Changing the Rules: A Pragmatic Approach to Product Development

by

Paul K. Huntwork, Douglas W. Muzzey, Christine M. Pietras,
and Dennis R. Wixon

ABSTRACT

Developing quality software rapidly and at low cost has been an elusive goal. Nevertheless, meeting this goal is essential in today's competitive environment where more and better products appear at accelerating rates and customers demand systems that support "what users need to do" in a natural and cost-effective manner. This paper discusses the processes used by the TeamLinks for Macintosh project team to achieve customer focus throughout the development of a groupware office product. Listening to customers radically reshaped the product and led to more rapid decisions, shorter development cycles, higher quality, and greater customer satisfaction.

WHERE WE STARTED

Product Overview

TeamLinks software allows Windows PCs and Macintosh computers to be integrated into enterprise-wide networks. The product utilizes Digital's extensive line of network applications and services, such as electronic mail, file sharing, workflow procedures, and work group applications.

The TeamLinks product also makes use of the latest personal productivity and client-server technology as a platform for comprehensive office solutions. Just as Digital's ALL-IN-1 Integrated Office System (IOS) allows organizations to rapidly develop organization-wide network applications in a timeshared environment, TeamLinks software provides capabilities that allow the creation of companywide client-server office applications tailored to meet the needs of any operation.

TeamLinks software provides customers with an intuitive graphical user interface that integrates their powerful personal productivity tools, such as word processing and spreadsheet applications, into local and wide area networks. This feature is independent of whether the user's desktop system is a Windows PC or a Macintosh computer.

Product Goals

For enterprise-wide work group computing strategies to have customer appeal, they must address both PC and Macintosh desktop computers. The introduction of TeamLinks for Windows during the spring/summer of 1992 further highlighted the need to immediately introduce similar functions on a Macintosh platform. The use of inside-outside strategic planning identified three primary factors that required consideration during the development of admissible product delivery strategies.[1]

First, we must satisfy the wants of the potentially available market. Customers require both Windows and Macintosh desktop solutions for their enterprise work group computing. Both the TeamLinks Program Office and customers requested a Macintosh platform that supported the core TeamLinks services of mail, ad hoc workflow, and filing, with product availability within six to nine months.

Second, we must deliver an acceptable solution with the available resources. Macintosh users are frequently recognized as demanding consumers of software applications. Although the breadth of experience in developing Macintosh products within the group was limited, the development team consciously planned objectives aimed at satisfying demanding consumers. The team's goals consisted of satisfying customers' basic office needs and having the product recognized as a quality TeamLinks implementation on the Macintosh platform.

Third, we must develop a product within the opportunities and constraints of today's environment. In many development environments, the reality of budgets with minimal and ever-decreasing resources is rapidly becoming today's normal mode of operation. Changing strategies, requirements, and management infrastructure are also particularly characteristic of current development environments.

Product Strategy

After resolving our initial project goals, we developed strategies to satisfy the goals. We chose to establish design partnerships with customers to iteratively obtain comments to use as a basis for refining the project's specific deliverables.

Most problem-solving strategies are simple variations of (1) define the problem, (2) develop solutions, (3) test, and (4) refine the solutions. The TeamLinks project team chose an iterative and concurrent adaptation of this strategy.

First, we identified our implicit working assumptions. Initially, the project assumed that all components present in the TeamLinks for Windows product would simply be ported to the Macintosh platform and retrofitted with a Macintosh user interface.

Second, we developed product plans based on our initial goals and implicit working assumptions. Iterative design techniques require prototypes that customers may evaluate and comment on. The project's initial product plans were utilized as the first product prototypes for collecting customer responses.

Third, we verified and refined our plans based on validated information. As product prototyping got under way, the team analyzed information from competitive products, industry consultants, and customers. A key consideration for the development team was that throughout the life cycle of the project, specific product deliverables would be changed as customer opinions became clear. As incoming data evolved into information, the cost and benefits of each change would be carefully weighed against the project's goals.

Product development thus proceeded on two fronts: one formulated in advance, the other created in response to new developments, customer comments, and experience with successes and failures of the plan.

Select the Best Work Model

Since the emergence of the software industry and continuing through the present, the ability of software groups to produce high-quality software has fallen far short of customer needs and demands. In response to this condition, government and academic specialists proclaimed a "software crisis" in 1969 and endorsed a concept of software engineering based on authoritative, hierarchical organizations and sequential application of specialized functions.[2] This model of software engineering is still prevalent in textbooks. Ironically, the model was created at a time when the competitive advantage of total worker participation in cross-functional teams, an outgrowth of Deming's approach to management, was being demonstrated in other industries.[3] The cross-functional approach is now widely recognized as a superior method of new product development. Figure 1 shows how cross-functional teams speed up work. Twenty-four years of the sequential model have not diminished the software crisis. We feel privileged to have been able to apply the cross-functional model to the development of the TeamLinks for Macintosh product. Descriptions of other best practices used by the TeamLinks team follow.

Find Out What Your Customer Needs

Determining the needs of our customers involved field research, quantitative research, and design justification through grounding.

Field Research.

One of the most powerful rationales for field

research is the realization that effective design begins with the discovery of exactly what users and customers want and do. Field research methods are designed to provide such in-depth understanding. These methods emphasize openness to user experience and create a dialog with users about that experience. Direct contact with users at early stages of design is viewed as an essential step, and the barrier between users and designers has been cited as a significant cause of suboptimal design.[4,5]

Quantitative Research. Given that discovery is the first stage to effective design, the next stage is decision.[6] Most likely, a team will not be able to respond to all user needs. Thus, it needs a systematic and objective way to make decisions. Quantitative methods provide a basis for decisions because they establish a dimension along which features can be compared.

Grounded Design. Unfortunately, many designs have an insufficient basis. Third-hand information, brainstorming, anecdotes from trade shows, and speculative talk about "what the customer really wants" within an isolated team all contribute to designs that do not meet customer needs and designs that do not reflect customer work. To ground a design means that all aspects of the design are rooted in customer data rather than in speculation. Providing mechanisms for this grounding is critical to producing an effective design.

Design Your Product Based on What You Learn

Demand pull, customer involvement, and design metaphors all contribute to a customer-focused product design.

Demand Pull. Using customer interaction to pull design features out of the development team greatly reduces the number of design decisions and the time required to make these decisions. A customer focus on work essentials and not on "bells and whistles" provides unambiguous feedback that supports direct decisions.[7]

Customer-driven Design. Design is a process of refinement and elaboration embedded in a cycle of creation and evaluation. Customer-driven design involves the evaluation of a tentative design (the creation) with the customer's evolving understanding of their work vis-a-vis the product.

Design Metaphors. Metaphors are an effective way to generate a design from customer work and technical capabilities. Examples include the "desktop" metaphor that drives much user interface design today. Although often criticized, metaphors have been shown to be very powerful and fundamental to human

thought.[8,9,10]

Refine Your Product with Customers

Using an iterative approach to product design combined with prototyping helps refine the product design.

Iterative Requirements. The need to break the development of complex software into manageable pieces has led to schemes such as "separation of concerns," "top-down development," and "step-wise refinement." Iterative design addresses this problem with a "basics first" approach. A basic idea is embodied in a prototype implementation and reviewed with customers. The iterative approach allows solutions to come into being and quickly converge to finished products under the influence of user interaction, even while users are discovering what they need. Detailed requirement specifications are not necessary to begin implementation, so there is no time lag between gathering requirements and providing solutions. This approach minimizes miscommunication and eliminates obsolete requirements.[11]

Prototyping. Prototyping supports a customer-driven design process, providing customers with an effective medium to respond to current system thinking.[12] For instance, user interface designs embody a theory about the way users work.[13] The most straightforward way to get feedback on the theory is to express it in a prototype. A prototype allows users to try the system directly instead of translating their work into an unfamiliar symbolic language.[14]

WHAT WE DID

The project team developed customer partnerships early in the project life cycle. Through Contextual Inquiries, focus groups, and artifact walk-throughs, the team internalized customer needs and requirements. The new data helped establish a shared understanding among team members and manifested itself in a new product design. Vector Comparative Analysis (VCA) data summarized team learnings and provided the foundation for new designs. Figure 2 diagrams this process.

Find Out What Your Customer Needs

Cross-functional Teams. The team comprised product managers, engineering managers, engineers (including some from companion products), account managers and support people, customer personnel, and specialists in marketing, human factors, graphic design, user publications, and competitive analysis. This

cross-functional team took training, visited customers, analyzed data, and made decisions as a whole or in cross-functional subgroups. The mutual understanding that grew out of the shared experience and the shared data enabled faster, more stable decisions and shorter schedules.

Customer Partners. We formed product-life-cycle partnerships at the start of the project with customers who represented the four industries that most heavily use PCs on the desktop: U.S. government contractors, manufacturing, pharmaceuticals, and banking. Within these industries, we identified Digital customers from the office partner group who used Macintosh PCs. Working with the account teams and the customers themselves, we selected partners who represented their industries. Each partner designated a specific person to coordinate their participation.

These partnerships allowed more interaction, better follow-up, clearer communication, and more consistent direction. For example, we could model their work in detail in later versions of the prototypes, and the partners could perform complex evaluations. Since we were familiar with their work and they were familiar with our product, no one experienced a high cost of learning at any stage of the project.

Contextual Inquiry. We decided to train the team in Contextual Inquiry methods so that they could interact more effectively with customers. Contextual Inquiry techniques are adaptations of the methods used by anthropologists and sociologists to understand other cultures. The Contextual Inquiry framework emphasizes three principles: (1) context, i.e., study user work in its natural environment; (2) partnership, i.e., engage customers as co-investigators to help develop your understanding; and (3) focus, i.e., clarify your interests and assumptions and be willing to change them based on what customers tell you.[15] Contextual Inquiry techniques have been used widely at Digital and have shown a positive impact on market penetration and revenue.[16]

Customer Survey. Information from customer visits was organized into a single hierarchy with benefits and needs at the top and desired capabilities and features at the bottom. A questionnaire was created to obtain quantitative customer importance weights for each node and leaf of the hierarchy. The questionnaire was sent to the customer partners. We encouraged multiple responses from each partner to get data from both Information System professionals and end users. We also collected importance weights from an industry consultant and additional customers beyond the partners. Figure 3 shows a typical question from the questionnaire.

Customer Day. Representatives from the four customer partners brought completed questionnaires to a customer day. We inquired about their experience with the questions, looking for omissions and refinements. We asked them to describe their top 10 issues and explain why they are important in their environment. The customer day information provided additional insight into user needs as well as a sanity check of the quantitative survey data.

Competitive Benchmarking. We created a score sheet from the features at the lowest level of the hierarchy developed for the customer survey. Engineers on the TeamLinks project, an industry consulting firm, and customers scored our existing products, alternative versions of our planned product, and competing offerings. The scoring by engineers directly contributed to their understanding of customer requirements. The information also fed the VCA process. Figure 4 shows a typical question from the score sheet.

Cross Validation. To minimize investment risks and to maximize the return on the wealth of information obtained from the data-gathering exercises, we revalidated the information to determine its applicability to the project. The information was cross-validated by comparing multiple sources, including the competition, industry consultants, and customers. We verified that we could understand different responses as true expressions of different needs before we used the data.

Vector Comparative Analysis. We input the customer importance weights from the questionnaire and the feature scores from the score sheet into the computer-based VCA tool.[17] This tool rolls the feature scores up through the hierarchy by a method of weighted averages to provide a score at each node. VCA can create a vector diagram for each node showing graphically how well each product satisfies the user needs represented by the node. Figure 5 shows the top few branches in the TeamLinks VCA hierarchy. Digital developed VCA for use with or as an alternative to Quality Function Deployment (QFD). For the TeamLinks project, no QFD was conducted.

Artifact Walk-throughs. Based on Contextual Inquiry principles, artifact walk-throughs allow a design team to look at processes that take place over time and that occur among groups of people. The name is derived from the approach of asking customers to bring the actual artifacts of a process, e.g., notes, memos, forms, and documents, into the walk-through as a reminder of the full complexity of the process. In the presence of the artifacts, we ask for the overall process goals, any known issues and problems, and a list of process steps. For each step of the process we ask, Who makes requests? Who does work? Who approves? What is the cost in person effort, materials, and equipment? What

is the normal cycle time? and What problems and issues exist with this step? Each type of information is recorded on a colored Post-it note and assembled into an annotated flow diagram of the process. Thus, these walk-throughs emphasize articulating a process in detail, grounding it in a specific customer example. We chose artifact walk-throughs as the natural approach to gathering data in order to customize our prototypes to each customer situation. At the same time, the walk-throughs uncovered additional general requirements.

Design Your Product Based on What You Learn

Team Discussions. The Contextual Inquiry results contained surprises. Even though the inquiry focus was on office products, customers expressed more requirements about cost containment than about product features. The messages, discussed in detail in the section What We Learned, were clear in the raw data and became the basis for revised plans even as the rigorous VCA was being completed. At this time, an early prototype, seen only by the development team, was redirected. Real customer data enabled rapid consensus within the team on changes to the project's direction.

Competitive Positioning. The survey and benchmark data, which was processed by VCA, allowed us to track our competitive position at all times. We could say, for instance, "If we build this alternative, we will satisfy more customers than competitor A but will need more mail features to compete with B." In addition, when the engineers performed the benchmarking in person, they learned more than just scores. One engineer decided to keep the competitive product he benchmarked as a working tool until our own replacement product was ready, because the competitor's product was better than the tools he had been using. Such experiences challenge the engineers to build better products.

Trade-off Analysis. The computer-based VCA tool allowed precise numerical comparisons to be made on demand. Many alternatives, ranging from the most probable plan, through minor variations, to wild "what-if" scenarios, could be analyzed. The graphical displays allowed the trade-offs between alternatives to be understood at a glance. Low customer-impact branches of the hierarchy could be identified and ignored during the period when basic directions were being established, thus simplifying the design process. Figure 6 is a representation of a VCA display, annotated to clarify how the charts are to be read. In particular, the importance of an item is indicated by the angle of the vector representing it --- the more important the item, the nearer the angle is to vertical. The length of a vector shows how well the item is realized in a given plan --- the better the

realization, the longer the vector. Therefore, long vertical vectors represent important items that are implemented well, and short horizontal vectors represent unimportant items that are not implemented well.

Refine Your Product with Customers

In addition to the techniques already described to bring customer input into the design of TeamLinks for Macintosh, we used four cycles of prototyping to confirm and refine our designs. In preparation for the third cycle, we conducted artifact walk-throughs with each customer partner as described earlier. The walk-through information enabled us to simulate real processes during the final prototype cycle, thus putting our products to an ultimate test. The four cycles are shown in Table 1.

Table 1 The TeamLinks Prototyping Cycles

Cycle	Content	Presentation	Data Collection
1	User interface facade	Macintosh Powerbook	One-on-one contextual interviews
2	User interface and limited functionality	Client software only	Sample tasks (scenarios), user diaries, and phone calls
3	Usable workflow, filing, and basic mail	Client and server software	Customer forms and work tasks, user diaries, and phone calls
4	Full functionality	Client and server software	Daily use, visits by team, and phone calls

WHAT WE LEARNED

Significant changes in functionality and the user interface were made based on user reaction to the prototypes. This section discusses these changes.

Unlearning Things We Thought We Knew

Throughout this paper, we focus on three main themes: (1) find out what your customer needs, (2) design your product based on what you learn, and (3) refine your product with customers.

The previous section of the paper discussed tools and techniques that we used to achieve these goals. Before actively gathering data, we developed a set of assumptions about our customer's needs and preferences for working. On subsequent visits we discovered that some of our assumptions were flawed and that we needed to change our original plans to better satisfy customer demand. In this section, we describe our initial assumptions, discoveries made throughout the data-gathering process, and new designs derived from our discoveries. Table 2 lists a comparison of our original and revised designs.

Table 2 Comparison of Original and Revised Designs

Original Design	Discovery	Revised Design
Mail		
Develop new X.400 TeamLinks mail client for Macintosh.	"Build one mail client and do it right."	Leverage existing X.400 mail client and focus on developing mail-enabled workflow applications.
Workflow		
Develop information manager application that contains routing services.	"Help us utilize our available desktop resources." "Build a 'real' Mac product."	Develop independent components that work well with existing Macintosh applications.
Filing		
Develop information manager application, in addition to Mac file system.	"Document management should look and work like a Mac."	Provide access to ALL-IN-1 IOS file cabinet as an extension of the Macintosh file system.

Lesson One

Our initial assumption was that customers need an information manager to navigate and to view file cabinets. TeamLinks for Windows provided an information manager to assist Windows users in viewing, naming, and navigating the ALL-IN-1 IOS and DEC MAILworks file cabinets. The file cabinet is a logical container based upon the physical metaphor of a filing cabinet. It enforces a hierarchical relationship, providing drawers that contain only folders and folders that contain only documents. The file

cabinets represent the central storage areas for all objects within the TeamLinks environment.

To parallel the TeamLinks for Windows environment, the team proposed an information manager for the Macintosh product. Figure 7 shows the proposed information manager window. Users would be presented with a single, world view of the file cabinets through the information manager. This proposal adds an additional document management layer on top of the native document management. The team planned to display the information in a manner as similar as possible to the Macintosh desktop display.

However, our customers stressed: "Document management should look and work like the Mac." The Macintosh desktop presents a single, world view to the users. They do not want a replacement. Our partners urged us to support document views and navigation that is native. After attending the Apple Developers Conference, the project leader also concluded that we would build a noncompetitive application if we followed our proposed plans.

The team decided not to build an integrated information manager. The revised design in Figure 8 shows how users can access the remote ALL-IN-1 IOS file cabinet as they do remote network volumes. In this approach, the ALL-IN-1 IOS file cabinet becomes an extension to the file system. This paradigm builds upon the Macintosh user's prior knowledge, making the interface comfortable and familiar.

Lesson Two

Our initial assumption was that we should follow the TeamLinks for Windows lead and create one tightly integrated application. Given the TeamLinks for Windows working model, the team proposed to develop a similar application for the Macintosh platform. Original plans detailed a large, integrated application. The information manager window would provide the central world view of the file cabinet. This window would have its own set of menus and a tool bar. All other services would be available through the information manager menus and tool bar. Mail messages, workflow packages, and other documents would be stored in file cabinet folders. Users would open these objects by double-clicking to invoke the appropriate editor.

Each service would be represented by its own window with unique menus and a tool bar. Services would include mail, workflow, address book, directory lookup, and distribution list editing.

Rather than enhancing the existing X.400 mail client, DEC MAILworks for Macintosh, the team planned to create a new mail client for the TeamLinks product. This decision would have resulted in two competing mail clients.

However, our customers stressed: "Help us utilize our available

desktop resources." Digital's office products need to work with existing Macintosh applications. Customers want to use their existing word processing, graphics, and other business applications while working with our office applications. The customers emphasized that TeamLinks components must work well together.

Throughout our interviews we heard: "Build a real Mac product." Our customers stressed that our Macintosh office products must look and feel like Macintosh applications as well as adhere to the Apple Human Interface Guidelines. They encouraged us to take advantage of color, direct manipulation, and point-and-click paradigms. In following these standards, we enable users to transfer their skills from one application to another, thus reducing training costs.

We also heard: "Build one mail client and do it well." Customers want consistency across our applications. If two Digital office products provide X.400 mail support on the Macintosh platform, each should present the same user interface. This practice will help reduce customer costs by eliminating additional user training. From Digital's perspective, it makes good business sense to take advantage of existing products and resources where appropriate. Our customers cautioned against developing a new X.400 mail client for the TeamLinks product when DEC MAILworks for Macintosh already exists. They encouraged us to direct resources toward developing a single, strategic mail application that is simple to use, X.400 compliant, reliable, and available for the popular desktop computers. They mentioned mail-enabled applications, such as workflow, conferencing, and time management.

The team decided to take advantage of existing components. Rather than build a new mail client, the TeamLinks and DEC MAILworks for Macintosh project teams collaborated to enhance the existing DEC MAILworks client and provide workflow support.

The TeamLinks team focused on developing the workflow component that would assist users with routing forms and documents for review and approval. As a result, the TeamLinks design migrated from a large, integrated application to components that work well together and allow users to exchange information that they have created with other popular Macintosh applications. Depending upon specific needs, customers can purchase a mail-only package, a workflow package, or a comprehensive package with mail, workflow, remote ALL-IN-1 IOS file cabinet access, and conferencing applications. Throughout development, the team refined designs, adhered to Macintosh guidelines where possible, used color to add value, and implemented point-and-click paradigms.

Lesson Three

Our initial assumption was that time management is important, but

we still have time before missing the opportunity to implement this feature. Although time management was viewed as an important product requirement, the team did not fully appreciate the consequences of not implementing a time management solution. Due to limited resources, the team relied on another internal group to deliver these services. If a time management product were to become available before the TeamLinks release date, it might be integrated into the package.

However, our customers stressed: "Help me manage my time." Customers often described their struggle in trying to schedule a meeting with a group of people and quickly followed this description with a request for time management support. People spend a great deal of time trying to manage their calendars. Two of our four partners rated time management support as their top priority. People want to browse one another's calendars, get assistance in finding common meeting times, and schedule resources and events across their organization or company.

One partner stated that they would not be able to migrate their ALL-IN-1 IOS users to TeamLinks for Macintosh until a time management solution was in place. VCA data indicated that if TeamLinks for Macintosh had an integrated time management model, the product would be in better competitive standing.

An office industry consultant told us that we had only six months to release an integrated time management module. If we delayed any longer, we would miss the opportunity.

The team had been considering third-party time management providers, but negotiations had stalled. The team decided to reemphasize negotiations. A contract was signed within a short time.

Lesson Four

Our initial assumption was that we would port TeamLinks for Windows to the Macintosh platform and Mac users would like the results. We originally planned to port the TeamLinks for Windows application first and then retrofit a Macintosh user interface. The team proposed an initial design that contained a rich set of functions identical to those in TeamLinks for Windows but gave little thought to what Macintosh users really wanted from a groupware office application. The importance of simplicity and ease of use was not clear to all team members.

However, our customers stressed: "I don't learn new functions unless I see clear value to my work." "[The] most valuable tool is the one you [already] know how to use." "Less is better." "All I want to do is create mail and read it." "Build a real Mac product."

People use tools and applications to simplify work tasks. Tools

should support existing work rather than create new work. People use tools if they add value; otherwise, they quickly abandon them. Customers want simple, elegant solutions.

Porting TeamLinks for Windows to the Macintosh platform would not succeed even if a user interface that resembled an actual Macintosh user interface were provided. Macintosh users easily spot and freely reject a ported Windows application. Vendors who have ported Windows applications to the Macintosh platform have failed to gain product acceptance.

The team decided to adopt simplicity as a theme. Although mail and workflow add value, they must be simple to use. We decided to take advantage of our users' previous knowledge of electronic mail and the postal mail metaphor in the design of our workflow package. The team first concentrated on designing the most frequently used functions and then on refining them.

Our VCA results indicated that we had an opportunity in the workflow area but that the window of opportunity was quickly closing. To complete our designs and develop customer-specific templates for prototyping, we needed to learn more about our customers' business processes. We used artifact walk-throughs to study three workflow examples: a manufacturing procurement request, a pharmaceutical regulatory submission, and a banking credit approval.

Rather than port the Windows application, the team created a new design utilizing user interface prototyping tools. We adhered to Macintosh guidelines, incorporating standard system fonts, point-and-click selection, standard text selection routines, standard menus and accelerators, consistent button placement, and dialog layout.

Discovering Delighters

Through the discovery process, several of our initial assumptions proved to be inaccurate or misguided. As a result, the team changed plans to better satisfy customer requirements. We learned from the experience and adapted appropriately. The team also discovered that certain product attributes delighted customers.

Button Bar. Surprisingly, the button bar or tool bar within the TeamLinks components is a delighter among customers. The buttons provide point-and-click access to frequently used mail and workflow functions, reducing menu navigation and recall of keyboard accelerators. Colorful icons indicate button function. Context-sensitive help is also available as users pass the mouse pointer over buttons in the bar.

Workflow Automation. Data from Contextual Inquiries, artifact

walk-throughs, and VCA revealed that business process reengineering and automation is an emerging opportunity within the office automation market. Today, businesses lose time and money tracking materials through approval life cycles. Tools that support workflow automation can potentially yield substantial savings for a corporation. In some industries, trimming one hour from a process can save millions of dollars.

One customer expressed his interest in workflow support as follows: "It will mostly save everyone's time which is now wasted in tracking down who has the material and who still needs to sign it. It should speed up things, because it doesn't have to physically be sent from office to office (sometimes even different states) for approval. I would think it could save time at year end for summary reports."

The development team capitalized on this information, focusing the corporate office strategy on developing leadership workflow tools. Rather than provide a set of "me too" features, the team decided to concentrate on a specific customer problem and provide a simple, well-done solution. The TeamLinks Routing product is the outcome of these efforts, and the group intends to focus the marketing message on its tracking capabilities. Six months later, leading competitors are now hastening to announce workflow product offerings.

Refinement during Prototype Review. Our VCA results indicated that customers place great value in ease of use. Items from the benefit hierarchy such as "Make the product usable --- match the way I work," "Make the UI consistent within itself," and "[Make a] product [that] adds value to my work" were all rated as highly important by our customer partners. Users are specifically interested in minimal keystrokes, consistent interfaces and functions across components, point-and-click paradigms, adherence to Macintosh user interface standards, and short-cut keys.

The team focused on satisfying these requirements within the TeamLinks components. We employed a design methodology that involved users throughout the development life cycle, allowing users to see product improvements on a monthly basis. During early prototyping, the team conducted one-on-one sessions with users to study concept learning and ease of use. Feedback from these sessions was used to progressively change the design. Subsequent testing revealed that the design modifications improved ease of use. A summary of specific design changes follows.

Redesign of Main Window for TeamLinks Routing. A user receives new packages for review and approval in the mail in-box folder. To view the package, the user double-clicks on the package in the in-box folder, opening a window. The original screen design for the TeamLinks Routing package window appears in Figure 9.

Prototype testing demonstrated that users had difficulty focusing on important information in this window. The button bar immediately caught their attention, and their eyes were then drawn to the distinctive "Routing List..." button and the corresponding list of names. Several users overlooked the list of attachments at the bottom of the window. Many users were unable to locate their role instructions, which outlined their specific tasks. Finally, several users commented that important information, such as, What do I have to do with this? When do I have to respond? and What's my role? was not visible on the main screen.

Users had difficulty understanding that the window represented a package that contained several attachments and signatures. Users were familiar with mail messages. They easily understood the concept of message attachments and the postal metaphor as it relates to electronic mail. They associated a workflow package with a special type of mail message that needed approval, yet the package window did not resemble the familiar message window.

Users overwhelmingly liked the button bar, because frequently used functions were more accessible and visible.

After going through several design iterations, the package window now appears as shown in Figure 10. The team applied the mail metaphor to workflow, rearranging some of the information to create distinct header and attachment areas as seen with mail messages. The header contains Initiator (From), Initiated (Date), To, and Subject fields. Additionally, we added a Role field to the header in response to user requests. Text labels are displayed in a bold font to improve readability and to help users focus their attention.

We simplified the window by removing noncritical information. For example, although the data in the routing list is important to users, they do not require this information in the main window, as long as it is available with a single mouse click. Therefore, we added an Edit/View-Routing-List button on the left-hand side of the tool bar. Users are also able to quickly view the routing list by double-clicking on the To field. In addition, we removed the Routing List button, which needlessly distracted users.

The graphic designer created smaller buttons and used subtle shades of gray to create a three-dimensional look. Shading was used to invite users to press the buttons. Icons were designed to be understandable in international settings. Below the header, shading was used to define the attachments area, and a paper clip icon was added to reinforce the metaphor.

To address the difficulty users had in locating role instructions, we placed them in the attachments list. If instructions are present, they always appear as the first

attachment and are denoted by a distinct document icon. Users simply double-click on the list entry to find out what they need to do with the package.

In subsequent evaluations with the prototype, customers commented: "I think it's pretty good. Once you get into it, it's pretty easy to use, pretty logical." "I was already somewhat familiar with it because I saw base-level one. It was pretty easy coming back to it. Just from using it the first time, it became familiar. I had some problems with the last one [base-level one], and I think you've solved a lot of the problems with this one [base-level two]." "Anyone familiar with a Mac shouldn't have a problem."

In designing the package window to look more like a mail message, we enabled users to transfer their mail knowledge to workflow. The concept of creating a package could be related to the concept of creating a mail message, namely, addressing the workflow package, attaching documents to the package, and typing in a subject. These changes help to reduce the need for user training.

By simplifying the main window, we enabled users to focus on important information, i.e., their role instructions and the attached work materials. Providing icon buttons for frequently used functions helps to minimize keystrokes and save time.

Terminology Review. The original TeamLinks Routing product used a series of technical terms in the title bars of package windows to identify packages and states. These terms were not very meaningful to users. The original terms are listed in column one of Table 3.

Table 3 TeamLinks Workflow Terminology

Original Title Bar	Revised Title Bar
TeamRoute - Template	Template - <document title>
TeamRoute - (Master, Routing)	Original - <document title>
TeamRoute - (Master, Completed)	Completed Original - <document title>
TeamRoute - (Master, Unsent)	Draft - <document title>
TeamRoute - (Master, Sent)	Original - <document title>
TeamRoute - (Routing Copy, Pending)	Routing Copy - <document title>
TeamRoute - (Routing Copy, Sent)	Carbon Copy - <document title>
TeamRoute - (Carbon Copy, Read)	Carbon Copy - <document title>
TeamRoute - (Tracking Report, Read)	Latest Copy - <document title>

Team members working on the Windows and Macintosh platforms agreed to review terminology with the goal of reaching consensus on simple terms that users could immediately identify. The team reflected on the traditional terminology for routing paper

packages to develop the new terminology. The new terms are listed in column two of Table 3.

By using terms that reflect the paper process, users can immediately identify packages they receive and understand the appropriate actions to take. The terms Template, Original, Carbon Copy, and Routing Copy describe both package type and status in simple, familiar terms rather than in technical terms. The package name is placed in the title bar of the package window and is readily visible to the user. The revised terms help to minimize new learning and reduce frustration. Consistent use of terminology across platforms allows users to speak in common terms with colleagues using alternate desktop systems.

Focus on the Package. The team made a concerted effort to focus on all components of the TeamLinks Office package: mail, workflow, filing, and conferencing. As discussed earlier, the process of iterative design yielded excellent results with TeamLinks Routing. Studies of prototypes demonstrated that the use of buttons, color, larger fonts and professional graphics, the mail metaphor, and adherence to Macintosh standards all contributed to ease of use and acceptance of the TeamLinks Routing product.

VCA results indicated that our customers viewed consistency across components as essential to minimizing training and increasing accessibility. Given this information, our goal was to produce a family of products with a consistent look and feel. The team spent six weeks working on mail enhancements, modifying the screens to be more consistent with TeamLinks Routing. For example, the graphic designer created more meaningful icons for the buttons, adding color to reinforce metaphors and make the buttons more distinct from one another. The team agreed on consistent button placement across components, moving all buttons to the top of mail windows. Similar font styles and sizes were used across components to increase readability. Figure 11 shows the original mail file cabinet window. Figure 12 shows the same window with the enhancements just mentioned.

In addition to focusing on consistency across user interfaces for mail, workflow, filing, and conferencing, the team employed the same graphic for the on-screen "About" boxes and for the packaging and documentation cover designs.

Consistency across product components and with other Macintosh applications received rave reviews from customers: "I liked the buttons across the top real well. Real nice." "The fact that it's consistent with other Mac applications is the best news." "Support for point-and-click --- you did a good job here."

By creating a similar look and feel across components, the team reduced customer training needs by increasing the transfer of learning. Employing the same graphics for all components created

a recognizable product identity for the TeamLinks family.

Filing. The original design to access the remote ALL-IN-1 IOS file cabinet on the Macintosh replicated the TeamLinks for Windows information manager. The VCA process demonstrated that this design would not be competitive nor would it satisfy customer needs.

The team developed a more viable solution by visualizing the ALL-IN-1 IOS file cabinet as an extension of the Macintosh file system. Team members developed a TeamLinks file cabinet extension. Users connect to the ALL-IN-1 IOS file cabinet through the chooser window. Once a user is connected, a volume, visually represented by a file cabinet icon, appears on the user's desktop. The user double-clicks on the file cabinet volume to view the contents in a new window. ALL-IN-1 IOS drawers and folders are visually depicted as their real-world counterparts, as seen in Figure 13. Users can manipulate files in a familiar fashion.

By using the standard Macintosh user interface to manipulate drawers, folders, and documents in the ALL-IN-1 IOS file cabinet, users do not need to learn a new paradigm. This approach minimizes new learning, increases accessibility and ease of use, and adds value. This design is compatible with the future Apple Open Collaborative Environment (AOCE) and will create a better return on investment for the program team.

CONCLUSIONS

The success or failure of any product can normally be attributed to the product's initial plans and the implementation of those plans. For this project, one can evaluate the development strategy against the initial project goals and against the customer needs.

The development strategy satisfied the program's goals. The initial version of the product was delivered in less than a year of development time and with minimal resources. By-products of the development strategy allowed the team to take additional "informed" risks (seven months into the project, the team received additional responsibility for delivering the mail client), to deliver three separate products with minimal resources, and to better engage and motivate the development team through consistency of purpose.

As for the customers, they say it best in their own words:

Major government contractors: "I thoroughly enjoyed testing the product. I am definitely going to buy it --- our company is committed to TeamLinks...." "Excellent adherence to Mac Interface."

Major manufacturing companies: "Simple enough to use and it works." "I'd say yes [in response to a question regarding whether they would purchase the product], it ties in well with ALL-IN-1 and meets the needs."

Major pharmaceutical companies: "Logical enough to use without the need to read documentation." "We're very excited and encouraged by these changes. Looks like a Winner!!!!" One customer stated publicly in ComputerWorld that TeamLinks/DEC MAILworks is their standard.

Selected government agencies: "Really like mail; like the graphic UI, color, bit buttons, the file cabinet...." "Easy to use." "I love this! Our whole branch will want this." "It is exactly what I've imagined and desired for months." "They [customer's users] are going crazy over it. They love it!"

ACKNOWLEDGMENTS

Many people were involved in the development of TeamLinks for Macintosh from its inception to its shipment. The authors would like to acknowledge the following contributors: Dave Brown, Dave Burns, Gary Floyd, George Gates, Sabrina Prentiss, Janna Rhodes, Charles Robbins, David Stutson, John Wise, Nam Hoang, and Eunice Zachry (account and support managers); Jennifer Dutton, Nina Eppes, Peter Laquerre, Terry Sherlock, Ricky Marks, Barb Mathers (documentation); Paul Clark, Debbie Christopher, Beth Doucette, Jim Emmond, Mark Grinnell, Steve Hain, Dean Jahns, John Lanoue, Bruce Miller, Stan Neumann, John Quimby, Tom Rogers, Larry Tyler, and Steve Zuckerman (engineering); Peggy Doucet, Mike Pfeiffer, and Beverly Schultz (management); Robert Lehmenkuler, Steve Fink, and Steve Martin (marketing); Phil Gabree and Meg Lustig (product management); Keith Brown, Tina Boisvert, Rick Palmer, Tim Sagear, and Tony Troppito (quality assurance engineering); and Peter Mierswa, for leading the team to develop customer-focused products.

Special thanks to our customers, without whose involvement none of this would have been possible.

REFERENCES

1. D. Ziemer and P. Maycock, "A Framework for Strategic Analysis," Long Range Planning, vol. 6, no. 2 (1973): 6-17.

2. P. Naur and B. Randall, eds., "Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee," North Atlantic Treaty Organization, 1969.

3. J. Womack, D. Jones, and D. Roos, The Machine That Changed the World, ISBN 0-89256-320-8 (New York: Rawson Associates,

1990).

4. J. Gould, S. Boies, and C. Lewis, "Making Usable, Useful, Productivity-enhancing Computer Applications," Communications of the ACM, vol. 34, no. 1 (January 1991): 75-85.

5. J. Grudin, "Systematic Sources of Suboptimal Interface Design in Large Product Development Organizations," Human Computer Interaction, vol. 6, no. 2 (1991): 147-196.

6. T. Gilb, Principles of Software Engineering Management (Reading, MA: Addison-Wesley, 1988).

7. T. Ohno, The Toyota Production System (Cambridge: Productivity Press, 1988).

8. F. Halasz and T. Moran, "Analogy Considered Harmful," Human Factors in Computer Systems Proceedings (March 1982): 383-386.

9. S. Pepper, World Hypotheses (Los Angeles: University of California Press, 1966).

10. G. Lakoff and J. Johnson, Metaphors We Live By (Chicago: University of Chicago Press, 1980).

11. Subcommittee on Investigations and Oversight, Committee on Science, Space, and Technology, Bugs in the Program: Problems in Federal Government Computer Software Development and Regulation (Washington, D.C.: Government Printing Office, September 1989).

12. J. Wilson and D. Rosenberg, "Rapid Prototyping," Handbook of Human-Computer Interaction (New York: North-Holland, 1988): 859-873.

13. J. Carroll and R. Campbell, "Artifacts as Psychological Theories," Behavior and Information Technology, vol 8. (1989): 247-256.

14. P. Ehn, Work-Oriented Design of Computer Artifacts (Stockholm: Arbetslivscentrum, 1988).

15. K. Holtzblatt and S. Jones, "Contextual Inquiry: Principles and Practice," Technical Report DEC-TR 729 (Maynard, MA: Digital Equipment Corporation, October 1990).

16. D. Wixon and S. Jones, "Usability for Fun and Profit," Human Computer Interface Design: Success Cases Emerging Methods and Real World Context (San Mateo, CA: Morgan Kaufman, Spring 1994, forthcoming).

17. J. Gilmore, Jr., "A Quantitative Comparative Analysis Technique for Benchmarking Product Functionality and Customer

Requirements," Eleventh International Conference on Decision Support Systems (Providence, RI: Institute of Management Sciences, June 1991).

TRADEMARKS

BIOGRAPHIES

Paul K. Huntwork Paul Huntwork is a consultant engineer in Digital's Software Engineering Technology Center, an organization that collaborates with development groups to adapt or invent world-class methods for use in their product development activities. He joined Digital in 1987 after leading reengineering projects in software development, verification, manufacturing, and distribution at Computervision. Paul also led proto-SEI process assessment and maturation drives in Control Data Corporation using techniques drawn from IBM's Federal Systems Division.

Douglas W. Muzzey Doug Muzzey is a software engineering manager in Workgroup Systems. He is the development manager for the TeamLinks for Macintosh and TeamRoute workflow products, and he sponsored the usability and customer partnering for the TeamLinks product family. In prior work, Doug contributed to communications and systems products and managed programs in Software Manufacturing, License Management, and Corporate Programs. Doug joined Digital in 1979. He holds a B.S.C.S. (1978) from Florida Technological University and an M.B.A. (1991) from Rivier College.

Christine M. Pietras A senior engineer in WorkGroup Systems, Chris collaborates in designing effective business solutions, incorporating field research data about customer work throughout the software development process. Her concentration is in user interface design and usability evaluation for TeamLinks products on the Macintosh and Microsoft Windows platforms. Chris joined Digital in 1985, after receiving an A.B. in mathematics from Smith College. In 1991, she earned an M.S. in industrial engineering and operations research, concentrating in human factors engineering, from the University of

Massachusetts at Amherst.

Dennis R. Wixon Dennis Wixon has worked in the area of user interface design for 20 years. He helped design the VT200 series keyboards and the DECwindows, Motif, and most recent windows interfaces. Currently a principal engineer in the Usability Expertise Center User, Dennis manages the Contextual Inquiry Program and conducts user needs analysis training. Before coming to Digital in 1981, he designed and programmed statistical analysis tools at Clark University. Dennis holds B.A., M.A., and Ph.D. degrees in psychology and has published more than 25 papers in psychology, statistics, and interface design.