

WHITE PAPER



USABILITY AS AN ERP SELECTION CRITERIA



CONTENT

PROBLEM: EVOLUTION WITH NO ROADMAP	2
SYSTEM ARCHITECTURE AND USABILITY	3
WEB-TYPE NAVIGATION FOR ERP	4
AFFORDANCE IN ERP DESIGN	6
THREE SELECTION CRITERIA	7
1. Modern SOA-based Architecture	7
2. Enterprise Application Search	8
3. Intuitive Web-Like Navigation	8



USABILITY AS AN ERP SELECTION CRITERIA

BY DAN MATTHEWS, CHIEF TECHNOLOGY OFFICER, IFS AB

In an article published by MIT's Sloan Management Review online and in its Fall 2007 print edition, consultant and author Cynthia Rettig claims that "enterprise software in large organizations has not delivered on its promise to fully integrate and intelligently control complex business processes while remaining flexible enough to adapt to changing business needs."

"Instead," the synopsis of her article continues, "ERP systems—including both software applications and the data they process—are variegated patchworks, containing 50 or more databases and hundreds of separate software programs installed over decades and interconnected by idiosyncratic, Byzantine and poorly documented customized processes."

Rettig seems pessimistic about the future of enterprise applications, and stresses the need for efficiency and simplicity.

I would tend to agree with Rettig's characterization that complexity and difficulty navigating and using enterprise applications is the main barrier that prevents enterprise resources planning (ERP) and other enterprise-wide software systems from delivering their potential benefits. Indeed, particularly in instances when ERP systems are in fact composed of disparate products that have been purchased and marketed together as an integrated suite, the resulting complexity severely compromises usability and overall serviceability. When packaged software is extensively modified for an individual user or haphazardly over time by a vendor, usability is likewise hampered. But Rettig does not address the significant advances being made in the area of ERP usability, or the movement towards more logically planned enterprise applications that can adapt to change without adding unnecessary complexity.

In this whitepaper, we will discuss how the complexity of a system like ERP can be reduced, and how the huge spectrum of functionality and information that an application encompasses can be made easier to navigate. In short, we will discuss ERP usability, with special attention to allowing you to consider usability as a criterion in your enterprise application selection process.

We will also make a case that enterprise applications should start to look, feel and function more like functionality found on the Web. Too often, enterprise software is built in the same way as narrower, more limited software programs. But as the power and scope of enterprise applications has grown to encompass a very broad spectrum



of business functionality and data, these applications have become too large, diverse and extensive for a single person to master. This means we need ways of navigating the features and information within our enterprise applications that are more similar to the way we navigate on the Internet, which similarly is a vast collection of functionality and information that no single person can fully understand.

Consider how you search for information in a typical software program. Search and query functions are typically predicated on the notion that the user knows how that data and the application are structured. Within a traditional software program, users need to be familiar with the system to express search conditions properly, and need knowledge of which module, screen or form within the system the query should be entered. This is in contrast with the way a search is conducted on the Internet, where nobody executing a search knows where the information is or how it is structured. Enterprise Applications that are built as scaled-up versions of smaller, more limited software packages may have a hard time adapting to new usability methods that harness Internet principals to tame enormous amounts of information and functionality.

PROBLEM: EVOLUTION WITH NO ROADMAP

The situation within many ERP products that were originally developed in the 1970s and early 1980s illustrates the problem with large, monolithic systems that evolve over time. Let us, for a moment, compare ERP systems with a building that has been expanded and added onto over the years. Anyone who has lived in or tried to remodel a house or structure with several additions may feel some of the same pain points that Rettig mentions in her criticism of ERP systems.

Imagine a 2-bedroom vacation cottage built near a pristine lake. It was designed as a summer home, and fulfilled that purpose very well. But then the home is sold, and the new owners want to live in the house year-round. They build an addition, and upsize the furnace accordingly, running ductwork at odd angles to service the addition. The house is sold again, and the new owners finish the second floor, which requires more ductwork and plumbing for a second bathroom. Over the decades as this work has been done, building codes and materials have changed, and there are few records of what has been done to the house, who has performed the work or what conventions were followed in designing the additions. It is no wonder that the water pressure is poor, the floors are uneven from one room to the next and some rooms never seem to get enough heat. Figuring out how to fix the problem is a complicated mess.

Old houses with numerous additions probably keep mechanical contractors very busy trouble-shooting and fixing various problems. Rettig's criticisms, and the growing popularity of middleware designed to make it easier for all of these confused, disparate systems to talk to each other, suggest that ERP users are facing similar problems.



SYSTEM ARCHITECTURE AND USABILITY

Whether you are trying to make sense of the tangled knot of functionality between your ears or a sloppy mess of functionality that you are trying to use to run a mid-sized business, it should be immediately apparent that consistent, predictable systems are easier to figure out, understand and use than inconsistent, convoluted or inconsistent systems. Imagine the difficulty faced by a management team learning a newly-implemented ERP solution, particularly when the finance module, manufacturing functionality and distribution tools all have different architectures and were originally developed by different companies prior to being acquired and lashed together into a “suite.” In these situations, there are different commands, different organizational structures and different file structures for each segment of the ERP system.

Other ERP applications are more “pure play” in their design, relying on internally developed functionality rather than consolidation of acquired products. But because the functionality within the system is so reliant on the rest of the system, layering, redundancy and complexity is the natural result as the system grows and expands through the addition of new functionality over the years.

Fortunately, newer methods of software development allow gradual evolution and change of systems without resulting in redundancy and complexity. Thanks to the advent of object-oriented software development and service oriented architecture (SOA), leading-edge enterprise applications vendors can today build their applications using a large number of small components. As the functionality and technology used by the system changes, these components can be swapped out to accommodate a gradual but rational evolutionary process. If we consider, again, our analogy of the house with many additions, we see the importance of rational and proactive planning to accommodate change. Our cabin on the lake analogy also illustrates the pitfalls of monolithic construction versus modular, flexible design. In a monolithic system, where every component part is completely dependent on the other, you can not change one part of the system without adversely affecting the rest of the system. But proper planning and granular design allow ERP systems to change without adding complexity over time.

Rational planning for change within a modern ERP product is possible because SOA-based design allows the numerous components to be loosely coupled with each other and still function as a cohesive system. Each component will be listed in an overall “directory” that really serves as the controlling “nerve center” of the application. Each component has a well-defined job to do within the application, and hides all of the complexity of how it goes about doing that job from the other components. This allows the system to change with a minimum of disruption. What that means to the end-user is an ERP system that is more logically thought-out, less redundant and easier to change and reconfigure over time without making the system more complicated or convoluted to use. And to the extent that ease of implementation, upgrade and adding of functionality are considered usability issues, a granular SOA



can be a tremendous asset in making an ERP suite easier to use.

Moreover, an SOA-based application can make it easier for an application vendor to make usability advances and improvements to the user interface. The “loose coupling” that SOA allows means that the user interface can also be changed and modified over time, once again without directly affecting the underlying business logic that underpins the applications.

What this all means is that usability depends on more than the user interface itself, but rather on the basic building blocks and architecture of an enterprise application. Systems that are built by lashing together fragmented, disparate products will automatically present usability challenges as users must in essence learn a handful of products instead of just one. Big, monolithic software programs tend towards complexity as more and more functionality is added over time. More modern applications built of smaller components united through a SOA are, in contrast, better positioned to offer a combination of simplicity and the ability to easily adapt to new usability conventions as the expectations of users change.

WEB-TYPE NAVIGATION FOR ERP

In an effort to understand how IFS could modify our products to enhance usability and increase our customers’ productivity, in late 2007 we performed a study of 250 IFS customers and 100 non-IFS customers. When asked to name the main barriers to productivity within their ERP packages, the top source of non-value-added time, the non-IFS customers said the top challenge they faced was the fact that different parts of their system worked in different ways, had different commands and required different types of interaction. The second most frequent response by the non-IFS customers was that difficulty in transferring information between parts of applications was a problem. We are proud to say that IFS customers did not complain about these two things to the same extent as other respondents, likely because we have always worked to present a consistent user experience across our applications, and to ensure that context-sensitive information follows a user from one functional area to another.

But the survey response most common to both non-IFS customers and IFS customers was that usability was hampered by difficulty in finding information and understanding how to navigate through the applications. Anyone who has implemented and used an ERP product knows that it can be a chore for any one person to quickly learn the small portion of an application they will be using on a daily basis. A certain amount of training is typically necessary to teach people how to perform their daily job functions in the new enterprise environment and how to find information and the functional tools they need to use. But today, we are faced with the additional challenge of fewer people in most organizations dealing with the same amount of, or more work, than they were a few years ago. More and more, rigid job descriptions are being abandoned in favor of flexibility and multi-tasking. This means we can expect that people will need to be able to navigate beyond the small section of ERP



functionality with which they are intimately familiar—making enhanced usability more important than ever.

We were wondering what we could do to make enterprise applications more intuitive and easier to use, so we asked people during our recent study what types of software tools they thought were most intuitive and easiest to use. A sizable 35 percent plurality said that Web-based applications were most usable. Even before this result came in, we had seen some of the handwriting on the wall, and introduced the industry's first Enterprise Application Search (EAS) tool in a service pack to the then-current version of IFS Applications, IFS Applications 7. EAS is more than an embedded, Google-like search tool. Rather, it is a whole new way for people to navigate within an enterprise application. Using EAS, a user can locate not only specific pieces of information by type, but can locate pieces of functionality necessary to undertake tasks—all without having to be trained or told where to find information or how to navigate to the correct functional screen.

But the user interface of the future will need to take intuitive navigation to whole new levels, well beyond what a search function alone can deliver. While we have always considered ease of use a top priority in our user interface design, but in recent years recognized the need to enhance usability so our applications begin to approach the simplicity people have become accustomed to with their consumer technology.

So, in designing IFS' next-generation interface, Aurora, we decided to use Web principals as our guide, and structure things to work in very much the same way that you would work on the World Wide Web. After all, there are real similarities between an ERP application and the Web. Both are very large systems of information, too large for any one person to master the information and functionality they encompass. Unfortunately, most ERP user interfaces are designed as upsized versions of traditional software tools—software tools that are limited enough in scope that one could ostensibly master them in their entirety.

Systems found on the Web, on the other hand, are designed under the assumption that all instruction necessary to use the features and functionality lies within the interface for the application in question. There is so much information and so many features on the Web that no one is expected to know where everything is or how everything is supposed to work. The distributed and diffuse nature of the Web forces system developers to simplify navigation to the point that they become intuitive to use. Furthermore, resources on the Web are designed to be accessed in standard, predictable ways—and by virtue of this can be seen as a rough analog of an SOA-based business application. Online resources like Web pages, applets, programs and other functionality can be considered services that are accessed by service consumers—specifically Web browser programs. As is the case in a modern SOA-based enterprise application, Web sites and functionality are loosely coupled with the browser, which is the Web equivalent of a user interface.

The degree of simplicity and standardization on the Web has taught people new



ways of interacting with information and functionality. In the near future, people will expect their business systems to offer similar navigational structures like forward and back navigation, search bars and visual cues to guide them to where they want to go and what they need to do. Gone are the days when we can expect ERP users to happily immerse themselves in abstruse classes just to learn the basic requirements for navigation within an application.

Because we are convinced that 21st Century ERP tools need to fully embrace Webtype navigation, we are designing Aurora to replicate the look and feel of a Web browser. Users will click on links to navigate to different places, use forward and back buttons and have access to a history of where they have been in the applications.

We also feel it will become more and more important to offer different styles of navigation to suit different individuals in various industries. One other thing our usability study and long experience working with customers have taught us is that different people in different organizations have different preferences about how they would like to work within an ERP suite. ERP is no longer a one-size-fits-all world, and vendors will need to increasingly provide more than one way to work within their applications. In some cases, vendors may need to develop very specific user interfaces for specialized, repetitive tasks—something that once again becomes more feasible when the user interface is loosely coupled to the underlying functionality. Whether it is through developing specialized interfaces, search-based navigation or other means, users will be looking for multiple ways of navigating not only the functionality but the information within the application. Applications will need to provide features like simple shortcuts and well-placed and complete link pages to related functionality. In the meantime, the traditional tree-based navigator should remain for those who want to use it.

AFFORDANCE IN ERP DESIGN

As ERP begins to increasingly mimic the Web in its look, feel and in how it makes information available to users, the disciplines of Web site usability and ERP usability will begin to converge. One concept used in Web design that will become extremely important to those designing enterprise applications is the concept of affordance. Affordance involves using basic stimulus-response thinking in the design of objects, situations or systems. By applying the concept of affordance, a designer can generate a user interface that can be intuitively figured out by the user. We see this in common objects every day. When you look at an object like a pair of pants, it should be very obvious what they are for and how to use them. The two pant legs are a rough analog of the objects they are designed to fit over, and most pants are very difficult to put on backwards due to how they are stitched and cut to accommodate other parts of the human form. New varieties of popular zip-closure freezer bags are using the concept of affordance to let people know when the zipper is completely fastened. The yellow and blue sides of the zipper combine to create the color green. Most successful consumer technology ranging from remote television controls to light switches

to the dashboards of cars use the concept of affordance to ensure that they are easy to use. When products are designed properly, no manuals or instructions are required.

In designing the way users interact with an enterprise application, application designers will have pay more attention to how people focus their attention on the screen, and how different variables draw attention to the workspace. If the active section of a screen flashes or offers interaction in some other way, it can focus attention on the correct part of the work area while not demanding as much attention from the user. Particularly when a work screen is divided into multiple parts, there must be an understandable progression across the page to make it easy to determine how to interact with the various fields and segments.

Graphical or text elements will, increasingly, need to present a visual hint or a clue as to how a user might interact with them. In some cases, a cursor changes when hovering over a screen element to show you there is something to interact with. In the design of Aurora, we are paying a lot of attention to providing these types of visual cues. For instance, to find out if there are any pieces of correspondence attached to a customer record in an older version of IFS Applications, you would have to click on a button in the record, opening a new window that would tell you if any attachments are attached to that customer record. But in Aurora, we will be working to make it immediately apparent if there are any attachments from within the customer record. An attachments button will blink for a moment and tell you that there are three attachments, presenting a clue to the user that there is something here that can be further explored.

THREE SELECTION CRITERIA

So where does this leave us when considering usability in the enterprise applications selection process? I would suggest three general criteria by which an application could be evaluated to ensure that it is either eminently usable currently or will be evolved towards greater usability in the immediate future. Ask vendors pointed questions not only how the current version of the application incorporates key usability features, but the extent to which they have a detailed roadmap for moving the application towards greater usability in the future.

1. Modern SOA-based Architecture

Without a flexible, well-thought-out and modern design, an enterprise application will not only be harder and more complicated to use, but will be harder for a vendor to evolve towards greater usability in the future.

Pay attention to whether the look and feel and applicable system commands are the same as you progress from finance to manufacturing to distribution or service management modules of an application. Also watch for redundant and seemingly repetitive screens or commands that can indicate that the business logic running beneath the application has been evolved haphazardly over time.



2. Enterprise Application Search

At this point in their history, enterprise applications have been complicated and large enough that no single person can be expected to understand their entire depth and breadth of functionality. That means that in most cases, using an application will feel very similar to the Web before the advent of search tools like Google—you can not find anything unless, paradoxically, you already know where it is.

While some application vendors also market separate enterprise search products that can be bolted onto the outside of an application, search tools that are integrated into the fabric of an application can present much more intuitive and powerful search options, offering in essence a whole new way to navigate through the functionality of an application (contact IFS for Dan Matthews white paper on enterprise application search).

3. Intuitive Web-Like Navigation

Rather than requiring users to learn and remember how to navigate through an application, the modern, usability-optimized application must present and make apparent options that are available and guide users through the tasks they need to complete. In the various application screens, look for clearly-labeled links to other parts of the software or functionality, links that relate to the information you are working with to other tasks that can be performed on the information.

In business software, you will see more of that style of navigating. Instead of the user needing to know which certain, proscribed steps to perform specific functions, the application will need to present a set of links to those specific functions that relate to the information in question.

Look at the basic feel of the application. Does it feel similar to the way you interact with information on the Web? Or does it feel similar to the old character-based systems, where you have to key in the order header, then go to a different screen to enter the order address, and otherwise work in a non-intuitive way?

Dan Matthews is the Chief Technology Officer of IFS R&D. He has been with IFS since 1996, working with software architecture and IFS technology platform. Dan holds a MSc. in computer science and software engineering from Linköping University, Sweden.

ABOUT IFS

IFS is a public company (OMX STO: IFS) founded in 1983 that develops, supplies, and implements IFS Applications™, a component-based extended ERP suite built on SOA technology. IFS focuses on agile businesses where any of four core processes are strategic: service & asset management, manufacturing, supply chain and projects. The company has 2,000 customers and is present in more than 50 countries with 2,700 employees in total. Net revenue in 2009 was SKr 2.6 billion.

More details can be found at www.IFSWORLD.com.
For further information, e-mail to info@ifsworld.com

AMERICAS	+1 888 437 4968
ARGENTINA, BRAZIL, CANADA, MEXICO, UNITED STATES	
ASIA PACIFIC	+65 63 33 33 00
AUSTRALIA, INDONESIA, JAPAN, MALAYSIA, NEW ZEALAND, PHILIPPINES, PR CHINA, SINGAPORE, THAILAND	
EUROPE EAST AND CENTRAL ASIA	+48 22 577 45 00
BALKANS, CZECH REPUBLIC, GEORGIA, HUNGARY, ISRAEL, KAZAKHSTAN, POLAND, RUSSIA AND CIS, SLOVAKIA, TURKEY, UKRAINE	
EUROPE CENTRAL	+49 9131 77 340
AUSTRIA, BELGIUM, GERMANY, ITALY, NETHERLANDS, SWITZERLAND	
EUROPE WEST	+44 1494 428 900
FRANCE, PORTUGAL, SPAIN, UNITED KINGDOM	
MIDDLE EAST AND AFRICA	+9714 390 0888
INDIA, SOUTH AFRICA, SRI LANKA, UNITED ARAB EMIRATES	
NORDIC	+46 13 460 4000
DENMARK, NORWAY, SWEDEN	
FINLAND AND THE BALTIC AREA	+358 102 17 9300
ESTONIA, FINLAND, LATVIA, LITHUANIA	

www.IFSWORLD.com

THIS DOCUMENT MAY CONTAIN STATEMENTS OF POSSIBLE FUTURE FUNCTIONALITY FOR IFS' SOFTWARE PRODUCTS AND TECHNOLOGY. SUCH STATEMENTS OF FUTURE FUNCTIONALITY ARE FOR INFORMATION PURPOSES ONLY AND SHOULD NOT BE INTERPRETED AS ANY COMMITMENT OR REPRESENTATION. IFS AND ALL IFS PRODUCT NAMES ARE TRADEMARKS OF IFS. THE NAMES OF ACTUAL COMPANIES AND PRODUCTS MENTIONED HEREIN MAY BE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

IFS AB ©2010