Understanding Context through a Comprehensive Prototyping Experience: A Testbed Research Strategy for Emerging Technologies

J. Ramon Gil-Garcia, Theresa A. Pardo & Andrea Baker
Center for Technology in Government, University at Albany, SUNY
{jgil-garcia, tpardo, abaker}@ctg.albany.edu

Abstract

Information and Communication Technologies (ICTs) are rapidly changing and new technologies, processes, and skills are constantly emerging. An important challenge for the research community is to gain knowledge about these emergent technologies in specific contexts, sometimes before they are actually implemented. This paper draws on our experience in the use of comprehensive prototyping as a methodology for building understanding of emerging technologies in new contexts[1]. A Testbed research strategy combines various prototyping, business analysis, team work, and training techniques to understand the specific characteristics of a technology and the context in which it is going to be embedded. The paper presents three cases of Testbed research approaches developed within a 10 year period and presents some insights based on those experiences to inform the efforts of both practitioners and researchers.

1. Introduction

Information and Communication Technologies (ICTs) are rapidly changing and new technologies, processes, and skills are constantly emerging. In order to understand the complexity of these emergent technologies in different social contexts, digital government researchers are employing different strategies such as the use of multiple methods and theoretical lenses [11, 34]. This combination of research methods in multi-method approaches has the potential to strengthen results through the power of triangulation and develop a more comprehensive understanding of the phenomenon under study by taking advantage of the capabilities of individual methods [13, 23, 37]. Similarly, the use of multiple theoretical lenses could provide a more integral explanation of the phenomenon. However, the insight gained through these efforts is limited, in general, for cases in which the phenomenon has not taken place in the context of interest. This bounded understanding presents a challenge to researchers as they work to understand these emerging phenomena before they actually take place in certain organizational or social contexts. Many traditional research methods are very effective for situations in which a retrospective analysis is possible, but few of them can capture or recreate the complexity of situations that have not happened yet. The use of scenarios and role-playing is a valuable starting point, but they are still limited in their capacity to realistically recreate the context for respondents [4]

Prototyping has been shown as a powerful way to gain knowledge about technical capabilities, benefits, and limitations in relative uncertain conditions [2]. We argue that this same logic works for a more comprehensive experience, in which people are encouraged to take into consideration not only the technical and business process aspects of an ICT initiative, but also managerial, institutional, and environmental factors. Through interactive training sessions, workshops, and team assignments in which participants use tools to understand their goals, benefits, challenges, and stakeholders, among other important aspects, they learn not only about the technology and its features (by prototyping an information system), but also about their own organizations and the contexts in which they are embedded. Therefore, this comprehensive prototyping experience has the potential to recreate the context (e.g., state agency setting) for a technology application (e.g., XML for content management) before that technology is actually implemented and used in that specific context. It also allows for different disciplinary lenses and traditional methods (ex., interviews, surveys, and direct observation) within the Testbed framework and, therefore, it can potentially

[1] A previous version of this paper was presented at the Pre- ICIS International Symposium on Research Methods, International Conference on Information Systems, Las Vegas, Nevada, USA, December 11-14. The authors want to thank the participants at this workshop for their valuable comments. Any mistakes or omissions are the sole responsibility of the authors.

1530-1605/07 $20.00 ©2007 IEEE

improve our understanding of emerging technologies in specific relevant contexts.

Drawing on our experience in the use of comprehensive prototyping, this paper argues that a Testbed methodology, combining prototyping, training, group dynamics, and traditional research methods, is a useful way to building understanding of emerging technologies in new different contexts. The paper presents a description of three Testbed projects conducted at the Center for Technology in Government (CTG) in the last 10 years and explains how this methodology has been used and refined during this time. It also offers some recommendations for future research and use.

The paper is organized in six sections including this introduction. Section two highlights some complementarities between prototyping and a more comprehensive view of systems development, implementation and use. Section three describes the methodology used in this paper. The paper is based on three detailed cases in which a Testbed strategy was used. Sections four presents a detailed description and analysis of the components of each Testbed project, as well as its results. Section five highlights the evolution of the Testbed strategy and its potential as a venue for research. Finally, section six provides some final comments and suggestions for future research.

2. Comprehensive Prototyping: Incorporating Environmental, Institutional and Organizational Factors

Information systems in general, and electronic government in particular, are complex socio-technical phenomena. This complexity is reflected in a high rate of failure [6, 14]. For instance, Heeks [14] estimates the rate of IT project failure in public sector settings in industrialized countries, is in the range of 80 percent, which is similar to failure rates in developing and transitional countries. Practitioners and scholars alike have consistently sought to uncover reasons for unsuccessful IT projects and create strategies to prevent systems failure.

One of the approaches to systems development has been prototyping. Considered a forward-looking approach to testing, the purpose of creating a prototype is to test system requirements before implementing a full-scale design. Prototyping approaches to application design and systems development have greatly evolved and changed since their popularity in the early 1980s. However, in general terms, the approach continues to focus mainly on the technology. In contrast, recent research about systems development, implementation and use argue that including environmental, institutional, and organizational factors is important to the adoption and success of information systems [26].

2.1. Systems Development and Prototyping

Throughout many periods in the history of the IS discipline scholars have sought to understand systems failure. Early on, researchers frequently found complex reasons for systems failure. While these insights proved useful to the evolution of the IS discipline, early analysis was mostly retrospective. Scaled down to its basic elements, the common method of systems development, referred to as the Systems Development Life Cycle (SDLC), consists of a requirements gathering phase, a design phase and an implementation phase [2]. Development was generally done by an IT analyst and turned over to users after completion [2].

This process became a target of criticism in the 1970s when researchers realized that the approach was in large part responsible for systems failure. Senn [35] determined that management was generally left out of the process; instead analysts developed the entire system. Analysts were criticized for approaching systems development as a problem solving approach for organizations without exploring the current system or the organizational environment. As a result, Senn [35] concluded that systems, which were often too complex, were not successful. Data was difficult to access and information was structured inefficiently from a use perspective. Senn's observations and insights led him to question the relationship between analysts and users. He suggested that communication, more specifically the lack of communication was "the most basic and crucial factor causing the problems" (p.31). In their case study of system development failures, Schmitt and Kozar [33] also concluded that weak relationships between users and analysts led to problems in systems development. In their study of a state land management agency, the authors found that organizations can lack knowledge about their information needs or lack the ability to transfer that knowledge to systems developers. This issue and an over-reliance on developers can lead to a poor fit between the system and the organization.

Soon after this line of inquiry, prototyping was employed as a method to counter system failure [1, 2]. Prototyping had long been used in the engineering field, but its application to the IS field was new in the 1980s [1, 15, 25]. At the time prototyping was viewed as an improvement to systems development because it was a window to gather user feedback and make revisions if necessary before implementing a full-scale

system [2]. In addition, the technique increased the speed at which new technologies were evaluated [25].

The key reason for using prototypes is to evaluate system requirements before design and implementation [38]. This frequently involves the creation of a pilot [20] and according to Naumann and Jenkins [25] involve three types of actors: the designer, the user and the system. A pilot is generally a model of a portion of the system to be implemented in the future as part of the SDLC. The pilot may be used as a test of concept only and "thrown away" after initial use, or as a base model for further development often called the "evolutionary" approach [20]. Prototyping approaches include everything from using paper-based models to simulation in virtual environments.

Today it is difficult to imagine systems design without prototyping. There are multiple types of prototyping techniques, but they generally fall into two broad categories: systems development and software development [5]. Techniques include rapid application, incremental development process, and simulation [2, 4, 20, 32].

Early approaches to prototyping include Naumann and Jenkins' four step approach [25]. Their approach began with identifying user requirements and proceeding to developing a "skeleton" of the proposed system in a short turn around time, to testing the prototype and finally refining it. They stress that the principle underlying prototyping is to create a flexible system fast so that users can test it in their own context and adapt it to their needs.

Prototyping has become an accepted strategy in the IT community mainly because of the benefits the method offers. Some of the benefits of prototyping include helping designers understand user requirements by allowing them to interact with a "working example" of a system, demonstrating the potential return on investment of the technology as well as decrease the costs of initial problems [5]. Kordon and Luqi [20] suggest rapid prototyping is needed to help alleviate potential problems once a system is deployed. Boar [2] states that prototyping creates an environment where designers can research the system requirements, an "evolutionary discovery as opposed to omniscient foresight" (p. 5). This discovery process provides an initial product that can be evaluated and refined, which helps increase the potential for organizational acceptance.

In their research on early prototypes in context, Reilly and colleagues [32] suggest adopting an approach that will draw on the actual experience of the intended users. By understanding the context in which the technology will be utilized, designers should be able to better meet user requirements and thus increase the chances of the success of the technology. "Determining appropriate levels of fidelity and granularity in a prototype is critical before developing and evaluating it" [32, p. 49].

Sutcliffe and Maiden [36] suggest that role playing the user scenario in a virtual environment allows analysts the opportunity to observe potential problems with requirements analysis. They call the type of prototyping "immersive scenario based requirements engineering" (IRSE). In a case study involving aircraft maintenance training, analysts tested the requirements for an aircraft maintenance process by "walking through" the task within a virtual environment. The analysts played the role of the users. Through this process the analysts discovered "requirements problems" with their design. The IRSE prototype provided the researchers "insight into the requirements defects in the virtual prototype while identifying usability problems that interfered with accurate requirements capture" [36, p. 108]. Although the authors suggested that ISRE is beneficial when testing multimedia applications, they also mentioned that the approach is very costly.

Buchenau and Suri [4] propose a similar approach to application development called "experience prototyping." This technique requires developers to experience the users perspective by inserting themselves into the situation where the technology will be used. Experience prototyping can include a deployment of a pilot of the technology or it may only test the context where the technology will be used. For example, Buchenau and Suri describe a simulation exercise in which pagers were distributed to designers of an Internet enabled cardiac telemetry system. Before the team designed the technology they wanted to gather information about the potential impact it would have on the patients. The pagers were used to simulate patient experience of a "defibrillating shock that would be sufficient to knock a person off their feet" [4, p. 426]. The experience allowed researchers to better understand patients' needs and designers to discover social and contextual factors wouldn't have otherwise known without this form of prototyping [4].

Overall and regardless of the technique used to create a prototype, the analysis of user requirements focuses mainly around the design of the system and the IT artifact. The prototype is used as a method to evaluate the effectiveness of an information technology, before a full system is implemented, but environmental, institutional and organizational factors have not been adequately integrated into the

prototyping logic while conceptualizing and developing a system.

2.2. Organizational and Social Factors in Systems Development, Implementation and Use

While prototyping and other system development techniques may be able to provide insight into systems' effectiveness, the organizational and social factors surrounding that context cannot be ignored. Newman and Robey [26] discussed two techniques for incorporating the analysis of organizational and social factors into system development models: factor research models and process research models. Factor models consist of relationships between predictors and outcomes. In contrast, the conceptualization of process modeling is based on a social approach to testing information systems. In this approach designers introduce the technology and then attempt to understand "how and why the results of the development efforts are achieved" [26, p. 250]. Process models require measurement or analysis over time, measuring encounters and episodes. Ultimately, they conclude that process modeling opens a window into user perceptions of the effectiveness of the technology and implications on their actions. The process model can "show how parties interact, how they collectively agree on future courses of action, and how they perceive constraints on their action" [26, p. 262]. The authors suggest the model has both descriptive and predictive capabilities and suggest that the approach can be insightful for researchers studying information systems development.

Zhu [39] offers a similar approach to understanding information systems development called WSR (wuli-shili-renli). The approach is based on Oriental philosophy and combines the technical, social and contextual elements of systems design. Wuli is translated to material and technical. Shili means cognitive and psychological and renli translates into social and political. Prototyping in this approach is part of a process that explores systems design from initial user inquiry to design and evaluation of the design. Zhu [39] says that the holistic approach is more challenging than traditional methods because "[developers] have the responsibility to learn more methods and interpersonal skills, and to become more sensitive towards uncertainty and complexity in organizational situations" (p. 191). Zhu's research also discusses the integration of business process analysis into information systems development. He suggests that technology is a "means to an end... not an end itself" [39, p. 197] and that successful projects incorporate both business and IT elements.

Recognizing the importance of multiple factors for a more comprehensive view of the phenomenon is not unique to systems development and extends to IS research in general. For instance, based on a review of research topics and findings in *Information Systems Research*, Orlikowski and Iacono [28] identify five meta-categories of how researchers conceptualize information technologies: (1) the tool view, (2) the proxy view, (3) the ensemble view, (4) the computational view, and (5) the nominal view.

The ensemble view includes the more holistic approaches to information technologies and organizations. According to Orlikowski and Iacono [28], the ensemble view refers to technology in four different ways: development project, production network, embedded system, and structure. Some examples of these more integrative approaches are: socio-technical systems theory [3, 22, 24]; social informatics [16, 18]; technology enactment theory [8, 9]; the structurational model of technology [27, 29]; and adaptive structuration theory [7, 31].

Using different but related concepts, these theories argue that there is a dynamic interaction between social structures and information technologies [10, 28]. Information and communication technologies (ICTs) have the potential to change social and organizational structures, but at the same time, they are affected by these structures in their design, implementation and use [9]. In addition, for these ensemble-view theories, information technologies are not only the physical artifacts, but also the social and organizational structures around those artifacts [28]. The technology is only one component of a more complex socio-technical system [17, 21, 30]. Other components can include commitment, training, and policies, among others [19]. These social and organizational structures can be thought of as the factors and relationships around the technological artifact. In different specific theoretical models, the construct social structures may include individual, project, organizational, institutional, legal, regulatory and/or environmental factors, as well as their interrelationships.

Based on a fairly comprehensive review of the current literature, Gil-Garcia and Pardo [12] classify many of these important factors into five categories: (1) information and data factors, (2) information technology factors, (3) organizational and managerial factors, (4) legal and regulatory factors, and (5) institutional and environmental factors (see Figure 1). As mentioned earlier in this paper, many of these factors and their interrelationships have been

recognized as important elements to take into consideration, but they have not been fully identified and incorporated into a comprehensive prototyping strategy.

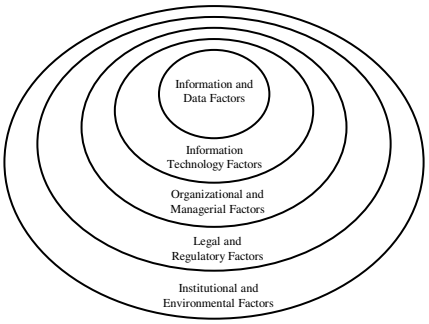

Figure 1. Factors affecting government IT initiatives

Therefore, several important questions remain unanswered: In order to study emergent technologies in new contexts, is there a better way to incorporate and understand the context? How can researchers and practitioners recreate technologies and contexts as a way to gain more comprehensive knowledge about emergent technologies? Are there strategies that successfully combine the strengths of prototyping and innovative ideas from systems development and IS research? Through the review of three cases in which a comprehensive prototyping strategy was used, this paper attempts to provide preliminary answers to these and other important questions.

3. Methodology and Case Introductions

This paper is based on the analysis of three cases in which a Testbed research strategy was used to help participants understand their business problem, the requirements of a solution strategy, as well as the relevant managerial, institutional, and environmental factors and to help researchers examine questions beyond those possible in a traditional prototyping environment. The cases were developed through a combination of content analysis, interview transcripts, and survey data. A comparative analysis of the cases was conducted to examine the use of comprehensive prototyping as a strategy for reducing the risk of failure of the adoption of emerging technologies and to providing researchers with access to more robust opportunities to study emerging technologies in context. The paper presents a description of three Testbed projects conducted in the last 10 years and explains how the methdology has been used and refined during this time. The paper concludes with recommendations for future research and use.

4. A Testbed Research Strategy

The Technology Testbed Program at the Center for Technology in Government was created as a specific implementation of a comprehensive prototyping strategy. The program provides an effective framework (technology and context) for examining an emergent technology in a specific context, prior to full-scale implementation. It has evolved along a number of dimensions to exemplify comprehensive prototyping.

Table 1 presents the purpose and focus of the three Testbeds of interest in this study. All Testbeds were conducted with a core set of common characteristics some based on empirically supported system development techniques and others based on accumulated knowledge and experience of both Center staff and Testbed participants. In terms of prototyping the context, participants were engaged and focused on a real project and were asked to think as if they were going to implement the project. All exercises and activities focused on the formulation of the context in various explicit ways such as models of problems, power and authority relationships, and resources. At the beginning of the Testbed, each team was required to have a specific proposal and team composition (including all actors involved in the web publishing process from content creation to publishing on the Web). For example, all Testbed teams are required to involve a mixture of professions such as information technology and program area specialists as well as mixtures of level of authority such as top and middle management and line staff. In addition, Testbed teams are required to look beyond the specific questions of technology capability and into questions about factors affecting their efforts (see Figure 1).

The following sections present a brief description of each of the Testbeds including some of the results. The discussion section focuses on two primary areas of interest; the evolution of the Testbed Strategy, specifically in terms of comprehensive prototyping and its impact on understanding factors affecting IT initiatives and on the Testbed as a research venue.

Table 1. Three Testbed cases

	Purpose	**Focus of Activity**
Groupware 1994	Investigate groupware technology	Work flow analysis and examination of ease of development and use, system efficiency and

		interoperability
Internet Services 1996	Identify the policies, management tools, information products, organizational structures, and business processes necessary to take advantage of the technology to achieve important public goals.	Develop, test, and evaluate prototype Web sites for each agency and identify the technology, management, and policy barriers they encountered and lessons learned in the prototyping process.
XML 2004	Determine the barriers and enablers to using XML as a web site content management strategy and examine the organizational capacity required to do it.	Series of presentations, training sessions, workshops, and discussions to support the development of agency-specific XML prototypes and business cases.

4.1. The Groupware Testbed

The first Technology Testbed, launched in 1994, investigated new technologies that support workgroups and teams. Staff from several state agencies focused on the investigation of several business problems faced by agencies, using specific products and technologies that offer solutions to those problems. The state agency built a prototype system for tracking and routing executive correspondence. The professional association prototyped a system for supporting collaborative writing of documents. The research center used a prototype group decision support tool to hold "any time, any place" decision conferences on its research activities. The projects engaged in work flow analysis, process reengineering, and examined questions about ease of development, ease of use, system efficiency, and interoperability. The Groupware Testbed culminated in a public seminar presenting the prototypes and the project results.

4.1.1. Comprehensive Prototyping Results

Through the Testbed the agencies became aware of the critical steps of work flow analysis and reengineering. Up to the testbed they had not included work flow analysis in their development efforts. Agencies indicated a new understanding of critical steps. Further the Testbed created a set of conditions where agencies became more familiar with technology due to the low-risk environment of the Center. The most significant result of the Groupware Testbed project was the organization learning. They became more aware of cross-division and cross-agency effects of moving to new technologies. Participants report that users and upper management became more interested and involved in the Testbed than they had been in previous system development activities.

4.2. Internet Services Testbed

In 1996 government agencies were again witnessing the emergence of a potentially transformative technology; electronic networks were becoming an increasingly more important means of communicating in society. However, there was very little experience in using the Web as a service delivery channel and it was unknown if the traditional methods agencies had used to define, design, and develop information systems would work in this highly public, networked environment.

The issue under investigation was do we have or can we develop policies, management tools, information products, organizational structures, and business processes to take advantage of this technology and direct its use to achieve important public goals? Will departments be willing to share pertinent and timely information? Will agencies be willing to relinquish solitary control over programs? Can traditional hiring and training practices allow the public work force to acquire and maintain new skills?

The project activities focused on developing, and evaluating prototype Web sites for each agency and identifying the technology, management, and policy barriers they encountered and the lessons they learned. The activities included ten events organized by the Center, including seven full-day workshops. Each workshop focused on a major component of the development process. The workshops reflected the collaborative and cross-organizational nature of Web-based work. Electronic communication among project participants and access to the Web itself was provided through Internet access accounts with a local service provider. Group collaboration software was also provided so all participants so they could share ideas, questions, and concerns.

Six of the seven agencies completed prototype Web sites during the project. The development and evaluation process uncovered a number of obstacles that the agencies worked to overcome.

4.2.1. Comprehensive Prototyping Results

The project proceeded in a series of workshops where the participating agencies worked together and

individually on service definition, development, refinement, and evaluation. The project methodology was designed to guide them through a process of aligning Internet technologies with the programmatic objectives of their organizations. Over the course of the project, most of the agencies revised their service goals and changed the membership of their development teams. Virtually all of these adjustments were made in response to new insights gained during the workshops and the subsequent development work taking place in each agency. The result was a multi-faceted analysis of each proposed project and in some cases major rethinking of the original proposal and its feasibility given organizational and programmatic realities.

The CTG project team worked with the agencies throughout the project to identify the barriers they encountered and record the lessons they learned as Web service developers. Time was set aside in each workshop to discuss these issues. The agency teams also participated in a brainstorming session to identify and classify the barriers they encountered in their work.

4.3. Web Site Management Using XML: A Testbed Project

XML is generally understood to be a technology that supports effective data exchange between applications. However, it also offers a viable long-term solution to many of the shortcomings of HTML because it structures and describes Web content in a meaningful way. Despite clear advantages, agencies confront many obstacles to the adoption and implementation of XML-based Web site management. These include the need for technical training and infrastructure readiness, but more importantly, the needs for solid business case justifications, understanding the impact of organizational change, leadership buy-in, and a firm understanding of where to begin. The purpose of this Testbed was to understand how XML could be used for website management in government settings.

This was the first time New York State (NYS) agencies were considering the use of XML for Web site content management. Similar to most organizations, NYS agencies are currently creating and managing their web sites using HTML or XHTML. Few webmasters have knowledge of XML in general or its application to the web site content management and publishing process. The XML Testbed project involves a series of presentations, training sessions, workshops, and discussions to support the development of agency-specific XML prototypes and business cases.

Two training sessions were used to transfer knowledge about XML and its advantages and challenges when used for website management. In addition, a two-day training session provided the basics of project management, including defining a problem, identifying stakeholders, analyzing workflows and business processes, developing cost-benefit analysis, and writing a business case, among others. Six workshops were organized for participants to present their progress in both the business cases and the XML prototypes. They were assigned specific tasks that each agency team needed to perform between workshops. In addition, at each workshop there was a presentation by an expert from a private sector company, a Webmaster with experience using XML, or a university professor that highlighted some of the potential applications, capabilities, and limits of XML. At several of the workshops, discussions between and within the teams were fostered and there were some sessions for people in similar roles (content providers, content reviewer, web developers) to exchange ideas and concerns. Finally, communication among team members was encouraged with the objective of solving common questions and problems.

4.3.1. Comprehensive Prototyping Results

Through the development of their business cases, participants gained the necessary knowledge about their problem, business processes, relevant stakeholders, current executive support, resistance in some organizational units, among other important factors. Agency teams were also trained in the use of XML and were asked to develop a prototype based on their original agency proposals. Overall, they had the opportunity to further their knowledge about XML for content management, but this learning did not happen in isolation of their real organizations and the potential enablers and constraints they would face if a XML project would happen.

5. Discussion

5.1. Evolution of the Testbed Research Strategy

Comprehensive prototyping attempts to take into consideration the most important elements early in the process. The prototype is not created in isolation and then tested in the organization. The prototype and a greater understanding of the organizational environment are developed and co-evolved together, with the participation of all relevant stakeholders and the guidance of an external organization (CTG).

Environmental, institutional, legal, and organizational factors are carefully analyzed before and throughout the development of the prototype and business cases (or business analysis).

Several components or activities played an important role in the Testbed strategy. Each is a different aspect of the comprehensive prototyping experience. Some of these components were present in the first prototype; others have emerged through experience. The following list presents these components and provides some information about their evolution and use.

Well-defined and commonly agreed upon project objective. The amount of attention directed toward the development of a well-defined project objective component has increased over the three Testbeds. In particular, because the initiatives have become increasingly more complex involving more processes and more partners. In one project it took several iterations of group discussions to come to consensus on the specific objective – in particular in these cases because the technology of interest is emerging and not generally well-understood. A well defined business problem is a particular necessity in these initiatives.

Team Composition. The need to speak to users about system requirements has long been recognized by the IS community. However, the first Testbed went beyond this idea and required that users be members of the team. This core design element has enabled the comprehensive prototyping experience, at least in some cases, to be guided from a user perspective. The team work efficiently and do not have to go somewhere else to get understanding of context from a user perspective. The perspective existed on the teams and could quickly be factored into discussions and conclusions. The teams need to have the right members.

Awareness and Training. In the Groupware Testbed investments were made in building awareness about the technologies of interest and about the need for a comprehensive prototyping experience to guide decision making. In the debrief of the Testbed however, participants stated that awareness was not sufficient – training on both the emergent technology was necessary. Formalized technology training was added to the Internet Testbed. However, additional assessment data from that Testbed indicated that formalized training in the analytical tools used for examining context was necessary as well. In the XML Testbed technology training was provided for both technical and program staff. In addition, training on tools to manage the prototype project, to evaluate the technology and to examine the context was provided in multi-day sessions outside of the regular Testbed workshops.

Team Assignments. Team assignments emerged as a component of the comprehensive prototyping experience during the Internet Testbed. Agency teams were being asked to continue the use of the analytical tools beyond the time provided in the workshops. Some teams did and some didn't. Therefore, the gap between what some teams were learning and others widened. In the XML Testbed agencies were notified as part of the proposal process that teams would be expected to complete homework and present the results of the analysis in the workshops to their colleagues from other agencies.

Knowledge Sharing Culture. Participants in the Groupware Testbed noted that the design of that Testbed did not allow for the sharing of knowledge across agency teams. The Internet Testbed was therefore refined to include workshops where agency teams could both focus on their own initiatives and work together to build understanding of context. Workshop discussions were facilitated to ensure maximum sharing of knowledge, exercises and "homework" was completed with the expectation that everything would be shared through presentations among the participants. In addition, vehicles for knowledge sharing, such as listserves and discussion boards were developed and made available to participants in both the Internet Testbed and the XML Testbed. The evolution of this idea and its impact can be best seen in the XML Testbed. Two new types of discussions were regularly facilitated among team members (in relation to both the business case and the prototype) and among participants from different teams playing similar roles (ex., for XML for Web site management: content creators, content reviewers, Web developers).

Public Commitment of Deliverables. The visibility of the deliverables evolved over the three Testbeds. During the groupware Testbed the deliverables were primarily the prototype and then a comparative lessons learned report from the Center. By the XML Testbed participants were expected to produce both a prototype and a formal business analysis document. The delivery of this formal business analysis provided significant incentive to participants to focus on achieving as full an understanding of context as possible. In the first two Testbeds the analysis of the context we primarily focus on producing a robust prototype, by the XML Testbed, the focus had flipped and the development of the prototype was primarily pursued as a way to inform the most robust context analysis.

5. 2. The Testbed as a Research Venue

Comprehensive prototyping as a research strategy is able to recreate the context more realistically and obtain better and more accurate data about the features of the technology and the characteristics of the organization in which it is going to be deployed. Consequently, the results obtained from traditional research methods used within a Testbed framework provide more accurate knowledge about what would happen with an emergent technology in a specific situation.

Table 2. Examining emergent technologies in context

	Groupware Testbed	Internet Testbed	XML Testbed
Information and Data Factors	Medium	High	High
Information Technology Factors	High	High	High
Organizational and Managerial Factors	Low	High	High
Legal and Regulatory Focus	None	Medium	Medium
Institutional and Environmental Factors	None	Low	Medium

A Testbed as a research venue has the potential to improve our current capacity to do prospective analysis of emergent technologies. A Testbed design includes activities such as training, workshops, short presentations, group discussions, among others that can help participants to understand the technology in their own context. The Testbed design along with a range of data collection and analysis methodologies provides the most comprehensive understanding of context. Table 2 shows the insight gained through the specific use of the comprehensive prototype Testbed research strategy.

6. Final Comments

The comprehensive prototyping experience provides a robust venue for the formal evaluation of users' perceptions and understandings about a technology and the intended context for its use, providing new opportunities for empirical work on emerging technologies in context. The practical guidance delivered through training sessions and workshops provide participants with tools to analyze how technology will interact with their business processes and organizational environments. We argue that prototyping logic can be usefully applied within a comprehensive prototyping experience toward a goal of increasing the success of IT initiatives. Future research will be necessary to determine if comprehensive prototyping increases practitioner insight of the interactions between context and technology and if that new insight has the power to create a more robust research venue.

7. References

[1] Alavi, M., "An Assessment of the Prototyping Approach to Systems Development", *Communications of the ACM, 27*(6), 1984, 556-563.

[2] Boar, B. H., *Application Prototyping: A requirements Definition Strategy for the 1980s*, John Wiley & Sons, New York, 1984.

[3] Bostrom, R. P., and Heinen, S. J., "MIS Problems and Failures: A Socio-technical Perspective Part II: The Application of Socio-technical Theory", *MIS Quarterly, 1*(4), 1977, 11-28.

[4] Buchenau, M., and Suri, J. F., *Experience Prototyping*. Paper presented at the Symposium on Designing Interactive Systems Archive. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques, New York, 2000

[5] Clear Methods, *Rapid XML Web Services*, 2004.

[6] Dawes, S. S., Pardo, T. A., Simon, S., Cresswell, A. M., LaVigne, M., Andersen, D., and Bloniartz, P. A., *Making Smart IT Choices: Understanding Value and Risk in Government IT Investments*, Center for Technology in Government, Albany, NY, 2004.

[7] DeSanctis, G., and Poole, M. S., "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory", *Organization Science, 5*(2), 1994, 121-147.

[8] Fountain, J. E., *Enacting Technology: An Institutional Perspective*, John F. Kennedy School of Government, Harvard University, Cambridge, MA, 1995.

[9] Fountain, J. E., *Building the Virtual State. Information Technology and Institutional Change*, Brookings Institution Press, Washington, D.C., 2001.

[10] Gil-Garcia, J. R., *Do Citizens Expectations Matter for E-Government? Exploring the Determinants of the Functionality of State Web Sites.* Paper presented at the 2005 APSA Annual Meeting, Washington, DC, 2005, September 1-4.

[11] Gil-Garcia, J. R., and Pardo, T. A., *Multi-Method Approaches to Digital Government Research: Value Lessons and Implementation Challenges.* Paper

presented at the Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06) - Track 6, Kauai, 2006, January 4-7.
[12] Gil-García, J. R., and Pardo, T. A., "E-Government Success Factors: Mapping Practical Tools to Theoretical Foundations", *Government Information Quarterly, 22*(2), 2005, 187–216.
[13] Hammond, C., "The Wider Benefits of Adult Learning: An Illustration of the Advantages of Multi-method Research", *International Journal of Social Research Methodoloy, 8*(3), 2005, 239–255.
[14] Heeks, R., *Success and Failure Rates of eGovernment in Developing/Transitional Countries: Overview*, 2003, from www.egov4dev.org/sfoverview.htm
[15] Janson, M. A., and Smith, L. D., "Prototyping for Systems Development: A Critical Appraisal", *MIS Quarterly, 9*(4), 1985, 305-316.
[16] Kling, R., "Learning about Information Technologies and Social Change: The Contribution of Social Informatics", *The Information Society, 16*(3), 2000, 217-232.
[17] Kling, R., and Lamb, R., IT and Organizational Change in Digital Economies: A Sociotechnical Approach. In E. Brynjolfsson & B. Kahin (Eds.), *Understanding the Digital Economy. Data, Tools, and Research*. The MIT Press, Cambridge, MA, 2000.
[18] Kling, R., Rosenbaum, H., and Hert, C., "Social Informatics in Information Science: An Introduction", *Journal of the American Society for Information Science, 49*(12), 1998, 1047-1052.
[19] Kling, R., and Schacchi, W., The Web of Computing: Computer Technology as Social Organization. In *Advances in Computers* (Vol. 21, pp. 1-90), 1982.
[20] Kordon, F., and Luqi, "An Introduction to Rapid System Prototyping", *IEEE Transactions on Software Engineering, 28*(9), 2002, 817 - 821.
[21] Kraemer, K. L., Dutton, W. H., and Northrop, A., *The management of information systems*, Columbia University Press, New York, 1980.
[22] Kraemer, K. L., and King, J. L., "Computing and Public Organizations", *Public Administration Review, 46*(Special Issue), 1986, 488-496.
[23] Mingers, J., "Combining IS Research Methods: Towards a Pluralist Methodology", *Information Systems Research, 12*(3), 2001, 240-259.
[24] Mumford, E., "A Socio-Technical Approach to Systems Design", *Requirements Engineering, 5*(2), 2000, 125-133.
[25] Naumann, J. D., and Jenkins, A. M., "Prototyping: The New Paradigm for Systems Development", *MIS Quarterly, 6*(3), 1982, 29-44.
[26] Newman, M., and Robey, D., "A Social Process Model of User-Analyst Relationships", *MIS Quarterly, 16*(2), 1992, 249-266.
[27] Orlikowski, W. J., "The Duality of Technology: Rethinking the Concept of Technology in Organizations", *Organization Science, 3*(3), 1992, 398-427.
[28] Orlikowski, W. J., and Iacono, C. S., "Research Commentary: Desperately Seeking the "IT" in IT Research - A Call to THeorizing the IT Artifact", *Information Systems Research, 12*(2), 2001, 121-134.
[29] Orlikowski, W. J., and Robey, D., "Information Technology and the Structuring of Organizations", *Information Systems Research, 2*(2), 1991, 143-169.
[30] Pasmore, W. A., *Designing Effective Organizations: The Sociotechnical Systems Perspective*, John Wiley, New York, 1988.
[31] Poole, M. S., Jackson, M., Kirsch, L., and DeSanctis, G., *Alignment of system and structure in the implementation of group decision support systems.* Paper presented at the 1998 Academy of Management, San Diego, CA, 1998
[32] Reilly, D., Dearman, D., Welsman-Dinelle, M., and Inkpen, K., "Evaluating Early Prototpes in Context: Trade-offs, Challenges and Successes", *IEEE Pervasive Computing, October-December*, 2005, 42-50.
[33] Schmitt, J. W., and Kozar, K. A., "Managament's Role in Information Systems Developemnt Failures: A Case Study", *MIS Quarterly, 2*(2), 1978, 7-16.
[34] Scholl, H. J., Mai, J.-E., and Fidel, R., *Interdisciplinary Analysis of Digital Government Work (Birds-of-a-Feather Session).* Paper presented at the 7th International Conference on Digital Government Research (dg.o), San Diego, CA, 2006, May 21-24.
[35] Senn, J. A., "A Managament View of Systams Analysts: Failures and Shortcomings", *MIS Quarterly, 2*(3), 1978, 25-32.
[36] Sutcliffe, A., Gault, B., and Maiden, N., "IRSE: immersive scenario-based engineering with virtual prototypes", *Requirements Engineering, 10*, 2003, 95-111.
[37] Tashakkori, A., and Teddlie, C., *Mixed Methodology: Combining Qualitative and Quantitative Approaches*, Sage, Thousand Oaks, CA, 1998.
[38] Zhang, J., and Chung, J.-Y., "Mockup-driven fast-prototyping methodology for Web Application development", *Software Practice and Engineering, 33*, 2003, 1251-1272.
[39] Zhu, Z., "WSR: A Systems Approach for Information Systems Developemnt", *Systems Research and Behavioral Science, 17*, 2000, 183-203.