INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION, *14*(3&4), 503–534

The Use of Stories in User Experience Design

Dan Gruen
IBM Research
IBM Corporation

Thyra Rauch
Tivoli Software
IBM Corporation

Sarah Redpath
Lotus Development
IBM Corporation

Stefan Ruettinger
IBM Software Development
IBM Corporation

Stories capture the characteristics of the design space and audience that designers and engineers need to understand to build a complete and useful software experience. A story is a design communication tool that transcends the cultural divides of multidisciplinary teams and intertwines a technology with its user's goals. This article describes how stories are powerful tools in software design, defines the elements that make a compelling story, and presents the use of stories at IBM from the authors' experience. It also explores the benefits at each phase of the design process and how stories evolve throughout the design process.

1. INTRODUCTION

To design a system that will delight, provide value, and "feel right," designers and engineers need a deep understanding of the people for whom they are designing, their goals and values, the settings in which they live and work, and their activities. Stories capture this understanding in a rich and meaningful way, in a form that the various members of a product team can relate to and use.

The authors of this article thank Maida Eisenberg for her editorial assistance. We also acknowledge others who helped in creating the examples we used: Stephan Feger, Ursula Hart, Marion Behnen, Gary Macomber, and Tina Adams.

Requests for reprints should be sent to Dan Gruen, IBM Research, 1 Rogers Street, Cambridge, MA 02142. E-mail: daniel_gruen@us.ibm.com

Stories have been used successfully by a number of groups at IBM, and their use as an element of the product development process is growing. We represent three different groups who have found story-based methods valuable throughout the product lifecycle, from inspiring the earliest vision, through concept definition, requirements specification, design, prototyping, development, and product introduction. Stories have helped us ground the design of a new system in the users' business context and environment by immersing designers and developers in the situations in which their systems will be used. Stories help designers determine which functions will be useful; how they should be presented; and what integration with other tools, people, and information will be important. Stories can prompt innovation by revealing new opportunities to provide value to end users. Because they are powerful communication and teaching tools, stories can also help multidisciplinary teams work together, and help end users understand and discuss how a system would fit into their lives.

In this article, we describe what we mean by stories and outline the elements needed to build compelling stories. We explain why stories are a powerful artifact in human–computer interaction (HCI) work. We then give examples of how stories have been used by several different groups at IBM and describe how a story evolved during the different stages of a project process. We conclude by discussing issues to consider when using stories as design tools.

2. WHAT ARE STORIES AND WHY ARE THEY POWERFUL?

As the terms are used differently by different practitioners, it is important to define here what we mean by *stories*, and how we see them as differing from many *scenarios*. In general, scenarios focus on developing the right sequence of actions needed to capture and convey an activity, largely as a way of defining requirements. Scenarios around a technological artifact typically focus on the way a system is used to perform a specific task. Scenarios can vary greatly in level of detail, though many do not include detailed descriptions of the people involved in a task, their motivations, values, or goals. Scenarios also often lack the plot development and drama integral to a compelling story.

For example, a scenario showing the sequence of actions and screens used to fill out an online expense report might not give detail about who is using the system, how the expense report task fits into a broader context and goal, or why it is beneficial to use an online system. Scenarios have been used extensively in training (Schank, 1997) and in HCI for communication and requirements definition (Carroll, 1995; Rosson, 1999).

Stories, on the other hand, are very specific. They include fleshed-out characters and settings, dramatic elements, well-formed plotlines, and enough detail to understand the people who will use a system and the value it will bring to their lives (Burroway, 1999; McKee, 1997). It is important to note that the lines between scenarios and stories are by no means firmly drawn, and many practitioners create scenarios that do contain some story elements. Our point is that we have found elements like plot, drama, and character development—elements optional to a scenario but critical for a compelling story—extremely valuable in our HCI work.

2.1. Elements of Stories

A good story has detailed characters with whom the audience can empathize; rich, contextualized settings; goals (what the protagonist is trying to accomplish and why); causality; and obstacles (what problems the protagonist has to overcome to accomplish the goal). Dramatic elements such as time locks (constraints on the time in which the goal must be accomplished) or option locks (constraints on the actions or items that can be used to accomplish the goal) heighten the dramatic impact of the story (Brannigan, 1992; Burroway, 1999; McKee, 1997). It is interesting to note that many of the same elements required to tell a compelling story also are needed to design a useful and compelling system.

Fleshed-out characters. Good stories have fleshed-out characters with details that allow an audience to understand, relate to, and empathize with them. This includes having a sense of their values, fears, weaknesses, and overall goals: knowing what is important to them and what they would like to avoid. Similarly, in design, it is important to understand, in depth, the people who will use and be impacted by a proposed system. Understanding the broader goals and values of these people, and being sensitive to their limitations, is important as design decisions are made. For instance, knowing that a user might have a high school degree and will get only one week of training can guide decisions about the content and structure of the system.

Detailed settings. Compelling stories generally include details of time and place that help the audience situate themselves in the setting in which the story takes place. Similarly, in design, it is important to understand the environment in which a tool or system will be used and the other artifacts and activities with which it must fit. For example, if the designers understand that an environment is crowded, loud, and fast paced, they can address those sensitivities through systems that do not rely on audio feedback yet provide quick access to system information.

Goals and obstacles. The plots of compelling stories typically are based on a conflict or obstacle that the characters overcome to accomplish a goal. Similarly, a designer should have a clear sense of the goals accomplished or the problem solved by using a system. In essence, this is the *raison d'etre* of the system: why it is being built in the first place and how the proposed solution is better than others the user could choose.

Motivation. Compelling stories also require a sense that the actions of the characters are clearly motivated. It is critical for the audience to be able to understand the reasons for their behavior. Stories can provide a "motivation walkthrough" similar to a "usability walkthrough," in which designers ask not only if users can perform a function or use a feature, but why they would choose to do so. For example, the fact that certain users at customer support centers are timed by their management to determine their compensation shows one source of motivation and would likely be a key design factor.

Causality. Stories are more than lists of unrelated events. Events in stories are connected through causal relations, although these relations sometimes only become clear at the end. Similarly, design stories should show the proposed system playing a causal role in furthering the goals of the people who use it, not simply being used in an incidental way. For instance, in researching how users interact with software systems, we have discovered "peripheral tasks" outside of the systems, where users refer to company directories, written notes about solutions to problems, or whiteboard calendars to complete a task. Information from these sources, which are external to the system, and the actions associated with their upkeep, have everything to do with the user's actions with the system. This is a view where the user's complete model of the task, not just the system elements of it, is represented with all the relations intact. This view can introduce opportunities to incorporate into the software more of the information and artifacts of the complete task as the user sees it.

Dramatic elements. Dramatic elements and plotlines make stories interesting and emotionally engaging. Dramatic elements heighten the sense that something is at stake and reveal the characters' core values. Similarly, in system design, anticipating crises and critical situations can lead to systems that are focused and that support essential functions robustly in a variety of situations.

2.2. Our Growing Recognition of the Importance of Drama

The importance of dramatic elements in design stories came as a revelation to us. Initially, our sense was that our stories should focus on describing systems used in typical, unspectacular activities, as a way of emphasizing their value in everyday situations. There was a fear that including crises and other exceptional events would detract from the generalizability of the stories, making it harder for people to map them to their own, presumably more controlled domains. We have learned, however, that including dramatic elements such as time and option locks adds interest to stories and helps focus attention on the essential benefits of the systems they depict. Including such elements leads to stories that involve situations in which the tool will make a critical difference.

For example, a story created around a proposed expertise-location system began by showing the system being used to locate people to work on a sports Web site that would let viewers monitor the positions of athletes during an upcoming triathlon. The dramatic effect of the story (and the value of the system to its users) was heightened by the information that the person in charge of the project had advocated increasing the company's investment in Web technology and that he was under pressure to prove that his recommendation was a good one. A later scene involved one of the company's public relations representatives, who gets a call from a journalist asking her to comment by five o'clock that evening on possible health effects of the technology they were planning to use to monitor the athletes' locations. The public relations person had not heard of the project before, but knows she will be blamed if a negative story appears in the press. With sweat running down her spine, she uses the system to locate the people working on the triathlon project to determine if any health issues exist and to prepare a response for the journalist.

The Benefits of Using Stories in the Design and Development Process

Stories are valuable artifacts throughout the product design and development cycle for both cognitive and social reasons. From a cognitive perspective, stories represent events and experiences in a coherent way through schemas that capture the relations and structures connecting individual details (Schank, 1990). This facilitates comprehension and memory and enables the recall of similar events with similar underlying structure (Cohen, 1989; Klatzky, 1980). For example, a story of how a complaint with a toy was handled by one company might remind someone of how a problem with a room was resolved by a hotel, despite the fact that there is little overlap in the specific details. From a social perspective, stories are a key mechanism through which human experience has been shared for generations. Given the right social environment (such as a group of colleagues over lunch recounting encounters with problem customers), people can find sharing stories to be a natural, effortless, and compelling experience (Ochs & Capps, 2001). Stories are thus powerful tools not only for capturing the situations in which technologies will be used, but also for encouraging others to recall relevant situations from their own experience.

A focus point for customers. Because of their cognitive and social power, stories can be valuable tools for customer research and user feedback. Stories focus customers on their experience instead of on an unfamiliar technology, prompting more accurate, detailed, and honest feedback. A common reaction when someone is shown a story set in one domain is to say, "The same thing happens here, only it's different." This provides an opportunity to probe deeper, have the person tell the story of his or her similar experience, and discover places where a depicted technology might need to be modified. Stories can help uncover potential problems with a planned design that stem from personal or cultural differences; as people hear a story they can reflect on whether it feels natural and if they could see themselves doing the same thing. Field studies and ethnographic research familiarize designers with the specifics of the people and settings for which they design, much as an author would engage in background research before writing about an unfamiliar domain.

Due diligence. Because stories involve many of the elements that will be important for a successful product, the act of creating a story can be a way of ensuring that critical issues have been addressed. Constructing a story at the early stage of a project can ensure that the research homework necessary to design an effective solution has been done. Furthermore, it helps the team understand the results of such research, not just as individual pieces of data, but in a coherent, causally connected way. To phrase it somewhat harshly, if you cannot tell a compelling story about how a system you are designing will be used and the value it will bring to the people who will use it, you should question why the system is being built in the first place. In practice, we have found that much of the effort required to construct a story is effort that the product team needs to make anyway. Constructing a story can ensure that this work is done early in the process.

Prompts for innovation. Stories can be used to prompt innovation as designers, forced to consider individual users with specific goals, characteristics, and situations, uncover opportunities to enhance the user's experience in novel ways. Thinking about a specific person performing an activity in detail and asking, "How can my tool help the most at each point" can point to features and functionalities that were not part of the initial conception of the product. We have even found this approach to lead to innovation on the level of business model, when, for example, the team asks a question like, "Wouldn't our protagonist prefer to just get that function as an externally hosted service?"

Establishing a shared vision. Stories are also effective tools for teaming and establishing shared vision around a project and for addressing many of the problems associated with multidisciplinary software design. Stories are understandable across the different cultures and languages of software business. Because they are not the traditional tools of any one of the established software development disciplines, stories are readily accessible to all members of a multidisciplinary team. Using stories in multidisciplinary storyboarding sessions helps teams overcome politics and cultural differences to create a shared vision around the customer.

Story-based methods carry the customer goals as a reference thread throughout the product lifecycle. Used at all decision points, they can ensure a focus on providing customer value where technology would otherwise be the deciding factor (what's easier, more available, etc). They can also serve as a lens through which to see and explain business problems that otherwise would be overwhelming, and can be an effective tool for conveying the value of a design to marketing.

Compelling stories, compelling designs. Designing a story to show how a new offering will be used captures and represents these elements in a more cohesive, comprehensive, and meaningful way than do lists of functional requirements or proposed features. To some extent, a story can be seen as a template to ensure that the team has done due diligence on thinking through each of the elements required for a successful project and then combined them into a whole with the proper relations described.

Used in this way, stories and storyboarding serve not just as a tool for communication, but as a methodology for design. In essence, designing the story is really a way of designing the offering itself. With large, poorly constrained, multifaceted problems (multiple market segments, many possible product and service offerings, multiple business models, many dimensions of demographic and market data), it can be especially useful to focus in detail on designing for specific, fleshed-out, individual users and settings.

Management buy-in. A critical success factor for software development projects is that not only do the developers feel confident about the product (which is mostly based on technical details), but also that all levels of the management support it. Stories can help tremendously with that because their focus is not on technical details, but on a level that is easy for everybody to understand. All persons involved in the project, including development, marketing, planning, and management, can understand and agree on the common goal of the software development project that the story describes.

Starting point for other product externals. Stories are very useful in all phases of a software development project, from the initial research, task analysis, and audience definition to design validation. They can, however, be used even more—as a starting point for product externals such as tutorials and marketing flyers.

For example, after we used a story early in the development cycle to validate the task analysis and audience definition, we immediately recognized a lot of interest in this approach, especially from marketing and information development. Marketing often needs a way to explain complex software products in an easy-to-understand way to potential customers. In this case, the story was an excellent starting point for them to create flyers and marketing material. The same is true for online documentation and tutorials, where there is a need for easy-to-understand examples to explain the product. Therefore, a story can be used not only to support the software development process, but also as a perfect starting point for product externals.

Two Major Types of Stories

This article describes two approaches to creating stories and suggests when and how to use each. It is the reader's decision which approach would be best to accomplish his or her goals.

Fictional stories. Fictional stories are created by the design team based on the design team's best understanding of the goals for a future vision solution. Fictional stories can be useful when the design problem is a new space, for instance, when you are creating a solution for a problem that customers do not yet know they have. For instance, imagine trying to describe the usefulness of a television or a FAX machine to people before they ever anticipated the need for such a thing or had a concept of what those devices were. It would have taken a story to help customers see beyond their daily activities to the use and value of these inventions.

We have found fictional stories to be effective tools for eliciting feedback from potential users of a technology, even at its early stages. Fictional stories can benefit teams in the very early stages of vision and strategy phases, especially in areas of new technology and newly combined cross-divisional solutions. Because of the flexibility and immediacy of this approach, fictional stories offer a very attractive option for simply initiating the story-based approach and getting a team to think in story form. If used at the market definition phase where business cases are explored, the process of creative storymaking can uncover new business uses or potential audiences for a particular solution.

Although it is best to involve storymakers who have some understanding of the intended audience, a team that is somewhat naïve in this regard can offer completely fresh ideas. The recommended follow up when using this approach, however, to validate the stories and related storyboards with real customers before investing time and resources in further development.

Customer stories. Customer stories are based on customer experiences. One approach is to use the real stories and anecdotes you have gathered, as is. Another method is to visit multiple customers and extrapolate across common subjects to create a *metastory.*

Using a single story intact can work when you want to illustrate a specific case or when business processes are well established and you know that there is little difference in the practices between customers. The extrapolated approach is effective when you need a more encompassing description than exists at one customer site. It is also useful when there is a need to protect the privacy of customers and a directly recorded anecdote is too revealing. Finally, it is a good approach to use for generalizing the roles or tasks to orient customers away from individual details and validate a high-level process.

When the customer is intimately familiar with the tasks and the associated problems they encounter, it is best to take this data and use it to construct real customer stories. Customers have a mental model of their work, however inefficient that might be today, and it is best to respect this and use it as a place to begin, even if you change aspects of it in the end. Given their current tasks and processes, they will have lots of anecdotes about what they are currently doing, what they wish they could do, and what happens when things go wrong.

Later, when doing solution validation with them, the customer stories approach is also valuable. When customers recognize that they have a problem and can critique a proposed solution based on that experience, they can react concretely and specifically to this reference point and use it to tell us whether the solution will solve their problem or not.

Customer stories may require more planning than fictional stories; there must be the time and resources to plan and document real customer stories. However, we have found cases where the real customer stories are more dramatic and compelling than anything we could have imagined.

3. EXAMPLES OF THE USE OF STORIES

This section presents examples that illustrate the value of using stories at the various stages of product design and development. Ideally, stories can be used throughout all phases of a project. However, not all projects are ideal, and teams frequently need to choose the phases in which they can introduce the use of stories. For this reason, we have chosen specific examples of how we have used stories at various phases of the design process. In some cases we also include actual stories that we have used. The final example shows how stories were used throughout the entire design project and how the story evolved in each of the design phases.

Although we assume that the general phases and sequence of the design process we describe here are familiar, it is useful to point out that people take personalized approaches to it. You will see some evidence of our personal approaches in the following examples. And, because these examples come from the teams who created them, they are each personal accounts by nature.

3.1. Stories in the Design Process

For the purposes of this article, we define the four main phases of the design process as *research, design articulation, evaluation,* and *execution.* (See Figure 1 for a complete illustration of this process.)

Because stories come in many flavors, it is important to analyze the story content used at the various stages to determine if a particular story has the correct characteristics for the phase of which it is a part.

Stories in the research phase describe the problem as defined by the research into customers, technology, and business parameters. The stories resulting from the research phase that feed into design include who the customers are for this solution, what the technological requirements and timeframe limitations are, what is happening in leading solutions today, and so forth. We have called the stories used in the research phase *problem statement stories.*

For the design phase the team focuses on the solution, so aspects of their future vision will be apparent in a story used in this phase. We call the story in the design phase a *solution story.* The characteristics of a solution story are the same as a problem statement story, but with the pros and cons of the technical environment and the details of the flow and composition of a proposed solution added. A storyboard may become an added story artifact in this phase. A storyboard is a series of visual diagrams to illustrate what a user might see and do, accompanied by a narrative “script” of the

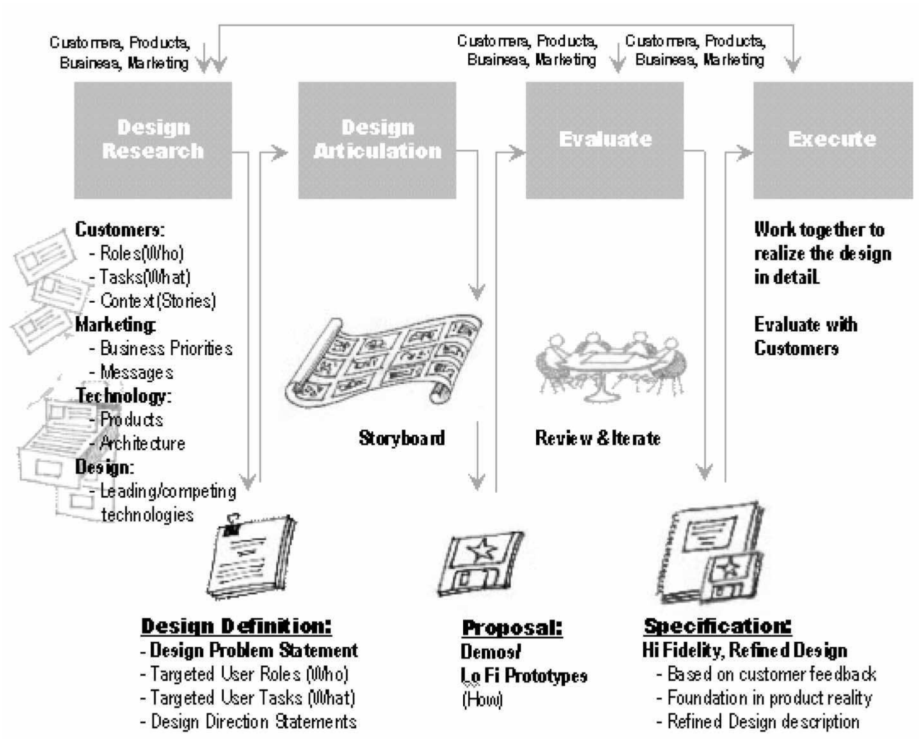

Customers, Products, Business, Marketing
Customers, Products, Business, Marketing
Customers, Products, Business, Marketing
Design Research
Design Articulation
Evaluate
Execute
Customers:
- Roles(Who)
- Tasks(What)
- Context(Stories)
Marketing:
- Business Priorities
- Messages
Technology:
- Products
- Architecture
Design:
- Leading/competing technologies
Storyboard
Review & Iterate
Work together to realize the design in detail
Evaluate with Customers
Design Definition:
- Design Problem Statement
- Targeted User Roles (Who)
- Targeted User Tasks (What)
- Design Direction Statements
Proposal:
Demos/
Lo Fi Prototypes
(How)
Specification:
Hi Fidelity, Refined Design
- Based on customer feedback
- Foundation in product reality
- Refined Design description

FIGURE 1 The design process.

story. Because of the benefits of exploring multiple solutions, the team may produce several storyboards or document multiple paths in a single storyboard.

During the evaluation phase, customers, product team leaders, marketing representatives, and key managers discuss revisions to the solution and make agreements and qualifications. These documented decisions bring the unique content to the story here. These rounds, which happen more than once and often include different groups, uncover new technical and task questions and problems to be solved. To articulate these changes, the team may choose to create more detailed descriptions and prototypes. They may also produce detailed visual design and information architectures. In the end, through this series of dialogs, what may begin as a solution story changes into what we call a *detailed product design flow.*

In the execution phase, stories and storyboards are used as a blueprint. For this reason, a story at this phase will have a high level of technical and customer detail. There will also be a level of agreement to the solution across the organization. Although it is still possible to encounter technological problems that would require fundamental changes to the solution, most of the core problems will have been covered by this point. With the technical assessment and details fleshed out, the story now becomes a *product design specification.* This can be thought of as a detailed prototype and accompanying scenario that focus on the steps the user would need to take to accomplish a particular goal. In addition to the technical details, the visual design and information architectures will have their own detailed designs as part of the product design specification. So although the story is still the framework, by this phase it is almost outweighed by all of the accompanying material produced to support it.

3.2. Research Phase: Aiding Customer Research

A common problem in gathering customer research is dealing with customers who want to jump right into the solution discussion. In this case, stories can focus them on the details that define the problem. For example, you can ask the customer to describe the last time they had a problem with "x" or something similar. This will usually produce a compelling story. Another strategy is to ask customers to imagine things by saying, "Let's pretend that resources and technology are no object. Now, if you could have anything you wanted to be able to do 'x,' what would it be, and what would it be like?" In response, they will tell you a sort of story. If they begin to describe what you have been designing, you can sketch something for them and ask if is this is what they had in mind.

3.3. Example 1: Conveying Findings From Customer Interviews

The following is an example of a metastory that was assembled from several customer experiences. The story was created, based on these real situations, by the customer research architect to represent all the elements of the software design problem at hand.

> Bill is a software distributor working for the IT [information technology] System group in a large global bank. Generally speaking, his job is to distribute software packages across functions at the bank. This can mean sending new versions of software out to the banks' employees (i.e., a new version of Office) or patches to existing versions of software when they are published by the software companies. Usually, these kinds of distributions can take place during the day with no disruption to banking operations. However, in this case, the software being distributed is a large core-business application and needs to be distributed after normal banking hours, when the check processing system can be shut down for 4 hr.
>
> Everyone expects this to be a routine distribution. However, at 2 a.m. on Monday morning, toward the end of the distribution, Bill finds himself staring at a flashing error message that is outside the scope of his knowledge and experience. He clicks on the help button, but it is no help. He goes to the online help for the distribution tool, and discovers that they do not publish the error codes online. He realizes that it might not be a problem with that software, but with some of the network software. Because Bill is not a network specialist; he is really in over his head.
>
> Bill knows that if Jack were here, this would be no problem because Jack is the team's source of obscure bits of technical trivia. Jack always knows the right person to call. Bill rushes to Jack's cubicle to rummage through Jack's sticky notes, stacks of tech-

> nical documentation, company procedures, contact numbers, and shelves of bound manuals. He calls some of the contact numbers, but no one answers. One of the contact numbers is no longer valid. He finally calls security to have them unlock Jack's desk to see if there are any more up-to-date clues on what do to, which wastes an additional 45 min. After 2 hr of searching, Bill gives up. Nothing there seems obviously applicable. He faces the cold fact that in 3 hr his management will be there, and there is no time left to complete the distribution. Bill's only hope is to get the old version of the software running in time for the check processing system to be up by the start of banking hours. If only the information he needed had been placed where Bill could find it easily.

Research phase: Task and audience definition. Task analysis provides core information on which to base an offering design. It focuses on understanding users, the environment in which they work, the tasks they currently perform, the tools they are using to do their jobs, and how they imagine they would like to do their work in the future.

Because stories are told from the perspective of specific individuals, they can ensure that a team considers the elements critical to the success of a system. Teams working from traditional design documents such as functional specs might otherwise ignore these elements. Following are two examples of projects in which creating story elements, including detailed characters and settings, problems to solve, and motivated actions led the team to see their audience and tasks in new ways.

3.4. Example 2: Defining Audience and Tasks for a New Product

Large organizations often purchase software through paid subscriptions that allow a specified number of people to use the software for a set period of time. Subscriptions are renewed periodically (typically annually), a process that often includes modifying the mix of products, number of users, and other provisions of the contract.

The group responsible for subscriptions for a major software product wanted to create a Web front end to allow customers to purchase and renew subscriptions, and to let customers manage their accounts online. They also wanted to open the subscription model to smaller businesses. Early on, the group decided to create a storyboard showing the various screens the customer would access to perform basic tasks such as selecting, verifying, and modifying details of an existing account; understanding alternatives; obtaining pricing information; and renewing subscriptions. They believed the storyboard could help justify funding for their planned Web design.

The group had an extensive flowchart (Figure 2) posted along the wall of their meeting area outlining the process the company followed internally to initiate and renew software subscriptions. They planned to use this process as the starting point for the structure of the Web site by adding a Web screen or dialog box for each task in the flowchart.

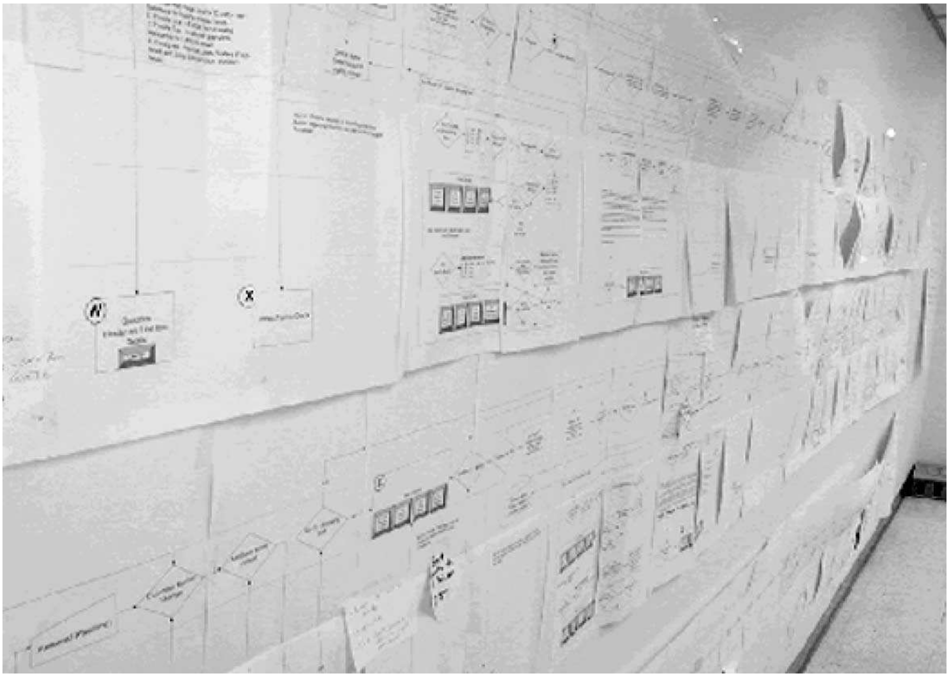


FIGURE 2 The flowchart covering an entire wall, depicting the internal process required to manage software subscriptions.

When we discussed creating a visual storyboard with this group, we emphasized that a critical element was getting the story right. This meant knowing in detail who would use the system, why they were interested in the software in the first place, what their concerns and goals were, and so on. When we first asked the group who their customer would be, they replied, “a small to medium-sized business,” citing the rather generic terms that appeared in their marketing studies. The customer’s goals were similarly described broadly, in terms such as “reducing cost” or “speeding access to information.”

As we worked with the group to define the detailed setting, characters, goals, and problems needed for a compelling story, questions arose that pointed to the need for the group to include participants from other departments (including sales and customer service) and to conduct a series of customer interviews.

Ultimately, we created a story around the fictional setting of a growing playground equipment manufacturer, which was representative of the intended users (Figure 3). The problems faced by the company included the cost of updating and maintaining printed catalogs and materials. The cast of characters in the story ranged from the company’s nontechnical CEO, to the accounting person who handled purchase orders, to a sales manager whose salespeople were complaining about the difficulty of getting up-to-date sales materials to their customers (Figure 4). Each of these characters was given a name, a face (through photographs), and a set of goals and concerns.

FIGURE 3 A depiction of the company created as the setting for the story around the new software subscription system. Though fictional, details were representative of actual companies and intended users.

A major contribution of the story was the realization that the team needed to think beyond a monolithic customer to consider the individual needs of the various people at a customer site who would be involved in purchasing and managing a software subscription.

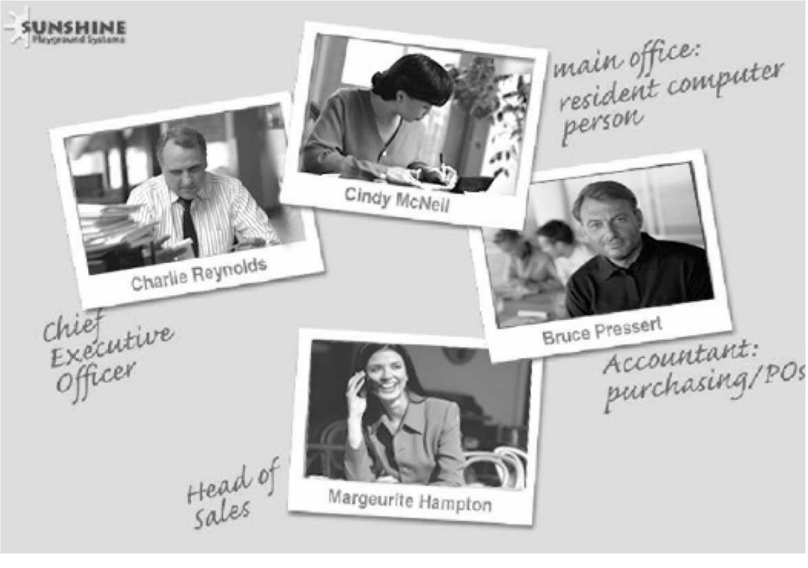


FIGURE 4 The characters in the software subscription story.

3.5. Example 3: Modeling Users and Tasks for a Follow-Up Product

The goal of this project was to deliver a follow-up version of MQSeries Workflow, a software tool that supports the modeling and execution of business processes and related IT and organizational infrastructure activities. This face-lifted product would reflect a changed marketing focus, address new market segments, and accommodate new usage scenarios.

It was important to know from the beginning of the new development cycle whether our existing task analysis and audience definition would still be valid for the changing market needs. The MQSeries Workflow product is very complex. It has many functions and different user interfaces for installation, configuration, administration, modeling, and monitoring. This was acceptable for the existing version of the product because different users in a company used different parts of the product. However, would this also be true for the new target audience? And, would the new users expect these different functions to be integrated into their existing software frameworks such as the administration console of Microsoft's Management Console presentation service?

Faced with these concerns, we needed to do more to define the audience and tasks than simply list them in a table. Instead, we wanted to verify our assumptions, not only with our existing customers, but also with users who did not have much experience with workflow software. To explain the complex product to the various target audience groups, we chose a story-based approach.

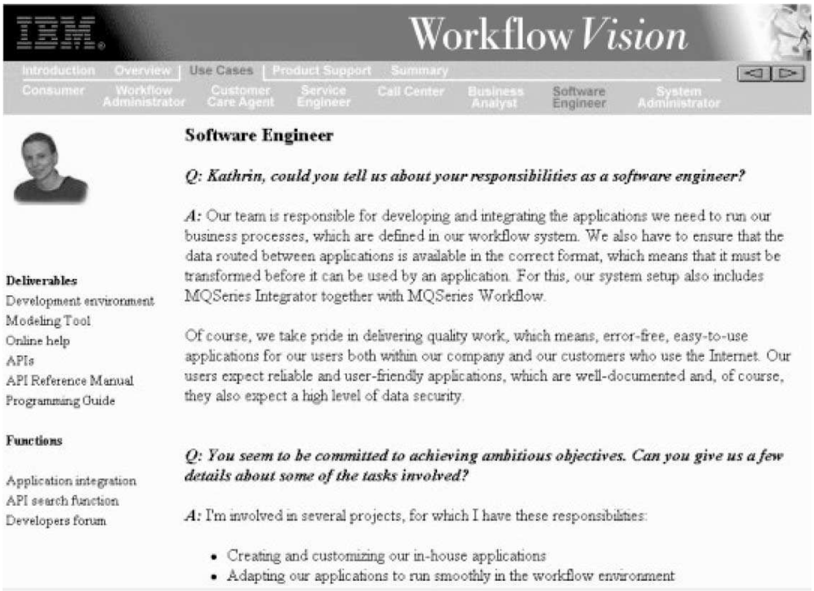
We presented the story using an interview style. In the story, we asked people (representing the new tool's audience) in a fictitious company what they wanted to do with the product (their tasks), what their current problems were, and how IBM MQSeries Workflow could help them with their job in the future.

Because we wanted the product vision to serve sales and support personnel as well as the primary users of the product, we also looked at the design of the product externals, such as marketing brochures, product presentations and Web pages—things that are often neglected when developing a new product. As we designed and wrote the story, we tried to incorporate as many of these product externals as possible. To enhance the story, we used existing screen captures or prepared mockups of the newly designed user interface to gather early feedback. The story was written in HTML so that it could be distributed and viewed readily, making it easy for people to evaluate the product concept. Figure 5 shows one of the pages of the story.

We asked existing and new customers as well as the sales, competence center, and support teams within IBM for feedback, both on the product and the overall story approach. The feedback we received was very positive and covered all aspects of the story; for example, ease of use of the defined tasks as well as the audience description and the relevance of details in the story.

Here is a sample of feedback from two of our customers:

> Your media is a very good way of communicating and connecting to the content very easily. Very good opportunity to pin down the content and design of the products to users' expectations and tasks.



Software Engineer

Q: Kathrin, could you tell us about your responsibilities as a software engineer?

A: Our team is responsible for developing and integrating the applications we need to run our business processes, which are defined in our workflow system. We also have to ensure that the data routed between applications is available in the correct format, which means that it must be transformed before it can be used by an application. For this, our system setup also includes MQSeries Integrator together with MQSeries Workflow.

Of course, we take pride in delivering quality work, which means, error-free, easy-to-use applications for our users both within our company and our customers who use the Internet. Our users expect reliable and user-friendly applications, which are well-documented and, of course, they also expect a high level of data security.

Q: You seem to be committed to achieving ambitious objectives. Can you give us a few details about some of the tasks involved?

A: I'm involved in several projects, for which I have these responsibilities:

- Creating and customizing our in-house applications
- Adapting our applications to run smoothly in the workflow environment

Deliverables
Development environment
Modeling Tool
Online help
APIs
API Reference Manual
Programming Guide

Functions

Application integration
API search function
Developers forum

FIGURE 5 Sample page of the story.

> I very much like the fact that the vision is driven from the user's expected experience. It describes what the vision is now from what the user experiences (performing their business tasks) are expected to be when the product is available.

The story helped our multidisciplinary development team gain new insights on their new audience and tasks beyond the feedback they had received on the existing product design. The story provided high-level descriptions, which helped us at an early stage to get a common understanding of what the product had to offer. This helped us check our assumptions, both with customers and within the development, marketing, and management teams. At the end of the process, the teams felt reassured and were glad to have a common vision for the new version of the product.

Design and development phases. Starting with design and continuing through development, practitioners from a variety of disciplines need to collaborate closely. This is where the value of a story in providing a shared vision and common language truly comes to light.

Different disciplines have different ways to approach a project based on their training, experience, and specific goals for the design process. To use three common players as examples, developers, marketers and human-centric designers often see the world through related lenses, but with vastly different emphases. Developers are the ones who will be tasked with actually building the software, and so they traditionally have a very pragmatic, schedule-based approach with functions and requirements as their baseline measurements. Marketing professionals focus on customer segmentation, pains, and messages. Human-centric design professionals exist somewhere in between, concerned with tasks and users. Good professionals in these areas will be passionate about the importance of their view of the world, and sometimes this passion can lead to conflict.

Stories provide a neutral avenue for representatives from each of these disciplines to articulate the key aspects of the concerns they see, and to weave them together into a shared vision. For instance, a solution story will have its context based on the audience targeted through marketing research, and will be based on the key pain points that the marketing arm determines. These pains will be detailed on a human level in the story through user and task descriptions. And, technical details from development will fill in these problem spaces described in the story with achievable solutions. Using the story based approach, all three disciplines can drive the design from their perspective. All three can satisfy their need to control that the design is on the right track from their perspective.

How does that happen? Stories help to "put a face" on the user for all the varied team members. Stories weave together laundry lists of requirements into a cohesive whole that reveals the customers' motivations and priorities. These motivations and priorities help designers make decisions as they discuss design trade-offs with practitioners from other disciplines. They help software engineers understand for whom they are developing and why. The concepts of personas and stories go hand-in-hand. Defining a persona (Cooper, 1999), or type of user, provides each member of the team with someone concrete to understand and identify with and helps him or her understand more about these types of users (e.g., their work environment, training, and preferences). Attaching stories to that persona makes them real. It becomes much easier to ask, "How would this design solve Sue's problem?" when put into the concrete context of a story.

Sometimes there may be several different audiences with specific tasks that must be addressed. In these cases, stories can treat each audience and task set as a distinct group, allowing designers and developers to focus on designing to meet each group's needs.

3.6. Example 4: Giving Functional Requirements Context and Priority for Developers

Working on EDMSuite, a tool for Enterprise Document Management, we visited a customer and followed several of their users around, watching them work. The customer had a number of content management products that dealt with online documents of various types, mostly forms and images. Our job was to find ways to integrate the products into a suite that would make it easier for users to perform their tasks than the individual products did. We discovered some wonderful integration opportunities in gathering our stories. One particular user, Sue, was working with an online document in one of the applications. At one point, Sue printed the document and walked over to the printer. She then picked up the printed document, walked over to the scanner, and scanned it into another product. When we asked why she had just done this, especially when both applications worked with the online documents, Sue replied that that was the only way she knew to get the document from one of our applications to the other.

Sue's story clearly illustrated to our team the importance of providing integration between the products in the solution. If the solution to the problem had been simply listed as just another requirement, the development team's motivation for implementing this feature might have been diluted. When we can all see why our solutions are necessary in the real world and how the requirement ties back to our customer's business, the urge to solve the problem becomes compelling.

Stories are also important in the design phase because they put things in user terms rather than technological or business terms, or in terms of any particular implementation or solution. Maintaining the user perspective in the early stages of the project provides the space and mindset to explore various solutions. In this way, stories can open the team to creative possibilities. Too often, design teams jump straight from an idea to its implementation without fully exploring whether there are alternative solutions or if the idea addresses the root problem to solve. Without guidance to the contrary, we naturally try to define things in terms that are familiar. Customer stories can help provide the guidance to go beyond initial conceptions.

3.7. Example 5: Stories as a Bridge Between Marketing and Development

This example illustrates the value of the story-based design approach in establishing strong communication links to marketing. Communicating product direction from one discipline to another is not an easy task. Everyone is trying to do the right thing, yet the differences in language, approach, and intent between marketing and development can create stressful friction at times. Even more dangerous is the fact that we sometimes found ourselves building functions to solve the wrong customer problem. This is where we have found stories and storyboarding to be unexpectedly useful.

At Tivoli, the marketing organization creates a market requirements document (MRD), which outlines the product's market, business problems, and goals. Ideally, the MRD should outline the parameters of the problem, including the constraints, at a level of detail that will leave the design possibilities open for the design process. However, it is a challenge for the authors of the MRD to maintain an objective view of the problem space when they receive input from vocal executives and customers about what the solution should contain. Technical details often creep into the MRD and become product directive, even though the underlying problem might be solved better in another way.

This example describes how the Tivoli Presentation Console design team introduced a skeptical marketing team to the idea of story-based design and as a result gained powerful advocates for the approach. Initially, the relation was tightly managed by the marketing executives who were very concerned about our interaction with customers and wanted to have a great deal of control over every conversation. As is appropriate, we were directed by marketing to focus first on the change management problems as the highest level of market opportunity. Our marketing team helped us select the customers to visit, scope the problem space, and guide the main points of the research.

Because the marketing team was understandably cautious about including new people in their customer relationships, we needed to gain their confidence in our ability to work with their customer contacts. We were careful to include marketing in each phase of the design. For example, we formally presented and gained their approval for our research approach with customers, got their buy-in on the resulting stories, and invited them to participate in the storyboarding. Later, when they saw that the type of research we were doing had nothing to do with selling solutions and was focused on the problems their customers were having, they trusted us to manage the customer communication and resulting documentation without their constant presence.

We maintained this close communication with the marketing team while we framed the design goals. When marketing saw the resulting storyboard, it was clear that it illustrated the customer problem in a way that they understood and approved. Moreover, it illustrated a high-level direction for a solution that they could easily understand. Whereas the MRDs of the past contained laundry lists of technical requirements, this approach, which was created in partnership, provided a vision of the design as a whole. Marketing was ecstatic at how the storyboard helped them communicate a cohesive vision clearly to customers, as well as to others in the company. And, because this storyboard was created in collaboration with the development leads, it was technically feasible, accurate, and apolitical.

The storyboard then became the MRD. Gone was the thick text document, replaced by an illustrated flow created by the multidisciplinary team. Although it included technological information that made it real, understandable and applicable to the development audience, it did not constrain the development teams to an unnecessary level of detail. It showed a general design direction while the actual detailed design was left to be solved by the experts. Our marketing representatives went so far as to require that all future Tivoli MRDs be in the form of a storyboard. They had become not only converts, but advocates.

Using stories through all phases of the design process. In the introduction to this section, we defined the types of stories that can be used in each phase of the product process. The following, extensive example gives the reader a view into a possible approach to an overall story-based design process. It also illustrates how one story is transformed from a problem statement story to a product design specification by the different needs and considerations in each phase of the process. The reader can see here the details and artifacts resulting from each phase that make the transformation process valuable, while maintaining connection to the original customer accounts.

3.8. Example 6: A Story Evolves Throughout Design

This example illustrates the evolution of one of a series of stories through the design process for the Tivoli Presentation Framework Console. Because our customers needed to use the console with many different products, we had to look at how information, security, and decisions were affected from product to product and from user to user. As a result, our stories spanned several products and involved many users who interacted. The integration layer between products is an area with a high risk of design failure and, therefore, one where stories can be their most powerful.

Our story comes from research on the entire flow of how large enterprises plan and execute the deployment of new software into existing business systems. In doing this research, we observed the exchange among eight types of users of Tivoli software: the executive administrator, coordinator, application expert, inventory expert, package builder, distributor, Level 3 help, and end user. Although our final story spanned eight pages and all eight of the aforementioned roles, we limit our example here to pieces of the distributor, change coordinator, and Level 3 tasks. An ellipsis (…) indicates portions that have been omitted.

Our original problem statement story is the result of exploring the problem during the research phase. It begins this way:

> Banking U.S.A.'s executives have mandated that everyone in the corporation will have PeopleSoft on their machines by June. This company has a well-defined process for scheduling changes, so … .
>
> The distributors, George, Jake, and Ian, go through the process of doing the detailed schedule of the software distribution for next weekend after the affected employees have gone home. They determine which machines need which software packages. They now have to go check to make sure that the targeted machines are available. There are many details in this task, such as how much bandwidth they have available, decisions on what to do about single failures, and doing a contingency plan.
>
> Next weekend, Ian, the distributor on duty, gets notified that the distribution is due to start automatically in 1 hr. As the distribution is happening, there is a team, including the change coordinator, Bob, who will check the progress of the distribution. Ideally, Bob would like to be able to monitor the distribution remotely and intervene during the process if necessary.
>
> As it turns out, one of the repeaters fails during the first hour. Ian sees the problem on his display, opens a problem ticket, and calls the Level 3 on duty. Stan, the Level 3, has already detected a problem with a server from his monitors. This server, which contains the repeater in question, has crashed. Stan reboots the server, checks to see if the problem is resolved, and closes the problem ticket … .

The story is transformed into a solution story through the design articulation phase, when our multidisciplinary team of human factors, visual design, development, information architecture and marketing representatives described the solution they envisioned. It was key that all or most of their perspectives were incorporated in real time at this stage because each had unique creative input, and their exchange enriched each other's ideas and the total solution. In our example, we began with the customer story and added the solution proposals at a high level to produce a storyboard. The skeleton of the storyboard script was the original customer story; the flesh came in the form of additional technical details.

The solution story that follows is actually our second try, and this gives us the opportunity to make an important point about this approach. It is vital to have a solid mechanism for getting early customer feedback on design direction. Our design team members had done a lot of early thinking about the change management space, and the design direction contained a lot of well-founded preconceptions. Once we had a storyboard and script for our original proposed solution, we took it to customers to validate. Because the design direction was so clearly articulated in the storyboard, it was easy for the customers to understand this solution in terms of their environment and business processes. The customers did not agree with our initial proposed design direction and gave us specific feedback about where it fell short. In fact, they proposed a fairly radical redirection of our efforts. Because the design was simply a storyboard at this point, it was easy for the team to change direction. The importance of this lesson cannot be overstated. Even with a very experienced, knowledgeable, and creative cross-disciplinary team, we were reminded that we are not the experts; our customers are. No matter how good a team is, sometimes design instincts are wrong, and story-based design helps detect that early and economically. Instead of wasting valuable time and resources producing a product that would not have met customer needs, we were able to change course quickly, develop another storyboard and script to illustrate a different proposed solution, validate it, and move forward—all before any code was written.

Our second storyboard and accompanying narrative were as follows:

> … . after Sally, the executive administrator, creates the change ticket … .
>
> The distributor, George, gets a beep and opens his to-do item. Here, he has not only the software packages and the endpoints, but also the calendar reference and announcement note generated by the change coordinator, Bob. When George opens the distribution scheduler, he sees the master schedule. He should be able to drag and drop the packages onto the schedule. He uses Bob's calendar as a reference to know where to target. Henry, the package builder, has already created the packages to know what their bandwidth requirements and other constraints are. Therefore, George can overlap the desktop client installs so they run simultaneously with the server installs but independent of them. The Web client "knows" which must be done before others can be completed. Further parameters and constraints are set, as well as notification preferences. George can drag the announcement out onto the calendar as well so that it will be sent out 1 or 2 days ahead of the distribution. All this will happen 3 weeks from that point (see Figure 6).

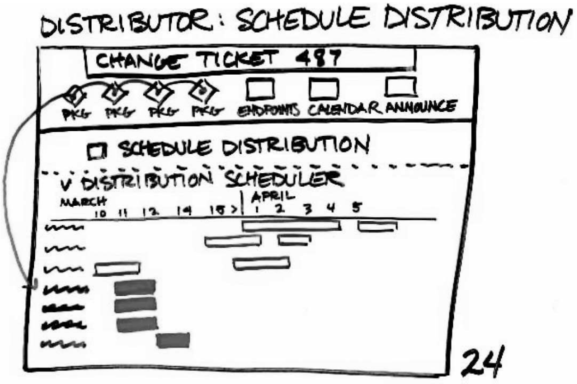


FIGURE 6 Distributor's view—schedule distribution.

Three weeks later, Bob is monitoring some distributions at home from a Web browser. From here he has a go–no go capability to stop the distribution for any reason. He can monitor the progress of the distributions. He also has quick access to the distributor's beeper number and can chat with the distribution team. The chat is contextual with whichever distribution is selected. Different distribution teams may work on different items, and he would want to chat with more than one potentially at one time (see Figure 7).

Meanwhile, back at the office, Ian, the distributor on duty, is doing some job scheduling when his to-do list beeped. He opens it up and sees that the Peoplesoft distribution is scheduled to take place that day. He also sees (Figure 8) that he has several other distributions that he needs to monitor … .

Bob is still monitoring from home. He may not have the fancy topology view available from the office, but at the very least he should be able to see (Figure 9) the same information about progress indicated in a table.

At this point, the distribution runs into a snag. One of the repeaters stops working. The software distribution product becomes aware of this event and makes Ian aware through the topology view. He also gets important error details on his Tivoli Assistant card. It tells what happened as well as the impact and gives him the option of opening a problem ticket. Ian creates a problem ticket and sends it on. Although we do not see it, we note that the problem ticket should be automatically populated with contextual information such as the name of the trouble repeater and the submitter name, based on the fact that the ticket is launched from this error message (Figure 10).

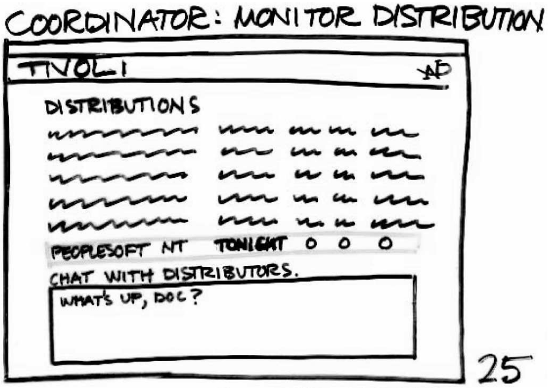
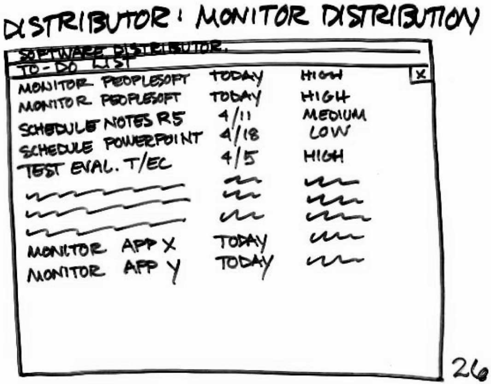

COORDINATOR: MONITOR DISTRIBUTION
TIVOLI
DISTRIBUTIONS
PEOPLESOFT NT TONIGHT O O O
CHAT WITH DISTRIBUTORS.
WHAT'S UP, DOC?
25

FIGURE 7 Coordinator's view—Monitor all distributions.



MONITOR PEOPLESOFT	TODAY	HIGH
MONITOR PEOPLESOFT	TODAY	HIGH
SCHEDULE NOTES R5	4/11	MEDIUM
SCHEDULE POWERPOINT	4/18	LOW
TEST EVAL. T/EC	4/5	HIGH
MONITOR APP X	TODAY	
MONITOR APP Y	TODAY	

FIGURE 8 Distributor's view—Monitor all distributions.

Stan, Ian's favorite Level 3 guy, receives the problem ticket and opens it. On the problem ticket itself, he has an attachment to the troublesome repeater, which has a red status (Figure 11). Directly from there, Stan should be able to do appropriate corrective action for a downed repeater, such as remotely rebooting the machine it sits on. After the reboot, the repeater is green, and Stan gets a message that it is back up and running. He has the option here to close the ticket or go on and do more diagnostics using a wizard, depending on which answer he gives … .

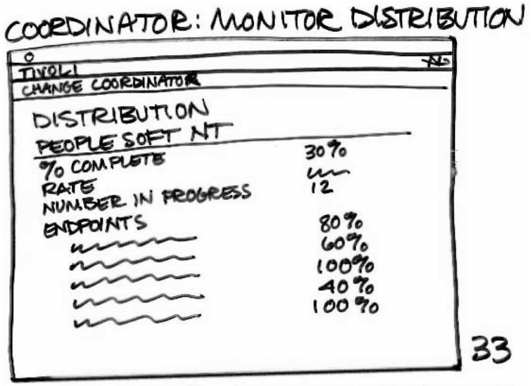


FIGURE 9 Coordinator's view—Monitor single distribution.

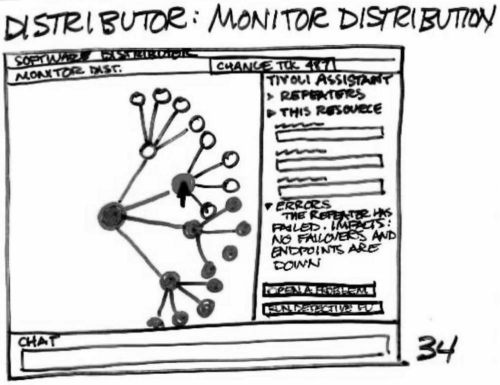


FIGURE 10 Distributor's view—Monitor single distribution.

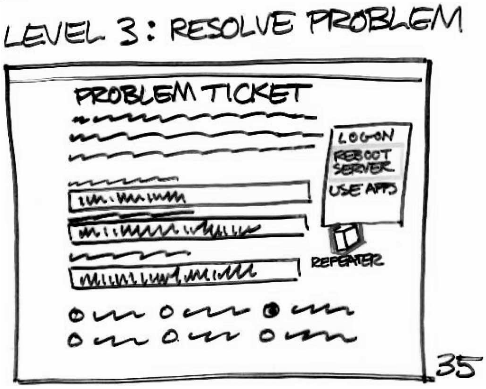


FIGURE 11 Level 3's view—Resolve problem.

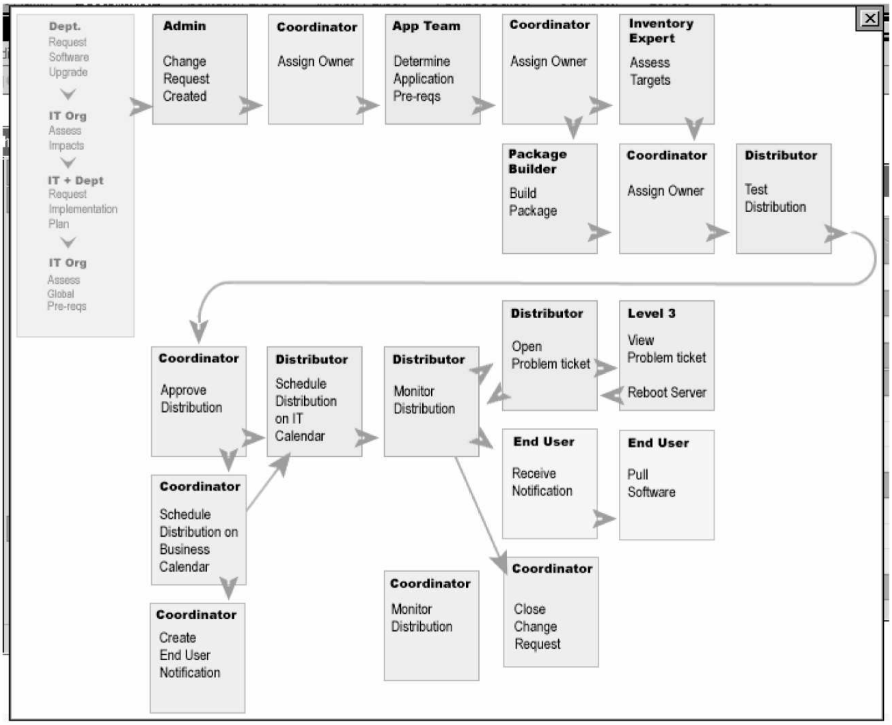


FIGURE 12 The process flow.

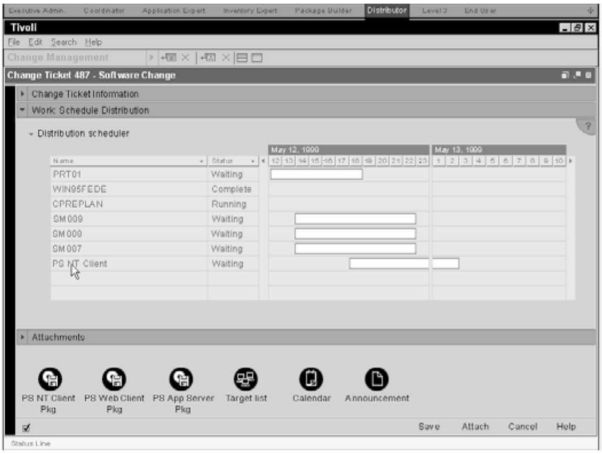
In the previous example, the lack of detail in both the words and the pictures is intentional. That detail will come later in the process. Note also that we picked a task flow that we thought would have broad applicability to our customers, as a major goal was to present a proposed solution to gather customer feedback.

Next, in the evaluation phase, the team mapped out the process flow from person to person, as this was considered to be critical to the whole solution (see Figure 12). Through evaluation with executives, customers, and extended team members, the solution story changed into a detailed product design flow. Although the form of the story did not change significantly during this phase, it underwent significant revisions. At the same time, we were also producing a version of the visuals that would include more details on the content represented, some initial interaction cues and flows, and initial style concepts. Where the original storyboard might have only a box with squiggly lines to indicate a control, in this phase the designers determined what kinds of objects and controls to use, what the relations between them should be, and initial directions for visual design and information architecture.

We rendered the storyboard in more detail and added interactivity using Macromedia Director, using essentially the same narrative. Many of the details we added were pulled from real customer examples and things they said they wanted to see. Later, you can see an illustration of some of these pieces of the detailed product design flow.

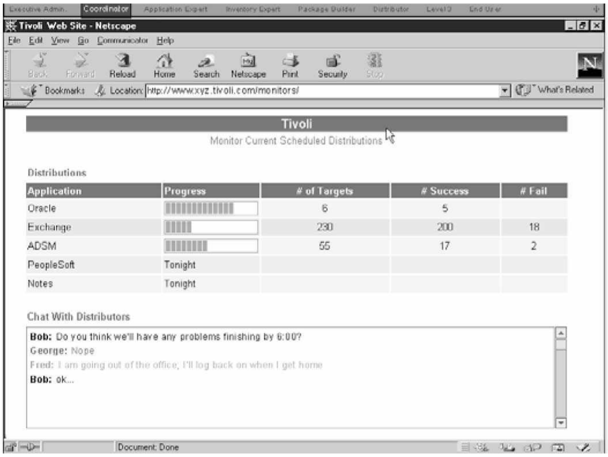
… .George, the distributor, gets a beep and opens his to-do list. Here, he has not only the software packages and the endpoints, but also the calendar reference and announcement note generated by Bob, the change coordinator. When he opens the distribution scheduler, he sees the master schedule. George can drag and drop the packages onto the schedule. He uses Bob's calendar as a reference to know where to target. Henry, the package builder, has already created the packages to know what their bandwidth requirements and other constraints are. Therefore, George can overlap the desktop client installs to run simultaneously with the server installs, but independent of them. However, the Web client "knows" that it cannot happen until the others are finished. Further parameters and constraints as well as notification preferences are set. George can then drag the announcement out onto the calendar as well, so that it will be sent out 1 or 2 days ahead of the distribution. All this will happen 3 weeks hence (Figure 13).

Three weeks later, Bob is monitoring some distributions at home from a Web browser. From here, he has a go–no go capability to stop the distribution for any unexpected reason. He can monitor the progress of the distributions, as well as have quick access to the on-duty distributor's beeper number, and can chat with the distribution team. The chat is contextual with whichever distribution is selected. Different distribution teams may work on different items, and he might want to chat with more than one team at a time (Figure 14).



Name	Status
PRT01	Waiting
WIN95FEDE	Complete
CPREPLAN	Running
SM 009	Waiting
SM 008	Waiting
SM 007	Waiting
PS NT Client	Waiting

FIGURE 13 Distributor's view—schedule distribution.



Tivoli

Monitor Current Scheduled Distributions

Distributions

Application	Progress	# of Targets	# Success	# Fail
Oracle	▮▮▮▮▮▮▮▮▮▮▮▮▮	6	5	
Exchange	▮▮▮▮▮	230	200	18
ADSM	▮▮▮▮▮▮▮▮	55	17	2
PeopleSoft	Tonight			
Notes	Tonight			

Chat With Distributors

Bob: Do you think we'll have any problems finishing by 6:00?
George: Nope
Fred: I am going out of the office, I'll log back on when I get home
Bob: ok...

FIGURE 14 Coordinator's view—Monitor scheduled distributions.

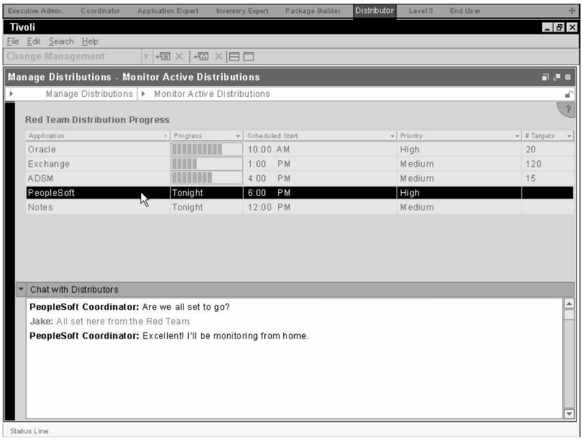
Meanwhile, back at the office, Ian, the distributor on duty, was doing some job scheduling when his to-do list beeped. He opened it up, and sees that the Peoplesoft distribution is to happen today (Figure 15). He also sees that he has several other distributions that he needs to monitor … .

The Peoplesoft distribution begins, and Ian is monitoring it (Figure 16) … .

Bob is still monitoring from home. He may not have the fancy topology view available from the office, but at the very least he should be able to see the same information about progress indicated in a table (Figure 17).

At this point, the distribution runs into a snag. One of the repeaters stops working. The software distribution product becomes aware of this event and makes Ian aware through the topology view. He also gets important error details on his Tivoli Assistant card. Here, it tells what happened as well as the impacts and gives him the option of opening a problem ticket, which he does. Ian creates a problem ticket and sends it on. The problem ticket was automatically populated with contextual information such as the name of the trouble repeater and the submitter name, based on the fact that the ticket is launched from this error message (Figure 18).

Stan, his favorite Level 3 guy, receives the problem ticket and opens it. On the problem ticket itself, he has an attachment to the troublesome repeater, which knows enough to have red status. Directly from there, Stan should be able to do appropriate

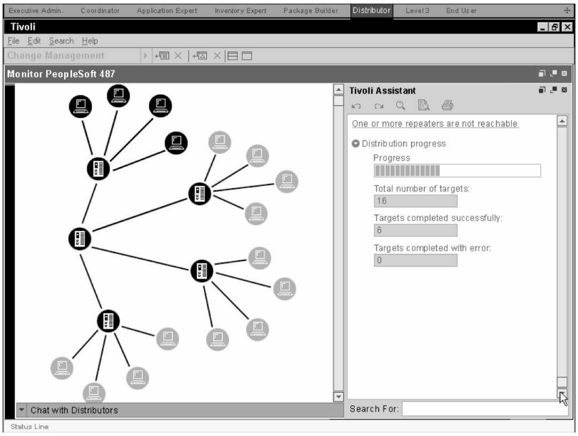


Red Team Distribution Progress

Application	Progress	Scheduled Start	Priority	# Targets
Oracle	▨	10:00 AM	High	20
Exchange	▨	1:00 PM	Medium	120
ADSM	▨	4:00 PM	Medium	15
PeopleSoft	Tonight	6:00 PM	High	
Notes	Tonight	12:00 PM	Medium	

PeopleSoft Coordinator: Are we all set to go?
Jake: All set here from the Red Team.
PeopleSoft Coordinator: Excellent! I'll be monitoring from home.

FIGURE 15 Distributor's view—Monitor active distributions.



One or more repeaters are not reachable

Distribution progress

Progress

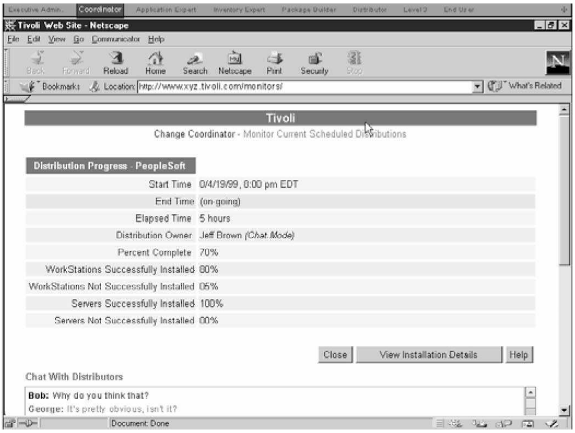
Total number of targets: 16

Targets completed successfully: 6

Targets completed with error: 0

FIGURE 16 Distributor's view—Monitor PeopleSoft distribution.

corrective action for a downed repeater, such as remotely rebooting the machine it sits on. After the reboot, the repeater is green, and Stan gets a message that it's back up and running. He has the option here to close the ticket or go on and do more diagnostics using a wizard, depending on which answer he gives (Figure 19) … .


Executive Admin.
Coordinator
Application Expert
Inventory Expert
Package Builder
Distributor
Level 3
End User
Tivoli Web Site - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks
Location: http://www.xyz.tivoli.com/monitors/
What's Related
Tivoli
Close
View Installation Details
Help
Document: Done

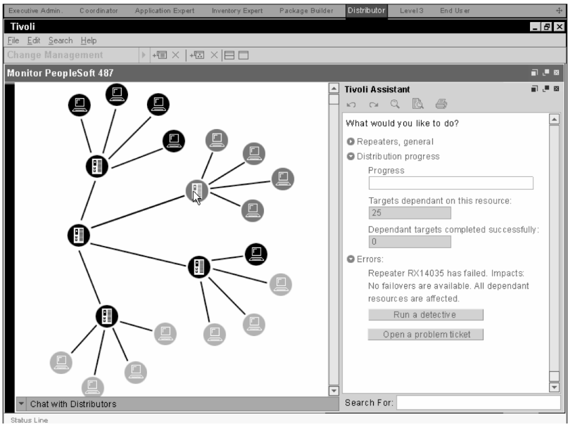
Change Coordinator - Monitor Current Scheduled Distributions

Distribution Progress - PeopleSoft	
Start Time	0/4/19/99, 8:00 pm EDT
End Time	(on-going)
Elapsed Time	5 hours
Distribution Owner	Jeff Brown *(Chat Mode)*
Percent Complete	70%
WorkStations Successfully Installed	80%
WorkStations Not Successfully Installed	05%
Servers Successfully Installed	100%
Servers Not Successfully Installed	00%

Chat With Distributors

Bob: Why do you think that?
George: It's pretty obvious, isn't it?

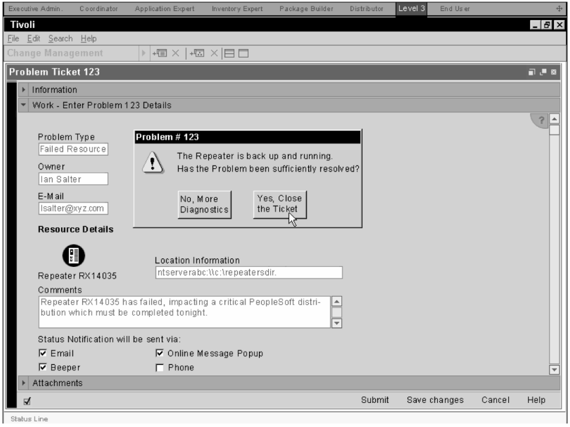
FIGURE 17 Coordinator's view—Monitor PeopleSoft distribution.



What would you like to do?

- Repeaters, general
- Distribution progress
  - Progress
  - Targets dependant on this resource: 25
  - Dependant targets completed successfully: 0
- Errors:
  - Repeater RX14035 has failed. Impacts: No failovers are available. All dependant resources are affected.

FIGURE 18 Distributor's view—distribution problem.



Problem Type: Failed Resource
Owner: Ian Salter
E-Mail: Isalter@xyz.com

Problem # 123
The Repeater is back up and running.
Has the Problem been sufficiently resolved?

Resource Details
Repeater RX14035
Location Information: ntserverabc:\\c:\repeatersdir.
Comments: Repeater RX14035 has failed, impacting a critical PeopleSoft distribution which must be completed tonight.

Status Notification will be sent via:
☑ Email ☑ Online Message Popup
☑ Beeper ☐ Phone

FIGURE 19 Level 3's view—problem ticket.

For the final incarnation of the story in the execution phase, namely the product design specification, the physical artifacts depend on how the individual product development team works.

In our case, the development team adopted the storyboard and reverse-engineered it to determine what the technical requirements would be and who would work on it. Initially, they worked directly from the hand-sketched storyboard illustrations and added their own notes to the panels. (Unfortunately, no documents showing this use of the storyboard still exist.) As the design progressed, we produced many more artifacts to explore details of the design. These examples illustrate the level of discussion that occurred between the designers and developers during this phase.

Figure 20 is a section of the design specification that documents details of interaction. These details were all based on information from the original customer research and stories.

> Universal Navigation and Windows Management
>
> Users can alternate between these views through some interface element, be it the "view" item on the menu bar or a toggle switch on the toolbar of the console. All of these views should be available, and the console should remember which view each user most recently used.
>
> The MDI "workspace" model: (Figure 20).
>
> In which there are several subordinate windows floating in a master navigation workspace or window.

In addition, there was a need to decompose and label the overall design so it could serve as a roadmap for the cross-divisional teams of developers who were learning the design and would need to code to it. Figure 21 shows one example of the design in progress, with labels showing what the individual elements were and how they would be used.

Such details as how the status line should be rendered are covered at this stage. For the developer, every small design element has many design questions that, ideally, they will want to discuss with the entire design team. Figures 22 and 23 show what existed already and the proposed redesign. Many detailed design questions

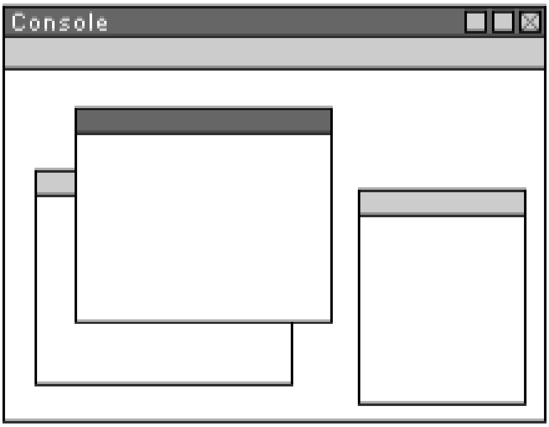
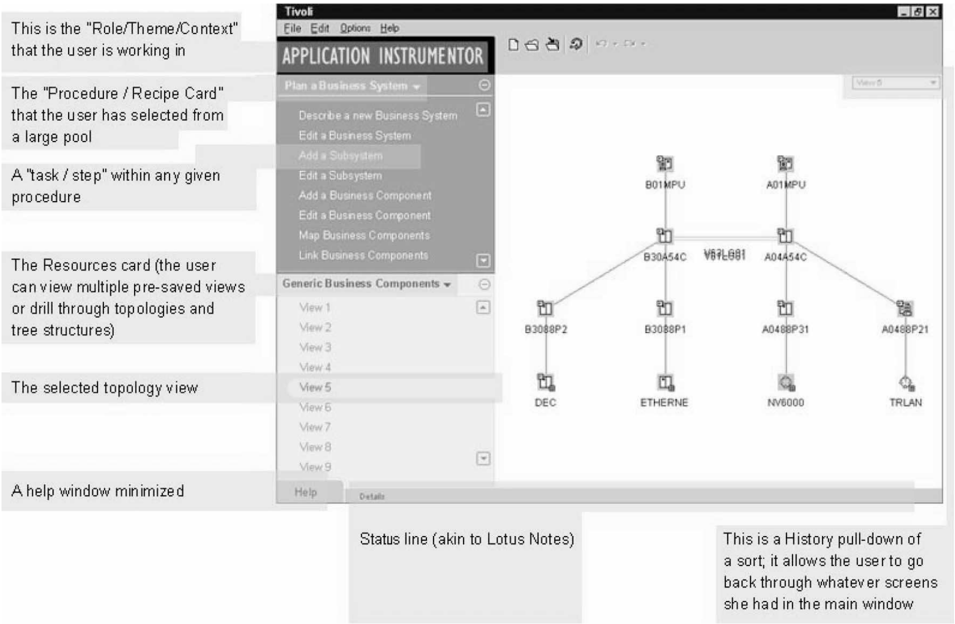


FIGURE 20 Illustration of design specification.



This is the "Role/Theme/Context" that the user is working in

The "Procedure / Recipe Card" that the user has selected from a large pool

A "task / step" within any given procedure

The Resources card (the user can view multiple pre-saved views or drill through topologies and tree structures)

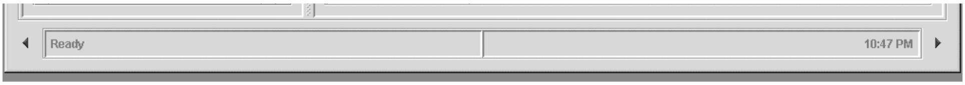
The selected topology view

A help window minimized

Status line (akin to Lotus Notes)

This is a History pull-down of a sort; it allows the user to go back through whatever screens she had in the main window

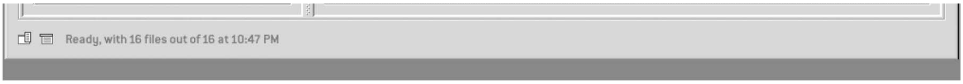
FIGURE 21 Design in progress.



Ready

10:47 PM

FIGURE 22 Status line, before.



Ready, with 16 files out of 16 at 10:47 PM

FIGURE 23 Status line, after.

arose in this phase, such as, What information should be presented? Should the information be in formal or informal voice? Should there be icons and how should they look? Our stories helped here as well.

4. CONSIDERATIONS WHEN USING STORIES IN DESIGN

Three considerations frequently arise when using stories as design tools. The following recommendations are based on our own experience and conversations with teams who were considering using stories in their design work but expressed common reservations.

4.1. Accuracy

As discussed previously, a story used to help guide development and design must accurately reflect situations and concerns in the real-world settings in which the technology will be used. Even if not true in every detail, the essential elements that would impact the design should be accurate. A common fear is that a team will be led astray designing to a story based on a naïve or incorrect view of what really happens in the user community. To protect against this, it is critical that stories be informed by significant real-world observation. In the case of fictional stories, this means involving informants who are knowledgeable about the task domain and setting in crafting the story. Equally important is the need to validate the story by reviewing it with people in the task domain and modifying it to reflect their experiences.

4.2. Generalizability

A related concern is the extent to which stories are generalizable and reflect the breadth of situations in which a technology will be used. Because stories are, by definition, about specific users and settings, there is a fear that a story could lead a team to focus on a single setting and activity at the expense of others. On the other hand, there is a conflicting fear that without a story, it would be hard for the team to really understand how the technology they are building will be used. There are several ways of addressing these concerns:

1. Use the results of field and marketing studies, demographic analyses, and other sources of data to understand the range of domains and situations for which the technology is expected to be used. Use that knowledge to guide the selection of settings and activities for the stories you create.
2. Do not rely on a single story. Instead, design several stories to ensure that you cover the key settings, domains, and tasks in which the technology will be used. Working with a series of different stories can point out the extent to which a single tool can satisfy all the broad needs and the extent to which the tool must be flexible and customizable.
3. Understand that stories are not the only tools to use in the design process. They should be accompanied by field studies, usability testing, surveys, and the like.

4.3. The Ease of Using Stories Effectively

Although most people can understand, appreciate, and empathize with a good story, not everyone knows how to write one. Having people with experience using stories on the team is valuable. Fortunately, we have found that people can learn to be better storytellers and storycrafters through practice and training, and by becoming sensitive to the elements that lead to effective stories. People who have used stories in the past can help the team understand the right level of detail for characters, settings, and plot lines. They can help ensure that plot lines effectively probe the value of the tools being designed and the broader motivations and goals that they are meant to accomplish. They can also help determine the most effective way of representing the story (as text, visual screens, or a movie) based on audience, goals, and constraints on time and resources.

5. CONCLUSIONS

Stories are powerful tools for ensuring the overall value and positive user experience of designed systems. In many ways, designing a compelling story means doing much of the same work one must do to design a compelling solution. For stories to be effective, they must include many of the same narrative elements that contribute to a compelling novel, short story, or movie. In practice, this means that teams using stories as a design tool can benefit from including participants with experience or skill in crafting stories.

As we have shown through our examples, teams at IBM have found stories useful at every stage of the development process, from early vision and innovation, through customer research, audience and feature definition, prototyping, user interface design and development, and marketing and rollout. This strength is based on the value of stories as rich tools for capturing and communicating human experience combined with the social power of stories that makes them an effective resource for teaming, eliciting feedback, and sharing common vision. The story-based design process not only knits the solution to the real problem, but can weave the team's contributions together in a way that builds the level of communication and trust of the team. When each discipline can see their contribution to the end solution, the level of satisfaction and empowerment grows.

Because effective storycrafting requires a sensitivity to human issues that are critical to overall user experience, thinking in terms of stories has made us better, more perceptive designers and developers. We suggest that the bookshelf of every user experience designer contain, along with texts on HCI, cognitive science, ethnography, graphical technique, and a few good books on crafting stories.

REFERENCES

Brannigan, E. (1992). *Narrative comprehension and film.* London: Routledge.

Burroway, J. (1999). *Writing fiction: A guide to narrative craft.* Reading, MA: Addison-Wesley.

Carroll, J. M. (Ed.). (1995). *Scenario-based design.* New York: Wiley.

Cohen, G. (1989). *Memory in the real world.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Cooper, A. (1999). *The inmates are running the asylum.* Indianapolis, IN: SAMS.

Klatzky, R. L. (1980). *Human memory: Structures and processes* (2nd ed.). San Francisco: Freeman.

McKee, R. (1997). *Story: Substance, structure, style, and the principles of screenwriting.* New York: HarperCollins.

Ochs, E., & Capps, L. (2001). *Living narrative: Creating lives in everyday storytelling.* Cambridge, MA: Harvard University Press.

Rosson, M. B. (1999). Integrating development of task and object models. *Communications of the Association for Computing Machinery, 42,* 49–56.

Schank, R. (1990). *Tell me a story: Narrative and intelligence.* Evanston, IL: Northwestern University Press.

Schank, R. (1997). *Virtual learning.* New York: McGraw-Hill.