Jörg Beringer, Karen Holtzblatt

Designing Composite Applications

Driving user productivity and business information for next generation business applications

SAP PRESS

Contents

Back in the 90s, Contextual Design helped SAP to improve the usability of business applications by understanding individual user needs and work practice. Contextual interviews became a standard technique for gathering user-centered requirements and analyzing work group collaboration.

Contextual Design captures in-depth knowledge of the context in which products will be used, a critical pre-requisite to deliver products that provide high value and are intuitive to use. Insights gained from Contextual Design Work Models last far beyond one particular development project or product release. They can be used to improve products by renewing them—leaving their essence intact, but strengthening them substantially, or by driving product innovation.

Heinz Roggenkemper,
EVP of Business Process Renovation, SAP Labs, Palo Alto

Being a pilot customer for SAP xApps was a win-win opportunity. The process of understanding business needs, and ultimately functional requirements, was facilitated through Contextual Design activities, such as interviews conducted in the user's daily work environment and later through prototype feedback sessions. This user-focused approach enabled SAP to gain direct insight into business drivers, and it challenged us to push beyond the status quo and think about better ways of working.

Once delivered, the solutions provided capabilities that crossed traditional functional lines; a good example is xPD (xApp Product Definition), which draws together users from multiple business areas to develop product concepts. With the rapid maturing of the NetWeaver platform, we are looking forward to the prospect of a rich set of business integration, and analytics services that xApps can utilize.

xApp Pilot Customer

About This Book

The design of business applications is undergoing a major paradigm shift; with new technologies and new demands, business can drive innovation into their organizations reliably. Enterprise resource planning (ERP) applications used to be focused on transactional data processing and process consistency, and designed to support trained professional users. Today, core ERP processes are automated and streamlined. The bulk of the work of running businesses has shifted to information workers challenged to find the information they need to support decision-making and ad-hoc exception handling. Modern business applications can improve business efficiency by enhancing the productivity of these individuals and their collaboration within their work groups.

Technologies, like the SAP NetWeaver business process platform, help the enterprise run its traditional business processes, identify new opportunities, and quickly adapt to changes. The core platform secures the ongoing business, but composite applications built on top of the platform address the need for improvements and new opportunities. Composite applications require an application development framework to orchestrate platform services into modern business solutions. SAP® xApps™ built on the SAP NetWeaver® platform and the SAP Composite Application Framework (CAF) are SAP's first products delivering the technology required to support the development of modern business applications.[1]

But technology alone is not sufficient. Successful business applications must also address the experience of its users. Applications will not be adopted and new processes will not be implemented if they don't work for both the business *and* the user. Next-generation composite applications that support users, their work groups, and enterprise processes can offer productivity tools and collaboration in the context of business processes. These applications require a robust requirements

1 For more information on SAP NetWeaver and related technologies, go to *www.sdn.sap.com*.

gathering and design process to ensure that the needs of both users and business processes are supported. SAP xApps can deliver a more goal-oriented user experience aligned with business objectives while leveraging SAP's new technology for optimal use.

This book introduces the concepts of composite application development and an associated user-centered design process streamlined to support the development of composite applications. We adapt the Contextual Design[2] method, a well-established customer-centered design process used by companies and taught in universities all over the world, to the needs of creating a better user experience in business applications based on reusable user experience building blocks in the SAP NetWeaver platform. Contextual Design (CD) is grounded in ethnographic studies of the user and business process, represents the data in appropriate work models, and develops the design through early prototypes tested with users in the field.

Throughout the book, we provide you with selected examples of an SAP xApp™ product development project: The SAP xApp Product Definition (SAP xPD) is an SAP xApp supporting collaborative idea generation and requirements management within a product innovation context. Understanding the user-centered design techniques of Contextual Design and the user experience building blocks provided with the SAP NetWeaver platform, you can start quickly on the road to packaged composite applications development.

Audience

This book is for anyone developing composite applications that can take advantage of modern business platform technologies. Contextual Design is a team-based process that brings together all who are responsible for the development to create a shared understanding of users' needs, process redesign, and ultimate system design. This book is relevant to marketing, product management, UI designers, developers, and usability professionals developing a business solution for a

2 For more information on Contextual Design and InContext services, go to *www.incontextdesign.com.*

market or for a specific customer. It is also relevant to business analysts, process engineers, developers, and usability professionals developing composite applications for internal business use.

Acknowledgements

We wish to thank all those people whose hard work went into the creation of this book. Thanks to the SAP xPD team for using the Contextual Design method and for producing a successful application that we can use as an example. We also want to thank the SAP xApp design group for providing input and snapshots of the actual design process. We especially want to thank Hugh Beyer, a wonderful writer and cofounder of the Contextual Design process, for his writing, editing, and negotiating support, without whom this book would not exist.

Introduction

Enterprise applications are in a state of crisis. The demands of businesses on IT groups have never been greater. The intense pace of competition that comes with globalization is forcing businesses to look for improvements in every process that they implement and in every project that they undertake. Yet the techniques available for developing systems have not grown with the demand. Consequently, IT groups are getting further and further behind. "Whatever you recommend," we were told on one reengineering project recently, "just remember, the IT group has a four-year backlog."

But now new systems concepts and new technology are ready to transform enterprise systems development. This book introduces the composite application development paradigm, supported by new technology that is now available. The SAP service-oriented business process platform promises to radically reduce development delivery times and dramatically simplify the work of defining, designing, and delivering an enterprise-level application

What kind of problems beg for this new approach? Consider the following examples:

- In 1997, the Kyoto Protocol imposed new constraints on industrial emissions. National governments are responding with new emissions control and monitoring regulations and your company will have to conform. How quickly can you develop systems that help people track emissions, report on them, and identify and respond to deviations, without having to disrupt legacy systems and integrate these new systems with existing work systems?
- You're in the business of producing a consumer product—hair dryers. In this commodity market, it's critical that you can identify an unmet need, invent a new product concept to address that need, and get the product on store shelves quickly, that is, within 3–6 months. Yet the whole process of product concept development is unarticulated and unsupported. The business needs a system to help manage product concepts—the generation, development, and eval-

uation of these concepts, culminating in moving promising concepts to product development. You need a solution fast. How quickly can you respond? (Hint: "Four years is *not* a good answer.")

Why are these hard problems? Introducing new systems and business processes into the complex environment of a modern business is a challenge. The risks are large—most companies depend on critical legacy systems that must not be disrupted. Companies have layers of support and management software in place, supporting highly complex business processes. And—as always—they are broken down into multiple cooperating and competing silos that somehow get the work done.

Furthermore, many of the new opportunities are of the sort characterized by our two examples: They seek to leverage strategic, collaborative, decision-making processes characterized by a high degree of communication across traditional silos. This is a new breed of application that is expanding the scope of automation in the enterprise.

Businesses *must* introduce new systems to respond to external changes and internal goals. An effective approach for developing applications is desperately needed. This approach must be flexible enough to deal with the range of challenges and rapid changes facing business today. It must be powerful enough to support the new kinds of solutions businesses need. And it must support the teams and work groups that make business happen without disrupting the systems and processes that are already in place.

Science fiction? No, the technology is available today, but technology alone is not sufficient. With this book, you'll learn how to discover and define the needs of the business, assemble the components of a coherent solution, integrate legacy databases and systems, and provide the user with an integrated environment in which to work—all based on tested and proven design principles, and all at the speed necessary to serve today's business.

This approach to design is based on three fundamental concepts: *packaged composite applications*, *user-centered design*, and *development with*

UI building blocks. So let's take a moment to introduce these concepts and describe how they fit together to make up a solution to our problem.

Packaged Composite Applications

Packaged composite applications (PCAs) describe the new kind of applications we are focusing on in this book. Rather than attempting to design and deploy a new system as a self-contained whole, PCAs view the new system as an organic extension of a system platform that already exists.

If we take the term apart, we're talking about *applications* that:

- Provide new functionality that goes beyond simply integrating existing systems
- Expand support to flexible, configurable, collaborative processes that go beyond the scope of existing applications
- Allow analysis followed by action, exceeding read-only aggregation of corporate information

They are *composite* because they:

- Aggregate functionality of existing systems, as exposed through Web services
- Cross traditional application boundaries
- Create a comprehensive process and information model

And being *packaged* results in:

- Products with a lower total cost of ownership (TCO) than custom development
- Products with integration costs leveraged over a large customer base
- Products supported with new releases and maintenance

In other words, a PCA is an application that uses underlying enterprise services to unlock the value of a company's existing systems. It gathers the information from all the heterogeneous legacy applications into a

unified, homogenous form, and then uses that information to build a new, focused solution based on a comprehensive view of the enterprise.[1]

This new application paradigm offers a way around the roadblocks that typically hamper the design, development, and introduction of new systems. Composite applications build on (legacy) systems rather than replacing them; work across silos, providing overall monitoring and cross-departmental coordination; and are especially strong in extending automation to new areas such as decision support and strategic systems.

But a "paradigm" isn't enough. You can't generate a new enterprise application with a paradigm. The industry needs concrete development platforms that instantiate this paradigm—that make it possible to design and develop composite applications for real-life systems.

That's where the SAP business process platform comes in. SAP NetWeaver provides tools and services that make it easy to build packaged composite applications like SAP xApps, which package new process support and improved user experience with flexible back-end links to legacy and new data stores. Rather than treating every project as a new problem, composite applications build in a number of well-understood process patterns and user interface paradigms, ready for you to tailor to help you resolve your business problem.

The collaborative emphasis of SAP xApps and its ability to unify information across existing systems allow applications to support strategic processes that require a comprehensive view of the enterprise. This avoids time-consuming assembly and a rollup of information from component systems, records the decisions for later steps in the process, and manages group communication. All of this contributes to strategic efficiency, the essence of which is making decisions as fast as possible based on the right information.

1 Dan Woods: *Packaged Composite Applications: An O'Reilly Field Guide to Enterprise Software.* O'Reilly & Associates, Sebastopol 2003.

SAP xApps expands organizational awareness across departmental boundaries. While many versions of a specific enterprise application, such as human capital management (HCM), supplier relationship management (SRM), or customer relationship management (CRM), may exist in a company, each class of application is a world unto itself, separate from other classes. CRM and HCM both have employee information, but the functions that each application performs and the information stored are often completely distinct. The job of the SAP business process platform is to integrate those core enterprise resource planning (ERP) components and provide a service layer that can be used by SAP xApps to spin a web across classes of applications and create a comprehensive model of information and functionality from each. These cross-functional processes can be the key to larger efficiencies or competitive advantages.

Developing composite applications extends the ERP platform and minimizes disruptions to business-critical functions, because it doesn't rely on alterations to core application business logic or data structures. Businesses depend on the continued operation of their core ERP processes, often supported by legacy systems. Changes can introduce bugs that make doing business impossible, and as systems become increasingly more complex, the cost of making the next change increases exponentially. Composite applications allow new functionality to be delivered without modifying hard-to-change base systems. Once this new functionality has been established, it can be integrated into the core platform once again to become an additional service for other applications.

User-Centered Design

The second key concept behind developing composite applications is *user-centered design*—the discipline of specifying a new system based on an in-depth understanding of the detailed work practice of its users. Composite applications are cross-functional, highly interactive systems—far more complex than traditional forms-driven interfaces supporting database transactions—and are far more enmeshed in the day-

to-day work of their users. Successful deployment of these systems cannot happen by management declaring that a new system must be employed by users—users have been working around systems that don't work well for them for years.

Management defines the project and states what's required. "Reduce travel costs," or "Meet the new environmental reporting regulations." If the system doesn't deliver on these expectations, it is a failure.

But it's the users of the system who determine whether or not the system will actually work. People organize their work to make their tasks efficient; they have intents (i.e., goals) that they are trying to achieve; they use strategies to achieve these intents and accomplish these tasks. Some of these tasks, intents, and strategies are explicitly defined; others are implicit and grow out of experience from doing the work. When looked at collectively, these tasks, intents, and strategies constitute the users' *work practice*. If the system supports and extends the work practice, it will be accepted; otherwise, users will subvert it. With the best of intentions, users will circumvent the system because they *must* get the job done somehow. "I know policy is to go through purchasing," one user might say. "But their vendor is more expensive—I'm just putting this purchase on our credit card." Or, "I'll do the bookkeeping work required by the system, but my *real* records are here in my trusty spreadsheet."

Understanding who the users are, how they work, and what their issues are is critical to meet the expectations of a project (not forgetting that we have to worry about all the users—including indirect users such as managers who depend on reports from the system and all the stakeholders who can affect its success.) We do this by adapting techniques from Contextual Design, the industry-leading process developed by Karen Holtzblatt and Hugh Beyer. Their book[2] describes the full Contextual Design process—here, we'll introduce a simplified and stream-

2 Hugh Beyer and Karen Holtzblatt: *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers, Inc., San Francisco 1997. For more information about CD and InContext services, go to *www.incontextdesign.com*.

lined design process tailored to the design of composite applications that addresses a business pain point within an enterprise.

Applying user-centered design to the development of composite applications starts by considering the *project scope*—the extent of change envisioned by a new project. Some projects envision fundamental change to the business practice; others seek only to streamline or simplify the way in which business is done. Composite applications can run the gamut from simple, fast-turnaround projects to fundamental business restructuring that may go so far as to redefine the relationships between businesses.

When analyzing project scope, we start by looking at the underlying *intent* of the business practice that the composite application will address. The intent is the goal or ultimate purpose of the practice, whether explicitly stated or entirely implicit. Some process intents include:

- *Creating customer value* in manufacturing or product creation
- *Making that value known* to customers in marketing and sales
- *Conforming to a set of externally imposed standards* like U.S. Food and Drug Administration (FDA) regulations, environmental compliance, or Global Trade Security
- *Providing an employee service* in human resources and benefits
- *Running the business itself* with accounting and financial tracking
- Redesigning business processes requires an understanding of such high-level business intents. The starting point for improving business processes is a clear and unambiguous description of the "as-is" practice—the way the business *currently* accomplishes the work. Such a concrete representation supports conversations about changes by getting everyone on the same page from the start.

Individual user data always plays an integral role in understanding the intent of the business process and the as-is practice. Process innovation requires that an organization is capable of seeing its own real-work practice. Once the practice has been revealed, it can be supported,

extended, transformed, rationalized, or fixed to meet the requirements of the business.

A composite application also supports people's work tasks in a day-to-day, minute-by-minute fashion. The project stakeholders may have an abstract, high-level view of the work of a business process as it flows through their part of the organization, but this abstraction idealizes what really goes on in the daily life of the organization. Too much of what really happens in daily life is tacit; workers act in the context of their culture, using formal and informal procedures that have become habitual, responding to the usual and unusual daily tasks, through ongoing collaborations that make up the fabric of working life. These habits, expectations, and implicit rules of practice are no longer conscious. Traditional interviews with stakeholders are simply not the best source of information about the real business practice and they do not lead to the detailed data that interaction designers need.

For example, in the product innovation process, the formal process descriptions described a clean stage/gate handover process that ensures the standardization and quality of new product development. But it did not describe how individual product managers struggled to collect and consolidate new product ideas, represent the requirements clearly, and prioritize new features so that all would agree to the final product definition.

So to redesign the business process with attention to how things really work in practice, look at the day-to-day activities of real people and work groups. To understand the actual work practice of people, gather low-level detail about daily work life. The Contextual Design process reveals such work practice by observing real people doing the real jobs that keep their companies running.

Revealing the work practice of individuals as it is actually done requires two things:

- **Contextual Interviews**
  Collecting observational data about what people really do by watching people work and talking with people in the context of their daily

life. Such field data is collected from all the key roles that make a process work. This data reveals what is really going on across the organization.

- **Work Modeling**
  Techniques that lay out the original field data in the form of diagrams or other notations to represent the structure of the work, replete with all the variation of the real business. These models show the high-level business process as it really is with all the actual breakdowns and opportunities to guide redesign. They also show the low-level detail about roles and tasks that allow for design at the lower levels.

Spectacular failures have resulted from enterprise system implementations that failed to respect the work practice nuances and business model variations of the implementing organization. The key to successful requirements definition is to build detailed work models from reliable field data. The models articulate what people are doing, why they are doing it, and how the activities of one set of people impacts and drives the activities of others. This high-level picture of the as-is business process gives designers the tools to see what needs to be changed or improved. The detailed data embedded in the work models can be used to re-engineer work practice and to help the designer create a new system solution, define the concrete functionality, and pick the *UI building blocks* to use.

Development with UI Building Blocks

The third key concept of creating composite applications is development with UI building blocks*:* a set of platform components that can easily be assembled into a coherent system. These components help to synthesize user interface, workflow, and underlying information flow into a system design that is simple, effective, and intuitive for users.

UI building blocks are similar to using design patterns, which have become a popular theme in software development. Few problems are wholly new; in every area of design, we see the same patterns appear

over and over again. In the design of enterprise systems, a few patterns of work practice repeat frequently. People organize their action items into task lists; they create triggers as reminders to do work; they follow step-by-step procedures to perform complex actions; and they monitor the state of the work that is under their control.

It's a waste of time and development resources to address these problems time and again, as though they were new. It's far better to design a powerful, effective solution for each work pattern only once and then, reuse this solution when needed. This is what the UI building blocks are, namely, an encapsulated software solution supporting a work practice pattern. Designed and tested with user-centered methods, they build effective user interface design and application flow into the system that incorporates them.

Projects based on UI building blocks focus on the business problem they are solving rather than on the details of UI design. Many development teams do not have the deep UI design skill required to design a good interface; even those that do rarely have the time in their project schedule to do more than barebones UI design. With all the other tasks of system design—structuring the data, designing workflows, figuring out how back-end connectivity should work—UI design tends to take a back seat. The UI building blocks help development teams create a system with a productive and pleasing user experience quickly.

We'll now introduce some of the key UI building blocks provided by the SAP business process platform, each of them supporting common work patterns:

- The *Control Center* provides a set of organizing views with an overview of the user's entire work history on a particular project. It collects work-related information into a set of distinct views, enabling the user to see what is urgent, track what is ongoing, and take action when necessary.
- *Work Centers* provide a single coherent place to do a set of closely related tasks. They collect the things being worked on, the views and

status indicators for managing ongoing activities and receiving new work triggers, and the tools required to take action.

- *Ad-hoc Activity Centers* provide a coherent place to do ad-hoc work on a specific task or project. They collect the views and status indicators to monitor the ongoing activity, as well as everything else required to accomplish the task. They save the user's context for the duration of the task, and ensure that it disappears when it is no longer needed.
- *Guided Procedures* support business processes that may cross people or organizations. They give an overview of the entire process, including who the contributors are and the current state of the process; they also give individual contributors activity centers in which to do their part of the work.
- *Object instance views* provide workspaces for manipulating the objects representing business concerns (a customer order, for example). They allow state and properties of the object to be viewed and modified (as appropriate) and provide access to functions that work on that object.
- *Dashboards* provide at-a-glance information about ongoing work and business objects that are important to the user at a particular moment. They summarize the status, show key information, and provide access to the object or work process itself.
- *Worklists* display the list of work items that the user is asked to act upon. These work items can be pushed ad-hoc requests, system-generated business objects that require attention, or standard tasks generated by workflow for which the user is responsible.

By taking advantage of these building blocks, a development team eliminates much of its design work. The building blocks provide a user interface framework along with key elements of that user interface; they offer developers with a common method of accessing and presenting legacy data; and they make available a structure and models into which any new user interfaces can be designed.

Assembling a Composite Application

When building a composite application, the technology platform does the job of understanding and managing the internal connections. The designer starts with a set of abstractions for the user interface, core services provided by platform components like content management systems, and enterprise services for functionality provided by existing engines or applications. With these at hand, design becomes a straightforward process of mapping requirements to preexisting UI components.

In order to be effective, such a pattern-based design approach depends on having detailed data about the real work practice of the people in the organization. Without such data, the designer cannot respond to the real problems in the work and requirements cannot be mapped to UI building blocks that support a seamless work practice for their users.

This book will help you to understand how to collect and model requirements in such a way that they can be smoothly mapped to reusable UI building blocks such as those provided by SAP NetWeaver. With this understanding of the work practice, supported by the SAP business process platform and the reliable design patterns in the UI building blocks, businesses can start innovating again. The bar for the support of new ideas is lower. The time to market is shorter, which means more ideas can be implemented and more ideas will have a positive return on investment (ROI). Furthermore, the existing infrastructure can be leveraged to provide more information and functionality to a greater number of users.

The following is an example of an SAP xApps packaged composite application that was defined using SAP NetWeaver to solve real-world needs. Throughout the book, we'll return to this subject, that is, the composite application, to illustrate various points.

SAP xApp Product Definition (SAP xPD) addresses the needs of concept development and requirements management in the product lifecycle management (PLM) domain. The project team was chartered to find a business case for an application in the PLM domain, a key SAP business

solution area and target software market. It was clear from the beginning that elements of traditional PLM—planning (such as product planning and implementation), design (such as computer-aided design tools), after-market provisioning (such as spare parts and service fleet management), and so on—were fairly mature market segments. However, the initial activities of the product lifecycle—things like idea capture, breaking down requirements, developing and evaluating a new concept, technical feasibility, and risk analysis—appear to have been largely ignored by other PLM software vendors.

Using a customer-centered design process similar to what we describe in this book, the SAP xPD development team produced an application for this area. It provides simple structures for surfacing and managing new requirements and product concepts, evaluating them through procedures that cut across multiple disciplines, and helping them to evolve into formal product proposals. The SAP xPD xApp coordinates the work of marketing research, business development, and product engineering to ensure that all perspectives contribute to the product concept. It provides summary management views of the number of product concepts under consideration and the state of those concepts. And it incorporates data from legacy systems without restructuring or re-implementing those systems.

We use the SAP xPD project as an example of the recommended design process throughout this book.

6 Creating and Iterating the Designs

UI building blocks are central to envisioning a new work practice for users and a new application design to support it. The consolidated models that we discussed in previous chapters enable the team to see existing work practice and to understand the intents behind the existing work practice at the user level and at the enterprise level. These insights and observations help to design a new solution that addresses business as well as individual needs. The new solution restructures the work practice and reengineers the business process with the help of UI building blocks.

Applications work best when they are designed from an understanding of how they can support a new way of working for the individual, the work group, and the business. Any application design benefits from developing a vision that describes the new work practice, considers how appropriate UI building blocks might be used, and decides what additional function and applications might be needed to best support the work. Depending on the type of project, this high-level vision can be detailed in storyboards or scenarios before mapping it to a particular UI design leveraging the UI building blocks. But no matter what type of project you are running—streamlining, surfacing, or invention—you want to be sure that the new system concept is tested with users and iterated until it meets their requirements.

In this chapter, we outline the steps of Contextual Design that can help your team produce a best-in-class application leveraging UI building blocks:

- **Visioning**
  Designing the new work practice of the organization as it will be supported by the new system; created as a story by the entire team
- **Storyboarding**
  Working out the details of the new work practice, step by step, using pictures augmented with text descriptions

- **Paper Prototyping**
  Testing and iterating the new design on paper, with users, by mocking up their own work scenarios in a rough paper prototype of the new application

In addition to driving the design, the team needs to be able to communicate the design to stakeholders. We end this chapter by discussing the role of *personas*, *storyboards*, and *scenarios* in sharing the redesign concepts and creating buy-in with stakeholders and users. This is particularly important when developing applications for users internal to the business.

6.1 Visioning a Redesigned Process

As we've seen, the complexity of the project dictates the complexity of the process that is required to handle the problem. A simple streamlining project might be able to do much of this reinvention work through building and testing the building block design in paper prototypes of user interfaces. But surfacing and inventing projects affect the work practice more radically; the team needs processes to help them think about the impact of proposed changes. Furthermore, because any project is likely to have its unexpected problems, where what seemed well-understood turns out to require a new design, it's useful for everyone to understand the processes for working through that new design from user data as a team.

Visioning is the process that supports this design step. It allows a team to work together on redesigning the work practice, inventing system support as needed, using the UI building blocks. Contextual Design represents visions as drawings showing the story of users interacting with the proposed new system and how the work practice is transformed by the introduction of technology. A vision depicts how manual practices, human interactions, and tools come together with new application design to better support the whole practice. Visioning identifies needed function in the context of the larger work practice. This technique ensures that the team members postpone lower-level decisions about implementation, platform, and user interface until

they have a clear picture of how their solution fits into the whole of the practice.

The primary intent of visioning is to redesign the work practice, not to design a user interface. Also, because a visioning session is a group activity, it fosters a shared understanding among team members and helps them to use their different points of view to push creativity.

Whether you're a team of two or six, the visioning process helps you work out the details of the new work practice for your users. It synthesizes design concepts into a new process that fits with the overall business process and the technical realities of the systems that will support it—be they legacy applications, new systems, or composite applications. If a project doesn't require much new work practice design, visions may simply give the team a way to lay out how the UI building blocks will be used to support the existing process. Streamlining projects are most likely to leave the basic structure of existing work practice unchanged, although for more complex projects, visioning at the task level ensures that the new design can support the existing process.

Visions can be "visionary" or follow the constraints of the existing business structure or technology. This decision is driven by the scope of the project, but it does not determine the level of innovation regarding the solution. The solution can be very innovative, even if traditional technology is used and vice versa. However, framing the vision by deciding about the vision scope increases efficiency and gets more buy-in from development.

Visioning need not be a time-consuming or complicated process. It's possible and appropriate to do a few quick visions in a couple of hours for a simpler project. After multiple visions, the team evaluates the options and synthesizes them into a single vision incorporating the best parts of each.

Throughout visioning and storyboarding (discussed in the next section), the team incorporates UI building blocks into their thinking. They use the vision and consolidated data to drive selecting appropriate

building blocks, filling them out with appropriate work instances, and tailoring them to meet the needs of users and the business process. During visioning, the use of building blocks is just sketched out—the details will be filled in when storyboarding takes place. Visioning is best done when the major data collection has been completed and the results analyzed.

Visioning is a relatively simple team process that works well to synthesize ideas into a coherent whole. It can be combined with more formal presentations and working sessions, but for many projects, visioning is a process that works well.

The Visioning Session

The first step to a visioning session is to review each model and the affinity diagram, in turn, immersing the team in customer data so their designs are grounded in the users' work. Each designer or stakeholder starts by reviewing the data individually, generating design ideas, or envisioning new technology that better meets the needs of the users, the business process, and the organization. Team members compare ideas and begin to get a shared idea of how to respond to the data. This individual activity helps to stimulate general team discussion about the potential redesign, which prepares everyone for the visioning session itself.

Anyone involved in visioning should participate in "walking the data" that is usually hanging up on the wall of a team room. Without this "walk," the process is no longer data-driven; anyone could arrive and offer his or her favorite design ideas based on feelings and prejudices. Walking customer data results in the selection and tailoring of preexisting ideas to fit the needs of the population. Since the visioning process evaluates visions based, in part, on their fit to the data, knowing the data is important for anyone participating in the process.

The team also reviews the UI building blocks. Just as walking the data puts the customer data in the designers' head, reviewing the design patterns puts the technology in their head. When customer data and

technological possibility come together in the head of a good designer, innovation happens.

To start the visioning session, the team picks a starting point and builds a story of the new work practice. A starting point may be the beginning of the known process, the story of one key member of a work group, or the story of a key task.

The team builds and draws the story on a flip chart, fitting ideas from each team member into the story as it unfolds. The story describes the new work practice, showing people, roles, systems, and anything else the vision requires. As the story touches each role and task, the team introduces appropriate UI building blocks; for example, when the user starts doing the work of a role, the team might introduce the Work Center building block to structure support for that role. The team does not worry about practicality at this point; all ideas are included.

During the vision and between visions, the team should ask itself whether it is accounting for all the data and design options it has available. Have all roles been considered? Are their key responsibilities and tasks well supported, with places in the design that allow the users to focus on the needs of that role or task? Have UI building blocks been used appropriately to organize and integrate the work of the different roles? Are there other designs, perhaps from previous projects in the organization, that should be considered and reused?

A team often generates several visions to cover all aspects of the problem and to drive its own creativity. The team may envision from multiple different assumptions how to restructure a work practice for increased efficiency or how to rely on increasingly leading-edge technology.

This results in a few (between three and five is good) visions for the team to work from. The team evaluates them with a simple process: for each vision, by first listing the good points—ways in which the vision supports the work practice or is easy to do organizationally or technically. Then the bad points are listed—ways in which the vision does not support users, depends on unrealistic technology, or is hard to do orga-

nizationally. While listing these problems, the team captures ideas to overcome these bad points. Finally, the team uses this evaluation to identify the best points of the different visions and synthesize them into one coherent work-practice solution.

Figure 6.1 shows an example from one of the vision sessions during which the team brainstormed how to support collection and evaluation of new product ideas. Instead of having one big pile of unsorted ideas, the system design helps to classify ideas and generate personalized inboxes for each product or topic owner based on personalization rules and role-based settings. For each of the ideas, the system offers the initiation of a standardized review process during which the ideas are assessed from various experts.

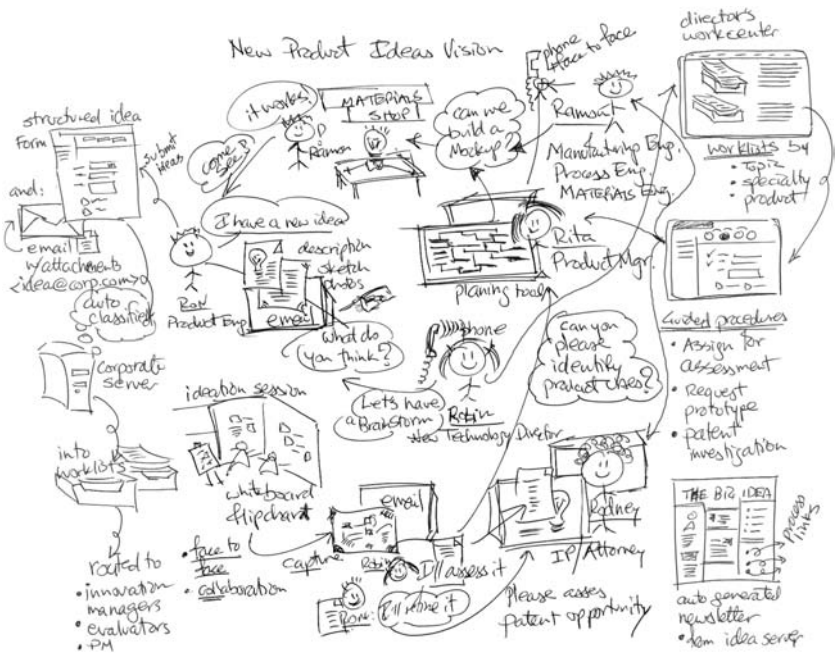

Figure 6.1 Sample Output from One of the Vision Sessions

6.2 Storyboarding the Details of the New Process

The consolidated vision represents a high-level idea of what the new system will be. To make it real, the team needs to work out the details—exactly what content will be available in which building blocks, what properties to expose, which values to watch in dashboards, and so on. *Storyboarding* or narrative scenarios is a way of working through these details that keeps work practice coherent without imposing a lot of process overhead. You need a story to test the flow of system interaction within and between building blocks.

Storyboards, like those used to plan a movie, combine pictures and text to describe the action. A storyboard cell is hand-drawn on a piece of paper (half sheets are about the right size) and depicts one step in the work practice. Some cells just show action between people; other cells show people interacting with the system being designed; and still others show what the system does internally to make user-visible behavior possible. The sketches are annotated with text to describe the action.

When a storyboard step shows the user interacting with the system, the interaction can use a UI building block. Where a building block exists to support the storyboard cell, designers sketch the building block and annotate it with the content the user needs at this point. Object instance views or Guided Procedure steps are sketched out in the action area of the building block. Where legacy systems provide a function that is embedded in the new system, those interfaces are sketched as they will appear in the new system.

Each storyboard describes one particular work situation or case. It shows how a task is accomplished following a particular strategy. If the task requires collaboration among multiple people or if it is handed off from person to person, one storyboard can describe the entire task. However, if you've observed two different strategies in your user population and the new design will support both, you need two storyboards—one for each strategy.

Storyboards are similar to midlevel use cases, focusing on the work steps of a task. But, unlike use cases, storyboards are pictorial—they allow designers to sketch out the interfaces they envision and embed those interfaces in the work practice they are designing. This ties together the whole user experience and supports the designers' need to see (literally) the system that they are creating. At the same time, the graphical nature of storyboards prevents the team from getting too detailed—complete specification of structure and function follows the step of working things out.

The steps and strategies of a work practice, being tacit, are easy to overlook. But sequence models capture the real moment-by-moment work activities. Working through detailed stories, guided by sequence consolidations and swim-lane diagrams, keeps the team honest and the design clean. Guided by the detailed models, the vision is made real. This ensures that the team does not overlook any intents and steps that are critical to the work. Even if the work is to be changed, the team has to think through the details of how it will be changed to ensure that adoption is easy.

Once the envisioned story and the set of building blocks are stabilized, you may capture the final version in the form of narrative scenarios that help you to test the flow of interaction. They also serve as use patterns for design sessions and usability testing. Such scenarios may be attached to personas to facilitate design sessions that should stay focused on real user needs.

6.3 Iterating Design

Regardless of project type, whether streamlining, surfacing, or inventing, each team must test the proposed user interface and overall system design. Managing risk means ensuring that the design works for the people and the business process, and is bought into by the people who will be using the application. Mock-up testing is the best way to simultaneously test the design, discover overlooked functions, and involve stakeholders in co-creating the final solution. For product companies, iteration of the design finalizes the requirements, tests the

design for its use across companies, and generates sales excitement among customers. No matter what the project type, mock-up testing is critical to a team's success.

Even when based on data, the design is actually a theory about how users can work better. To find out if this theory is correct, it must be tested—and it can be tested only with the people who will use it. But users—business workers—are not data modelers or systems designers. They do not understand the implications of an object model or a flow model on their own work practice. They cannot even articulate their work practice because it is tacit, so how can they tell if changes to the system and their work practice will work?

An accurate test of the new application depends on a representation of the system that users can both understand *and* interact with. This means the team must represent their ideas as a user interface. Mock-up testing lets users interact with user interface prototypes as though they were real. Users can provide reliable feedback on whether they like the work practice that results. During the mock-up interview, the designer can get feedback on low-level details of the system design that would be very difficult to confirm in any other way.

Low-Fidelity Prototype Testing

The fastest and most efficient way to build the first system prototype is in paper, using normal stationary supplies. Card stock provides a stable background to simulate the screen. Sticky notes effectively simulate anything that might be moved during an interview, such as menus, dialogs, or buttons. Sample content is put on a removable sheet so that users can replace it with their own, real content during the interview. Designs that include new hardware can use other kinds of props to simulate devices, robots, panel interfaces, and whatever else is needed. The final prototype may be rough, but it represents both the system's structure and its behavior.

Figure 6.2 shows an example of a low-fidelity paper prototype illustrating the idea of organizing requirements within the SAP xPD Work Center. Similar mock-ups along with other pieces were presented to the

users who pretended to use it for their real work. Changes to the design were made in the moment in response to issues. This process was used to reveal insights about what functionality is needed for an efficient management of requirements and how users wanted to organize their buckets of requirements, e.g. personal collections vs. formal requirements structures.

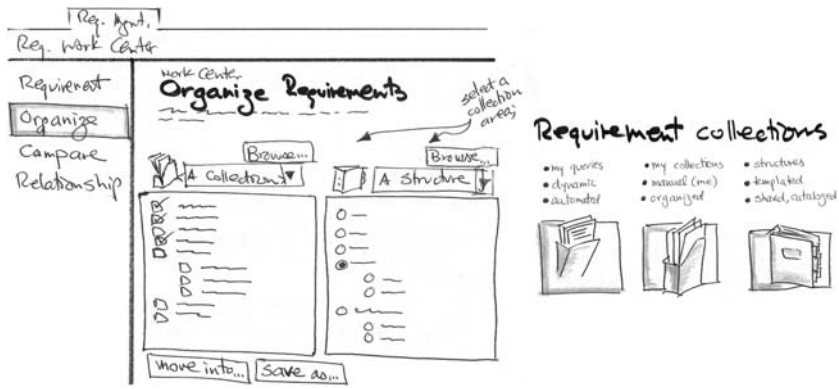

Figure 6.2 Example of Low-Fidelity Paper Prototype for Requirements Within Management Work Center

Mock-up interviews help designers understand why design elements work or fail and help identify new functions. These interviews are based on the principles of Contextual Interviews that we described earlier. The team tests the paper prototype with users in their own context to keep them grounded in their real work practice. Users interact with the prototype by writing in their own content and by manipulating and modifying the prototype. The partnership is one of codesign—as users work with the prototype following a task they need to do or did in the recent past, the user and interviewer uncover problems and in real time, they change the prototype to fix them. Together the user and interviewer interpret what is going on in the usage and come up with alternative designs instantiated in paper. Hand-drawn paper prototypes make it clear to the user that icons, layout, and other interface details are not central to the purpose of the interviews. Because the prototypes are rough, they keep the user focused on testing structure and function.

The mock-up interview is a field interview, but is conducted by two people. One team member manipulates the prototype, presenting new screens as needed by the task that the user is performing. The other person takes detailed notes of what is touched or changed, as well as the overall response to the prototype. The interpretation session captures issues and changes to each screen at the level of function, interaction design, and overall usability. The team pays special attention to whether UI building blocks are really working or need to be extended, to the support for collaboration and workflow, and to whether the full task activities and data needs are being addressed.

After the team tests and interprets the prototype with three to four users, the team redesigns it to respond to the feedback. Multiple rounds of interviews and iterations allow testing in increasing levels of detail, first addressing structural issues, then, user interface theme and layout issues, and finally, detailed user interaction issues. Over the multiple rounds, the application is tested with all the different roles that will use it. Before stabilizing the requirements and design of the system, we recommend three rounds of iteration with each business role that the system is supposed to support.

6.4 Visioning and Prototyping for Different Kinds of Projects

The kind of project you have affects how you use visioning, storyboarding, and prototyping. Here are some ideas for tweaking the process based on your project type:

Surfacing and *inventing* projects benefit from running visioning sessions after consolidating requirements and building an affinity diagram. Inventing projects, because of their wide impact, may run multiple visioning sessions with different stakeholders in the project. This is a useful way to get organizational buy-in, including buy-in from the client organization that has to adopt the change. By seeing and contributing to early ideas, the client organization's task of introducing the new application becomes much easier. Product groups need to con-

vince their development organizations to take a new direction—not only management and marketing, but also the teams working on related products.

The team should test their new design with paper mockups, using several rounds to incorporate feedback and flesh out the design. Once the design has stabilized, the team can move on to visual design and implementation. If they can generate a running prototype or HTML mockup, the design will benefit from additional Contextual Interviews with key users to catch low-level interaction and usability issues. And if the team can implement a working version on top of the underlying technology, then performance, real data lookup, and links can be tested.

Working with your users throughout the development cycle for any composite application project ensures that the function and layout is working at every level of detail. As always, more work with the user population ensures greater buy-in as the team expands the numbers of users interviewed with each round of iteration.

Streamlining projects can move directly from initial prototype (developed through the direct UI mapping process described in Section 3.2.3) to prototype iterations, but they can benefit from lightweight visioning and storyboarding to develop the initial prototype. More complex streamlining projects should at least storyboard the new work practice to ensure that the UI building blocks do indeed cover the activities of the necessary tasks. Market research and data from the initial interviews form the source data for the vision and storyboards. The "plusses-and-minuses" process helps the team weed out the weak ideas and develop a more robust design. Reviewing of storyboards by potential users provides high-level feedback and increases buy-in.

A streamlining project can start prototyping immediately. With only a few Contextual Interviews and initial analysis to flesh out the UI building blocks needed, the team has a reasonable cut at a workable system. During the first iteration, the team must stay vigilant, looking for new practice and different ways of approaching existing practice than was anticipated during the brief initial analysis.

For streamlining projects, we recommend starting with a standard Contextual Interview to get an overview of the users' work process and ensure that it does indeed map reasonably well to the mock-up as planned. If the interviewer finds the user going beyond the work practice as understood, he may need to expand this portion of the interview by collecting standard Contextual Interview data. As the interviewer understands how to modify the mockup in order to incorporate the unexpected work practice, the mockup is introduced and the interviewer makes the necessary changes in real time. If the work practice is too different (or the interviewer is less conversant with UI building blocks), the data can be brought back to the team for consideration, consolidation, and redesign.

If the new work practice is well-supported by the mockup, the interviewer simply moves to the standard mock-up interview. In any situation where a new work practice appears, if the interviewer can identify an existing UI building block that could support it, he may sketch a prototype on the fly and revise it with the user.

6.5 Communicating a Design

In real organizations, no project team works in a vacuum. They must coordinate with other teams and gain approval from management. Communicating the proposed design clearly is one of a team's most important tasks.

Because any design process creates tangible artifacts throughout, it's generally straightforward to build communications describing a design. Here are some methods we've found to be useful:

- The *vision* is the team's high-level statement of what the design is and how it affects the work. Not only does it keep various members of the team working toward the same goal, it helps communicate to project stakeholders. When summarized with data and displayed as a slide show, the vision communicates the redesign intention and shows how UI building blocks can be leveraged for significant change. Internal projects can show the client organization what they

are getting and how it will affect them, producing buy-in and getting feedback on the design. Product developers can show marketing what the product will do, both for buy-in and for preselling to customer prospects. The vision shows developers what will be required of them so they can start investigating technology issues. For management, the team can show what is being created and why there is a need for it. At this point in the process, many teams organize sharing events to communicate the status and direction of their work.

- *Storyboards* are a convenient way of describing and communicating the new work practice, because they walk through the user's work as a coherent process and integrate UI sketches into the work. The team can walk the storyboard with stakeholders to show what the new system delivers and how it works; this raises understanding and acceptance among those who are not appropriate candidates for prototype interviews.
- *Personas* and *scenarios* are ways to describe "typical" users so that those unfamiliar with the data have a concrete representation of their users. The consolidated flow model provides all the information required for robust personas. First, you identify core users of the proposed system and note their principal roles from the flow models. Then, you look across the range of users studied and build representative users drawing on those key roles. The roles show their intents, their tasks, and their concerns; building these characteristics into the personas results in a robust representation of the different user archetypes that you want to support.

  Scenarios describe how these personas work in the context of different tasks and work situations. They describe the "to-be" work practice, building in the work practice designed in the vision. Each scenario illustrates a key work situation and shows how the different personas work alone or together to get the work accomplished.

  Scenarios are built from the vision and storyboards. They describe the sequence of work steps and handoffs between people defined by the storyboard. Like a storyboard, each scenario focuses on a single task and work situation (for example, driving a modification of an existing product versus inventing a product). Scenarios work best

with personas to provide an easy-to-follow description of who the users are and how they will work.

Product Manager	
Needs	▶ Database with product requirements and market data ▶ Portfolio tools to oversee product innovation efforts ▶ Planning tools for managing product changes
Goals	▶ Make sure that no cool idea is lost ▶ Drive innovation at reasonable costs and efforts

Table 6.1 Example of a Persona-Like Description of a Product Manager

Personas, scenarios, and storyboards are good communication devices. Slide shows embodying the vision and the initial data also help communicate the team's direction. Put personas on posters; take digital photos of the storyboard frames and display them in slides for stakeholders to review. Because scenarios tell the detailed story of the user's interaction with the system in an accessible form, potential users and business stakeholders can understand the proposed process and provide feedback. This is especially useful for internal teams to generate buy-in.

Index