Understanding Face Value - Giving Users What They Want… And Still Meeting Their Needs

Kim Foster, CoreTech Consulting Group, Inc., King of Prussia, PA

ABSTRACT

The purpose of this paper is to provide a general overview of approaches to solving user issues with graphical interfaces (otherwise known as GUI design). This paper is intended for development or business managers responsible for providing deliverables, training, and ongoing support to the user community. Business intelligence tools will be identified and defined. Techniques on how to manage the development process are also provided.

INTRODUCTION

Outlined in this paper are not only ideas for providing successful, tangible deliverables but suggestions on how to improve the intangible services you are providing as well. These services are just as important as the product itself. You may be able to give the user exactly what they want but they may not like the road they had to travel to get there. Will they come back to you the next time around? Or, will they find someone else who may take a little longer to meet their needs but is much easier to work with?

Think back to the last few times you had a major appliance or automobile repaired. What type of individual or company did you choose? What happens when you have done everything possible yet you have an unreasonable client? How do you keep him happy without compromising your application's integrity or your sanity in the process?

If you have ever done development work, you have probably experienced a nightmare user. You know the one. He gives you a broad, generic scope of requirements and provides little feedback or interest along the way, then foams at the mouth when you provide him with, what he considers, a less than adequate picture of what he wanted.

Often, you feel as if users expect you to read minds. They accuse *you* of not communicating effectively. This usually happens after a team of very talented developers puts in hundreds of hours of work and a high level of creative effort. You bent over backwards to ensure that deadlines and budgets were met, quality assurance completed, and every task originally outlined was checked and double checked. But, they still refuse to use what has been built and a large amount of time and money could potentially go down the drain. Before you begin beating each individual user senseless with a mouse, take a deep breath.

THE CUSTOMER

There are some basic guidelines to customer service you may want to consider before responding to the situation described here. You have heard it a million times, but let's review the golden rule in customer service: ***THE CUSTOMER IS ALWAYS RIGHT***. Even worse, he knows it. You do not always have to agree with what he wants. In fact, you may be able to educate a user on more effective ways of doing things. Who knows, you may even be able to change a mind or two along the way. But, don't get too bent out of shape if it doesn't happen. Try not to put yourself in a position where there is a stalemate. You will lose hands down every time. This could label you or your organization as difficult to work with, lose potential projects and destroy relationships. The most damaging situations are those where the user, client, or customer quietly disappears. Without feedback, you cannot improve processes or development techniques. You may then set yourself up to make the same mistakes over again unknowingly with worse repercussions.

There are many reasons for these types of conflicts and you may find that you are one of them. If the user feels that you did not listen to what he was asking for, he is going to let you know. If you make him feel like an idiot or make him look like he doesn't know what he is doing, he may be a little less eager to share his thoughts the next time around. Alan Cooper said it best in *About Face The Essentials of User Interface Design*:

Don't make the user look stupid

Oren Harari states the consequences of such an assumption clearly in his article "A Tale of a Fence, or the Death of Commodities." He wrote that, 'In the new order of things, what will count for customers is the total experience they have with the vendor.' He points out several key questions that customers, or in the case of interface development, users will raise. I have added a few others:

- How easy is it to work with this particular vendor/consultant/person - before, during, and after a project?
- Do they really understand what I, as a user, am looking for?
- Do my primary contacts seem aloof or condescending? Or do they treat me and my staff with respect and take the time to explain things without throwing around a bunch of technical jargon?
- How responsive or concerned are they about my business need or particular problem?
- How willing are they to adjust to my sometimes unreasonable, unexpected, or fickle demands?
- How effective are they at coming up with a creative, easy-to-use solution to my problem?
- Did they do more or less than I expected?
- Overall, how satisfied am I with the experience of working with them? Would I do it again?
- What would I say if someone who wanted to use their services asked me for my opinion?

Sometimes it is difficult to step out of the role of developer and into that of a user. It is completely appropriate to pay attention to how technology and design come together to build faster, slicker, more powerful front-end applications. However, you have probably had few experiences where an application ended up exactly like the example in your coder's manual. No development is based strictly on tools or design techniques. You may use great tools and build fantastically effective techniques, but they didn't satiate the business need. Try to concentrate more time and effort on the desired end result of the work that you've been tasked to accomplish. Let the development team expend the necessary energy on the path that needs to be taken to get there.

Make sure you take the time to sit down with users often to find out what their goals are. Ask questions like: what are you trying to accomplish? What are you doing currently & why? What is lacking in processes you have in place now?
Effective design is one of the most critical mantras for developing PC based applications, screens, forms, reports, or analysis tools. This means striking a balance between flexibility, ease of use, reusability, and progressive design. You don't want the user to become bored with the application. You also don't want them to have to page through 20 screens and 10 different drill down paths to get to the information they want to see. There are different tools and techniques that can help you better meet user requirements based on areas of concentration. Make sure you use what best fits the business need.

THE TOOLS

You may find it impossible to provide one solution that meets every user requirement. However, the best approach is to develop or select a core set of tools that allows users to perform multiple functions and offers many options for data visualization and timely access to accurate information. These types of tools are categorized as **business intelligence tools**. They give the user the ability to turn data into information using visual techniques.

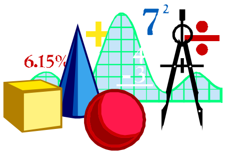
Once you determine user requirements, then figure out what types of tools provide the most effective solutions. There are many different categories of business intelligence tools based on different technologies. Let us take a look at four major areas:

Business Intelligence

Query & Reporting Tools

These tools can be used to perform basic spreadsheet type functions like frequency analysis, summation reports, and graphical capabilities like bar or pie charts. These types of tools are useful because in most cases they do not require the user to do programming or understand extraction logic for accessing data.

Query and reporting tools provide the user with the ability to access data stores and perform different levels of analysis based on queries or calls to the source data. They are used primarily to get a sense of what has occurred within a particular business area (ex. customer segment or product line).



On-line Analytical Processing (OLAP) Tools

OLAP tools pick up the ball where query and reporting tools leave off. Sometimes, it is necessary to do more complex statistical analysis or locate reasons why specific trends or frequencies occur. These tools allow the user to drill through data and make comparisons across or downward and increase the level of analysis capabilities. In many cases, OLAP works well when applied to a data mart or data warehouse because it provides functionality to access multiple, complex, and large data sources. It work well in situations where users want to:

- Move through hierarchical views of data (ex. from a customer based view of data to account or household level views)
- Do statistical queries (ex. chi-squared testing)
- Do multi-dimensional analysis (ex. see total sales over product line by month or quarter)
- Access many different data stores (ex. internal profitability analysis compared to industry averages obtained from an outside data provider)

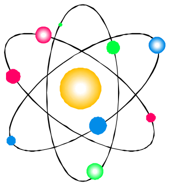
These tools are customizable and can be tailored to the specific needs of the business or user. However, you will need to remember that SQL programming or other procedural logic will likely be required in order to format or access different data sources. Naming conventions for data elements need to be understandable to the user. You may find it necessary to rename, reformat and massage data for usability and consistency sake. Object-oriented programming and development techniques are critical in supporting a front-end that is user-friendly and has reusable functionality.

Executive Information Systems (EIS)

EIS tools take OLAP functionality and apply it to a higher level of user (hence the word ‘executive’). These tools usually provide snapshots of a business line or particular area’s bottom line at a particular point in time. The visualization is concise and the analysis is highly summarized. Particular attention is paid to tool flexibility and user friendly design that allows for intuitive usage, and provides quick access to information. These tools are effective when large numbers of users need to access the same information on a regular basis (like weekly or monthly). The information should be at the user’s fingertips while still providing drill-down and across functionality. EIS tools may not be as customizable as other types of OLAP tools. However, the focus here is on the type of user accessing the information. Who is the audience? What are they looking for and how proficient are they at understanding the analysis being done? You want to save them as much time looking for information as possible and alert their attention to areas of concern or trends (loss of profits, increase in market share, etc.).

Geographical Information Systems (GIS)

GIS tools combine OLAP functionality, EIS quick access to more integrated information, and geographical analysis capabilities. They allow the user to view spatial data as it compares to business information and provides mapping capabilities to boot. If your users want to look at information over time, by products but also want to see how a particular market region or segment is performing, these tools are designed to give them just that. Users can view reports, graphs, or trend plots segmented by a particular geography after selecting a region on a map. Government agencies, financial organizations, and environmental based companies find these tools particularly useful in order to also view demographic information within a tri-state, county, or census tract area. Mapping tools can be quite large and take specialized programming techniques to support customized development.



Data Mining Tools

One of the fastest growing areas that has caught the attention of the technology industry (besides the zillions of Internet applications) is data mining. Data mining tools apply discovery techniques for analyzing data to find unknown or unexpected relationships or patterns. Where you drill-down using OLAP tools you would, conversely, drill upward through information when data mining. Ideally, most businesses make decisions based on an understanding of the relationship between their internal and external environments (i.e. business performance versus the economy) . Data mining tools help uncover critical *unknown* relationships or facts. Locating and identifying anomalies are also common characteristics. Users can see where their business may be taking a somewhat subtle turn in a not-so-positive direction and make decisions on what can be done to make adjustments through early detection. Predictive modeling, decision trees, fractal analysis (use of chaos), and neural networks are common examples where data mining is used. The function of these tools is to provide users with strategic insights on how they do business, what they need to do to change it, and how to identify potential effective approaches for moving forward. These tools are very powerful, take can a great deal of effort to design and depend heavily on statistical models for analysis. However, the results can be extremely valuable.

Business Intelligence Wrap-Up

Business intelligence tools are designed exactly as they are named. They allow users to make intelligent business decisions. At no time should they be developed to replace an existing transaction processing system. They are meant to provide visual support for business activities but not for every single transaction or field that the organization maintains. Decisions are made based on compilations of data. In order to make these decisions, a lot of information needs to be gathered, analyzed, and presented in a way that users can understand and use. Opening doors that provide hidden business trends and information is key as is incorporating the ability to apply self-learning techniques within the tools.

THE TECHNIQUES

I won't begin to outline the hundreds of programming techniques available for designing and developing applications. Suffice it to say that there are many different patterns, ideologies, methodologies, and programming languages that allow developers to transform mere data into information that blows the user away. As manager responsible for implementing such applications, your role is not to do the actual development coding, but to find out what the users are looking for and provide the guidance to the development team to meet those needs. However, what I will provide are some guidelines for using flexible, intuitive, and reusable techniques for managing design and development.

⇒ Meet with users frequently. Find out what the business need is before diving into data visualization needs. Get ongoing feedback for development and design. Let them assist in the tool evaluation process.

⇒ Don't choose tools or techniques before meeting with the users. You could end up with a great solution...to the wrong problem. In addition, they may feel forced into using something they did not participate in choosing.

⇒ Once all user needs have been identified, spend time doing your homework on the technologies out there. Don't just assume one package is better than another because one of your staff members just happens to know the software Make sure you understand the environment where the application will reside before making selections.

⇒ Go through a Request For Proposal (RFP) process for software vendors and consulting groups alike. Outline business goals, systems infrastructures, and budgetary/time constraints. Compare how different organizations, their products and services can meet your needs.

⇒ Make tool selections based on user needs, system and hardware infrastructures, costs (both tangible and intangible), the strength and history of the product line, the support available, the scalability/portability of the software, and the ratio of time & money the software company puts into research and development.

⇒ Emphasis reusability, flexibility, and limited maintenance requirements for development. If you are relying on consultants for development, remember that they may not be there in six months, so make sure that someone internally can support changes that may be necessary in the future.

⇒ How do you update data? How do you update or make changes to a particular screen? How will you distribute the application to users at different locations? Work through data flow diagrams and decide on how you want to update the application on a regular basis. Not all processes are automated. Make sure you are aware of how to manage those that are not.

⇒ Educate developers on how to be user friendly. Have them spend time with users to understand the user perspective. This will also give them a chance to sharpen their people skills, get some immediate recognition for their work, and have an opportunity to utilize feedback as a catalyst for change.

⇒ Keep a development environment. DONT MESS WITH AN APPLICATION WHILE IT"S IN PRODUCTION. Moving from a development to production environment should not be done when the users are logged on. Don't let developers start making changes to production when their might be users in the application. You could affect their session and they'll just assume the application doesn't work properly.

⇒ Make sure developers follow one set of standards, not twenty unique ways of design.

⇒ Manage vendor relationships effectively. Don't assume because you haven't heard anything in a while that everything is okay. Make sure you know how things are doing at all times. Hold weekly update calls or meetings to obtain status updates and trouble shoot.

- ⇒ Test and retest before putting anything on the user's desk. Nothing is more frustrating for both you and the user when he finds errors that developers should have caught.
- ⇒ Make it impossible for any type of error to occur. There should always be a way out, over, down, across, forward or back.
- ⇒ Provide on-line help and tutorials instead of hard copy as the first source of assistance to the user. If the user has to pick up a manual to use basic functionality of the application, usability suffers.
- ⇒ Assume nothing...if you don't know, ask.
- ⇒ Don't overdo visualization. Be concise and make patterns or trends as intuitive as possible.
- ⇒ There is always more than one way to accomplish something successfully. Have a back-up plan (or two) in case your first choice doesn't work.
- ⇒ Developers can be replaced...at a cost. They are not indispensable, but treat them as if they are.
- ⇒ Make roll-out a big production. Have count down communications and meetings, offer iterative views of work-in-progress, publish timelines via email.
- ⇒ Offer many training sessions for different levels of users, ongoing. Make them interactive, upbeat, and fun (try to limit them to a few hours).
- ⇒ Start up user group meetings. Have monthly or quarterly meetings where users can give feedback, show new ways they have used the tools, and developers can show off new features.
- ⇒ Don't assume roll-out means done. There is a lot of marketing needed. People can be hesitant to use new things. Give them time to warm up to what has been delivered and stay in their faces. Offer help lines, support visits, and on-line feedback capabilities. Users can let you know right away what they like and don't like. Then, changes can be made more efficiently and effectively. Remember: this is a process, not just a product...constantly moving, forever changing.

CONCLUSION

As you can see, there are many different factors that make delivery of an application successful. By understanding the customer, the technology, and the methodology, development managers have a better chance at surviving in this highly competitive market of delivering information. Taking the time to increase your expertise in these areas will only make you more
valuable in the long run.

REFERENCES

Cooper, Alan (1995), *About Face The Essentials of User Interface Design*, Foster City, CA: IDG Books Worldwide, Inc.,16.

Harari, Oren (1996), “A Tale of a Fence, or the Death of Commodities,” Management Review, American Management Association, November 1996, Volume 85, No. 11, Saranac Lake, NY: American Management Association Publications Division, 16-19.

USEFUL RESOURCES

Adriaans, Pieter and Zantinge, Dolf (1996*), Data Mining*, New York: Addison-Wesley Publishing Company, Inc.

Bhote, Keki, R. (1996), *Beyond Customer Satisfaction to Customer Loyalty The Key to Greater Profitability*, New York: AMA Publications Division.

Gamma, Erich, Helm, Richard, Johnson, Ralph, and Vlissides, John (1995), *Design Patterns Elements of Reusable Object-Oriented Software*, Addison-Wesley Publishing Company, Inc.

Booch, Grady (1996), *Object Solutions Managing the Object-Oriented Project*, Menlo Park, CA: Addison-Wesley Publishing Company, Inc.

SAS Institute, Inc. (1989), *SAS/AF Software: Usage and Reference, Version 6, First Edition*, Cary, NC: SAS Institutute, Inc.

SAS Institute, Inc. (1992), *Doing More with SAS/ASSIST Software, Version 6, First Eddition*, Cary, NC: SAS Institutute, Inc.

SAS Institute, Inc. (1993), *SAS/EIS Software: Reference, Version 6, First Edition*, Cary, NC: SAS Institutute, Inc.

Whitten, Neal (1995), *Managing Software Projects Formula for Success*, New York: John Wiley & Sons Inc.

AUTHOR CONTACT

Kim Foster
Application Technologies
CoreTech Consulting Group, Inc.
1040 First Avenue, Suite 400
King of Prussia, PA 19406-1336
Phone: 610-992-0800, ext. 3542
Fax: 610-337-2333
email: kfoster@coretech-group.com