DEC-TR 739
Software Usability Engineering

Linda Tse
Jay Bolgatz
Rich June

November, 1990

Digital Internal Use Only

Using Quality Function Deployment To Design the Colormix Widget

DIGITAL INTERNAL USE ONLY

Contents

Figures

1 Abstract

The design method, Quality Function Deployment(QFD), was used in designing the DECwindows' Colormix Widget Picker Model. This method allowed the project team to successfully incorporate *customer demands* into the design of the Picker Model. Through the use of QFD, the team was able to achieve these objectives:

- Verify the prototype design.
- Resolve design disputes, foster a team vision for the product.
- Develop implementation priorities that could be tied back directly to customer demands.
- Preserve customer needs that can feed into long term product direction.
- Achieve a shorter overall development cycle.
- Reduce rework during implementation, fewer QARs.
- Discover the '*delighters*' for the customer that can keep Digital ahead of the competition.

Fifty percent of the development effort was devoted to developing the prototype, and twenty five percent was devoted to doing the QFD exercise. They represented seventy five percent of the total development period. Since many of the major changes were made during the design phase when changes were less costly, only twenty five percent of the development period needed to be devoted to implementation, the implementation phase was more focused and more efficient. This report details a successful scenario of using QFD in Software Engineering. The Colormix Widget is currently being patented for its design.

2 Introduction

Quality Function Deployment(QFD) has been used for designing products such as automobiles and computer hardware. But does it work for designing software products? This paper describes how QFD was implemented successfully for designing the Picker Color Model in the Colormix Widget. It looks at issues and breakthroughs that occurred before, during, and after the QFD. This process is not the only way to implement QFD. However, an analysis of our implementation may provide insights for how QFD can be applied to software development.

First we will describe the project, and the events that led up to the decision to use QFD. Next the QFD process will be reviewed, followed by a profile of the development process. We will conclude with a retrospective look at what went well in this process and what we thought could be improved for the future.

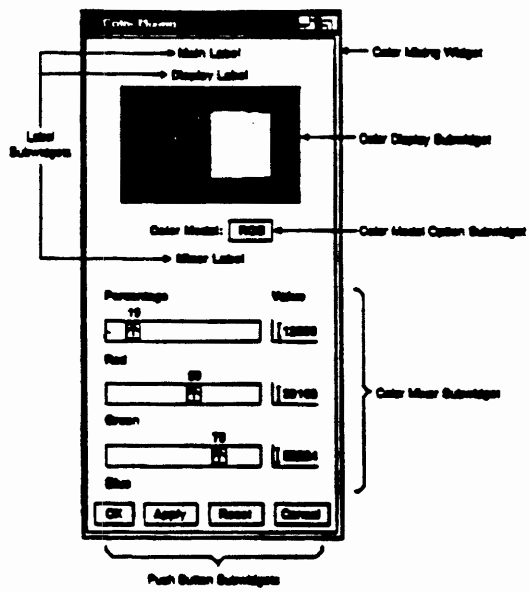
3 Problem

3.1 The XUI Colormix Widget

Originally scheduled for XUI V2, the priority of the Colormix Widget was lowered against higher priority items. However, color workstations were becoming more and more common and DECwindows applications were asking for color support. As the need for a color selection tool grew, a midnight project developed into the first version of the Colormix Widget. The following criteria were the design goals for the first version:

- intended to be general purpose - for the average engineer
- supports standard color models
- run on all DEC workstations that support DECwindows
- target hardware was a 4 plane color workstation supporting 16 colors
- simple interface for short development

Figure 1: V1 of the Colormix Widget



Using these design goals the first version supported two standard color models, the Red-Green-Blue(RGB) color model, and the Hue-Lightness-Saturation(HLS) color model. The application supported user customization in the DECwindows session manager, and could support direct color manipulation in applications such as DECchart.

3.2 Version 1 Limitations

The first version met its design goals but feedback from users indicated the Colormix Widget was limited. The two color models that it supported, RGB and HLS, were quickly becoming yesterday's technology. These color models were also not intuitive, using these models required training and assumed knowledge of color theory. Though minimally usable, the text entry fields and sliders implemented in the interface were primitive graphical interface objects and did not compare to more impressive color selection applications, like the Macintosh Color Wheel. Next to the Color Wheel, the Colormix Widget appeared dull and uninviting. Users felt color was supposed to be fun to use! While the need to improve the product was clear, the direction this improvement should take was not obvious.

3.3 Colormix Task Force

After feedback on a draft specification for the next version of the Colormix Widget, a task force was assembled to design a new color model for the Colormix Widget. The task force consisted of designers with color expertise and knowledge of DECwindows. They worked with the following design criteria that was established by the project leader:

- target to both ends of the user spectrum, novice as well as experts
- provide an intuitive interface
- target hardware is moved to an 8 plane color workstation
- stay ahead of the competition
- fun to use

The task force held four, two hour sessions between February and June. Three different specifications were drafted before a final prototype specification was accepted, but the final proposal left many open issues about the interface.

3.4 Problems During Design

By traditional software development standards the Colormix taskforce was a good start, but designing exclusively within a taskforce can be problematic.

One problem was the nature of the design team. The design team represented a variety of skills, backgrounds, and design styles. Each was a respected designer in their own right, but when put together, different approaches and strong personalities gave rise to conflicting designs, impeding progress for the team.

True consensus was almost impossible to reach. After each intensive design session, a different engineer volunteered to draft the notes of the meeting into a design specification. Each time the outcome would be that engineer's interpretation of the agreed to design. Another engineer would eventually reject the draft saying, "But this doesn't look anything like what we agreed to!" As a result, the team did not make much progress.

The design goals for the new Colormix interface were as ambiguous as the design goals for the initial interface. The goals provided Engineering's interpretation of what the the users wanted, but they did not give any insights into what the words *novice, expert, intuitive, and fun to use* meant in the context of this product. It turned out that no one on the team really knew their users. There were many potential users of the Colormix Widget and the only users that the design team could truly represent were engineers and people who knew about color theory.

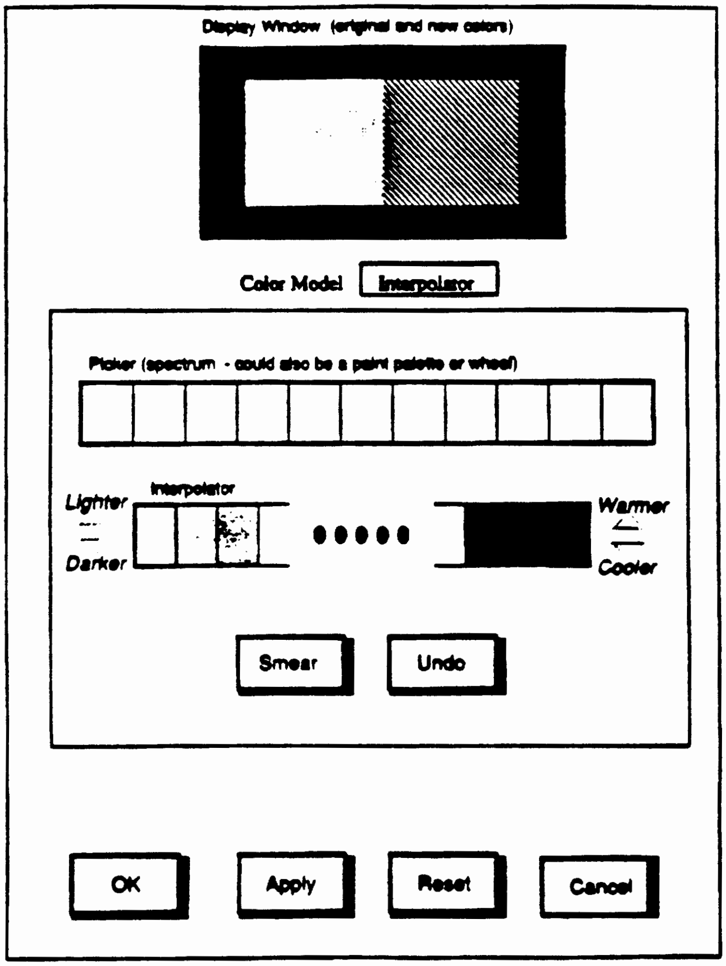
These issues are representative of what happens in most design meetings, but this process was no longer acceptable for the team. It proved to produce only limited results and was unable to generate a final design.

3.5 Verification Of Initial Design

During the design meetings, the question "But what do the customers want?" eventually arose so many times that the team decided no further decisions could be made before they received feedback from actual users of a running prototype. They agreed to commit sufficient time to soliciting customer input and to feed recommendations back into the design specification. The team hoped the following issues would be resolved in the allotted time.

- Are the lighter/darker and warmer/cooler controls useful, are they intuitive to use?
- Is the 'smear' label the right word to use for the interpolation functionality?
- Is the eye dropper metaphor useful and can people pick it up without help?
- Does the product need an undo function?
- Does the 'pick from spectrum and interpolate metaphor work for people who do not have in-depth knowledge of color?
- What colors should be allocated in the spectrum and are there enough colors to choose from?
- Can users actually achieve the color that they are looking for with this color model, do they get what they expect?
- Is this color model considered fun to use?
- How would users use the color that they have generated, do they need further support to use that color in their applications?

Figure 2: Picker Model Prototype Layout



3.6 Opportunity

Like many development projects the Colormix Widget underwent schedule changes. The development team's schedule moved up significantly when they found that there was an opportunity to include the next version of the Colormix Widget in DECwindows V3. With the tighter schedule the development team had a challenge; Could they get the information necessary to design an interface that met the customers' needs and still be able to stay within the new schedule? Their needs went beyond just gathering information from users, they needed to find a design method that would help them to prioritize these needs and help them to make the right design choices. After looking at several alternatives, they chose to try the QFD design method.

4 A Solution - QFD

4.1 History of QFD

Quality Function Deployment is a method for prioritizing, structuring, and incorporating customer needs into a product design. QFD originated in the U.S. during the Manhattan Project, but the methodology itself was not embraced by U.S. manufacturers until recently. Yoji Akao introduced QFD to the Japanese in the 1960s, and developed QFD into the systematic quality control charts that were first implemented in the Kobe Shipyards of Mitsubishi Heavy Industries in 1972.[1] In Japan, the word deployment, in the phrase QFD, means **diffusion**.[2] The Japanese use QFD to diffuse their product strategy, as defined by the customer, throughout the entire organization from staff to line, so that each person can understand the significance of his/her role in contributing to the overall product goals. For example, at Toyota, the assembly engineer understood that he should rustproof all of the bolts on the underside of the car in addition to the wheel wells, because it was extremely important to the customers to have weatherproof cars.[2]

4.2 QFD Benefits

The Colormix Widget design team chose to use QFD for the benefits that it showed in other design projects:

- It provides a way to resolve design disputes.
- It is a statistical tool for prioritizing and structuring customer needs.
- It prioritizes implementation alternatives to match the customer needs, providing inputs for the implementation specification.
- It provides engineers with metrics for decision making and for communicating strategy to management.
- It builds the team and creates a team vision.
- Both minimal design requirements and customer delighters are revealed in the analysis, they keep the company ahead of the competition.
- It produces clear design directives which helps to shorten the implementation cycle. This ultimately contributes to shorter time to market.

5 The QFD Process

The House of Quality model is made up of several components[3] (see appendix A). In the interest of time and the specific needs of the development team, we decided to use only portions of the House of Quality. We will discuss the sections of the model that were used and how the exercises were modified to meet the team's needs. The three exercises that were implemented were the Generating Customer Demands exercise, the Planning Matrix, and the Correlation Matrix. The results from each of the exercises feeds into the next exercise until a cohesive strategy is formed.

5.1 Gathering the Data

To gather data for the QFD, test drive sessions were held. Six users came to ZKO to use the running prototype on a workstation. The participants represented three cross sections of our target user population, two experts: CAD users, two intermediates: graphic designers, and two novices: administrative staff. The goal of the test drives was to see if participants could successfully create the color of their choice using the Picker Model. We designed a flexible test drive script that could be expanded by the user to meet their work criteria. We encouraged them to bring in tasks from their own work environments and try to use the prototype to accomplish their tasks.

The developer in this project was encouraged to participate in the test drives as well to observe the user's experience firsthand, and to engage in a co-design dialogue with the users. Engaging in a design dialogue with the users helped the developer to brainstorm creative implementation alternatives that can be fed into the QFD process. After each test drive, the team reviewed their observations and compared notes to verify users comments. They wanted to be as accurate about the user's words as possible since this was going to be their only data source and there wouldn't be enough time to transcribe audiotapes of each session.

5.2 Generating Customer Demands

The first exercise in the QFD asks the design team to list the top customer demands based on actual customer inputs. For this project, the inputs came from the test drives. Some of the customer's quotes were used verbatim, some of the demands were generated through an interpretation of the user's reactions while using the prototype. When the exercise was completed, there were 45 customer demands. Next the team used affinity diagramming, grouping by Post-its, to group the related demands into manageable categories and to filter out any redundant data. The team ended up with seventeen categories, with a list of related demands under each category.

This portion of the QFD is one of the most time consuming exercises in the process because of the level of interpretation and consensus that needs to occur within the team. The process was both painful and exhilarating, painful because the team was forced to scrutinize the data, and exhilarating because they came away with a clear profile of their customer base.

5.3 The Planning Matrix

Once the generation of customer demands was completed, this list was transferred over to the next part of the process, the Planning Matrix. In the Planning Matrix, the aim is to achieve a total demand weight for each customer demand, by also factoring in the considerations that are important to the company. Each factor is represented by a column on the planning matrix(see appendix B) . The team decided to give input to only two factors on the chart, they were:

- Rate of importance of the feature to the customer
- Sales points for feature improvement

The customer demands were then ranked using the totals of the columns. Other factors that could be included are:

- How do our competitors compare on each customer demand
- How does our company rank on each customer demand
- How much improvement do we want to plan for on this feature

The team decided not to use these factors for two reasons, they currently had no major color application competitors in the workstation space, and their ratio of improvement would have been flat for all customer demands because they wanted to improve all features in this version.

The first column, rate of importance of the feature to the customer, was ranked based upon how many times the customer voiced that demand, each customer counted as one vote no matter how many times they asked for that feature. For example, one vote represented a rating of 1 which means not very important to the customer. The following is the scale that the team used.

1 to 2 votes = 1 = nice to have,not very important to the customer
3 to 4 votes = 3 = moderately important to the customer
5 to 6 votes = 5 = extremely important to the customer

The second column, sales points for feature improvement, was rated based upon how many sales points the team felt Digital would receive if a solution was provided for that customer demand. The scale for rating this was very similar to the rating of customer importance.

1 to 2 votes = 1.0 = customer already expects this feature, would be dissatisfied if it was not available
3 to 4 votes = 1.3 = customer would be satisfied if they had this feature
5 to 6 votes = 1.5 = customer would be *delighted* if this feature were available

After both columns on the Planning Matrix were completed, they were multiplied together to form a total demand weight for each customer demand. This exercise helped to separate features that the customer expected, from features that would both delight the customer and keep Digital ahead of its competition.

5.4 The Correlation Matrix

The last exercise in the QFD involves brainstorming solutions for each customer demand, and lastly determining which solutions would satisfy the most demands.(see appendix C) For this project, the team decided the results would be more meaningful if engineering doability was also factored into the decision making process. Doability in this context meant 'how hard is the task to implement' and 'is the technology available for that feature today.' These considerations were very important to the engineers for input into the implementation specification.

First, the list of prioritized customer demands was transferred over to the Correlation Matrix. Next, the team brainstormed all possible solutions for each demand. Doability and resource availability were discounted during the brainstorm because the team wanted to document all creative ideas for long term strategy; things that may not be doable today, but may be doable in the future. The team came up with 40 alternative solutions to meet the customer demands.

Each alternative solution was then correlated with each customer demand using the following criteria:

- Does this solution have a negative impact on this customer demand?
- Does this solution have a positive impact on this customer demand?
- If it is a positive impact, is this solution doable?

Doability was factored in at this point to see what was possible for this version of the product. When the exercise was completed, the Correlation Matrix provided the team with a color coded chart that showed which solutions provided the most leverage for the product. Other beneficial outcomes were:

- The data could be preserved and re-used to settle any future implementation disputes.
- Although not doable today, several high leverage features were documented for the long term strategy.
- The chart gave the team a way of evaluating how many of the customer demands they would meet, if they were only able to implement a portion of the high priority items in the development timeframe.

5.5 Evaluation

A follow-up meeting was held to present the results of the QFD to the rest of the design team. The purpose of this meeting was to achieve consensus on the outcome of the analysis, for those who were not able to attend the QFD, to make some final design decisions, and to bring closure to the design process. The team was very successful in gaining consensus this time around and decisions were made in less than an hour, much unlike the prior design meetings. The project leader commented, "At first, I was skeptical that QFD was worth the extra time involved in preparing the charts. I was pleasantly surprised." The team felt that the following reasons contributed to the smooth closure:

- The statistics spoke for themselves.
- Every recommendation was traceable to a customer voiced demand, it's hard to argue what the customer has said.
- The onus of the ideas fell on the customers and no longer fell on one designer verses another.
- There was now a clear set of customer needs that could be used as a mediation device for design ideas.
- Having the list of alternative solutions in front of them, including the far out suggestions, gave the designers ideas for creative *intermediate* solutions to long term customer needs.

6 Results

The Team was able to verify the following functionality on the prototype design. The final design is shown on the next page.

- the spectrum selection/interpolation model is an effective model for specifying colors
- the warmer/cooler, lighter/darker controls were needed on the interface
- using the smear label on the interpolate button was more intuitive than using the words mix or interpolate
- the undo function is necessary in addition to the reset function
- the user interface is understandable and generally intuitive

The following high priority features were identified:

- increase the number of color choices in the spectrum
- increase the dimensions of the picker and interpolator tiles
- provide more direct manipulation alternatives for color selection
- add on-line help

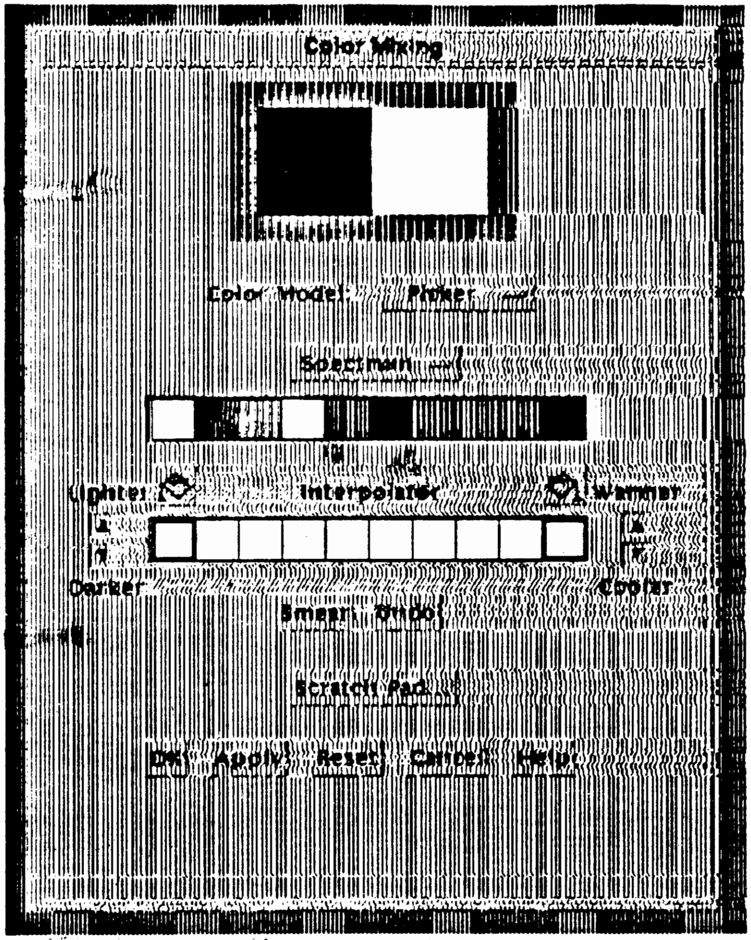
The following problem areas were highlighted by the QFD:

- the eye dropper selection method is not intuitive
- users needed a temporary holding area for holding colors they mixed along the way
- users wanted both range and reversibility in the warmer/cooler functionality

One high leverage feature was identified for future strategy:

- combine a full palette editor with the Color Picker Model. It turns out that palette editing is an extremely important feature for high endand intermediate users . Though it cannot make the V3 schedule, itwill be incorporated in the product strategy for the next version.

Figure 3: Final Picker Model Design



7 Looking At the Development Process

As shown in the following development cycle breakdown, seventy percent of the development process was devoted to the design phase(includes both prototyping and QFD process), which was very different from many projects which devote seventy percent to implementation instead. Because of the team's dedication to the design phase, most of the functionality and interface changes in this project were made during the design and prototyping stage, when changes dictated by users of the prototype were less costly to implement. Major feature changes did not have to occur during implementation when it may be too late or too costly to fix design flaws.

This resulted in fewer overall design and integration problems during implementation and a shorter implementation phase.

It is worth noting that the team spent as many person-hours in the initial design meetings as they did doing the QFD exercise, but were not able to arrive at an agreed upon design during the design meetings. If QFD had been implemented earlier in the design phase, the team might have been able to save more time and shorten the development cycle even further.

The following is a profile of the development process for the Colormix Widget V2. It is broken down into three segments, time spent on the prototyping phase, time spent on the QFD process, and time spent in the implementation phase. The totals are represented in person hours, and the time reported is elapse time.

7.1 Prototyping Phase

- Pre-prototype design meetings (4 engineers, 4 meetings @ 2 hours each):
  total of 32 person-hours
- Writing of 3 design specs (2 engineers):
  total of 40 person-hours
- prototype coding: 40 person-hours

Total prototyping phase = 112 person-hours

7.2 QFD Process

- Design of test script for test drives (1 engineer):
  total of 2 person-hours
- Contacting customers and setting up sessions (1 engineer):
  total of 8 person-hours
- Test drives (2 engineers * 6 sessions @ 1 hour each):
  total of 12 person-hours
- QFD process (2 engineers @ 16 hours each):
  total of 32 person-hours
- Meeting with project leader (3 engineers @ 1 hour each):
  total of 3 person-hours
- Final design closure meeting (5 engineers @1 hour each):
  total of 5 person-hours

Total QFD process: 52 person-hours

7.3 Implementation

- coding changes: 20 person-hours
- testing/debugging: 40 person-hours

Total implementation phase: 60 person-hours

8 What Could Have Been Improved

Looking back at the design process, the team saw several areas that could be improved for the future.

Rich, who participated in the test drives, felt that our sample customer base was a little too small to draw sweeping conclusions from. He felt we had ignored a large group of potential Colormix Widget users, the applications developers, who are workstation savvy and also need to use color, but aren't sophisticated color users. Rich commented that "Ultimately, Jay and I had to represent the needs of this group ourselves, which was difficult to do in an unbiased way. We both had certain features that we personally like, and 'knew' we should keep in no matter what the test drives revealed." Our sample base of users could have been larger if we were able to allot more time to the data gathering period. We might want to schedule more time into the development plan for user input next time.

During the QFD analysis, a large amount of time was spent creating the charts by hand. The end result was large charts that spanned the wall which were nice to look at and could be easily read by a large group, but these charts were tedious to create manually. Drawing the grids for the charts alone took several hours. The team often wished there was a statistical software package available on the workstation or PC that could support large charts and allow data to be plugged right into formulas in the charts. This would have cut the QFD process down by half a day.

The team would have preferred to be able to move the prototype software directly to the customers sites and have the customers use the prototype in their own work environments, but the Motif library was not installed at every customer site and an 8 plane color workstation could not be guaranteed at every site. Portability was a problem for us during the prototype testing phase.

9 What Went Well

The team found QFD to be an extremely flexible methodology. Columns could be added or taken away on every matrix based on the needs of the project. For instance, the team removed the comparison to competitors column in the Planning Matrix because at the time, they had no significant competition in the workstation space. The exclusion of that column did not prevent the team from producing meaningful results. QFD was also flexible enough to accommodate additions to the model without compromising the integrity of the results. The team was able to add doability as a criteria to measure against the alternative solutions in the Correlation Matrix.

Because of the insights that the results brought to the development team, The project leader felt that they were able to achieve a version 2 product within a version 1 timeframe. He also believes the product will have fewer QARs and there will be significantly less rework needed before the product ships to external customers. The team felt that QFD helped them to shorten the overall development process. They were happy to meet time to market for the product.

The flexibility of the development team also contributed to the overall success of the process. Schedule changes did not prevent the team from finding creative ways to complete the QFD and implementation in time for internal field test. Also, without the support of the team's managers, individuals would not have been able to reschedule prior commitments and find the resources that it needed to complete the project in such a short timeframe.

10 Conclusions

Having used QFD successfully in the design and implementation of the Colormix Widget picker model, we conclude that the QFD design method can be beneficial in software development.

A design method such as QFD that can prioritize customers needs and can communicate design objectives clearly, can contribute to the development of a better product in a shorter time frame.

For more information on the QFD process, contact Linda Tse in Software Usability Engineering, 4GL::TSE.

For more information on the impact of QFD on the Colormix Widget project, contact Jay Bolgatz in the Extended Toolkit group, RTL::BOLGATZ.

Acknowledgments

We would like to acknowledge the following people for their contributions to the design of the Colormix Widget V2:

Eric Rehm(CADSE)	- design team
Eliot Tarlin(SUE)	- design team
Rich June(Toolkit)	- coding, test drives, QFD
Jay Bolgatz(Toolkit)	- project leader, design team. QFD
Linda Tse(SUE)	- test drives, QFD, design team

We would like to acknowledge the following managers for their support to the project:

Anne Duncan(SUE)	- approved SUE resources
Steve Grass(Toolkit)	- approved Toolkit resources

We would also like to acknowledge members of the SUE group for their helpful reviews of this report.

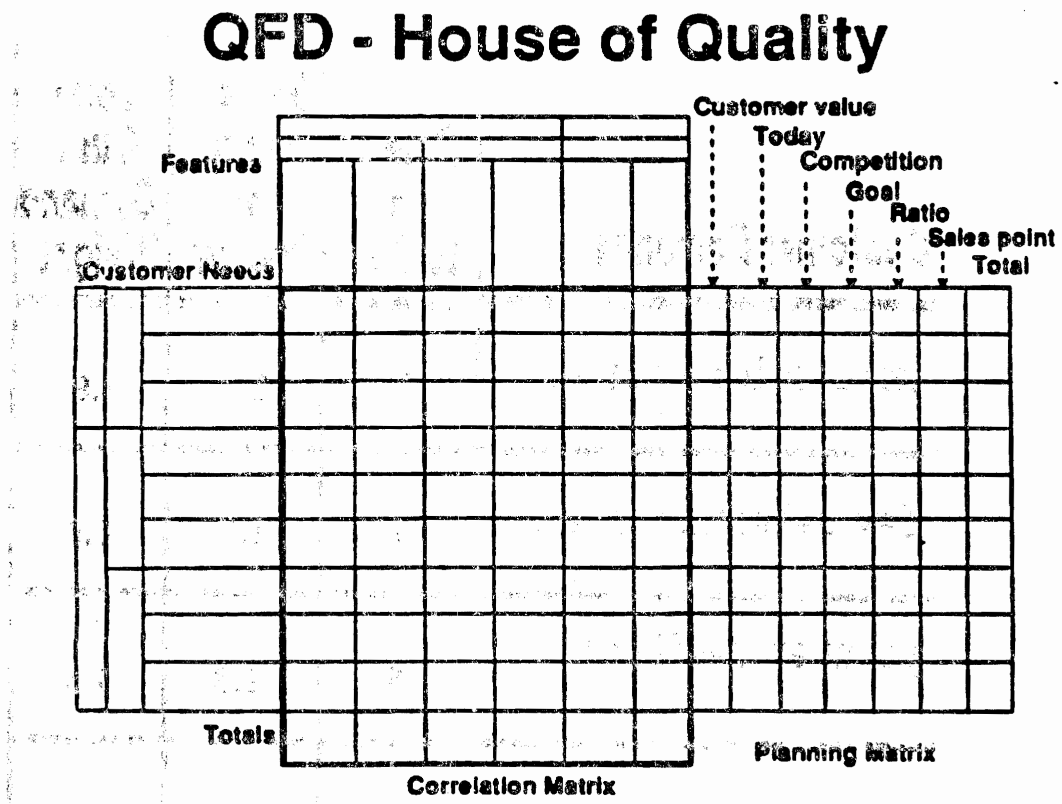
References:

1. King, Bob. Better Designs In Half the Time. Methuen, MA, 1989. Available from GOAL/QPC, Methuen, MA.

2. Course, Quality Function Deployment. Engineering Quality Technology Group Digital Equipment Corp.

3. Course, Contextual Inquiry. Software Usability Engineering Digital Equipment Corp.

Appendix A: House of quality



Appendix B: Example of Planning Matrix Used by the Team

Customer Demands	Rate of Imp to Cust	Sales points for Improv	Total Cust Demand Weight
more color choices	3	1.3	3.9
few selection steps	5	1.5	6.5
pointer feedback	5	1.3	7.5

Appendix C: Example of Correlation Matrix Used By the Team

Prioritized Customer Demands	Alternative Solutions		
fewer selection steps			
more color choices		Impact / Usability	
pointer feedback			
		Scores	