Terry R. Snyder, with more than 30 years of management and programming experience of software for large-scale, real-time computer systems, was appointed vice president of systems and software engineering for Hughes Aircraft Company in October 1994. In addition, Snyder manages the systems and software engineering division of Hughes Information Systems segment and is also its engineering director.

As vice president, Snyder is responsible for institutionalizing a common set of systems and software processes, methods, metrics and tools to increase reuse, reduce cycle time and cost, and increase employee mobility throughout Hughes.

Snyder has spent a significant part of his time over the past three years ensuring that over one million lines of high quality software would be produced in time to meet a key milestone on the Peace Shield Program for Saudi Arabia. The USAF has recognized this as one of the most successful programs ever and is adopting many of the Hughes processes and metrics as a standard.

Snyder's main responsibility as manager and engineering director of the systems and software engineering division is to support Hughes Aircraft Company with high quality software. The systems and software engineering division is a recognized leader in software development as indicated by its high rating in software process maturity from the Software Engineering Institute.

Snyder graduated from Pennsylvania State University in 1960 with a bachelor of science degree in mathematics, and completed the UCLA Executive Management Program in 1978. Snyder spent six years as a senior research engineer for the Autonetics Division of North American Rockwell, working on the Minuteman and Advanced Minuteman programs. He joined Hughes Aircraft Company in 1966.

Snyder is chairman of the company's Systems Software Engineering Council. As an active member of the software technology world outside of Hughes, Snyder is an advisory board member of the Software Engineering Institute, served as vice chairman of the Board of Directors for the Software Productivity Consortium and was chairman of the Embedded Computer Software Committee for the Aerospace Industries Association. He is Hughes' corporate key executive for California State University, Fullerton and is on the UC Irvine Information and Computer Science Department's Advisory Board.

Snyder was born on Dec. 27, 1938, in Allentown, Penn., and lives in Tustin, Calif.
Panel: COTS Considerations in Government Contracting

USC Center for Software Engineering Executive Workshop on "System Integration with Commercial-off-the-Shelf Software (COTS Integration)"

Terry R. Snyder
Hughes Aircraft Company
COTS in the context of reuse

Reuse in the context of DoD Policy (with influence of Government outside DoD)

Changes to development process

Composite systems a key issue

Issues are very, very complex
  - Problems are solvable
  - In Software and Systems Engineering context
  - Disciplined process is mandatory
DoD Software Reuse Initiative
Vision and Strategy

- Key Points: software reuse improves development and maintenance, and 2) outlines a specific agenda for mechanisms of accomplishing software reuse.

- Vision: "... foster software reuse through domain-specific, process-driven, architecture-centric, technology-supported approaches.

- Strategy: "... is based on systematic reuse ..."

- This vision and strategy is not just DoD; it is right in line with emerging commercial software industry consensus on how to go about reuse.
Hughes Aircraft Company develops complex systems by deriving new and modified components principally from existing assets rather than by the creation of new software.

There are three aspects of such development:

1. It is **process-driven**.
2. It is **domain-specific**.
3. It is **asset-based**.

It is implicit in the vision that we do not reuse software assets simply for the sake of reuse, but rather only where it reduces cost and cycle time.
Development Process Adapted from DISA
*Domain Engineering Process version 2.0*

Domain Management

- Domain Definition
- Domain Analysis
- Domain Design
- Domain Implementation
Evolutionary development
- COTS and legacy asset improvements can increase functionality and reduce custom code
- Open systems are key
- "Glue" or "wrapper" code is a distinctive feature
- Composite systems are vital to our business
Trade Study Issues (1)

- source of material to be reused
- compatibility with planned new designs
- number of source lines of code
- projected percent of modification needed
- performance
- effort to modify
- interface definitions
- reuse of all software products (e.g., requirements, design, code, test cases)
- rights (do we have right to use? can we pass on the rights?)
Trade Study Issues (2)

- security
- source development requirements (e.g., DoD-STD-2167A, DoD-STD-1750A) and process standards implemented (e.g., coding standards)
- current rights (e.g., NDI that is Proprietary, COTS, GFE or CFE)
- current stage of software development
- ability to demonstrate that the reuse software requirements are traceable to the program requirements
- existence of documentation which describes the characteristics of the reuse software design and code
Trade Study Issues (3)

- existence of test plans, cases and records to ensure the reuse software performed its required functions
- existence of software quality assurance certification records
- feasibility (e.g., transportability, context independence, low component complexity)
- availability of reuse items to meet the program's schedule requirements
- choice of computer languages
- choice of processors
Trade Study Issues (4)

- determination of system and software architecture
- analysis of the processes and methods used to originally develop the potentially reused software, and a determination of whether or not to adopt such processes and methods
- effect of the reuse decision on further development of reuse assets
- maintenance of reuse repositories for the assets

And here is another set of issues to be dealt with about risk analysis. And another set of issues regarding non-COTS NDI or modified reused software!
There is considerable agreement on effective processes for reuse, including COTS.
- We believe the process is sufficiently different that it requires formal change to our directive system.

The notion of composite systems solves some COTS implementation problems.

There are no easy answers either to selection of components or dealing with interfaces, configuration control, and so on.
- But there are answers.
- And they exist in the framework of traditional systems engineering, using trade studies and technical performance measures.