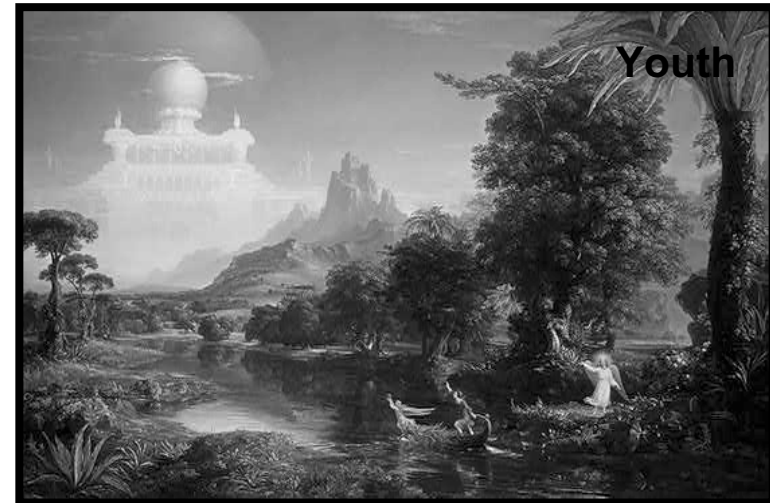


# Software Product Maturity in SIS Source Selection



Thomas Cole, *The Voyage of Life*, 1842, National Gallery of Art, Washington, DC

**Richard Turner**, OUSD(AT&L)/DS/SIS (George Washington University)  
Acquisition of SIS Conference, January 28, 2004

# Software Product Maturity in SIS Source Selection



*Thomas Cole, The Voyage of Life, 1842, National Gallery of Art, Washington, DC*

**Richard Turner**, OUSD(AT&L)/DS/SIS (George Washington University)  
Acquisition of SIS Conference, January 28, 2004

# The Requirement

- Section 804 of the '3 Defense Authorization Act requires that OSD

“assist MILDEPs by ensuring criteria applicable to selection of sources provides added emphasis on

- *Past performance of potential sources*
- *Maturity of software products offered by potential sources”*

# Defining SW Product Maturity

- No standard definitions/scales
- Not Software Technology Readiness Levels (TRL)
  - Maturity addresses a specific product
  - TRL addresses underlying technology
- Highly dependent on environment and application context
- Many dimensions of maturity

# The Approach

- Gathered a group of experts to:
- Review existing approaches
- Develop characteristics and information sources
- Develop guidance for source selection
- Develop RFQ/RFP language

# Focus on Source Selection

- **General maturity problem is extremely difficult**
  - Limited time and resources
  - Need for significant effort to work on development-based maturity
  - Some promising work (MDA, AF) but untried
- **Source selection was the Congressional emphasis**
- **Source selection bounds the problem to measuring existing, working software (e.g. COTS, GOTS, legacy)**

# Software in Source Selection

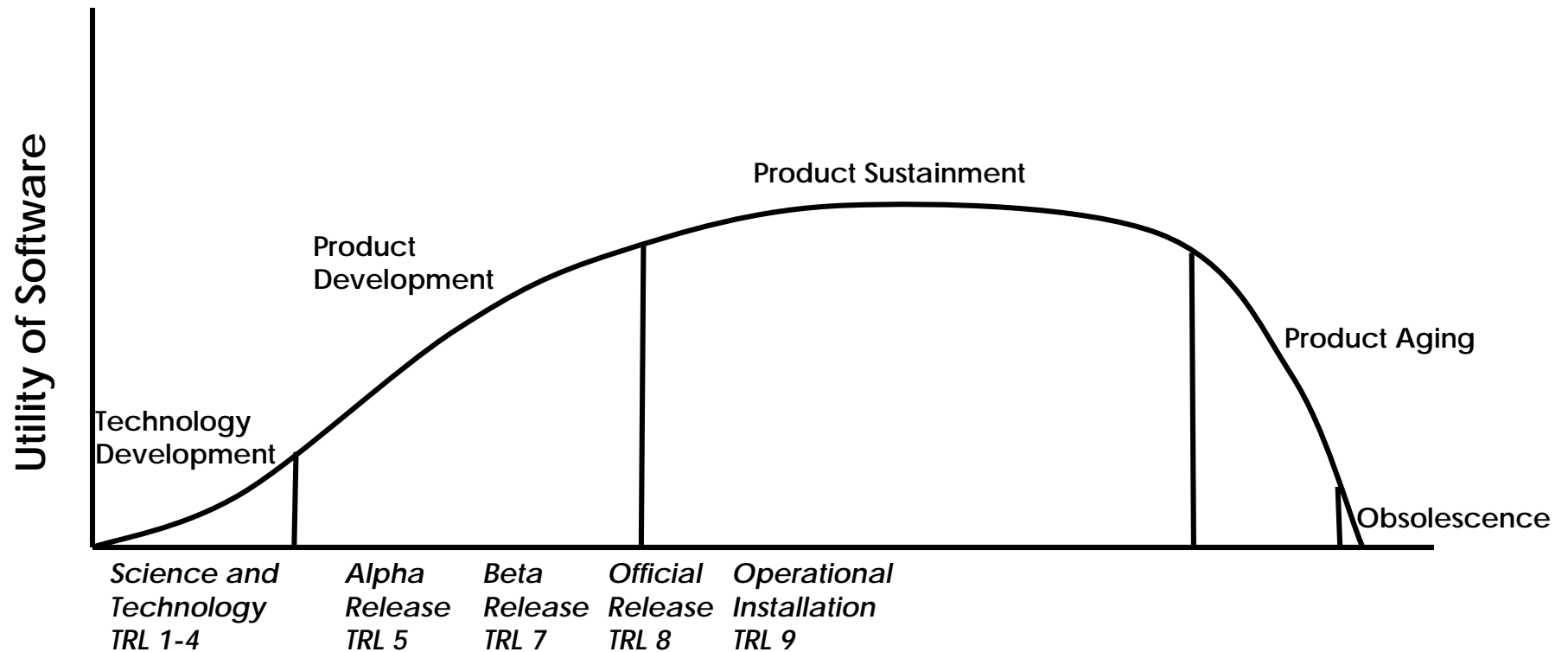
- Many different kinds of source selections
  - Greenfield vs. Upgrade
  - Traditional business-process IT system implementation vs. Command and Control or embedded software
- Different kinds of software in programs
  - Only software that exists has determinable maturity
  - Aggregations of existent and non-existent software are common
- Software doesn't exist (Not measurable)
  - Developmental software
- Software exists (Measurable)
  - COTS
  - GOTS
  - Prototype
  - NDI/Legacy
  - Experimental

# Observations

- **Software product maturity is value-neutral**
  - Mature software not better than immature software in some instances; must be interpreted in light of risk
    - *Web-pages*
    - *Proofs of concept*
- **Software can become senile**
  - Collective impact of changes overwhelm the architecture
  - Environment changes
  - Utility degrades
- **Level of understanding of context directly impacts risk and interpretation of maturity**
  - Poorly understood application environment or target makes risk assessment difficult



# Notional SW Maturity Lifecycle



# Candidate Characteristics

- Represent areas/dimensions affecting product maturity
- Must be considered both separately and as a group
- Weight of each characteristic may differ in any particular situation
- Must be evaluated against intended purpose

# Candidate Characteristics

- Understanding of the potential for latent defects within the product
- Understanding of the domain of product applicability
- Predictability of product behavior (within well-defined parameters)
- Product stability
- Product supportability
- Product pedigree

# Potential for Latent Defects

- Addresses the risk of undetected bugs
- Possible measures or information sources
  - Test regimen
  - Test coverage
  - History and trends of types/frequency of faults
  - Certifications and test packages

# Domain of Product Applicability

- Addresses risk of suitability of the product to the intended task
- Possible measures or information sources
  - Compatibility measures
  - Robustness (adaptability, scalability, portability, security, safety, integrity, etc.)
  - Availability and quality of design and maintenance documents
  - Certifications and test packages
  - Specific operational environment(s)
  - Limitations on product use

# Predictability of Product Behavior

- Addresses the risks associated with suitability of operational and functional quality
- Possible measures or information sources
  - Test regimen
  - Test coverage
  - History and trends of types/frequency of faults
  - MTBF
  - Availability
  - Recovery from faults
  - Compatibility measures
  - Accuracy
  - Completeness of features/functions definition

# Product Stability

- Addresses the risks associated with historic volatility that could re-emerge
- Possible measures or information sources
  - Release history and frequency
  - History and trends of types/frequency of faults
  - Obsolescence potential
  - Software aging characteristics

# Product Supportability

- Addresses the risks associated with continuing suitability of the product
- Possible measures or information sources
  - Availability of training
  - Availability of vendor/developer/consultant support
  - Recovery from faults
  - Mean time between failure
  - Availability and quality of design/maintenance documents
  - Dependency on events out of product control
  - Life expectancy
    - *First shipment date*
    - *End-of-life plans*
    - *Market share*
    - *Market trend*
    - *Rights granted on discontinuation of product*



# Product Pedigree

- Addresses the risks associated with the developers/sources for the product
- Possible measures or information sources
  - Installed base
  - Market share
  - Market trend
  - Maturity of underlying software technology(ies)
  - Customer references
  - Confidence in adherence to standards
  - History of upward compatibility

# Additional Factors

- Control over configuration/evolution
  - Does the acquisition need to determine when or how the product will change and the type of features that may be added or dropped?
- Predictability of evolution and obsolescence
  - Does the acquisition have a clear understanding of the direction and speed of product evolution and the time remaining in the product's likely supported life?
- Schedule
  - Does the acquisition understand when the product will be available or updated (such as availability of NDI or required product functionality)?
- Costs
  - Does the acquisition understand the full costs associated with the product, such as licensing, refresh, maintenance

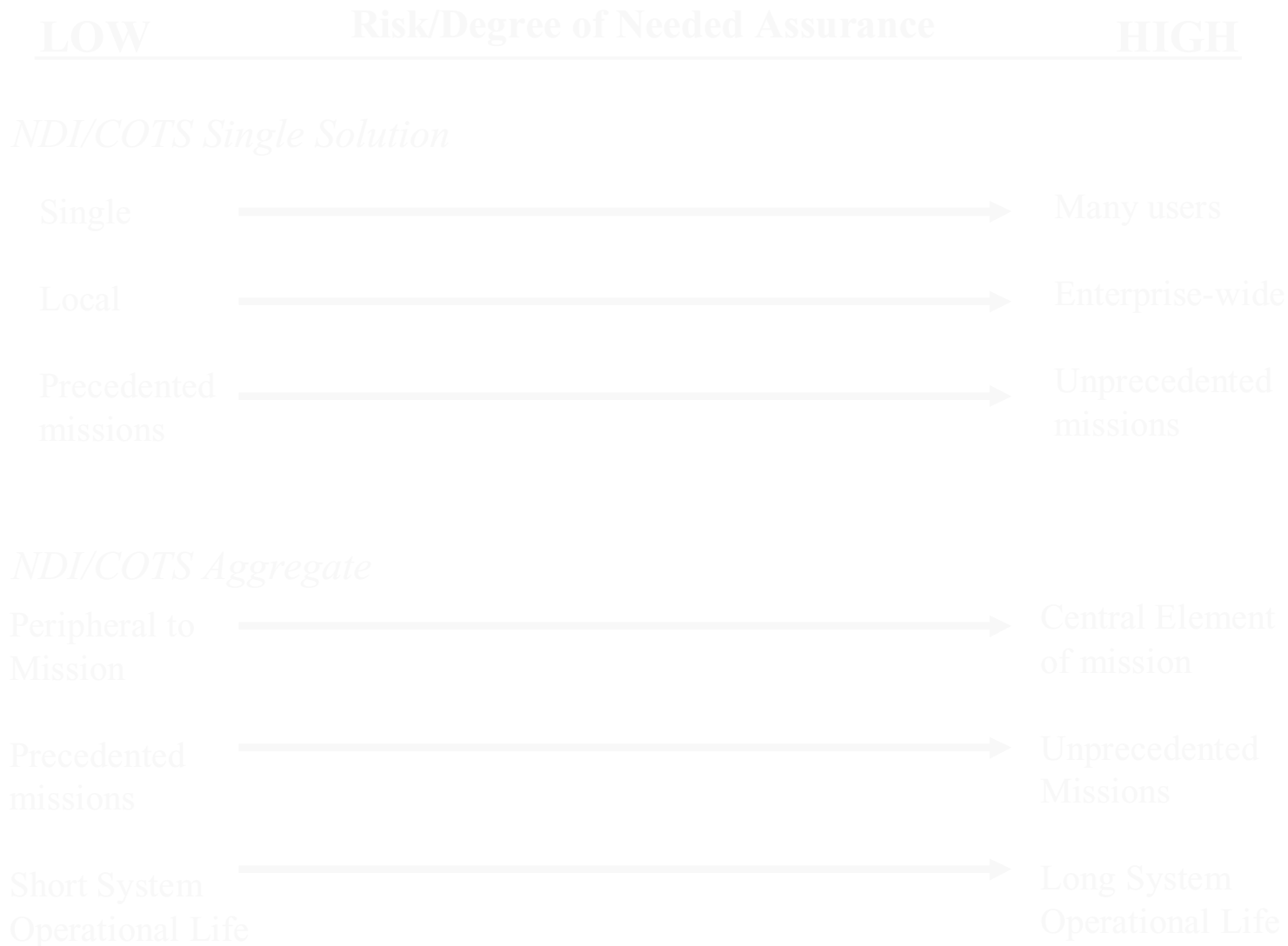
# Additional Factors - 2

- **Architecture**
  - Will the product require significant changes to an existing software architecture?
- **Operational Context**
  - Will the product fit the current or envisioned modes of operation, operational environment (e.g. platform) and process context?
- **Fitness for use**
  - Do the product characteristics meet the needs of the envisioned use (such as security, availability, and scalability)?
- **Modification of product**
  - Will there need to be modifications to the product that will prevent normal developer/vendor refresh?

# Additional Factors - 3

- **Release synchronization**
  - Will the vendor release schedule impact operations?
- **Pedigree of product developer**
  - Does the acquisition have confidence in the developer/vendor (including disclosure of subcontractors)?

# Context Impacts Risk



# Summary

- **Maturity can only be measured on existing software**
  - For source selection this means COTS, GOTS, NDI, prototype, experimental
- **Initial set of software product maturity characteristics defined**
- **Maturity evaluation complex - dependent on context and related factors**
- **Next steps**
  - Complete draft language for OSD Guidebook
  - Refine characteristics and measures
  - Continue to evaluate development maturity concepts

# Questions?

- **Contact Information**
  - [Rich.turner.ctr@osd.mil](mailto:Rich.turner.ctr@osd.mil)
  - 703.602.0851 x 124