NORTHROP GRUMMAN

Applying User Centered Design in the Development of Systems without User Interfaces

National Defense Industrial Association

14th Annual Systems Engineering Conference

October 27, 2011

Dr. W. David Reese, CSEP

Dr. Conrad B. Monson

Systems and Specialty Engineering

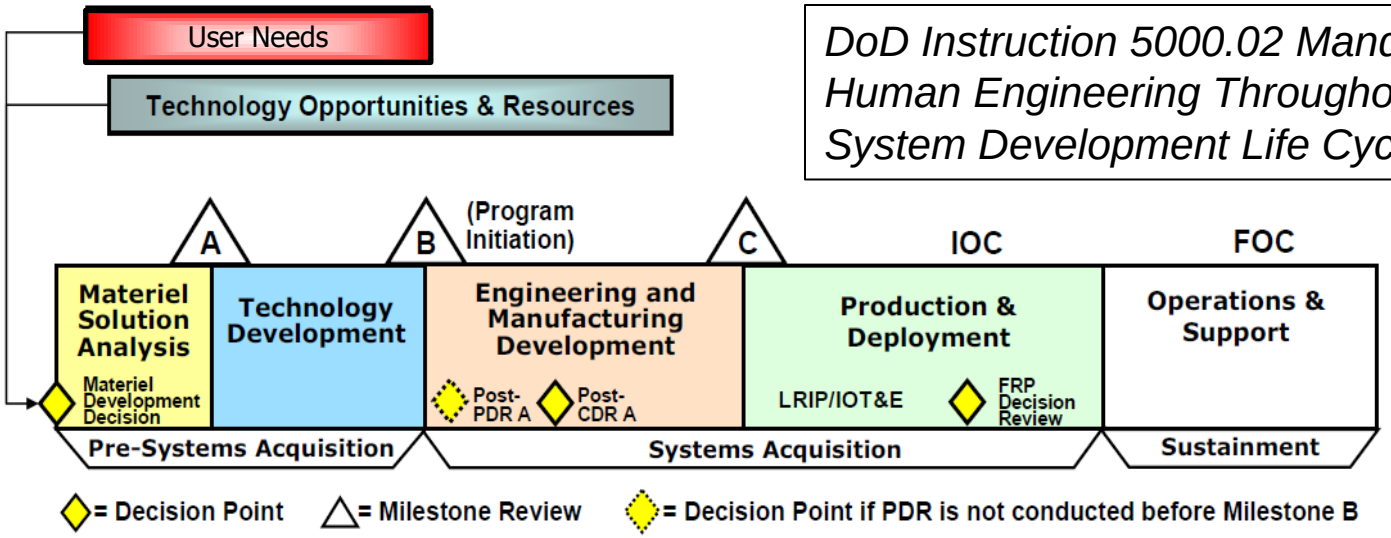
Northrop Grumman Electronic Systems

Agenda

NORTHROP GRUMMAN

- Human Engineering in System Acquisition
- User-Centered Design (UCD)
- System-User Interactions
- Today’s Acquisition Environment
- Evolving User Interactions – Case Studies in Missile Tracking
- UCD Principles to Consider in Systems Development
- Conclusion

Human Engineering in System Acquisition



DoD Instruction 5000.02 Mandates Human Engineering Throughout The System Development Life Cycle

- Human Engineering (HE) activities support the variety of system development efforts
  - Programs that begin before Milestone A and continue through Sustainment
  - Programs that are developed in only one or a few of the phases (e.g., an R&D effort that ends before Milestone B)
- Regardless of program, understanding user needs is not only key to the success of the HE effort but is also key to the success of the program

User-Centered Design

NORTHROP GRUMMAN

Identify Tasks

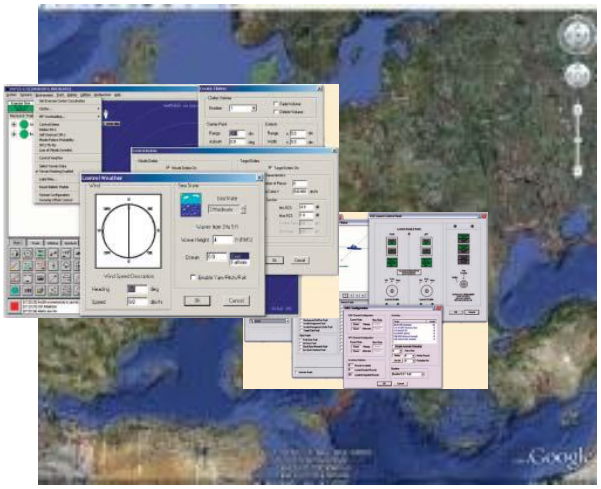
Use mission and goal -oriented task analyses, CONOPS to develop user tasks. Ensure architectures, requirements, system models , etc. reflect user operational needs and capabilities

Identify Information and Control Needs for Task Performance

What information does the user need, what are the display elements that provide that information, and what are the controls that enable users to act on the information?

Understand the User

What are the user goals, skills, experience and needs?



Create Prototypes

Develop cognitive task flows or sequences to organize display and control elements into prototypes with increasing levels of fidelity

Conduct Usability Tests and Evaluations

Evaluate and tests prototypes with users; re-design and re-test as needed

User-Centered Design (cont.)

NORTHROP GRUMMAN

Guiding Principles of the UCD process[1]

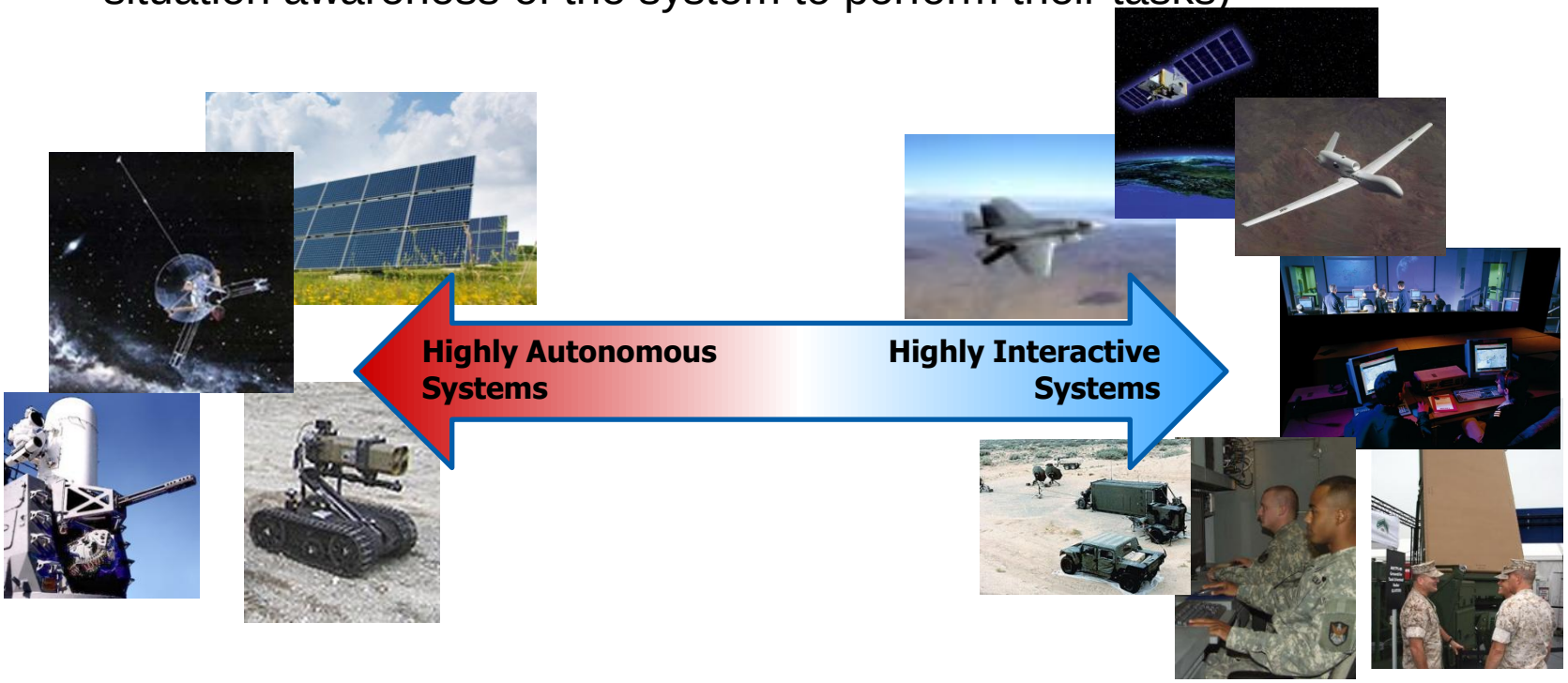
- Develop displays from user goals, tasks, abilities, information processing and decision making – don't let the technology drive the design
- Design systems that keep users in control and aware of the state of the system
- Ensure extensive user involvement in the design process
- Situation Awareness (SA) is key to effective design

  "If operators can achieve a high level of SA, they will be more effective … than if SA is denied or hard to achieve"

[1]Adapted from Mica Endsley, et. al. *Designing for Situation Awareness, An Approach to User-Centered Design*. Taylor and Francis, London and New York, 2003.

System-User Interactions

- **Autonomous Systems**
  - Minimal (if any) user interaction as the system functions autonomously in performing most (if not all) of the system tasks
- **Interactive Systems**
  - High levels of user interaction with some system functions performed autonomously and others performed by users (who must have sufficient situation awareness of the system to perform their tasks)



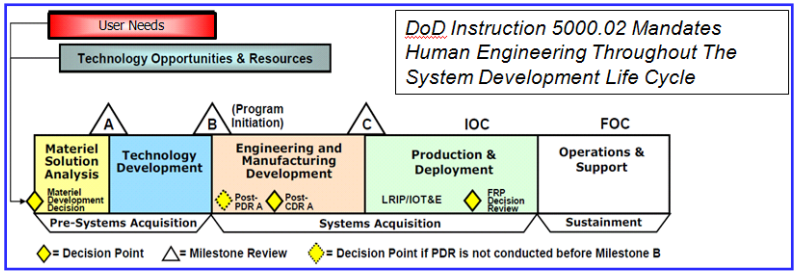
Today's Acquisition Environment for Military Systems

NORTHROP GRUMMAN

Systems need to cost less

…and be fielded more rapidly

Acquisition strategies are changing to meet these needs

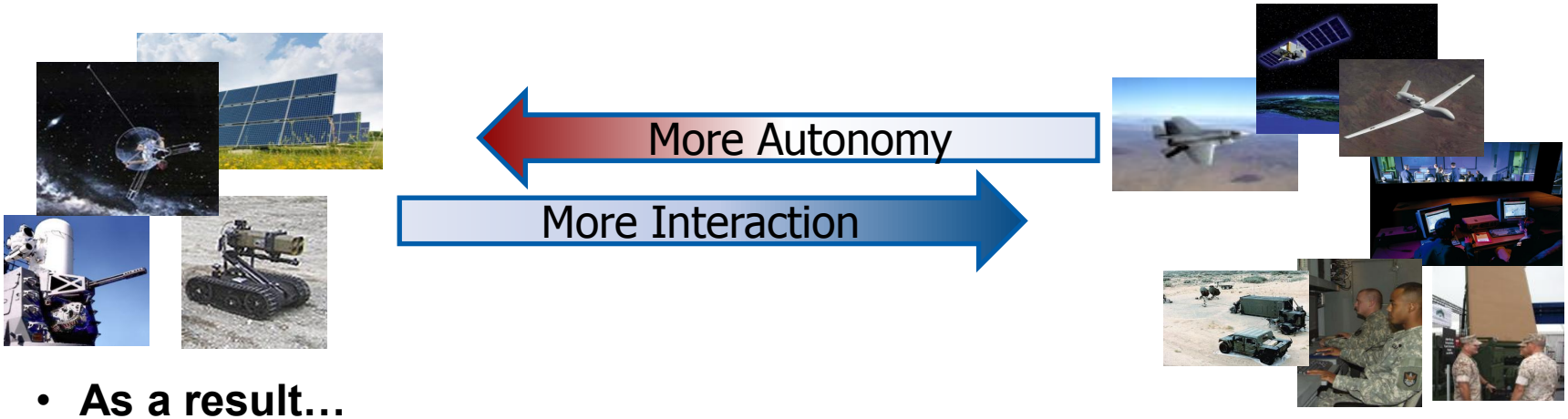


- There are fewer full cycle, system development programs (Milestones A-C)
  - Define and develop the system from scratch
  - Design, evaluate, test and refine user interactions and associated system interfaces
- There is more extensive re-use of existing system components
  - Increasing use of legacy components
  - Existing systems (or components) used in new ways
- Re-use can drive different—or entirely new—user interfaces
  - Re-use can change the degree of system autonomy or user interaction
  - This, in turn, may surface the need for an interface where none existed previously
  - User interface changes may also occur as a result of natural evolution as "next generation" systems are built

User Interactions Evolve to Support System Reuse

NORTHROP GRUMMAN

- **As systems evolve or are re-used, the user interaction normally changes**
  - Complex, highly interactive systems are made simpler with more automated elements
  - Automated, autonomous systems evolve towards more interaction with users



- **As a result…**
  - User tasks may need to change
  - Some user interfaces might be eliminated
  - Existing user interfaces need to be updated
  - New interfaces may need to be created

Evolving User Interactions
Case Studies in Missile Tracking

NORTHROP GRUMMAN

System-User interaction trends surfaced within several Research and Development efforts that support evolving missile defense systems

- An applied research program is evaluating the utility of new sensors and novel processing algorithms for missile tracking

- This multi-year effort is open ended with respect to the pipeline of new sensors and evolving algorithms to be evaluated
- Prototype track processing systems are used to gain confidence in results with the most promising improvements submitted as candidates for near term enhancements to operational systems
- A number of system-user interfaces have resulted from this program—many of which were not originally envisioned

Evolving User Interactions
Case Studies in Missile Tracking (cont.)

NORTHROP GRUMMAN

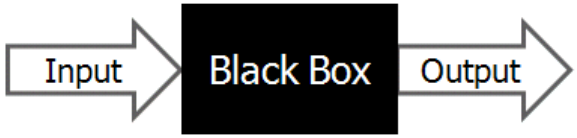
- Utility evaluation efforts led quickly to the need for a variety of tools
  - Scripting tools to automate or speed repetitive tasks
  - Logging tools to record outputs and behaviors
  - Simulation and playback tools to mimic external application environments
  - Analytical tools for characterizing performance
- Most tools were ad hoc creations—many were subsequently polished for use in algorithm refinement
  - Analytical tools were enhanced to generate graphical displays instead of columnar data
  - Simulation and playback tools were combined to simplify time synchronization needs
- Evaluation tools were originally designed without any intended user interaction
  - GUIs were sometimes added to allow junior (or less specialized) engineers to perform certain functions
- A subset of these tools proved useful to testers and maintainers of operational versions of the tracking systems and were further refined using more formal user interface design techniques

Earliest user interfaces targeted testers and maintainers

Evolving User Interactions
Case Studies in Missile Tracking (cont.)

NORTHROP GRUMMAN

- The originally targeted operational system was an autonomous “black box” that generated and passed through track data within a larger system of systems
  - Black box had no dedicated user or operator
  - Primary system interaction was with system maintainers
  - Provided rudimentary situation awareness via system’s health and status display
- Subsequent applications were found for the black box as less autonomous or even stand alone systems



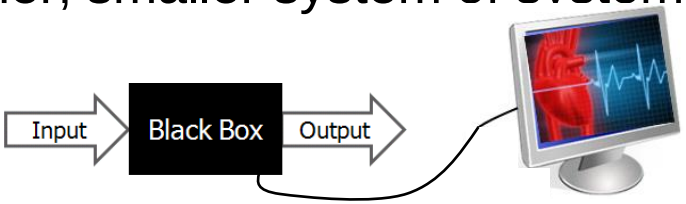
- Black box reuse…
  - Required the development of more interactive and informative operator interfaces
  - Significantly expanded need for user situation awareness
  - Previously non-existent user interfaces were suddenly critical to system utility
- Some interface functionality leveraged tools or displays previously built for software developer use
  - Graphical displays (e.g., mission globe) to visualize outputs
  - Numerical displays for performance cognizance

Had the potential need for a user interface been considered—as dictated by UCD—at the outset of the program, better interfaces could have been designed

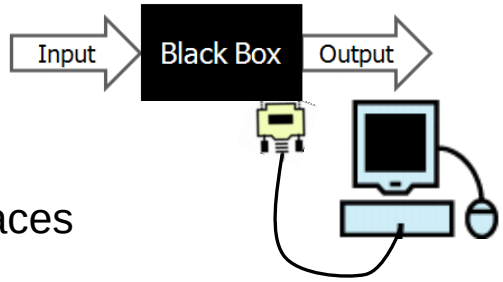
Evolving User Interaction Summary

- Tools developed during R&D phases often had utility and potential for reuse in later phases of system development, deployment or sustainment
- Efforts to improve black box within an operational system were not focused on the potential for reuse and failed to consider user interfaces that reuse might require
  - Ad hoc tools continued to be developed and or used
  - User interaction evolved to include system testers and maintainers

- Initial reuse of black box was within another, smaller system of systems
  - Users required some Situation Awareness
  - Health and status indicators needed
  - User interaction remained primarily at the more aggregated System of Systems level



- Later reuse was as a stand-alone system
  - User interaction with black box was now essential
  - Required introduction of (previously non-existent) user interfaces



Evolving User Interaction Summary (cont.)

NORTHROP GRUMMAN

- Our experiences with R&D-to-Ops development unveiled certain user interface development trends that may be typical in other development efforts
- The earliest interfaces tend to be informally designed and supported development or testing
  - Often evolve from simple tools and scripts
  - May include visualization aids or tools—especially when performance was key
  - Sometimes evolve into user interfaces for system operators or maintainers
- Initial user interfaces were oriented towards maintenance or situational awareness tasks (e.g. system health & status)
- User interfaces targeted at system operators required significant SE effort—especially when no operator interface was originally envisioned (i.e. when a system or component is repurposed for other uses)

UCD Principles to Consider in Systems Development

NORTHROP GRUMMAN

- Interfaces should be designed from user goals, tasks, abilities, information processing and decision making needs and abilities
  - Don’t let the technology drive the interface design
- System-user interfaces evolve over time
- *Potential* user interface needs or changes should be considered early in the design process—avoid designs that prevent interface changes or insertions
- Consider that user interfaces may involve a variety of users including
  - Developers
  - Maintainers
  - Testers
  - Operators / warfighters
  - Users of other, larger systems

UCD Principles to Consider in Systems Development (cont.)

NORTHROP GRUMMAN

- Today's acquisition environment makes component or system reuse more likely
- Sometimes component or system reuse may be accomplished by changes to user tasking without changes to user interfaces
  - Understanding system user's goals, skills, experience and needs – a key principle of UCD – could generate new or modified user tasks for component or system reuse that don't require a new or modified interface
- In many cases, there may be a need for interfaces that:
  - Support different kinds of users (developers, maintainers, testers, end users)
  - Provide a different degree of user interaction
  - Impart a different level of situation awareness (perception vs. comprehension of system operation or perhaps a projection of future system operations)
  - Are entirely new—where none had existed before ("exposed black box")

UCD Principles to Consider in Systems Development (cont.)

NORTHROP GRUMMAN

- Providing sufficient level of SA can "make or break" the operator or maintainer of a reused system or system component. Consider these requirements for SA in reuse[1]
  - **Perception level**: the user perceives the status, attributes, and dynamics of system elements
  - **Comprehension level**: the user integrates information to understand impacts upon system goals and objectives
  - **Projection level:** the user can project future impacts on system operations
- A reused system operating with high levels of autonomy may not even have a user interface… *SA must still be provided so that*
  - System health and status is understood
  - Over- or under-reliance on automation is avoided
  - User actions can be taken when the automation enabling the autonomy degrades or fails

[1]Adapted from Mica Endsley, et. al. *Designing for Situation Awareness, An Approach to User-Centered Design*. Taylor and Francis, London and New York, 2003.

Conclusion

NORTHROP GRUMMAN

- Today's evolving Acquisition Environment requires system components to evolve or get repurposed for other *uses…and the user interface needs to evolve as well*
- Case studies indicate there are potential efficiencies in planning for, or incorporating user interface "hooks" in designs to increase extensibility
  - Simplifies the future insertion of interfaces if and when ultimately needed
  - May enable leveraging of informal tools
- All systems interact with users at one or more levels
  - Developers, testers, maintainers…
  - End users at the System of Systems level
- With increased system re-use, applying User Centered Design principles to systems without user interfaces should become an increasingly important element of a modern day system development process

Backup

Design Processes Can Support Evolving User Interactions

NORTHROP GRUMMAN

Systems Engineering processes provide the framework for user interaction with the system

- Systems designed with Modular Open Systems Approach (MOSA)
  - Are interoperable
  - Support component re-use
  - Allow easy insertion of new capabilities (including insertion of user interfaces)
- Systems defined by DoDAF architectures include users in “viewpoints”
- Use Cases showing interactions between elements of a system, the “Actors”, can include interactions between users and other system elements

Designing Extensible Systems

NORTHROP GRUMMAN

- Extensibility Design Approach[1]
  - Use an up front design that allows for addition
    - Can't design everything in advance; provide a framework that allows for changes.
  - Make additions in small incremental steps
  - Separate work elements into comprehensible units
    - Keep each element at a human level … humans have to read and understand the system even if it is code generated by a computer

- Software Developments to Make Systems More Extensibility
  - Open Systems Architectures
  - Net Centric
  - SOA
  - MOSA

[1] Kelly, Allen. The Philosophy of Extensible Software, Overload Journal #50, August, 2002

Author Contact Information

Dr. W. David Reese
720-622-2211
Northrop Grumman Electronic Systems
William.reese2@ngc.com

Dr. Conrad Monson
720-622-2231
Northrop Grumman Electronic Systems
Conrad.monson@ngc.com