelf software. We went into these offices and captured the current operation using the FBAM methodology [McMaster & Voorhis, 1994]. This method involves repeatedly stating your current understanding of how the business operates with what you have analyzed, and then moving on to new areas of operation in a sequential, step-wise process. After this stage was completed, we then went back to the office with the team and drew a large picture of a mountain with a path up the mountain, and a parking lot at the other end. The analogical journey to the mountain was through trees and around lakes. The final presentation was the top of the mountain with the deliverable. The parking lot was where we were now. When and where was the final presentation? To whom would we be presenting? These are typical questions and the more questions to give the reality of the situation, the better. We could define some of these details right away. Then we placed what we thought were excellent milestones on our climb. As we worked our way back to the parking lot it became clear that we did not have all the skills required for the climb, and training was appropriately scheduled. The chart was excellent at keeping the final goal, the day of the final presentation, in mind. (There were of course, many presentations on the way "Up the Mountain")

At the start of every meeting, we would give the team members the clear vision of where we were going with the project and what the end product would look like. Then we get a picture of what we will accomplish for the next milestone. Then we get a picture of any additional skills that we will need to

acquire to reach the objective. It was also important to check with the team members at the start of the meeting regarding what was on their agenda as well as what was important to them at this time. It was also important at the end of each meeting to ask each team member what he or she thought that they had done well. We would run our "movies" through our minds of all the great results that they had achieved since the last meeting. This is called retroactive visualization and helps build a "reel" in their minds of all the best accomplishments [Lindal, personal conversation]. Also, we ask our team members to make entries in a journal that they keep of all their best accomplishments. Putting items into a journal significantly helps to reinforce the great things that they have accomplished in their development of the product to date and in their meetings with the client. The journal also helped the development team to individually keep track of their position in the project. The bottom line was to structure design activities so that what we discussed in our planning and design actions closely approximated what we are going to deliver and do with our clients.

Case #1 Summary. When you have a great picture in your mind of the end-point, you can see clearly what must be done to get there. It is important to connect the pictures of what will take place back to the present day. For example, you may want to visualize what the team is doing the week before the presentation so that the presentation is a wow. This may mean giving the presentations in practice sessions, or in a severe case to taking toastmasters! You then picture what is happening the month before that, then two months before that. Pick a timeline that works for you and ensures accountability and a timely result at each checkpoint.

When working with the team members, whether on a programming team or an analysis team, we ensured that each time we met that we reinforced not only how to visualize, but also how to critically analyze the journey to the end product. In the franchise MIS development project, one critical benefit of using this approach was how it demonstrated to us so early on in the process that we lacked the necessary programming skills to deliver the product that we wanted. Rather than finding out later in the development process and being tempted to 'make-do', we were forced to recognize shortcomings immediately. To this day, we are not confident that we would have recognized this point using a traditional waterfall approach.

We also coined the phrase "E.P.V. beyond", which means simply to go out a little further from the end of the design or construction stages. Having the team members see themselves leaving the customer presentations as well as imagine how they would feel seeing their work being appreciated is an important motivational tool. This means that you can take simple action steps such as asking one of the team members to book a room for the presentation. I would pace the vision out even further, and have the team members visualize themselves training the users. All team members are to run clear pictures in their minds the night before a meeting, so that they have clear in their minds what they would like to accomplish in the meeting and in the next phase of the project development. We would then check with the team members during the meeting as to exactly what they had pictured in their minds. Keeping the end point in sight at all times was a critical success factor; for example, we found it was important to have a white board in the office to record progress, to pose questions, and to encourage communication, which helped to ensure that the E.P.V. was always in the forefront.

Case Study 2: Upper Level University Classes are "Coached" by the Instructor to Develop Commercial Software using The E.P.V. Methodology.

Case Context. At Brandon University, some instructors follow an academic model where professors in Computer Science endeavor to earn an equivalent salary outside the university. This ensures that we

know and believe in what we deliver in the classroom. Hence, we deliver consulting services to business and are therefore aware of businesses that could benefit from student projects. These relationships provide valuable benefits to all those involved: the businesses have access to bright, motivated computer science students to develop business applications for them; the students have the opportunity to learn their skills in a real business context; and the instructors are able to test and refine new methodology models under field conditions while still maintaining some level of control over the outcomes.

We always ask our students to work with a local business in our city or within a 100 mile radius, and produce a report that analyzes the business and delivers a report capturing a comprehensive vision of what the end-point product should look like. Five traditional IS development tools were used to capture the end-point: FBAM analyses are carried out, allowing the analyst to understand the business and its requirements as well as creating a visual model that reassures the prospective business user that you understand a lot about operating their business; a dataflow diagram (DFD); an entity relationship model (ERD); sample user interfaces; and recommendations for the client. The results have been spectacular in that teams within a classroom will often create commercially viable software in a subsequent data base course. In previous years teams have created: a small library system (McMaster [1993]) a scheduling and tracking system for an air shuttle company, software for the operation of a chinchilla farm, and a complete report for the Canadian Diabetes Society.

Methodology In Use. As a first example of the application of this coaching methodology that was originally developed and tested in the work for We Care Health Services in Case Study 1, we applied the methodology to an upper level system analysis class. At the start of each new class we have someone from a previous year's team do a presentation using their report on the analysis of a company from the previous year, and I also ensure that the presentation involves the client. They can get the good "feeling" for the process and for the interaction with the client. This is the first step in visualizing the end-point. The students are coached to "envision" the end product, the required report that they will have to produce, and the presentation and the wonderful interaction with the client. We then work backward through the steps to get from the end-point back to the present day and indicate what we need to do at each stage in the project development. It is now also clear to the students what they are required to learn in order to succeed. Having the "end-point" clearly defined for them has been repeated seen as very motivating to them.

We then schedule the presentation time three months out from now (even if they have not met with their client yet!) since the analysis course runs over three months. We determine what they will need for the presentation. For example, if they are weak in presentation skills we recommend them to do "Structured Walk Throughs" during the course (See Yourdon [1988]). Echoing our experience in the first case study, it also becomes clear what skills must accumulated and on what schedule as we get ready to do the analysis of a brand new company.

Case Study #2 Summary. The EPV methodology permits us to coach students in two sequential, half term courses. We successfully develop significant software, and in many cases create useable working prototypes that are expanded on by the students outside of the course structure. The point is that students, in a short time frame, reliably and consistently produced software that was used by business.

Our major finding in case #2, and repeated over several iterations, was the extent to which traditional SDLC tools and practices such as using FBAM, DFDs, and ERDs were also applicable in the EPV approach. This finding reinforces our proposition of EPV as a design approach; the use of the existing

SDLC toolset, guided by the EPV approach to software design, resulted in successful software products that were used in business contexts.

The student projects gave us the opportunity to refine the initial methodology that was first used in the first case. As Nicklaus stated, "the picture of the ball landing and sitting up white" was key … to have a clear picture. With the students, we became better at coaching them to visualize a variety of situations and to do so more clearly. To illustrate, we would not hesitate to view a chinchilla-breeding program, where many factors are input to produce an optimum result, as a system designed to assist in producing a fine animal pelt. As with coaching a sports team, the practice sessions become better as the students realize that what is coached in practice is what also leads to superior real world results.

DISCUSSION

As described in case study 1, this methodology was first used to create and deliver an integrated HR and accounting software for a major health care corporation. Case study 2 summarizes how the methodology was refined through being successfully applied to other software environments and to classroom coaching.

The typical SDLC steps are usually specified as: analysis, requirements, design, construction, delivery, maintenance. What we are proposing is that the EPV approach be considered as a foundation for interpreting the SDLC, along with use of the Forms Based Analysis method [McMaster and Voorhis 1994]. Since much of software today has tremendous delivery challenges (not on time and not on spec.), the coaching of the team must be elevated as an important development factor so that the team has a system to constantly focus on achieving the goal. If it is part of the SDLC, then we believe that it will go a long way to alleviating many challenges in software development.

Toward the goal of further developing EPV and using it in practice, we have developed a preliminary protocol for the approach. While we stress that this procedure is in its early stages of development, we propose the following steps be taken in applying the approach in practice:

1. **Analysis**: Use the forms based analysis methodology (FBAM) to get a quick and understandable diagrammatic method from the end-user's perspective to understand the purpose and procedures of the business problem. This stage helps define the end-point from a business view versus a systems view.
2. **Run an EPV with the team**. To make it more realistic, this could be a mock-up of the final presentation to the client. Flesh out as many details as possible. What is the delivery date, and what steps, beginning at the end, are required to get from there to here? We have found that visual representations, such as drawing the mountain climbing analogical model and list items on the visual that must be accomplished with timelines (see Smith [1997]).
3. **Requirements**: Using traditional SDLC tools (eg. DFD, ERD, State Diagrams, etc), design the system to capture the characteristics developed in stapes #1 & #2.
4. **Repeat the EPV process at regular intervals.** It is important to state final objectives on a periodic basis for at least two reasons. First, we found it important to periodically re-engage development teams to ensure that everyone was working to the same end-point. Second, an EPV procedure was found to be helpful in dealing with the unexpected changes that occur during development. For example, changes to key requirements in response to changing business contexts were initially addressed with the question 'has our EPV changed, and if so, how?' We therefore advocate regularly scheduled EPV procedures during development to keep team

members synchronized, as well as remaining open to unscheduled EPV meetings in response to potential changes during design and construction.

Additional development steps between #3 and #4 are free to follow classic models. However, we cannot overstress the use of the EPV at every stage and in every meeting. The vision of the completed project must be constantly repeated and then the process of stepping back to now with a look at what we need to do at this stage.

To summarize, the coaching methodology clearly defined the deliverable first, even defining the final interaction with the client. The design team then began working backward to the present day rather than beginning with an analysis or requirements phase as suggested by traditional structured design methods. The question was always asked, "What can we do today and in the time interval to the next meeting to accomplish our task and be successful?" To stress the importance of the visualization process, a diagram of a mountain climb was used as a metaphor, and reinforced the current position as we moved along in the development process. What it meant to be on the top of the mountain was reviewed at each meeting. We also used "EPV beyond" by discussing with the programmers, just how they would feel when they flew back home after a successful presentation.

Discussion – Principles of EPV as an ISDA

We recall that "an ISDA is a class of methodologies which share fundamental concepts and principles for IS development." (Iivari et al. 1998). Let us highlight the principles of the EPV design and coaching process as used in Information Systems development.

- Start at the end and work back
- Continually ask what can be done today to reach the goal.
- Make clear visualization a part of the team learning and application process
- Consider the end point from multiple perspectives

The next question to ask is, "why does EPV work?" If we think of the analogy of an airplane, it is almost always off-course, but it knows where the end is and can constantly correct its position. If the end point is changed, it can adapt at that time and fly to another city for example. The system forces early goal identification early on. We have observed that EPV naturally improves both communication and the buy-in on the part of the individuals working on the project. It reminds the designers and managers of what the end-point should look like, so that they are not distracted and keep their efforts focused. As a result, more informed decisions are made along the way. People are motivated when they have the end in mind.

A final question would be, "How does the EPV system of software development differ from what we already have?" The waterfall method and associated methods go from the start to the finish, and we believe that this forces people to make important decisions up front, is an un-natural method, and takes away some of the incentive to make changes later on. From our perspective, we consider waterfall methods to define the IS deliverable in terms of the sum of its constituent parts, where analysis defines the scope of requirements, requirements define the scope of the design, etc. until the software product is delivered. Under this approach, the process drives the deliverable, and it is then unremarkable that the process moves from analysis to finished product. In contrast, EPV considers the IS deliverable as a holistic goal that defines itself from outside the IS design realm. Under EPV, the deliverable defines the

process; clearly defining the end-point and working back to the present day is therefore seen as the enactment of "deliverable drives process'.

CONCLUSION

Having used the EPV system to achieve great results in both classroom projects and on very large corporate software projects, we highly recommend the methodology to practitioners of software development. The manager becomes more of a "Coach" than a "Manager", and their task is much more enjoyable, and far less stressful. Additionally, we can state that the teams that have used this approach have achieved success rates much higher than the quoted 50% probability of delivery. We fully expect that once the E.P.V. system is mastered, the manager ("coach") will be surprised at how it will help them manage more readily, and clarifies the result that they want at each stage of the project. Additionally, practitioners will find that they will use the method in many more situations.

Some additional ideas that proved to be most useful in supporting the E.P.V. methodology was to continually create times for presentations of the software to the users by the programmers themselves. This provided great incentive to meet deadlines. The users were coached as to how to make their interaction a positive and a constructive one with the programmers. For example criticism was not an option, whereas constructive and positive suggestions were. We also drew a picture of a mountain (Visual timeline with tasks) with a flag being planted on top representing the delivery of the product. Where we were at the beginning of any project was far from the base of the mountain. By seeing the end and pacing it back to now we realized what language we were going to use, and since several of the programmers did not know the chosen language a learning time was built in. The analysis phase was included in here, since it is our finding that most intended users do not clearly know what they want in a system, and go through many iterations of problem definition before they get there. Colonel Rolf Smith[1993] in his excellent book also uses a variation on this visual model when leading major corporations on a "Thinking Expedition" which essentially helps a corporation discover their "Real" problem versus a "Perceived" problem and to describe it clearly. Having assisted Rolf on one 11-day expedition with the Halliburton Corporation, this challenge was clearly made evident as we progressed through 12 iterations of attempting to clarify and specify the problem, and the mountain climbing analogy was very helpful as a visual aid to mark our progress. This model created a certain degree of excitement and flair, and was a great reminder of how we were making progress to our final E.P.V.

This paper briefly outlines how we have adapted the EPV methodology used in professional sporting and traditional business contexts for use as a software development approach. It has been our experience that the EPV approach is an improvement over the traditional structured waterfall methods that have been criticized as being inflexible (Russo & Wynekoop, 1995). Nevertheless, our results are limited by the fact that we require more work be done in confirming the efficacy of this approach. We currently see at least two areas for future research. First, we must confirm that the design approach is an improvement over existing methodologies by comparing software development projects that have used the EPV approach with those that have used traditional methodologies in order to ascertain the advantage of one approach over the other. We should also test the EPV approach in more varied contexts, such as its use in larger development teams, pair programming, or in long-term software maintenance situations, to name just a few. We visualize as our end-point the creation of a more robust, flexible, and successful software design approach. Over time, and through an evolutionary improvement process, the path to a more successful ISDA will eventually come into view.

REFERENCES

Coad, P. and Yourdon, E., *Object Oriented Analysis*, Yourdon Press Computing Series, 1991.

Eriksson, D.M. "Multi-modal investigation of a business process and information system redesign: A post-implementation case study, Systems Research and Behavioral Science, 18 pp181-196 (2001)

Finkelstein, C. "Information Engineering : Strategic Systems Development", Addison-Wesley, Sydney: Australia (1992)

Fitzgerald, B. "The Systems Development Dilemma: Whether to Adopt Formalised Systems Development Methodologies or Not?" in Bates, W. (Ed.) Proceedings of the Second European Conference on Information Systems, Nijenrode University Press, Breukelen, 1994, pp691-709

Gerber, M., The E-Myth Revisited, Harper Business (1995)

Iivari, J.; Hirschheim, R.; & Klein, H. "Contrasting IS Development Approaches" Information Systems Research 9(2) pp164-193 (1998)

Lindal, Victor, http://hypbus.com/vicvball/coach/skill.html, referenced Sept 11, 2003.

McMaster, G.E.. *Programming Teams*, Proceedings of The Converging Technologies Conference of the Canadian Information Processing Society, May 1993.

McMaster, G.E., Voorhis, D., *Forms Based Analysis and Interview Method*, Proceedings of the Western Decision Sciences Conference, 1994

McMaster, G.E.; Wicks, S. "Success Characteristics of Female Entrepreneurs in Today's Rapidly Changing Business Environment" Proceedings of the Western Decision Sciences Conference, Kauai, (2003)

Nicklaus, Jack, "Golf My Way – Control Shots", Simon & Schuster, 1983

Russo, N.; Wynekoop, J.L. "The Use and Adaptation of System Development Methodologies," in M. Khosrowpour (ed.), Managing Information & Communications in a Changing Global Environment: Proceedings of the Information Resources Management Association International Conference, Idea Group Publishing, May 1995, p. 162.

Rosson, M.B., Integrating Development of Task and Object Models, Communications of The ACM, New York, 1999

Smith, Rolf, "The Seven Levels of Change", "The Summit Publishing Group, 1997

Yourdon, Edward, "Managing the Structured Techniques", Fourth Edition. Prentice Hall, 1988.

"The McDonalds of Golf?" The Canadian Franchise & Dealership Guide, (2001)