IBM’s traditions in human factors are rooted as far back as the 1950s. This article traces and describes how the challenges to the discipline have changed over the last 40 years, based on the personal and professional experiences of the authors. During that span of time, the focus of attention has shifted from traditional knobs and dials hardware, to system and application software, and now to consulting services.

This article discusses these changes and describes how user-centered design (UCD) has become a key component of the IBM Global Services (GS) information technology consulting approach. The UCD component of the GS method is described and illustrated with 2 case study samples from recent engagements.

1. BACKGROUND AND HISTORY

The earliest IBM usability work began in the 1950s in two areas—in support of military and government projects as part of the Federal Systems Division and as part of IBM’s Advanced Systems Development Division (ASDD) and Research Division. Earliest pioneers, among others, included Corwin Bennett and Dick Hirsch, one of the founding members of the Human Factors and Ergonomics Society.

In the late 1960s, our work was devoted to developing a range of computer systems called the System 370, the successor to the highly successful System 360. Originally, the System 360 was expected to sell only a few systems, but surprised many by selling many thousands. Most of the human engineering work during this period was spent on designing the controls and displays with which computer operator specialists interacted and ensuring that they were compatible among the various models developed at different lab locations.

The low end of System 370 was developed in Hursley, England; the mid-range in Endicott, NY; and the high end at Poughkeepsie and Kingston, NY. The challenges to developing a unified system revolved around (a) coordinating activities across locations and countries with limited communications; (b) trying to ensure that us-
ers of various systems would be confronted with a consistent user interface (UI); and (c) designing a system family that was cohesive, despite the fact that low-end buyers had a very small budget, and those at the high end had no such restrictions.

Direct access to computer systems during this period was restricted to those operators who ran the system from behind glass walls. This limited set of people were the users that we targeted for our UI design activities. We needed to ensure that as an operator moved from one system to another, he (and virtually all were male operators during this time) would see similar controls and displays that would minimize relearning and transfer of training problems.

Other users who had more indirect interaction with the system were programmers who were attempting to accomplish some work as a result of writing computer programs in a low-level machine language or “higher level” languages, such as Cobol or Fortran. Writing the programs and gaining the results were done in a discontinuous fashion. Batch processing was the norm. Jobs were submitted and the results would arrive hours or even days later, often replete with syntax errors.

During this period, the concept of timesharing was introduced as a way of shortening up this cycle, reducing the errors and helping people to be more productive. The closer users were able to come to more directly interacting with computers, the more successful they became.

The next phase of usability involvement dealt with developing operating system software to run these large computer systems. The human factors department anticipated the shift to software, and we began educating ourselves about this new critical interface. DOS was the software of choice for the range of computers we dealt with (OS ran the big systems). New and more user-friendly systems were planned, but not implemented at that time. Customer setup instructions sometimes required users to type in code as part of the installation process. Our unit became affiliated with banking and finance systems, and we shifted our focus to hardware and application software for these systems. This included automated teller machines (ATMs), check reader–sorters, and bank teller and administrative systems. ATMs were designed for walk-up and drive-up operation and had to be both easy to use and easy to service. Check reader–sorters were designed to be operated by one person and, subsequently, had to be able to read and process handwritten information—great challenges for human–computer engineers.

Human factors or usability within IBM has waxed and waned throughout the years. The emphasis on human factors peaked during the 1960s until 1968 with the hiring of many new professionals. Subsequently, hiring went virtually dormant for about 10 years. In the late 1970s, Lewis Branscomb, IBM’s chief scientist, sponsored and kicked off a corporate-wide event at the Brown Palace Hotel in Denver, Colorado to refocus attention on usability, which led to an increase in hiring and focus on issues of ease of use within the development community. Another 10 years or so passed until another corporate focus took place in the late 1980s to develop a broad focus on organization, tools, operations, leadership, and business linkages. IBM Vice President, Terry Lautenbach, made videotapes stressing the importance of usability, launching a “usability rollout” campaign. Much of this was de-emphasized with the subsequent overall corporate downsizing that occurred in 1993.
2. GERSTNER BRINGS A NEW ERA

A new upswing took place in the mid-1990s, when Lou Gerstner assumed the reins as the head of IBM. Mr. Gerstner approached his job not from the technology side, but as a former client of IBM. He championed ease of use as one of the top six major battles that needed to be won to achieve leadership in the industry. As a result of this re-emphasis, user-centered design (UCD) became a core element of IBM’s software strategy, instantiating it into the Integrated Product Development model. Processes were put into place to measure and monitor progress in providing resources for, and improvements to, the usability of mainline software development in the company.

On the other hand, development of specialized software waned within the company during this period. Evidence of this was reflected in the loss of monetary support from development on a year-to-year basis. In anticipation of an eventual phaseout, the human factors department in Charlotte, NC began supporting some of our banking clients directly, and at their locations. As a result, we were well positioned to move into the next phase of our careers—that of consultants to industry.

The consulting business grew from scratch at IBM. When Lou Gerstner arrived from Nabisco, he remembered that what he really wanted from us as a customer was our intellectual capital and expertise as leaders in the information technology (IT) world, something that most businesses did not have. From that point on he brought in experts to build this new consulting business, which has now grown to be the largest such organization in the world, while bringing in almost one half of IBM’s revenue. IBM Global Services (GS) is truly a formidable worldwide organization.

As human factors or usability practitioners, we took our approach and methods, which we had built over time to service internal customers (primarily developers of our own internal products), and applied them to help external clients build user-friendly applications.

3. THE BEGINNINGS OF THE USABILITY ENGINEERING (UE) PRACTICE

In the early 1990s, Jim Kubie, a sales and marketing executive who had just returned from an assignment in Japan, believed there was a sizeable market for UI design and evaluation services among IBM clients. This belief was based on seeing how UCD expertise had made the difference in a number of internal software development organizations across IBM. Looking for opportunity on his arrival back to the United States, he received the go-ahead to form a small boutique consulting organization within what was then known as the IBM Consulting Group. At the time, many specialty practices were being chartered as IBM Consulting geared up to become much more active in the IT services business, because the restrictions of the 1956 antitrust decree against IBM were finally being lifted. There was little risk for IBM in launching these boutique practices—The thought being that each would stand on its own or perish. Those that survived, however, would have free reign to grow at their own pace, relatively independent of senior management involvement.

Initially, the UE practice was quite small, consisting of five Ph.D. psychologists focused mainly on usability testing. In these early days of usability consulting, the
going was not easy, as few businesses had an appreciation for the need for UI services. In addition, the group focused mainly on design evaluation and testing. Although an important facet, it was difficult to maintain long-term engagements based solely on this limited activity. Another major growth impediment had to do with the lack of appreciation for the importance of specialized UI design services among IBM Consulting Group industry principals—those responsible for generating leads for the UE practice. Because of this, it was difficult to position usability up front in the sales cycle.

Often, our services were an afterthought at best, and only offered in dire cases, where usability of the system was atrocious and the customer was complaining vehemently. Even then, the UE practice was not often asked to get involved in projects, simply because many of the newly hired IBM industry principals did not even know our group existed. As a result, the practice struggled for a number of years. The advances we did make in furthering the UCD cause during these years were based on a grass roots effort; building our reputation around IBM and IBM clients from the ground up.

We realized however that our go-to-market strategy was in need of major revision or we would not survive as a stand-alone practice. During our years working with the financial services internal development organizations, we had learned that IBM customers saw quite a bit of value in design prototyping. At the time, our design prototyping focused on using UI development tools to create, evaluate, and refine UI designs. We realized that IBM Consulting Group clients might also find value in design prototyping. We began to craft an offering around design prototyping that consisted of a package of UI activities including requirements gathering, design iteration, and usability evaluation performed in a 10- to 12-week cycle. Our hunch was verified when we were able to engage on a number of small rapid prototyping contracts.

We also realized that our skill base needed to evolve from an evaluation focus, to one centered on design and UI development. We refocused our hiring activities on recruiting UI professionals with strong UI prototyping and programming skills.

A significant problem remained in getting the word out about our group. Not only was IBM GS indeed a large organization, but in the early 1990s was quite entrepreneurial, with groups hawking their wares independent of coordination with other IBM entities. In addition, it was growing rapidly, with a focus on hiring professionals from the “Big 6” consulting firms that existed at the time. These seasoned consultants were arriving at IBM en masse from very successful consulting firms, each with its own established consulting method. As can be imagined, this breed of consultant had strong ideas about how consulting engagements should be approached—Unfortunately, there was little agreement on what that should be.

Another complicating factor was that IBM addressed projects using matrixed management, where the project team is brought together representing many specialties and backgrounds. This arrangement created significant problems because of the inconsistencies in the way individual consultants were trained to address client problems. Our UCD practice was not immune to this problem. Among our UCD consultants, there were significant differences among the team in terms of
technique, process, and deliverables, although we typically hired consultants with significant UCD work experience.

Communicating techniques and approaches between consultants from other specialty areas was a serious problem, creating a veritable Tower of Babel, with each consulting practice having its own terminology, techniques, and activity descriptions. Consider the difficulty in setting client expectations as a result of signing a consulting agreement, when there was not even agreement among the IBM team consultants about what was to be produced. Add to this fray the occasional immense consultant ego, and IBM had a good recipe for project disaster. Not surprisingly, by the mid-1990s IBM had a significant number of troubled and very costly projects on its hands.

In late 1996, IBM GS, as it was now called, embarked on a mission to address these problems. One of these initiatives was to establish a consulting methodology so that all consultants would have a common approach when addressing software development. The goal was to create an end-to-end software engineering methodology to address the full life cycle of custom application development so that all consultants collectively would have a single blueprint for approaching engagements. A senior IBM executive who recognized the importance of usability on custom application development insisted that the creation of the consulting method have representation from our small practice. This development was a critical turning point for UCD and our current UE practice in the IBM services organization.

4. THE IBM GS METHODOLOGY

A small team of senior consultants was brought together from across IBM GS to develop the new IBM GS methodology (GSMethod). These consultants represented each of the main custom application development disciplines—business analysis, organizational change, application development, architecture, operations, and project management. Included in this small team were consultants from the UE practice. A benchmark of known existing methodologies was performed, and best practices identified. As a result, a number of key defining principles guided development of the method:

- Clients had identified poor execution and delivery as the number-one problem on our consulting engagements. In the words of one client, IBM Consulting had “too many talkers, not enough do-ers.” To overcome this problem, the methods team focused the effort on the deliverable—specifically, that which is produced as a result of the engagement, rather than the technique or process of delivery. A construct was created called the work product, which was identified as the basic building block of the methodology. A work product is the artifact or deliverable provided to the client as a result of performing a contract activity. Basing the method on work products focused attention on what IBM delivers versus what we say we will deliver. Identifying the work product as the operative measurable set the tone for a new delivery-oriented method. To formalize the work product construct, the methodology team then cre-
ated the work product description (WPD), which would serve as the detailed specification for the content of each particular work product.

- Conversely, the method would provide minimal guidance regarding the process or technique for creating the work product. The rationale for this decision was that consultants were assumed to have the requisite skills to execute the work for their particular discipline. Rather than rehashing already well-known steps for executing various tasks, the WPD focuses instead on specifying the format and requirements for a successful work product delivery.

- Relatedly, each WPD itself would provide criteria for determining whether the delivery meets criteria for excellence. This provides a self-check for consultants when determining whether the work product is ready for delivery.

- The method was to be asset based (i.e., it was intended that once the method became established, no work product should ever be created from scratch). Instead, an intellectual capital database shared by all of IBM GS consultants should serve as the starting point for any work product development effort.

Not surprisingly, bringing a group of 15 seasoned consultants together to create a methodology for 9,000 IBM consultants was fraught with many challenges, not the least of which was getting these experts to agree on a common footprint for the method. Much negotiation occurred in building a method that addressed each discipline’s activities and outputs, yet did not significantly overlap each other.

One hundred work products were eventually identified to serve as the core of the method covering the full range of consulting activity across five identified phases of development (see Figure 1).

Solution outline is a short problem understanding and scoping phase, representing perhaps 10% to 15% of the total project schedule. During solution outline, consultants work to understand the client’s problem, determine the high-level project plan, and perform detailed scoping for subsequent phases of the project. In addition, the release plan for the project is created, providing guidance on the staging of functionality by release.

![FIGURE 1](image-url)  
Global Services methodology phases of development.
During macro design, the high-level architecture for the to-be-developed system is created. The goal during macro design is to ensure that whatever architecture is developed will indeed serve as a scalable, extensible foundation for subsequent release development. Therefore, it is release-independent—It is not tuned specifically for the first release, but instead is designed to scale up for all subsequent releases—release 1, 2, and beyond.

The subsequent phases (micro design, build cycle, and deployment) are release dependent (i.e., detailed design is performed for each release for the micro design phase, followed by the build cycle, and then subsequent deployment).

5. CREATING THE UCD WORK PRODUCTS

Because the high-level phases had already been predefined, our first major effort was to determine the content of the UCD component of the method and then determine which artifacts that are typically developed during a UI design effort were truly worthy of being elevated to work product status.

In developing the UCD component of GSMethod, we conducted a “best practices” review of a number of different UI design methodologies, eventually selecting IBM’s UCD process, which had been integrated into the IBM software development lab’s overall software engineering methodology. Of the methods we investigated, we felt the IBM UCD approach contained most of the key pieces to ensure a successful UI design engagement. Specifically

- Focus on the user—Early and continuous focus is placed on the user in terms of all aspects that have an impact on the eventual UI design, including but not limited to user aptitudes, user computer experience, their work environment, user tasks, and user requirements for success.
- Design the total user experience by a multidisciplinary team—Many skills are necessary to create a successful user experience, starting with the initial learning stages of the solution, through to the point where the user reaches power user status. To build this user experience effectively, it is important that the correct mix of skills are brought to bear on the UI effort including not only UI designers, but also marketers, requirements analysts, visual designers, IT architects, and usability engineers.
- Relentless focus on the competition—For a UI effort to truly succeed, it is important to understand the benchmark for success. One way to benchmark the effort is to determine the success level of the current system and design so that the new system is as good as or better than the current system. Another way to benchmark the system is to identify meaningful competition to the to-be-developed system and ensure that the new system is as good as or better than its competition. For many of our projects, clients are creating customer care sites that are intended to offload traffic from their call centers. In this case, the competition is the telephone, not other Web sites, because if customers can transact their business faster using the phone, the investment in the developed system will be wasted.
• Frequent usability evaluation at all stages of design—This includes not only formal usability testing, but also other forms of formative design usability evaluations including heuristic review and cognitive walkthroughs.

Our internal UCD process, although explaining the steps that needed to be followed, unfortunately did not contain sufficient detail to identify the minimum requirements for what was to be delivered. This is not a problem unique to the internal IBM UCD process. For example, in reviewing a recent, very popular, and well-written book on task analysis, we found only a handful of pages dedicated to describing how the results of a task analysis should be documented. The vast majority of the book focused on describing the process of how various methods of task analysis should be conducted.

The initial list of UI artifacts numbered over 30; through a merge and eliminate process, this initial list was reduced to 15 work product candidates worthy of standing on their own as part of the overall 100 work products of the full services methodology. The work products fell into three main categories—planning, design, and evaluation (as shown in Figure 2).

UI planning work products consist of those work products that are important for planning the design of the UI. These work products effectively ensure that the project has an early usability focus and creates an overall strategy for ensuring the usability of the solution. UI design work products are specific outputs resulting from the design activity for the engagement. As a group, they ensure that the solution will meet the requirements identified by the UI planning work products. Usability evaluation work products focus on evaluation activities associated with the requirements analysis and design of the solution. These may be created at various points of the IBM GSMethod, from solution outline through the build cycle and deployment, depending on their content.

Figure 3 shows where UCD work products are created in the IBM GSMethod. Although traditionally the UE practice had been associated with end-of-cycle usability testing, the reader will note that many of the UCD work products were identified for inclusion early in the development cycle. We spent a great deal of time educating the other method team members on the advantages of performing UCD activities up front in the design process so that maximum value could be attained and rework due to poor design minimized. Note that work product development is

---

**FIGURE 2** Work product by category.
iterative, with many of the work products updated at multiple phases of development. In each subsequent phase, further detail is provided to these work products, as more knowledge about the to-be-developed system is revealed.

6. THE WPD

The specification for a work product is called a WPD. It contains the following sections:

- **Description**—describes the contents of the finished work product.
- **Purpose**—including the reasons for developing the work product, the impact if it is not developed, and reasons for when the work product is not necessary.
- **Notation**—serves as the outline for the finished work product. It identifies those sections one would expect to find in the work product. For example, if the work product is a usability test plan, the reader would expect to find among other things, a description of the methods, procedures, planned data collection and analysis approach, and test participant selection as part of the plan.
- **Brief example**—shows an example of what the finished work product may look like.
- **Development approach**—provides the high-level steps that should be conducted to ensure the work product is completed properly. Although it provides a
high-level, step-by-step description, the assumption is that the developer of the work product will have formal training and experience in completing the steps.

- Validation and verification—serves as a quality control check for the work product. A reviewer of the completed work product could use the validation and verification section from the WPD to use as a means to ensure the work product was constructed properly. For example, if a consultant had developed a usability requirements work product, a few of the items from the validation and verification checklist are as follows:

  - Requirements were gathered directly from representative end users in Joint Application Development, Decision Support Center sessions, or their equivalent
  - Prioritized and documented usability attributes to provide substantial guidance to the design and evaluation teams is included in the work product.
  - Performance criteria that are in a quantifiable format that would allow them to be incorporated into usability test plans and tested during usability testing are present.

- Advice and guidance—provides guidance on how to create a successful work product that meets the needs of the client.
- References—provides a bibliography of sources the reader can refer to when needing details about specifics of the work product construction, such as detailed guidance for creating the work product.
- Estimating guidelines—provides details regarding how big an effort the creation of the work product is, depending on the size of the project in terms of consultant hours and level of expertise.

7. THE WORK PRODUCTS

As mentioned earlier, many artifacts were identified as possible work products for inclusion, but 15 made the cut for the first release of the methodology. These work products were as follows:

1. Current solution evaluation—The current solution evaluation work product is an assessment of the solution that is currently in use by the general target audience. The current solution may be an earlier version of the solution under development, a competitor solution, or a manual process that accomplishes the same goals as the solution under development. The purpose of the current solution evaluation is to benchmark the usability of the current system.

2. User profiles—The user profiles work product is a document that contains detailed descriptions of the relevant characteristics of each user category. Characteristics include descriptions of users’ prior knowledge and experience; physical characteristics that may have an impact on job performance with the new system; social environment; jobs, tasks, and requirements; and cognitive characteristics. User profiles classify the different types of users who will use the new system.

3. Usability requirements—The usability requirements work product provides a documented set of measurable usability wants and needs for the new solution
under development. These requirements may specify the attributes of interaction (e.g., navigate to open problem ticket status in no more than two mouse clicks), display (e.g., target screen resolution of 1024 × 768 pixels), and measurable user performance or productivity needs (e.g., 2 min to complete a transaction).

4. UCD approach to use case modeling (UCM)—The UCD approach to UCM addressed a major disconnect between the way UCD practitioners and software developers specify functional design requirements. Typically, our UCD practitioners built detailed task analyses, including task affinity models, to help identify and model how users would interact with the new system. These analyses provided the UCD consultant with the impetus for identifying the application conceptual model and high-level navigation. In building the task analyses, our UCD consultants spent quite a bit of time conducting interviews, observations, and other requirements gathering sessions to identify this detail. In the meantime, as a parallel effort, the UCM team composed of business analysts and software engineers were also often conducting similar independent sessions in the form of joint application design sessions, interviews, observations, and surveys as they created their use case models. From this activity, the UCM team would first create a list of use cases (which specify the interaction between the to-be-developed system and various actors, such as other systems, devices, or end users). Later in the design phases, more detail in the form of a flow of events, inputs, outputs, pre- and post conditions, typical and alternative scenarios, and other step-by-step details were added.

We undertook an analysis of best practices in both the task analysis method and UCM with the hope of creating a unified method. From this analysis we concluded that there was indeed substantial overlap between the two methods, particularly for those use cases where the actor in the system–actor relation was the end user. We felt that the UCM did an exceptional job of documenting findings, but was much weaker in specifying the process by which information was collected. We concluded that although the task analysis literature was very strong in specifying various methods for performing task analysis, it was quite a bit weaker in specifying a consistent approach for documenting the findings. As a result, the UCD approach to UCM was created to address these issues. By integrating task analysis and UCM techniques, we were able to create an approach that was strong in both process and reporting, while providing the entire team with a single vehicle to clearly communicate design requirements among all team members.

5. Use case validation report—The use case validation report details the results of a prioritization of use cases. It describes the technique and criteria for prioritization and documents the findings.

6. Usability design and evaluation (UD&E) plan—A UD&E plan is a document that specifies the activities, deliverables, schedules, dependencies, process, team members, and other aspects of how UCD will be implemented throughout the course of the design and development of a solution.

7. UI conceptual model—The UI conceptual model forms the foundation on which the entire interface is built. It identifies the UI objects to surface to the user and provides the underlying basis for high-level navigation.

8. UI prototype—A UI prototype is a simulation of the UI of a system or application with which a user can interact. It is built early in the development cycle with
the express purpose of being changed and improved as part of the iterative design approach. Prototypes can be rendered in a range of fidelities (e.g., low fidelity and high fidelity), depending on the phase of development.

9. Early usability evaluation—The early usability evaluation work product was created to provide an assessment of the intuitiveness of early design prototypes. It could take the form of either a heuristic review or a cognitive walkthrough, but the intent of the evaluation was to provide design improvement feedback to help the design team improve and refine the design.

10. UI design specification—A UI design specification work product specifies the finished UI design that is produced at the end of the micro design phase. Besides providing detailed specification for global and pervasive screen elements and navigation, it also provides detailed page specification including widget and screen behavior; user exception handling and error messages; and all user assistance externals including all help text, supporting visuals, and wizards. The UI design specification and detailed UI prototype are often provided together as a means of fully specifying the UI. Depending on the team and the goals of the engagement, ideally, the UI prototype will contain screen layout code and widget interaction in a format that can be delivered to the development team and reused in the coding development effort.

11. UI design guidelines—The UI design guidelines work product is created to provide a consistent set of standards to assist developers in constructing work products related to UIs. The UI design guidelines work product explicitly documents mandatory, recommended, encouraged, and discouraged practices based on the most frequent types of errors that are made against this standard and the specific types of systems that are being built across the organization.

12. UI architecture—The UI architecture work product describes the rationale for and specification of software components and frameworks required to support solution-wide user interaction needs. The UI architecture is a subset of the overall functional architecture. The UI architecture work product specifies those architectural components that are pervasive throughout the UI. Common UI architectures include windowing and keyboard support frameworks. They may also include design work products and other diagrams and textual description.

13. and 14. Usability test plan and report—These two work products combined provide the details of conducting the solution’s usability testing. They include descriptions of participants, setting, materials, procedures, and usability test cases for the usability testing. They describe the plan for conducting the usability testing of a design. The usability test plan is a written document and is distributed before the evaluation to secure approvals from key personnel. The usability test report documents the results of the usability testing. It provides a permanent, historical record and summary of each evaluation conducted throughout the development cycle of a solution, in addition to the rationale behind conclusions reached and recommendations made.

15. Visual resources—The visual resources work product documents the library of all of the visual design elements (including images, gifs, and jpegs) that are created during the development of a new system. The visual resources work product also can take the form of a look and feel guideline by providing typography, foreground, and background color values and general visual design layout information.
8. TAILORING THE METHOD

Clearly with 100 work products in the entire method, and 15 devoted exclusively to the UI, it is the case that not all work products will be delivered for all engagements. Some engagements are either not large enough or the scope of the development work is not sufficiently end to end to warrant a full-blown application of the method. This knowledge gave rise to the notion of an engagement model. The idea behind an engagement model is that there exist certain patterns of UI design problems where it makes sense to apply but a subset of the work products to solve the problem.

The rapid prototyping engagement model is a frequently applied pattern of UI design in our practice. It is applied in cases where

1. Clients recognize they have a usability problem with a current system.
2. They are not sure what exactly is causing the problem, but they do believe that usability design refinement or even a brand new design will address the problem.
3. The client’s concern is that they will engage a design agency or a consulting firm to perform the redesign, and the engaged vendor will provide no evidence other than “expert opinion” that the design they create will improve on the existing system design.

Figure 4 shows the rapid prototyping engagement model. As can be seen from Figure 4, during solution outline, we work to understand the current environment (current solution evaluation), and also identify the main, high-priority tasks that will serve as the basis for high-level and detailed design in subsequent phases (UCD approach to UCM). At the end of solution outline, the work products that are delivered to the client clearly identify the usability problems that exist with the cur-

![FIGURE 4 Rapid prototyping engagement model.](image-url)
rent system and the primary tasks that are not well supported due to weaknesses in the current environment. In addition, we deliver quantifiable requirements for usability success in the form of the usability requirements work product. Specifically, measurements are provided to identify the optimal time-on-task for high-importance system functionality, acceptable error levels, and acceptable measures of user satisfaction. As a whole, these work products provide the client with a yardstick by which to measure the effectiveness of the designs we develop in the subsequent design phases. Also at the end of this phase, we provide the client with a detailed plan for how we intend to meet the usability goals in developing the design (usability design and evaluation plan).

During the macro design phase, we create and validate the high-level design for the system through the creation of the UI conceptual model. We typically create three conceptual models and render these models in the form of low-fidelity prototypes. These prototypes are then evaluated by conducting cognitive walkthroughs of usability test scenarios with actual eventual end users of the system. From these early usability evaluation results, we are able to iteratively identify one model that we will further refine during micro design.

During micro design, the model is further fleshed out using rapid prototyping techniques. The usability evaluations during this phase take the form of formal usability testing so that the usability success criteria identified during solution outline can be rigorously tested.

At the end of the engagement, a working prototype is delivered, along with documentation providing usability test evidence that the design hits the mark for meeting the usability needs of the client.

The next section details a number of case studies where we have effectively used our work product-based methodology to create more usable designs for our clients.

9. CASE STUDIES

9.1. Case Study 1: Internet Shipping

In 2000, an international shipping distributor wanted to expand their Internet shipping presence and engaged the UE practice to conduct an evaluation of, and provide tactical design refinement to, an online shipping Web site. The company had a go-to-market development deadline to meet and wanted to refine their beta design to ensure that new ideas could be generated, demonstrated, and tested in an 8-week period. Based on the client’s stated needs, we felt a rapid prototyping engagement model would provide the best value for the client. We assembled a three-member team that included two experienced, but new, members of our practice and a team lead that worked on the project on a part-time basis. None of the team members had ever worked together on a project.

Work product start and delivery dates comprised the checkpoints and milestones for the project. The team harvested existing intellectual capital to serve as deliverable examples and templates. The work effort was divided into three stages, as in Figure 5.
The engagement began with the delivery of a usability design and evaluation plan that outlined the activities that we would be performing over the next 8 weeks. The plan provided documentation to both the client and team to ensure a clear and shared understanding of the activities to be performed, and described the staging of UCD activities and milestones and the format of the final deliverables.

During solution outline, an analysis of the current work environment was conducted through interviews with stakeholders and by conducting evaluations of comparative and competitive Web sites. Because we had limited access to local client stakeholders, we asked the client to use the user profiles work product template to help us understand the audience for which this Web site was being designed. The template contained detailed descriptions of relevant user characteristics, including users’ prior knowledge and experience; physical environment and job requirements; and current and potential shipping patterns. A wider range of users was identified than expected by the client from these interviews, including a potential audience of users that currently do not use Internet shipping sites. These users expected a site in which simplicity, lack of jargon, and ease of task completion would be provided.

While the user profiles were being collected, we conducted current solution evaluations of the client’s Internet shipping alpha and beta sites along with three competitor sites that had established an online shipping presence. A heuristic evaluation of the alpha and beta sites was conducted using a set of guidelines to gauge the solution’s usability. The competitive analysis involved reviewing competitor sites and pinpointing areas of site distinction, as well as the strengths and weaknesses of each site. The goal of these evaluations was to identify problem areas for remediation, their impact on usability, and to provide recommendations for design refinement.

The competitive analysis findings identified deficiencies in the areas of visual design, user interaction, and navigation for competitors that provided online ship-
ping functions. These findings indicate that considerable opportunity existed for our client to become the best Internet shipping site in terms of usability by creating a site with a strong visual presence, simple and intuitive UI, and a clear and consistent navigation scheme.

A notable heuristic finding was that the overall current UI design and visual branding provided in the beta site was sound. However, a number of areas of improvement were identified that could greatly reduce the number of screens and the complexity of the site to which users would be exposed. Areas of improvement included the following:

- Reducing required scrolling and keystrokes.
- Providing a clear and consistent navigation scheme.
- Providing improved visual feedback regarding task flow and task completion.
- Reducing shipping jargon that was unknown to users outside of the shipping industry.
- Incorporating filtering into the system to eliminate invalid shipping options (e.g., ground shipping from the United States to Europe).
- Simplifying the interface by providing a default series of steps that targeted users who ship packages the same way the majority of the time.

We believed that the client could realize significant near-term benefits by redesigning the Internet shipping application without changing their current design structure and visual branding.

Our next step was to apply our findings to begin the macro design phase of the Internet shipping UI. During this phase, we identified key UI concepts and evaluated alternative conceptual models for the UI. We iteratively progressed through this phase by evaluating alternative designs in cognitive walkthroughs with end users, and then revisiting the designs to ensure that the design met the system’s usability requirements. The focus of this phase was on low-cost, low-fidelity prototypes that could be quickly created, rapidly evaluated, and easily changed. Early prototypes consisted of a series of paper-and-pencil sketches depicting key attributes of the interface.

Early usability evaluations were conducted as an assessment of the efficacy of the UI conceptual model. Two usability walkthroughs were conducted in a usability lab rented by the client at their headquarters. Client stakeholders were invited to review the task scenarios and to observe the studies being conducted. A total of 12 users participated in the studies. Participants worked under the direction of a test facilitator to complete task scenarios and answered directed questions regarding controls usage, navigation, and expected task flow. The participants were asked to document their first impressions, complete response sheets to questions asked by the tester, and to discuss their opinions. Users were tested in pairs and encouraged to utilize a think-aloud protocol. Each study was conducted in 1 day, with 2 days provided between the studies to provide time for design refinement. Two alternative design approaches were evaluated in these studies. The results identified the best prototype for further refinement in the high-fidelity prototyping phase.
The final stage of our involvement was to develop a limited function, interactive, UI prototype that provided a simulation of the Internet shipping UI. During this phase, key UI concepts were identified, evaluated, and refined through an iterative design–evaluation–refinement loop using rapid prototyping techniques. A single conceptual model was to be selected at the start of this phase. Although a model tested during macro design was preferred by users, some members of the development group that had produced the beta prototype still believed their approach to be the preferred approach. To resolve the disagreement, the UE team conducted a short comparative usability assessment of the two design approaches. A low-fidelity prototyping approach was used, consisting of screen print task flows. Data-driven success criteria based on keystrokes, errors, and subjective evaluations were defined prior to the conduct of the study. All parties agreed that the most highly rated design approach would become the plan of record. Results showed a user preference for the refined design approach proposed by UE, and the client relented and agreed to the development of a higher fidelity prototype of this alternative design.

An interactive HTML prototype was created as a working model of the selected concept. Key task scenarios were prototyped creating a “vertical” prototype of selected functions in which deviations more than one screen away from the selected path were not possible. The prototype was evaluated twice during its development at a usability facility near the client’s headquarters site. As in the earlier cognitive walkthroughs, these design walkthroughs were each conducted in 1 day. Unlike the earlier studies in which the test facilitator controlled the task flow, these assessments were user driven. Again, 2 days were provided between iterations to provide the UE team with time to refine the design based on user feedback from the first study. Test scenarios were created from typical task scenarios. The scenarios focused on user expectations regarding content, layout, navigation, and task flow. The scenarios were consistent across the usability tests, providing for a baseline of comparison across studies.

The Internet shipping project was very successful. The methodology and associated work products allowed us to develop and meet an aggressive schedule.

Using work product templates as the baseline for design activities and deliverables allowed us to meet a very aggressive schedule and deliver quality work products on time and within budget to the client. The work products provided a common language that can be used by team members who have never worked together. The methodology provided a common design approach, activities and goals under which to work, making management of this project relatively easy. In return, the client received a refined application design that provided more functionality and targeted a broader user group than initially envisioned. The resultant design provided users with an ability to set shipping preferences, provided the power user with a “quick ship” option, and filtered available options to eliminate the possibility of setting incompatible shipping options (such as ground shipping a package overseas). The UI was greatly simplified while still maintaining the client’s original design. The following are examples of page and keystroke savings:

- Quick ship of a package reduced the number of required pages from five to one and the number of keystrokes from 34 to 13.
For a detailed ship, in which many options may be selected, the number of pages visited was reduced from five to two, with the number of keystrokes required reduced from 34 to 26.

The work product deliverables, including the prototypes, were turned over to the client for implementation. Overall, the client was pleased. The resultant application is now available on the Web and has been well received. One very pleased user provided an enthusiastic endorsement by writing, “Very big thumbs up to whomever redesigned the Internet ‘2 click ship.’ I just sent 3 packages in record time.”

9.2. Case Study 2: Data Dissemination

IBM has been a part of an ongoing multiyear development effort with a government agency that is responsible for data collection and data dissemination of demographic and economic data. In recent years, this agency has moved to replace print as the primary means of disseminating data. With IBM’s help, they are building a data dissemination system for the Internet. Early prototypes were based on a systems technology approach. In this traditional approach, development is focused on the function to be provided and is driven by the technology available to present them. A defect view of quality is taken, with user measurements collected at the end of the development cycle. These early prototypes were also developed with a waterfall approach to requirements gathering, interface design, system build, functional testing, and beta testing. In this approach, development stages are conducted sequentially and independently, usually by independent teams with limited multidisciplinary cooperation.

This waterfall development approach generally does not lend itself to a focus on ease of use, as once a stage is completed, the deliverables are presented and the stage is generally not revisited. Design iterations are generally conducted within a stage, with design focused primarily on the incorporation of functional requirements rather than on ease of use. In this particular development effort, programmers and developers using a set of general usability guidelines performed UI design without concern for the eventual users of the site, nor an understanding of what their task goals were. Although some usability testing was conducted, results were passed on for consideration into the next major development release instead of being incorporated into the prototype itself. The design assumed that users were familiar with the data products, data organization, and the means available to perform the task. For example, finding the population of a particular geography required the user to know the report in which this information was published, the tool that provided access to this report, and the hierarchy in which the geography was located. This process resulted in a difficult-to-use system that did not match the users’ mental model of how the system should operate.

A shift to a UCD approach and creation of a UI design team was made when it was time to build the first production system, followed by a transformation to a work product-based UCD process. Usability requirements were defined, consisting of measurable usability targets and goals. These included targets for the num-
ber of steps to display data, transaction success rates, time to completion, and learning rates. These usability requirements were incorporated into the test process as nonfunctional requirements and are now a part of system testing.

Next, a current solution evaluation was conducted of the existing interface. The current solution evaluation included both a heuristic review of the application that had been released and a competitive analysis of comparable sites that disseminated similar data. A summary description of each site was provided, along with its salient features and usability shortcomings. A task that was common to all sites ("Find the population of a specific city") was included in both the heuristic and competitive evaluations to establish a common baseline across applications. A keystroke analysis for each site identified the wide range of actions required to complete the tasks on each site. This keystroke analysis was well received by the client and has been successfully used in other comparative and competitive evaluations.

The UI design team interviewed potential external users, along with internal users and client executives, to establish user profiles. Initial interviews focused on task analysis and skills assessments. The initial release of the application was based on profiles rooted in existing job categories. A user segmentation study conducted by IBM identified behavioral indicators of four customer segments wants and needs. This recasting of user profiles into task-based customer segments ranging from Web “surfers” to experienced data “extractors” replaced the job classification. This opened up the design to place greater consideration on users who were not current users of this type of data and their need for a simplified UI, “performance support” tools, and a sensitivity to data-related jargon and terminology.

Conceptual models were developed to map and organize the UI. A variety of metaphors were explored and iterated with internal end users, including client organizations that directly interact with the public. A description of each metaphor under discussion was provided, along with a representative look and feel, and pros and cons of its use. The team ultimately settled on a newspaper metaphor that provided flexibility in changing content and the “push” of items of interest to users. A lesson learned for future engagements is to avoid creating visual representations during the conceptual modeling stage that were “too real.” This realism bogged the review team down to discussions that focused on specific content of pages rather than on the concepts that they were attempting to represent.

A number of UI prototypes were created as a part of this development effort, ranging in fidelity from paper-and-pencil sketches, to PhotoShop screen shots, to high-fidelity HTML interactive code. These prototypes were used for internal evaluation and review and for demonstrations to the client and design teams. The prototypes were evaluated in usability design walkthroughs, electronic meeting-room environments (using GroupSystems software), and formal usability tests using a usability lab. Visual resources created as a part of these efforts were incorporated into the final design and were made available to the team for reuse on subsequent releases.

A usability test plan was developed for the conduct of a usability design walkthrough of an interim prototype and a usability test of the completed product. Test scenarios were based on common task scenarios and created use cases. These studies resulted in UI improvements that continue to be incorporated into ongoing scheduled releases. Until this project, usability tests had primarily been conducted
at the end of a development cycle and involved much planning, concern, and involvement on the part of the client.

The client had already established Internet standards and design guidelines, which provide a detailed overview of usability design principles and principles of interaction. These documents were not specific to a single or family of applications that comprises this client’s site. We recently released the first iteration of a Web-based UI style guide for the client to provide more detailed implementation rules for Web page presentation and navigation. The guide is intended for use across client sites and will evolve over time. Implementation rules identified during ongoing efforts will be incorporated into the style guide.

As the project progresses, continued focus will be placed on iterative usability tests and creating an integrated site development process that can be utilized by other client development groups. Prototypes will continue to play an important role in demonstrating UI enhancements and improvements. The value of the style guide as a design tool to create consistent UIs will also be promoted as emphasis is placed on integrating a variety of sites that disseminate data to the user.

10. RESULTS AND CONCLUSIONS

The establishment of the GSMethod has benefitted the UE practice greatly. Because all IBM GS consultants are trained in the GSMethod soon after arriving at IBM, all consultants are not only familiarized with the UCD work products methodology, but also provided contact information to the UE practice for the UCD domain of the method. As a result, the UE practice has more than doubled in size as it has been engaged in delivering solutions to IBM clients.

The rapid prototyping engagement model has been one of our most successful endeavors to date, having been applied at numerous clients, particularly in the call centers of the banking, brokerage, and insurance industries. Because these companies can easily quantify usability in terms of cost savings (e.g., one client calculated that every 1-sec delay in call handling cost the company $500,000 annually), it is very easy to build the cost justification for the rapid prototyping engagement model.