Seven ± Two Lessons Learned from the Development of Large-scale Systems

S. Gulu Gambhir, D.Sc.
SAIC Senior Vice President & Chief Technology Officer, Intelligence, Surveillance & Reconnaissance Group
The George Washington University, School of Engineering & Applied Science, Professorial Lecturer

11 September 2012

SAIC®

NATIONAL SECURITY • ENERGY & ENVIRONMENT • HEALTH • CYBERSECURITY

32	30	92	13
16	51	22	92
78	93	38	12
51	65	49	69
76	37	57	25

Miller's Magic Number

- "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information"
  - By George A. Miller (originally published in *The Psychological Review, 1956, vol. 63, pp. 81-97)*
    - Span of immediate memory
    - "Let me summarize the situation in this way. There is a clear and definite limit to the accuracy with which we can identify absolutely the magnitude of a unidimensional stimulus variable."
    - "I would propose to call this limit the *span of absolute judgment,* and I maintain that for unidimensional judgments this span is usually somewhere in the neighborhood of seven."

SAIC®

Agenda

1. Miller's Magic Number Applied to Systems Engineering
2. Requirements Elicitation
3. Requirements Specification
4. Design Practice
5. Test

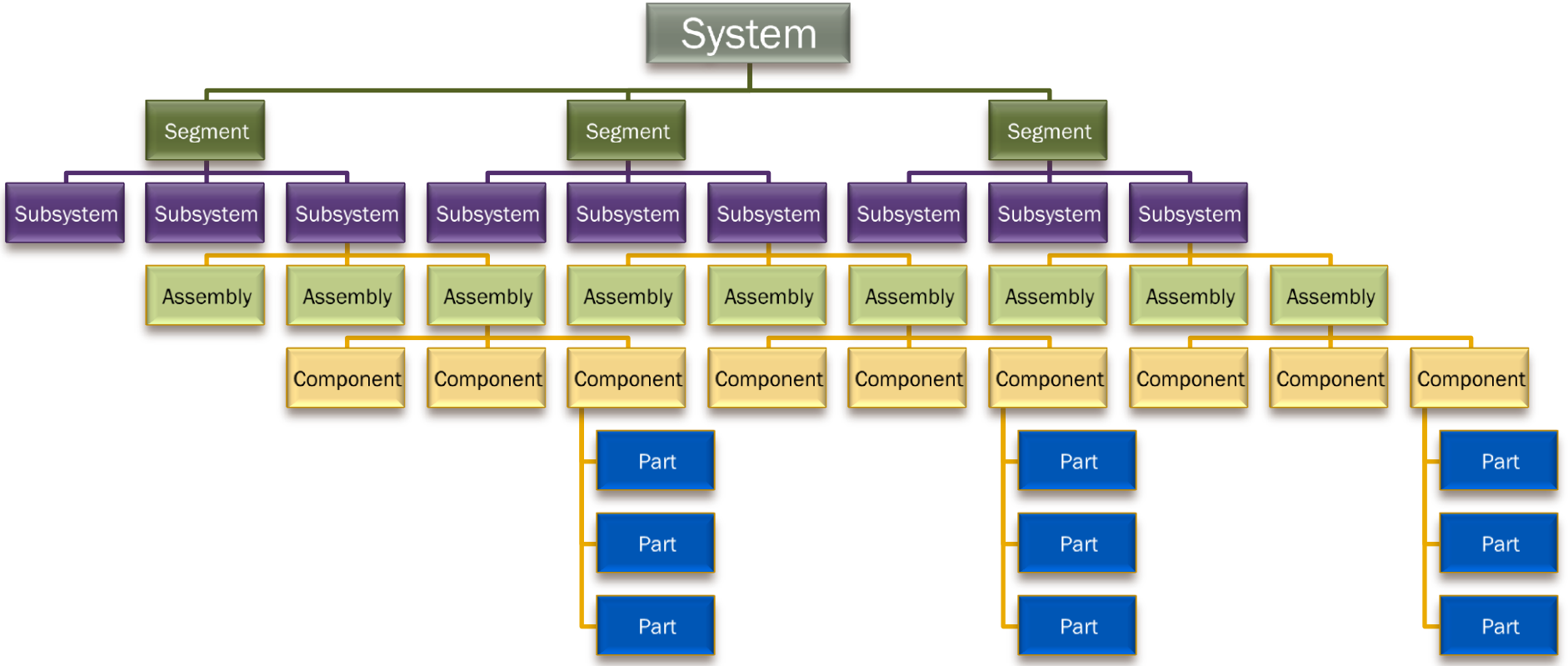
Developing Systems

- Reductionism
  - Foundation of most system developments … taking a large problem and breaking it up into a series of smaller problems
  - We hope that by solving the smaller problems, and integrating these solutions, we solve the larger problem
  - Systems thinking: component parts act differently when separated from the whole
    - Or … the system acts differently than the component parts would lead you to expect

- $T^2 = R^3$
  - The square of the orbital period of a planet is proportional to the cube of the semi-major axis of its orbit (Kepler's 3rd Law)
- V = IR
  - Current through a conductor is proportional to the potential difference (Ohm's Law)
- ?? = ??
  - 1st law of systems?

SAIC®

Decomposition



7±2 elements in each grouping

SAIC®

Trade Study Candidates:
RAND Analysis of Alternatives for KC-135 Recapitalization

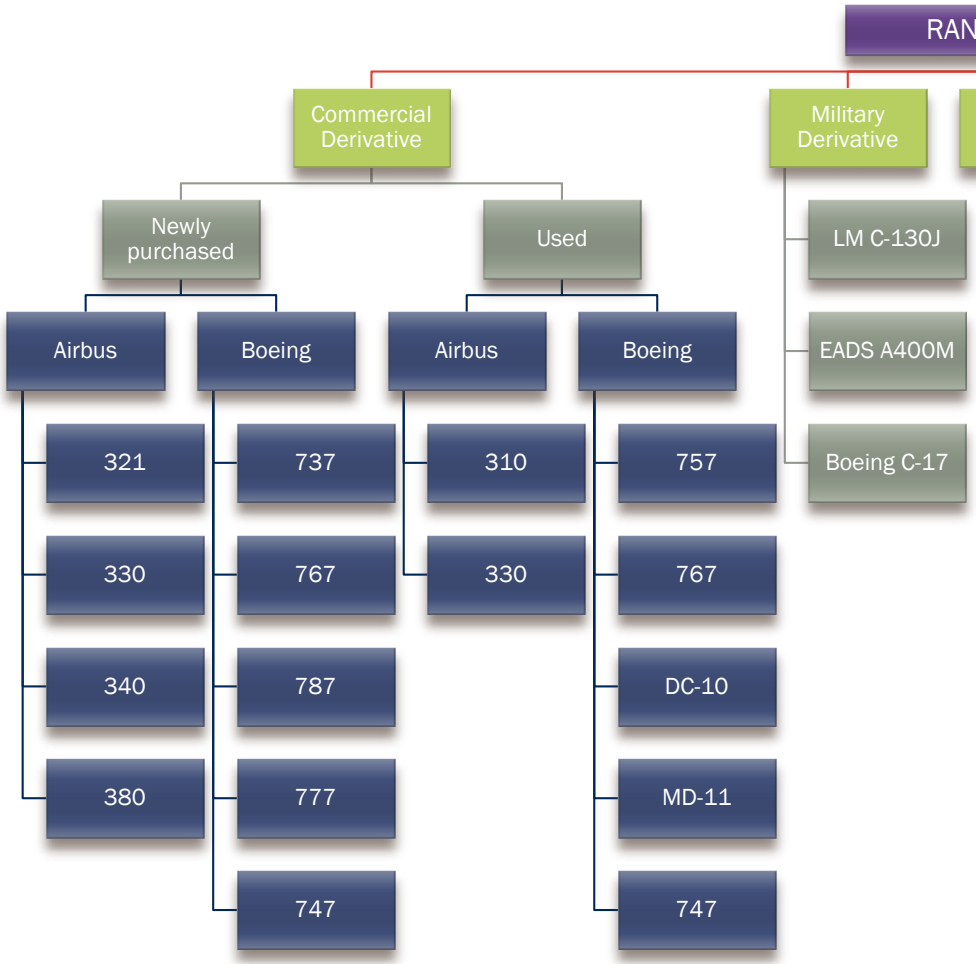
1. LM* C-130J
2. EADS® A400M
3. Boeing® C-17
4. Airbus® 321 (new)
5. Airbus® 330 (new)
6. Airbus® 340 (new)
7. Airbus® 380 (new)
8. Airbus® 310 (used)
9. Airbus® 330 (used)
10. Boeing® 737 (new)
11. Boeing® 767 (new)
12. Boeing® 787 (new)
13. Boeing® 777 (new)
14. Boeing® 747 (new)
15. Boeing® 757 (used)
16. Boeing® DC-10 (used)
17. Boeing® MD-11 (used)
18. Boeing® 747 (used)
19. Boeing® 767 (used)
20. Boeing® 767 (used)
21. New designs
22. Unmanned aerial vehicles as tankers
23. Stealthy tankers
24. Fleets with mixes
25. Commercial sources for refueling

*LM = Lockheed Martin Corporation. EADS is a registered trademark of the European Aeronautic Defence and Space Company in the U.S. and/or other countries. Boeing is a registered trademark of The Boeing Company in the U.S. and/or other countries. Airbus is a registered trademark of Airbus Deutschland GmbH in the U.S. and/or other countries.

SAIC®

Source: www.rand.org

Trade Study Candidates: RAND Analysis of Alternatives for KC-135 Recapitalization



7±2 alternatives in each grouping

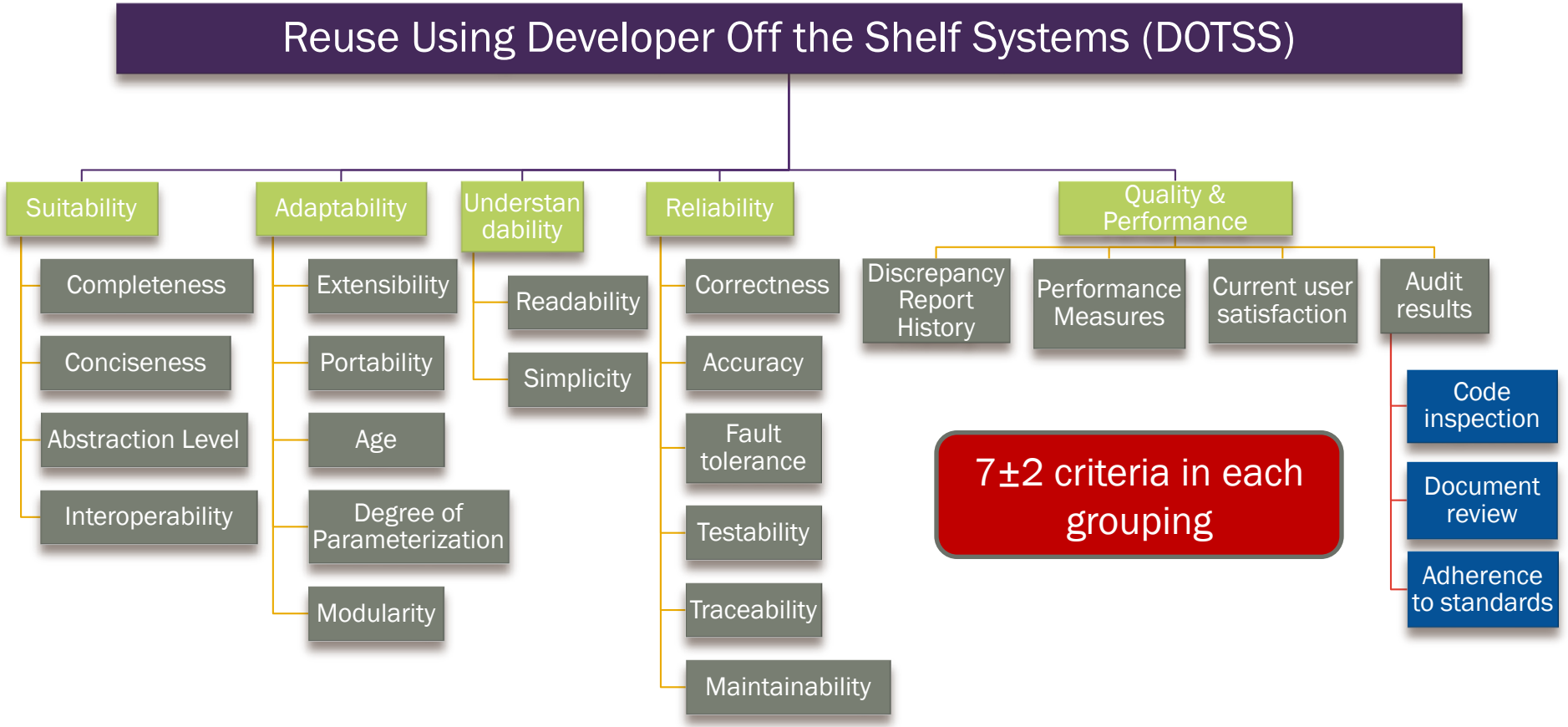
Source: www.rand.org

Trade Study Evaluation: Evaluation Criteria

- Should be tied to mission objective or need
- Should be measurable
- Should help differentiate among the alternatives
- Should be orthogonal to one another
  - Don't double count either strengths or weaknesses
- Can be qualitative or quantitative
  - Qualitative
    - Expandability
    - Quality of design
    - Appearance
    - Quality of construction
    - Ease of use
  - Quantitative
    - Called Measures of Effectiveness (MOEs)

SAIC®

Trade Study Evaluation: Evaluation Criteria (2)



Eisner, Howard "Reengineering the Software Acquisition Process Using Developer Off-the-Shelf Systems (DOTSS)" Proceedings of the 1995 IEEE International Conference on Systems, Man, and Cybernetics.

SAIC®

Applying Miller's Magic Number

- Systems engineering context
  - Functional decomposition
  - Review checklists
  - Trade study alternatives
  - Trade study evaluation criteria
  - Interface specification
  - Test plans
  - Training plans
  - Operations plans

SAIC®

Complexity – Technology

Source: http://firstflight.open.ac.uk

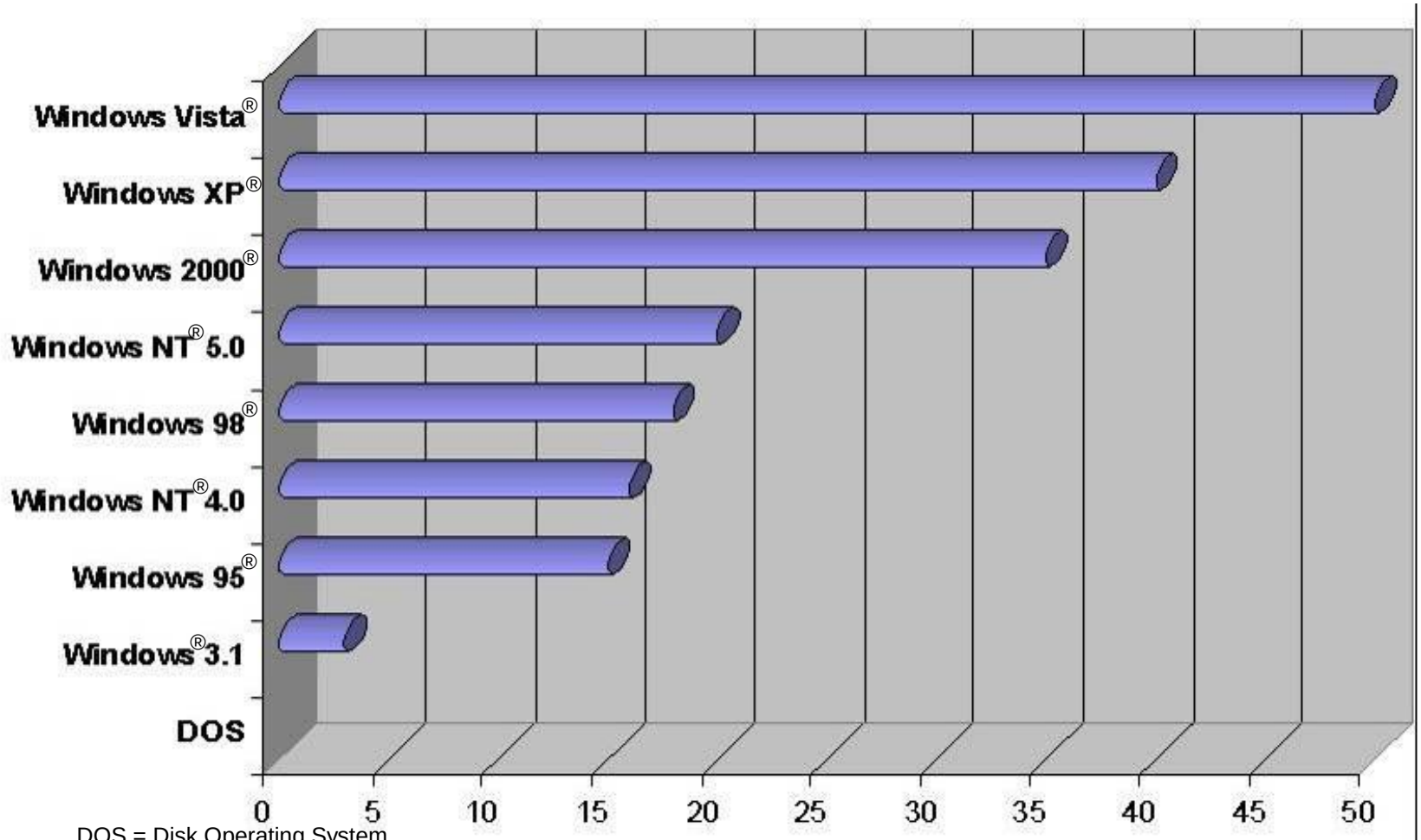
The rapid pace of technological advancement has required systems of increasing complexity to fully exploit available technology

- **Wright Flyer (1903)**
  - December 17, 1903, Orville and Wilbur Wright
  - Wood plus cloth construction
  - 12h-hp engine
  - Altitude of 10 feet, 12-second flight went 120 feet
- **Subsystems**
  - Structural, controls, propulsion

- **Boeing® 777 (1995)**
  - 281 passengers and 22,000 lbs of cargo
  - Range: 4,600 miles
- **Subsystems**
  - Structural, controls, propulsion, electrical, entertainment, food, toilet, etc.

Microsoft Windows® Evolution

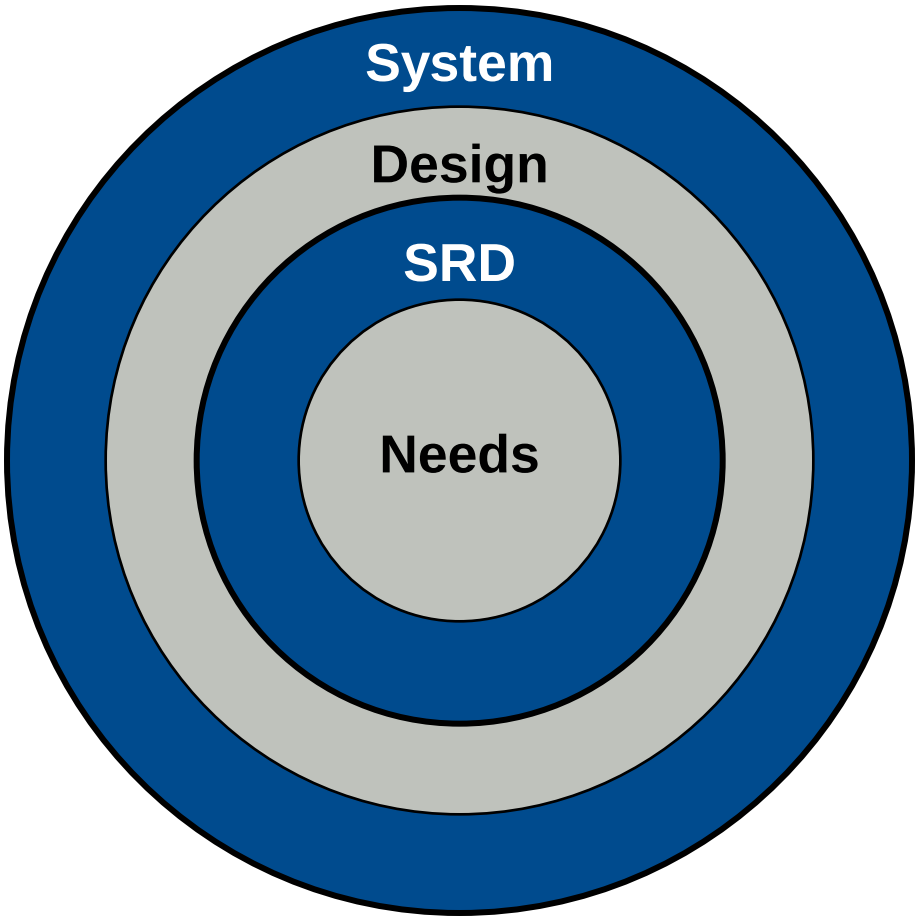
(Estimated Lines of Code – Millions)



DOS = Disk Operating System

Needs ... to Systems

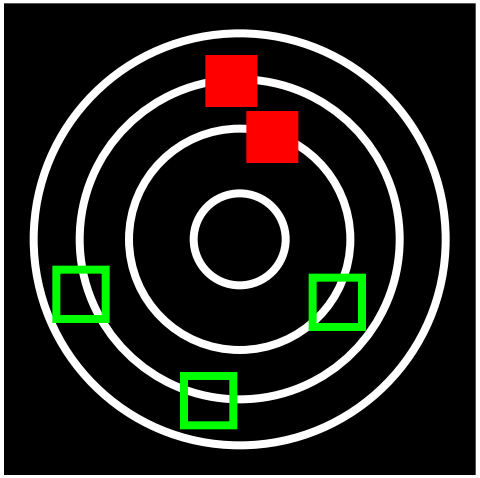
- Many large-scale system development problems have been traced to poor system requirements
- Requirement errors remain latent and are not “discovered” until late in system development
  - The later in the development life cycle that a requirement error is uncovered, the more expensive it is to repair
- Types of errors
  - Incorrect facts
  - Omissions
  - Inconsistencies
  - Ambiguities



SRD = System Requirements Document

Separating the "What" From the "How"

- Users and/or customers may not understand or be able to fully articulate their needs
- Example: air traffic controller
  - What the user says the need is
    - "I need to see these planes in red"
  - What the user means
    - "I need to be alerted that there is an alarm situation"
- Role of the systems engineer
  - Separate the what from the how
  - Distinguish between needs and wants
    - Every requirement carries a cost



Requirements Elicitation … What Do We Want to Learn?

….from our *customers*?

- Understanding of the problem in solution-independent terms
- The user groups that will be affected by the system
- Overall program goals
- Specific program objectives
- Motivations
- Concerns
- Expectations
- Constraints
- Priorities – performance, cost, schedule, risk

….from our *users*?

- Understanding of the problem in solution-independent terms
  - Domain-specific terminology
- Problems or limitations with the current system
  - Shortfall identification
- Domain knowledge
  - Functions, interfaces, performance issues, physical characteristics, operational issues, RMA, operability requirements
- Needs versus wants
  - New ideas

Gaining knowledge relevant to successfully solving the problem

RMA = reliability, maintainability, and availability

Requirements Elicitation ...
Example of Scenario-based Elicitation

- **Scenario-based elicitation**
  - **Scenario is a story that illustrates how a perceived system will meet user needs**
  - **Allows the users to more easily conceptualize the system**
- **Scenarios (or use cases)**
  - **Ask users what they do today and why**
  - **Ask users what they expect to do in the future**
- **Identify**
  - Actors
  - System elements
  - Activities
- **Develop matrix with customer and users**
  - Common understanding of how the system will be used and what it is supposed to do

	System Elements	
Actor	**Bank Line**	**ATM**
Bank Customer	Withdraw cash Check account balance Make a deposit	Make a deposit Withdraw cash Check Account balance
Bank Employee	Give account balance Give cash out Take deposits	Take deposits from ATM Add money to ATM

SAIC®

Elicitation Methods and Recommendations

Elicitation methods

- Interviewing (formal or informal)
- Scenario-based
- Form analysis
- Reverse engineering and SRD re-use
- Joint application design (JAD)
- Quality function deployment (QFD)
- Brainstorming
- Surveys and questionnaires
- Examine interaction with existing system or prototype
- Workshops
- Focus groups
- Panel groups
- Help desk feedback

Best Practice

- There is no single elicitation method well suited for every situation
- Select the most appropriate method that fits within the constraints of your problem
- Each elicitation technique may yield unique information

SRD = System Requirements Document

SAIC®

Specifying What You Want …

- No single list of factors that is well accepted by community
  - Balzer and Goldman (1979)
  - Yeh and Zave (1980)
  - Bethke et. al. (1981)
  - Lewis (1982)
  - Roman (1985)
  - Davis (1995)

- Davis includes 13 factors
  - Apply primarily to the SRD as a whole
    - Complete
    - Consistent
    - Modifiable
    - Traceable
    - Organized
  - Apply primarily to each requirement
    - Correct
    - Unambiguous
    - Verifiable (method identified)
    - Understandable by customer
    - Traced (rationale)
    - Design independent
    - Annotated (priority and relative stability)
    - Concise

SRD = System Requirements Document

Davis Requirements Quality Factor – Verifiable

- "An SRS is verifiable if, and only if, every requirement stated therein is verifiable." (Davis)
- "A requirement is verifiable if, and only if, there exists some finite cost effective process with which a person or machine can check that the actual as built software product (system) meets the requirement [IEEE 1984]."
- Typical verification methods:
  - Test
    - Performance measured during or after a controlled application of a functional and/or environmental stimulus
    - Quantitative measurements are analyzed to determine the degree of compliance
  - Demonstration
    - Items are observed, but not measured, in a dynamic state
  - Analysis
    - Comparing hardware or software design with known scientific and technical principles, procedures, and practices
  - Inspection
    - Static state examination of the hardware, software, and/or technical documentation and data

SAIC®

Davis Requirements Quality Factor – Traced

- "An SRS is traced if the origin of each of its requirements is clear." (Davis)
- Rather than including only the origin of the requirement (for example, reference to another document), include a rationale
  - Rationale describes why the requirement is included
  - Rationale should also describe why specific values were selected
    - Requirement
      - "The system shall be capable of accommodating six simultaneous satellite contacts."
    - Rationale
      - "We are currently operating four satellites, but believe that we may add two additional satellites to our operational constellation in the near future."

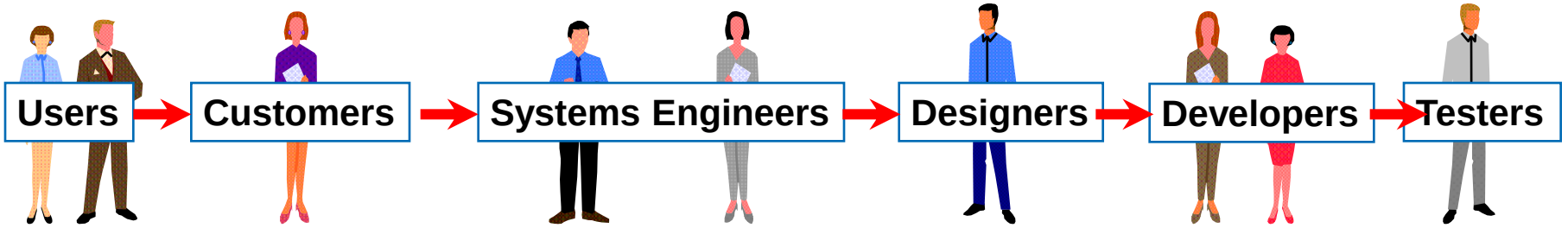
SRS = software requirements specification

Davis Requirements Quality Factor – Annotated

- Things we should know about this requirement
  - Relative necessity

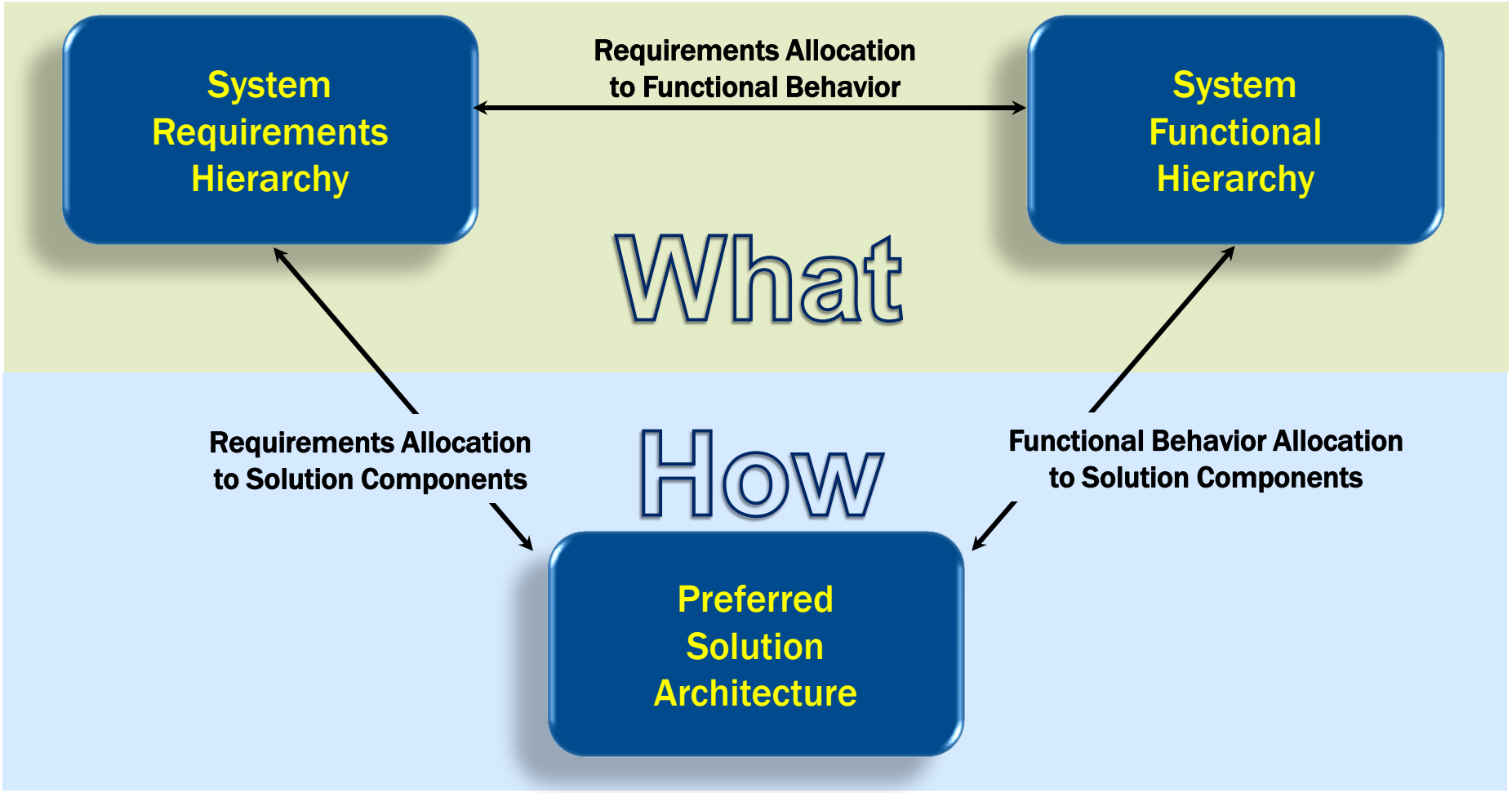
Priority		
Shall	Threshold	Key Performance Parameter
Shall, where practical	Objective	
Preferred or should		
May		

  - Relative stability
    - Likelihood to change in the future



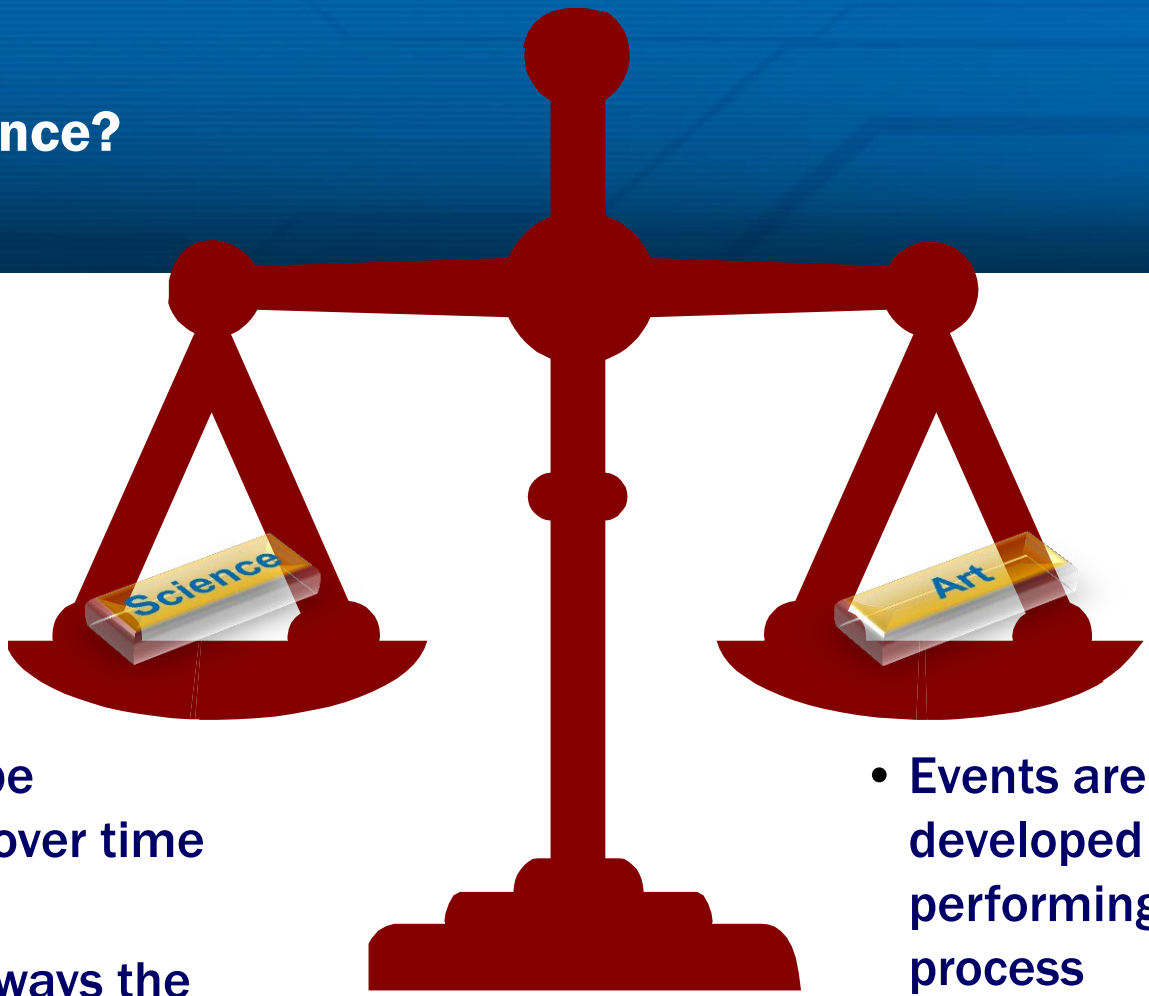
SAIC®

Design Practice



SAIC®

Art or Science?



- Events can be reproduced over time and space
- Answer is always the same each time the process is performed

- Events are uniquely developed even when performing the same process
- Results are not exactly the same; though not necessarily large, differences are discernible

Artists Don't Have a Monopoly on Creativity or Good Design

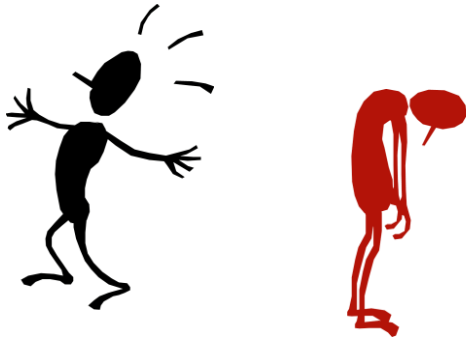
Dieter Rams' 10 principles of "good design"

1. Good design is innovative
2. Good design makes a product useful
3. Good design is aesthetic
4. Good design makes a product understandable
5. Good design is unobtrusive
6. Good design is honest
7. Good design is long-lasting
8. Good design is thorough down to the last detail
9. Good design is environmentally friendly
10. Good design is as little design as possible

SAIC®

Design Practice Practices

- **Trades and regrets**
  - Design is about making choices
  - Optimize globally as much as possible; recognize regrets" and the affected groups



- **Process and documentation**
  - A little bit of process is a good thing
  - Documentation – less is better than more

- **Challenging requirements and assumptions**
  - Ask why
  - Understand the approximately one dozen requirements that drive the system; what can be done to make development easier?

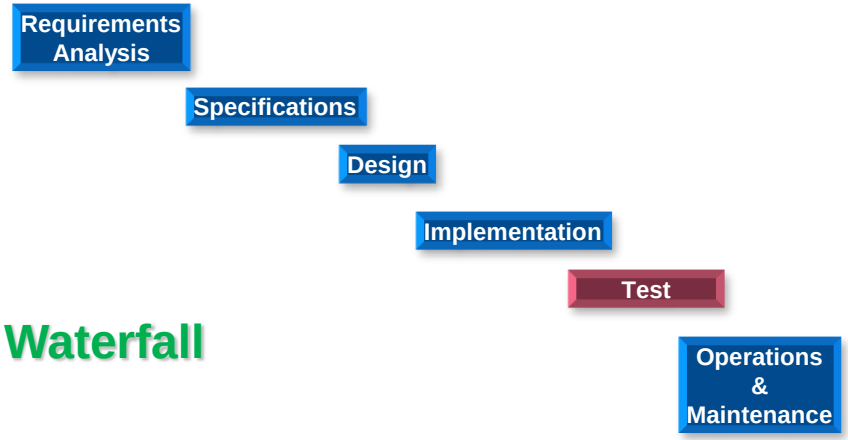
Design Practice Practices

- **Margin is good**
  - Start with some, use as needed, retain as you can

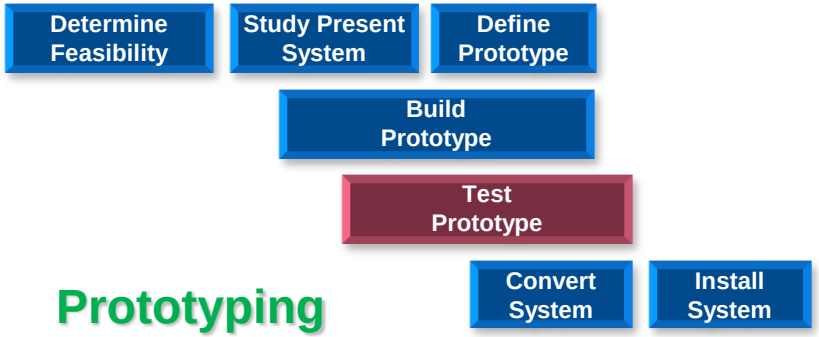
- **Interfaces**
  - Provide leverage and should be carefully considered
  - In partitioning, chose elements as independent from one another as possible
  - Strive for low external complexity, high internal complexity

System Testing

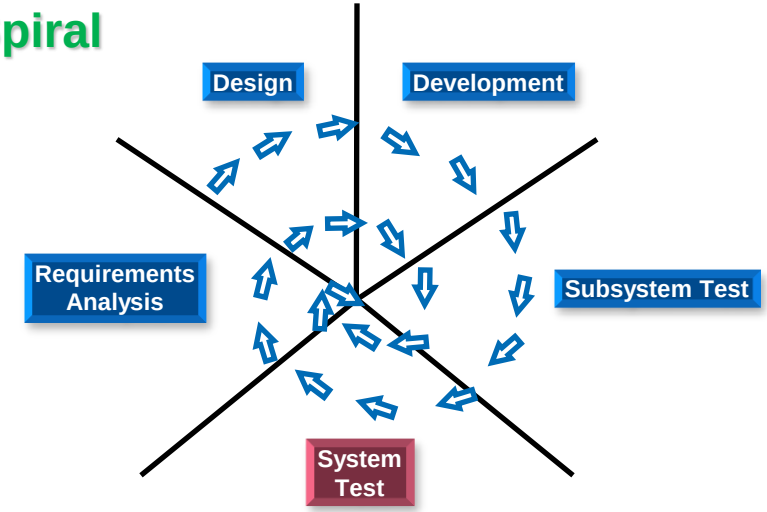
Waterfall



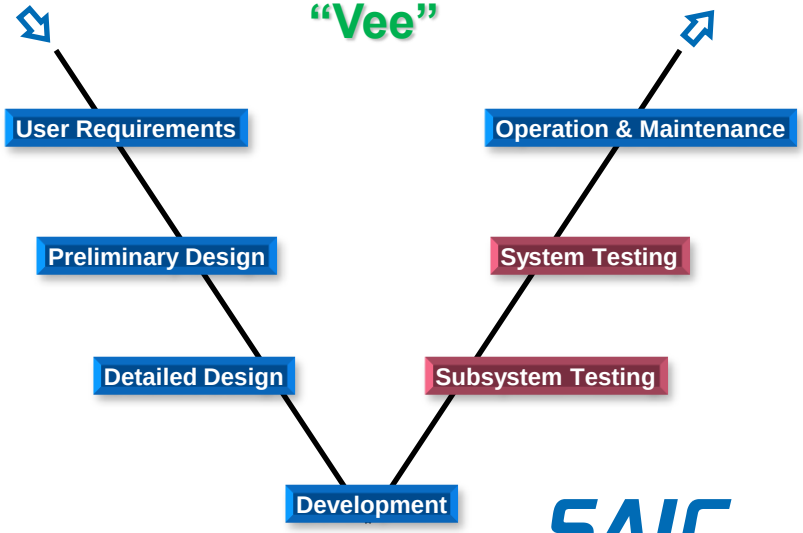
Prototyping



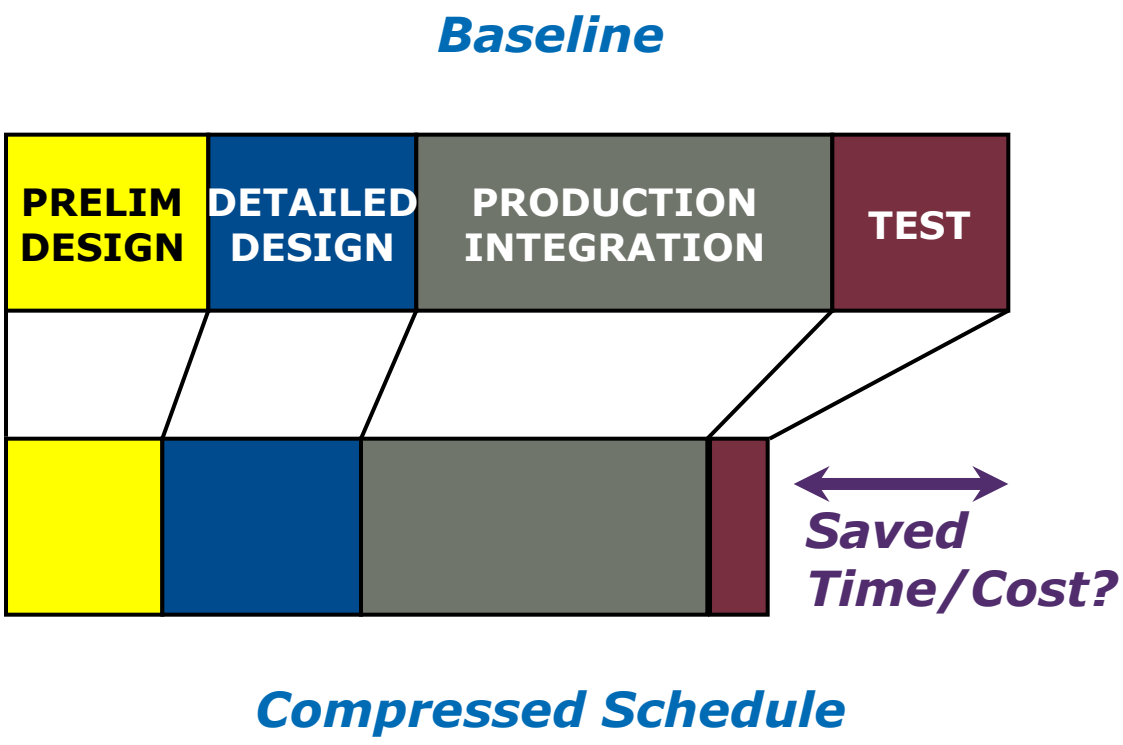
Spiral



"Vee"



System Testing



System Testing

APOLLO SPACECRAFT

George M. Low
Manager, Apollo Spacecraft Program
NASA Manned Spacecraft Center
Houston, Texas

Abstract

The flawless performance of the five manned Apollo flights is attributed to reliable hardware; thoroughly planned and executed flight operations; and skilled, superbly trained crews. Major factors contributing to spacecraft reliability are simplicity and redundancy in design; major emphasis on tests; a disciplined system of change control; and closeout of all discrepancies. In the Apollo de-

Spacecraft Design

The principles of manned spacecraft design involve a combination of aircraft design practice and elements of missile design technology: Build it simple; and then double up on many components or systems so that if one fails the other will take over. There are many examples in our spacecraft: ablative thrust chambers that don't require regenerative cooling; hypergolic propellants that do not

Source: George M. Low (Manager, Apollo Spacecraft Program) NASA Manned Spacecraft Center, Houston, Texas, 1969.

System Testing

Apollo Test Activities

The single most important factor leading to the high degree of reliability of the Apollo spacecraft was the tremendous depth and breadth of the test activity.

DEVELOPMENT AND QUALIFICATION TESTS

[Full-Scale Spacecraft Testing]

Escape motor flight tests	7 flights
Parachute drop tests	40 drops
Command Module land impact tests	48 tests
Command Module water impact tests	52 tests
Lunar Module structural drop tests	16 drops
Lunar Module complete drop tests	5 drops
Command and Service Module acoustic/vibration tests	15.5 hr
Lunar Module acoustic/vibration tests	3.5 hr
Command and Service Module modal survey testing	277.6 hr
Lunar Module modal survey testing	351.4 hr
Command and Service Modules thermal vacuum tests	773 hr
Lunar Module thermal vacuum tests	2652 hr
Service Module propulsion systems tests	1474.5 min
Ascent stage propulsion systems tests	153 min
Descent stage propulsion systems tests	220 min

Source: George M. Low (Manager, Apollo Spacecraft Program) NASA Manned Spacecraft Center, Houston, Texas, 1969.

Seven ± Two Lessons Learned From the Development of Large-scale Systems

1. Miller's Magic Number Applied to Systems Engineering
2. Requirements Elicitation
3. Requirements Specification
4. Design Practice
5. Test

SAIC®