N88-17231

USER ENGINEERING: A NEW LOOK AT SYSTEM ENGINEERING

Larry L. McLaughlin
TRW Defense Systems
One Space Park
Bldg. 02 Room 2779
Redondo Beach, CA 90278

ABSTRACT

The dawn of the missile and space era gave birth to System Engineering, and the evolution of human computer systems has spawned a new concept called User Engineering.

Because users are diverse, and their requirements are always subjective, and initially vague, unknown or incompletely defined, existing methodologies are insufficient to identify user issues which then cause major disruptions throughout the requirements, design and development of the system. Human thinking and decision making in critical environments such as space stations or space defense command and control centers demand new engineering approaches to tackle user complexity and reduce system development risk.

User Engineering is a new System Engineering perspective responsible for defining and maintaining the user's view of the system. It's elements are a process to guide the project and customer, a multidisciplinary team including hard and soft sciences, rapid prototyping tools to build user interfaces quickly and modify them frequently at low cost, and a prototyping center for involving users and designers in an iterative way. The main consideration is reducing the risk that the end user will not or can not effectively use the system.

The process begins with user analysis to produce cognitive and work style models, and task analysis to produce user work functions and scenarios. These become major drivers of the human computer interface design which is presented and reviewed as an interactive prototype by users. Feedback is rapid and productive, and user effectiveness can be measured and observed before the system is built and fielded. Requirements are derived via the prototype and baselined early to serve as an input to the architecture and software design.

INTRODUCTION

This paper details a methodology that has been used successfully to engineer complex human computer systems. It is a "process" technology, relating to how the government and corporations can go about conceptualizing, defining, and building systems. The motivation for this new process is described in terms of problems with existing approaches and the pressing need for a dramatic change in the nature and goal for these systems

The new process, called "User Engineering", is defined and related to the other perspectives of system engineering. Then it is described in terms of four primary elements: methodology, tools, team, and laboratory. Examples of systems where this technology has successfully been applied and the benefits are summarized.

PROBLEM

User interactive systems differ markedly from systems the government and aerospace companies have built in the past. Those systems were driven by critical simulation and/or data processing requirements, but required limited human interaction. With user interactive systems, the primary consideration is providing knowledge integration and fast problem solving, using powerful workstations connected to information sources supported by decision aids (Shneiderman, 1987).

Requirements for these systems are initially at best subjective and are often vague, unknown, and incompletely defined. Users do not know what they want until they see it, and frequently can neither describe how they do their work (Rasmussen, 1986), or communicate it to even the most able interviewers. Additionally, these requirements do not meet physical laws and can not be proven on paper in advance.

Classical approaches used successfully in the past on systems like the ICBM program or site defense, depend on functional partitioning and depend crucially on the study of interfaces. For human computer systems, these fail because of the need for symbiotic interaction between a primary subsystem, the human, and the computer. (Winograd and Flores, 1986)

The term "human computer interactive" that is now widely used to describe these systems, masks the incredible complexity and difficulties associated with trying to combine the powers of the human brain and computers. Only recently has the aerospace industry begun to uncover the reasons why efforts to build such systems have had so many problems.

Analysis of costs for these systems reveals that for every dollar spent in development, approximately two and one half dollars are spent in operations and maintenance. Of the software development costs, thirty percent (30%) is breakage within the waterfall. It is now known that these costs are primarily traceable to failure to have complete, clear and consistent user/system requirements. More startling, of the software maintenance costs, forty percent (40%) is user enhancements. In other words, these costs are associated with making the system do what users suddenly discover they really needed in the first instance. In the worst case, over one third of the life cycle costs for these systems could be considered wasted if better requirements could be established early (Boehm, 1981).

GOAL

The challenge for each project then becomes how to understand and provide customized support for diverse users, to translate this knowledge into user/system requirements that achieve the best interface for their dialog with the system, and to produce the best technical design solution to accomplish needed system support. The ultimate system engineering goals, thus are to validate these requirements with users early enough to affect system design, and to architect systems to meet user needs, not force fit users to system architectures.

APPROACH

Traditionally, system engineering is understood to have four perspectives (Figure 1.). Mission engineering produces the operational concept for the system and concentrates on "who, what, why, where and how." Requirements engineering defines requirements in terms of "the system shall..." statements for contractual and testing purposes. Design engineering produces the physical view of the system, its architecture and functional partitioning, and simulations of performance. Implementation engineering makes decisions about order of build, languages, test beds, and testing. To strengthen the human analysis, a new perspective, we coined "User Engineering", is added to complement the others. It is responsible for defining, validating, and maintaining the user view of the system. User Engineering is supported by rapid prototyping to identify and resolve issues early, and identify requirements and drivers for the architecture design. The process is risk driven, as opposed to document driven, and thus initially avoids paper specifications which are costly to produce, read, and understand, and which often amount to little more than speculation about the utility and usability of the final system.

METHODOLOGY

The User Engineering process comprises seven steps that provide a framework (Figure 2.) to guide the engineering tradeoffs which lead to a baselined prototype and subsequent set of system and software requirements. The central and most important phases produce analyses of the user, the tasks the system will support, and how the users will interface with the system to perform work. For purposes of description, the phases are top down and hierarchical, but in practice are iterative, recursive, and flexible as needed. Each of the phases has a specific goal and product, and produces prototypes from the project's inception onward. The prototypes provide focus for continual interaction with users for exercising working models of the system which look and feel like the proposed system.

Specifically, the process begins with the definition of an operational concept, and the gathering of user and system data about how the current system (if it exists) works and could be improved. A user model is formed from results of interviews, observations, and cognitive, workstyle and personality measures. Then task analysis is then done from the user point of view, to complement the partitioning of internal system functions. Key to the task analysis is the preparation of scenarios with users to reflect more completely how their work is done. Candidate user interfaces are defined and the scenarios are prototyped to involve users hands-on. Software algorithms for critical functions can be included in the prototype and can operate on real or simulated data. User and designer timely feedback identifies issues and provides the information to do effective tradeoffs and make clear decisions.

As the prototype matures, it begins to reflect the final system at the level of detailed design. It looks, feels, and behaves like the real system, including simulating response times exactly. It remains as a tool throughout the development to interpret the system, and respond to the inevitably changing user requirements. Additionally, it provides a mechanism for measuring the performance of the user on the system in advance, and for beginning the development of training concepts. The user interface is baselined with attendant requirements documentation at a User Design Review, a new review proposed to occur at or before the System Design Review.

TOOLS

Key to successful User Engineering is the ability to build prototypes rapidly and be able to change them frequently at low cost. No customer will support a lengthy expensive process no matter what reduction in risk is promised. Prototyping tools must facilitate rapid construction (initial prototypes within 14 to 30 days), have real time interaction, be transportable to the user, and provide for user performance monitoring. Referred to as "user interface management systems", these tools promote rapid screen generation and provide mechanisms to sequence the screens and couple them to special or library applications programs.

Additionally, the tools should (Figure 3.) allow User Engineers to select from any number of terminals, input/output devices, different user dialogues, and interchange these as required. Appropriate application programs and their associated data bases (e.g. map generation and display, or image processing) inherent to user interactive systems are included to facilitate effective modeling of systems.

TEAM

No single individual has the breadth of both "hard" and "soft" science expertise needed to analyze and design these complex human computer systems. Furthermore, a highly experiential process such as this requires a team approach (Figure 4.). Training in group process techniques is essential to effective utilization of both individual and team resources. Depending on the problem at hand, disciplines including organizational development, sociology, psychology, macroergonomics, human factors, man machine interface design, prototyping, and general system engineering must be represented. In addition, it is mandatory that the customers' end users and experts in the subject of the system be participants.

LABORATORY

It is also clear that computer centers, with noisy equipment and harsh lighting, are inappropriate settings for developing and gathering feedback about user interaction on the prototype. A laboratory, referred to as the "System Prototyping Center", must be established as a special environment (Figure 5.) keyed to the users experiencing of the proposed system. It has individual work areas into which proposed workstations or consoles are brought in and which are decorated to resemble the work place. The workstations house or are connected to prototyping and measurement tools. Another area in the center contains monitors, large screen projectors, video cameras and recorders, and storyboards for capturing and sharing ideas.

APPLICATIONS OF USER ENGINEERING

This technology is being successfully applied both in research trials and on contract at TRW for government customers (Figure 6.). These are all user based systems tending toward the most complex end of the spectrum of cognition, tasks, and MMI.

Ongoing analysis of these applications of User Engineering clearly indicates that the risks are dramatically reduced for requirements and design, and that users are more satisfied throughout the development and deployment of the system. Much less paper is prepared at the outset, as the prototype communicates the system concept quickly and effectively. User feedback is rapid, and meaningful, and issues are surfaced before the system is built and fielded.

Designers better understand what kind of architectures they should use, and the resulting systems are easier to extend and maintain. User productivity can be measured in advance, and training concepts can be explored on the prototype prior to system delivery. This approach can nicely complement more formal development processes which follow the User Design Review and are based on the specification derived from the prototype.

CONCLUSION

In summary, the User Engineering technology is the basis for a new way of developing heavily user interactive systems. Set within System Engineering, it focuses on the human stimulus response and the redefinition of the workplace. Prototyping is the key tool for rapid conceptualizing and communicating across customers, developers and users. This kind of experimenting provides the best approach to defining/validating requirements and getting hands on experience with the system before committing to large costly developments.

The incorporation of this technology into current practice is straightforward but requires modification of traditional modalities. Of utmost importance is obtaining active participation of end users, something not always deemed desirable by customers. Additionally, new procurement procedures are needed to insure these activities are conducted early. Formal documentation deliverables in contracts must initially yield to delivery of prototypes, and the analysis surrounding their development and trial use. These kinds of risk reduction techniques will in the long run not only improve user and system effectiveness, but will lower development costs as well.

ACKNOWLEDGMENTS

I would like to acknowledge Lance Valt for his full time support to the research and development of this technology and the contribution of ideas and words found in this paper.

REFERENCES

1. Shneiderman, B., "Designing the User Interface", Addison-Wesley Publishing Company, 1987

2. Rasmussen, J., "Information Processing and Human-Machine Interaction," North-Holland, 1986

3. Winograd, T., and Flores, F., "Understanding Computers and Cognition," Ablex Publishing Corporation, Norwood, New Jersey, 1986

4. Boehm, B., "Software Engineering Economics," Prentice-Hall, 1981

5. Norman, D. and Draper, S., "User Centered System Design," Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey, 1986

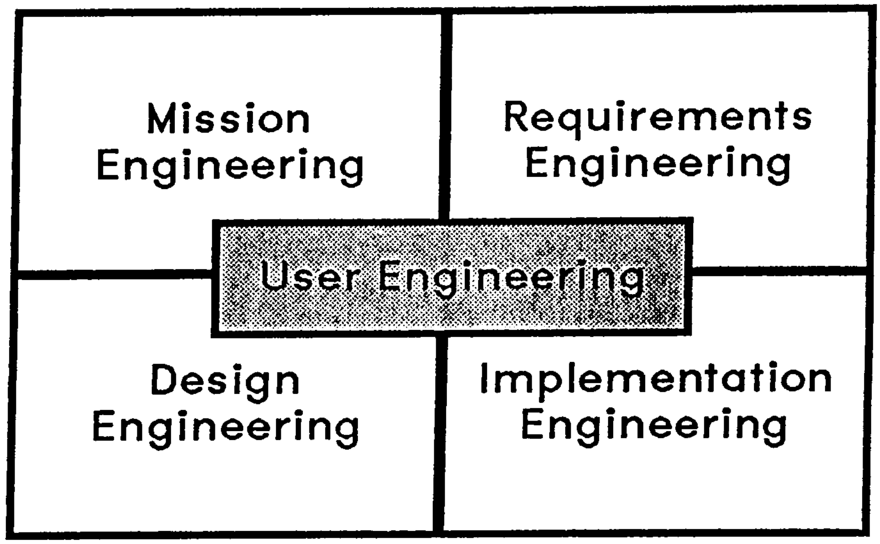

Figure 1. User Engineering Integrated Within System Engineering

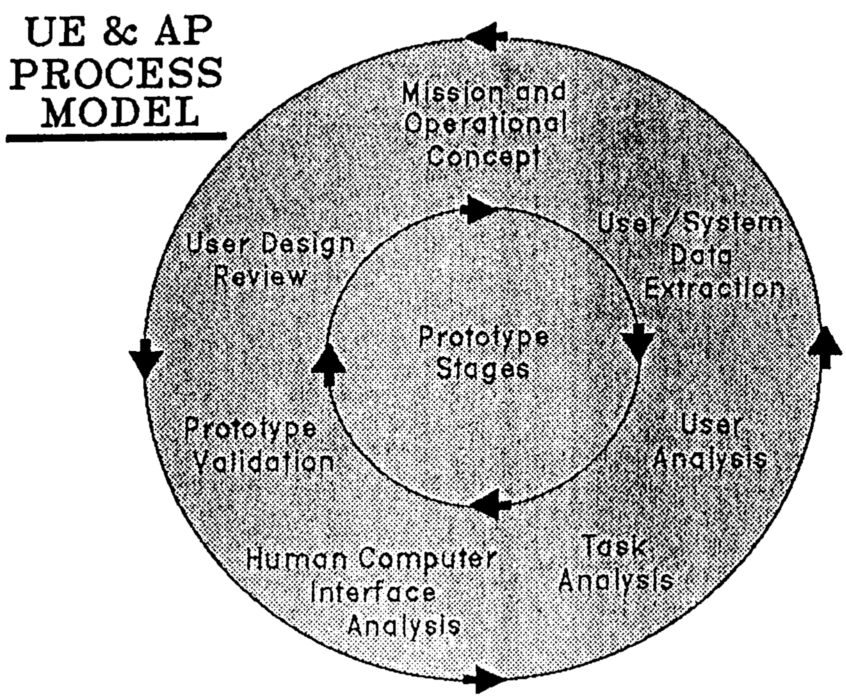

Figure 2. User Engineering Methodology

SOFTWARE PROTOTYPING TOOLS

Ruby GIDS FLAIR — USER INTERFACE MANAGEMENT SYSTEMS

Tool Capabilities	Applications Library
Generate Displays	Image Processing
Develop Dialogues	Signal Processing
I/F to Applications Libraries	Map Generation/Display
Provide Workstation Drivers	Knowledge Engineering
Support I/O Devices	Menu Processing
Interpret Script for Linkage	Time Simulation
Assist Application Development	Graphs/Plots
Monitor User Interaction	Geographic Display

Figure 3. User Interface Prototyping Tools

INTERDISCIPLINARY USER ENGINEERING TEAM

SKILLS:

- User Architect (leader)
- System Users
- Experts (sys specific)
- System Engineering

- User Engineering
  - User/Task Analysis
  - Prototyping

- MMI Design
- Psychology
- Human Factors
- Sociology
- Training
- Organizational Devel.

TYPES:

- Creative
- Intuitive
- Emergent
- Extroverted
- Sensitive to People

- Detailed
- Systematic
- Schedule Driven
- Introverted
- Sensitive to Technology

- Highly Motivated
- Tolerate Ambiguity
- Small in Number

Figure 4. User Engineering Team

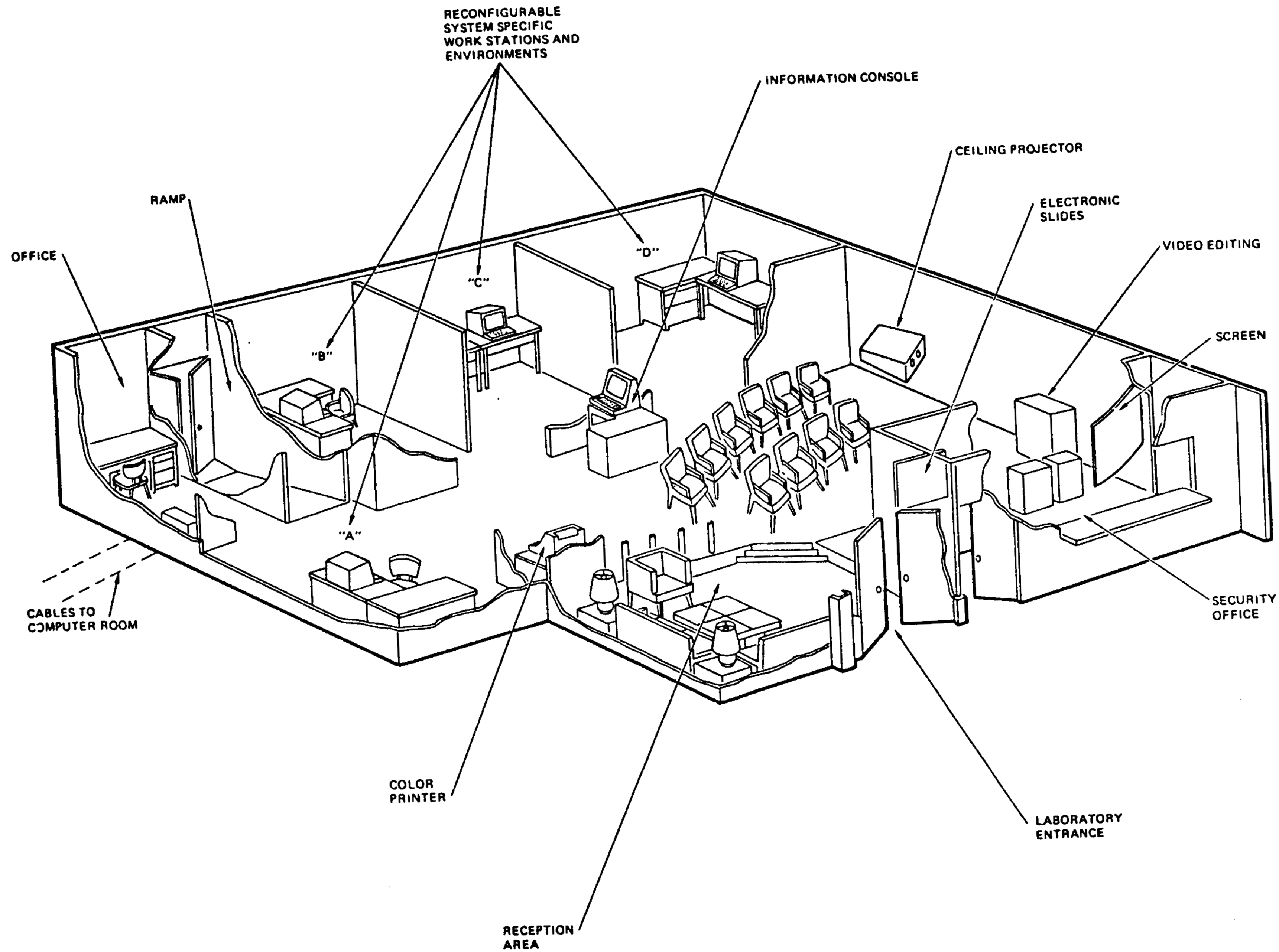

Figure 5. A Typical System Prototyping Laboratory

USER ENGINEERING EXPERIENCES

Name	Function	Customer	When
OBU	Ocean Surveillance	Navy	'82/12mo
SAFE	Intelligence Prod.	Class.	'82/3mo
SOGS	Space Telescope	NASA GOD.	'82/5mo
EXPR	Monoscopic Revision	TRW Res.	'83/1day
SPADOC	Space Threat Anal.	AF ESD	'83/18mo
CPGS	Map Finishing	TRW Res.	'84/.5mo
ASAS	All Source Anal.	Class.	'84/.5mo
BDIP	Bathymetric Integ.	Class.	'84/2mo
OMV	Orbiting Man. Veh.	NASA MAR.	'84/10mo
CMSS	Crisis Management	Class.	'85/4days
CHE	Space Station	NASA JOH.	'85/4mos
RMMS	Remote Maint. Mon.	FAA	'85/5mo
NTB	Space Defense	SDIO	'86/3mo
4700	Sensor Data Proc.	Class.	'86/6mo

Figure 6. User Engineering Applications