NORTHROP GRUMMAN

DEFINING THE FUTURE

Reuse Readiness Levels: A Framework for Decision Making

NDIA Systems Engineering Conference
October 20-23, 2008

Steven Wong

Northrop Grumman Mission Systems

Topics

- Reuse and Maturity
- Measures of Maturity - Technology Readiness Levels
  - Background
  - Applicability to Software
  - Limitations
- Reuse Readiness Levels
  - Motivation
  - Background
    - SEI
    - NASA
  - Northrop Grumman Approach
    - Reuse Attributes
    - Decision Analysis Resolution Process
- Outcomes

To Reuse or Not to Reuse Software?

- "Good reuse " economizes time and money; ensures quality
  - Increased dependability
  - Compliance to standards
  - Accelerated development
  - Economies of Scale
  - Reduce product and process risk
- "Bad reuse " introduces risk resulting in cost and schedule growth
  - Incompatibility
  - Obsolescence
  - Breakage
  - Requirements differences
  - Unfamiliarity

How can one make an *a priori* distinction between good and bad reuse?

DoD 5000.2-R, Jan 4, 2001

NORTHROP GRUMMAN

7.5. -- Technology Maturity

Technology maturity shall measure the degree to which proposed critical technologies meet program objectives. Technology maturity is a principal element of program risk. A technology readiness assessment shall examine program concepts, technology requirements, and demonstrated technology capabilities to determine technological maturity**.**

The PM shall identify critical technologies via the work breakdown structure (WBS) (see 5.3.1). **Technology readiness assessments for critical technologies shall occur sufficiently prior to milestone decision points B and C to provide useful technology maturity information to the acquisition review process.**

The Component Science and Technology (S&T) Executive shall direct the technology readiness assessment and, for ACAT ID and ACAT IA programs, **submit the findings to the Deputy Under Secretary of Defense (S&T) (DUSD(S&T)) with a recommended technology readiness level (TRL) for each critical technology.** In cooperation with the Component S&T Executive and the program office, the DUSD(S&T) shall evaluate the technology readiness assessment and, if he/she concurs, forward findings to the OIPT leader and DAB**.** If the DUSD(S&T) does not concur with the technology readiness assessment findings, an independent technology readiness assessment, under the direction of the DUSD(S&T), shall be required**.**

A Definition

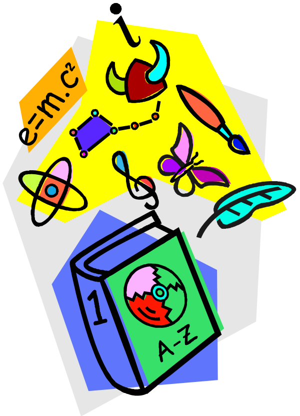
- *Technology Readiness Levels (TRL)* are used to assess the maturity of a practically-applied scientific/engineering invention (materials, components, methods, devices, etc.) prior to its incorporation into a system

- A method for assessing how much **<u>risk</u>** is potentially involved with adopting a technology

- TRLs assume that a technology is less suitable for immediate usage when it is newly invented or conceptualized

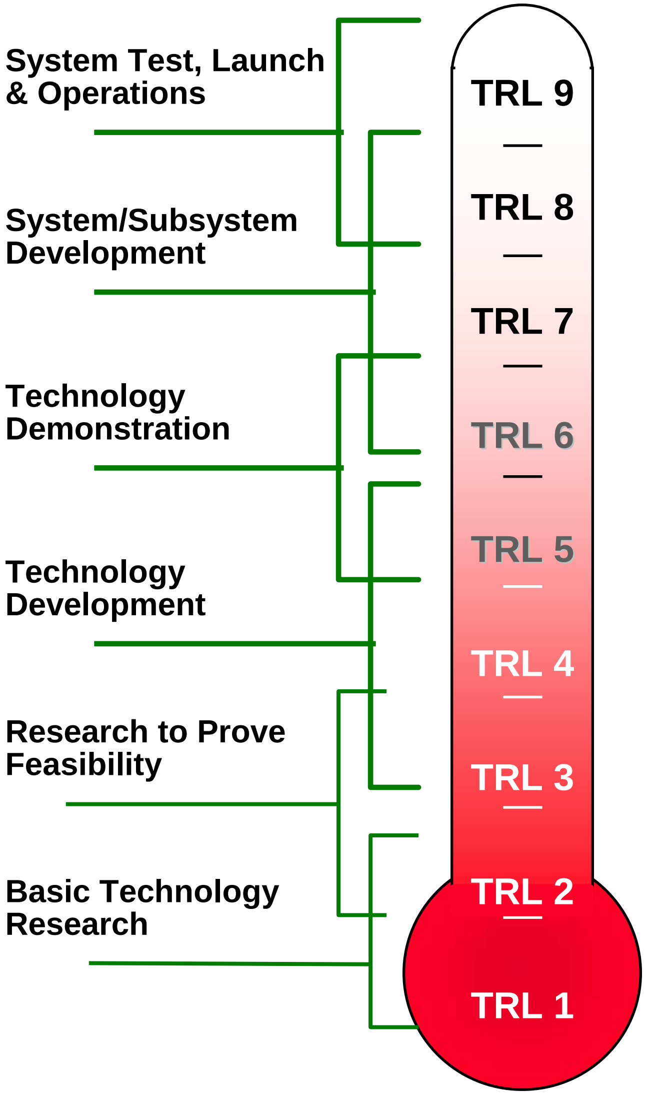
- A technology becomes sufficiently proven (i.e., mature) after being subjected to experimentation, refinement, and increasingly demonstrated and tested in a realistic environment

- Examples: Hardware TRL, Software TRL, Manufacturing TRL, Biomedical TRL



Technology Readiness Levels

NORTHROP GRUMMAN



9 - Actual system “flight proven” through successful mission operations

8 - Actual system completed and “flight qualified” through test and demonstration

7 - System prototype demonstration in a operational environment

6 - System/subsystem model or prototype demonstration in a relevant environment

5 - Component and/or breadboard validation in relevant environment

4 - Component and/or breadboard validation in laboratory environment

3 - Analytical and experimental critical function and/or characteristic proof-of-concept

2 - Technology concept and/or application formulated

1 - Basic principles observed and reported

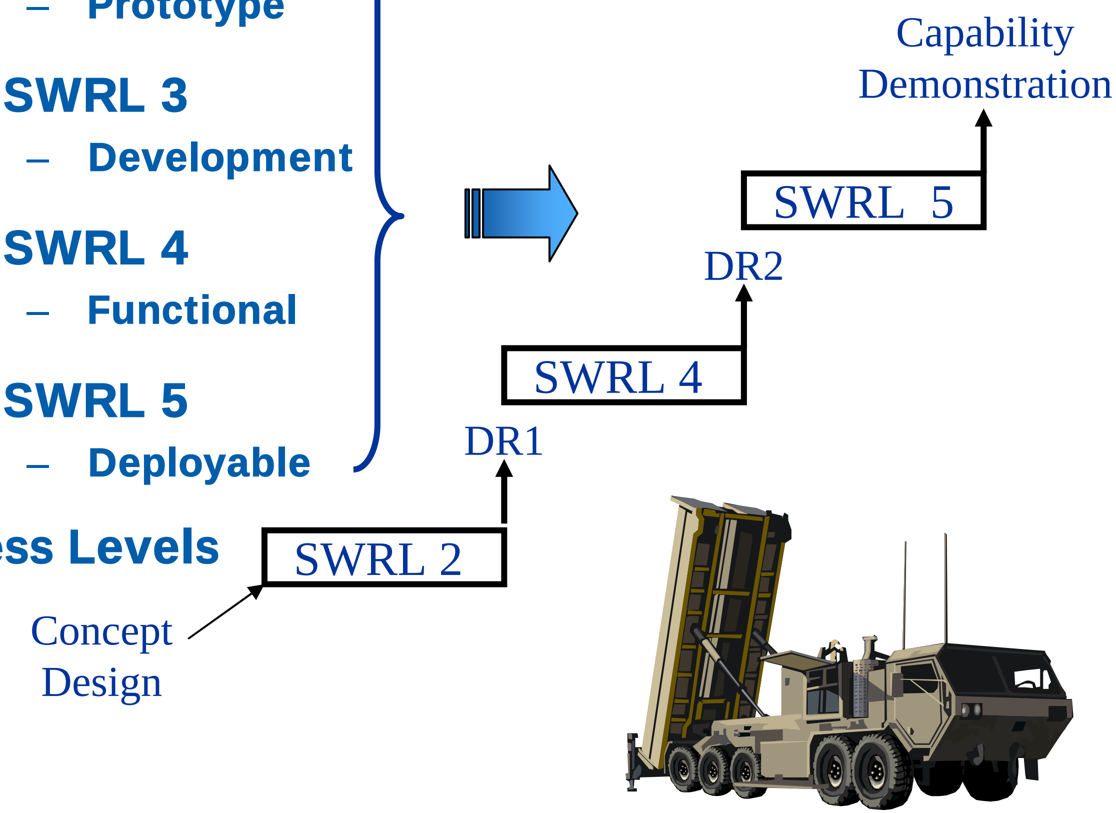
Software Readiness Levels (SWRL)
Missile Defense Agency (MDA)

Engineering Manufacturing Readiness Levels (Hardware)

- *EMRL 1*
  - *Breadboard*
- *EMRL 2*
  - *Prototype*
- *EMRL 3*
  - *Advanced development*
- *EMRL 4*
  - *Similar production*
- *EMRL 5*
  - *FRP*

- **SWRL 1**
  - **Concept**
- **SWRL 2**
  - **Prototype**
- **SWRL 3**
  - **Development**
- **SWRL 4**
  - **Functional**
- **SWRL 5**
  - **Deployable**

Software Readiness Levels



TRL Software Descriptions – DUSD(S&T) TRA Deskbook 2005

NORTHROP GRUMMAN

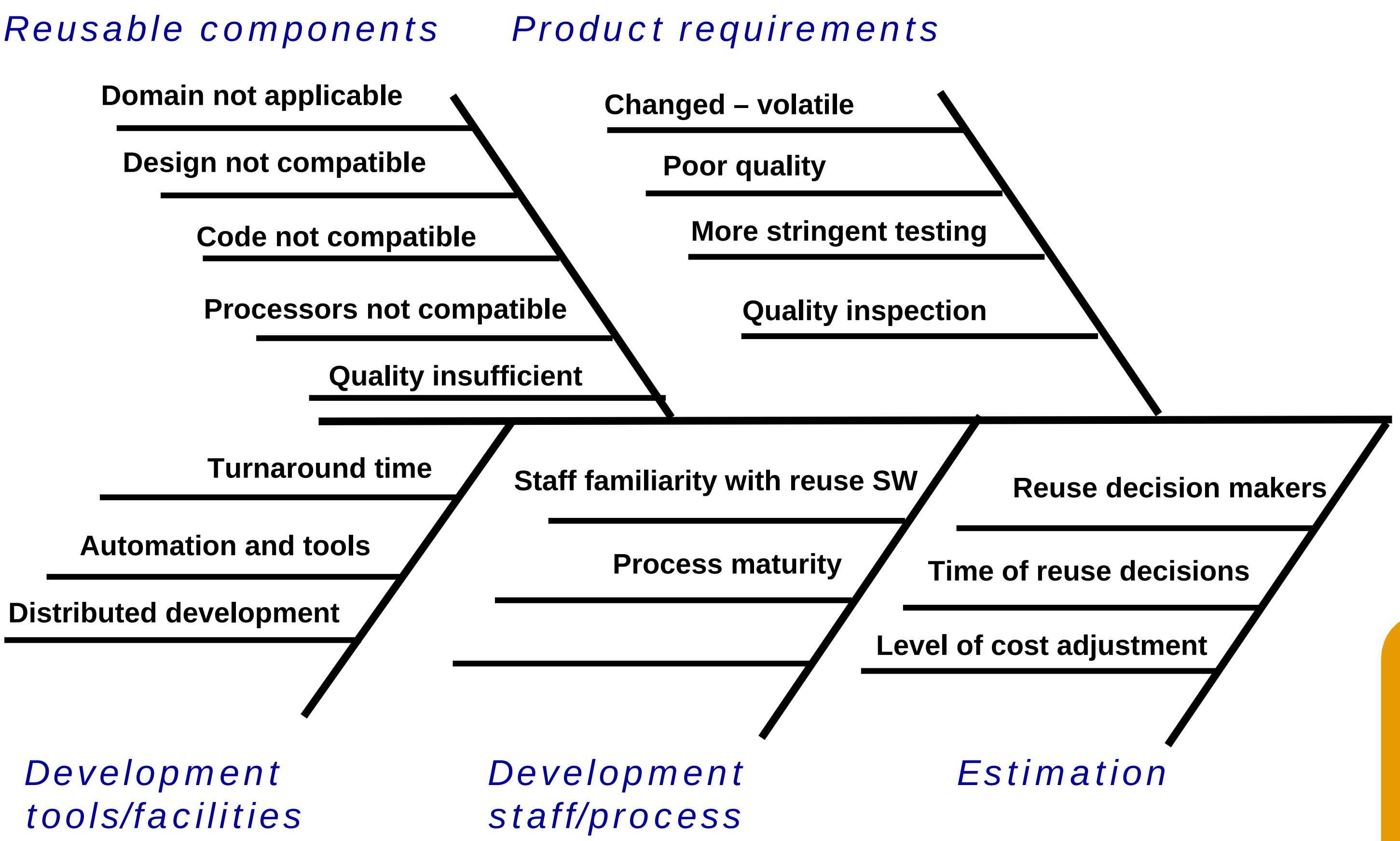
Technology Readiness Level	Software Description
1. Basic principles observed and reported	Lowest level of software technology readiness. A new software domain is being investigated by the basic research community. This level extends to the development of basic use, basic properties of software architecture, mathematical formulations, and general algorithms.
2. Technology concept and/or application formulated	Once basic principles are observed, practical applications can be invented. Applications are speculative, and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies using synthetic data.
3. Analytical and experimental critical function and/or characteristic proof of concept	Active R&D is initiated. The level at which scientific feasibility is demonstrated through analytical and laboratory studies. This level extends to the development of limited functionality environments to validate critical properties and analytical predictions using nonintegrated software components and partially representative data.
4. Module and/or subsystem validation in a laboratory environment (i.e., software Prototype development environment).	Basic software components are integrated to establish that they will work together. They are relatively primitive with regard to efficiency and robustness compared with the eventual system. Architecture development initiated to include interoperability, reliability, maintainability, extensibility, scalability, and security issues. Emulation with current/legacy elements as appropriate. Prototypes developed to demonstrate different aspects of eventual system.
5. Module and/or subsystem validation in a relevant Environment	Level at which software technology is ready to start integration with existing systems. The prototype implementations conform to target environment / interfaces. Experiments with realistic problems. Simulated interfaces to existing systems. System software architecture established. Algorithms run on a processor(s) with characteristics expected in the operational environment.
6. Module and/or subsystem validation in a relevant end-to-end environment)	Level at which the engineering feasibility of a software technology is demonstrated. This level extends to laboratory prototype implementations on full-scale realistic problems in which the software technology is partially integrated with existing hardware/software systems
7. System prototype demonstration in an operational high-fidelity environment	Level at which the program feasibility of a software technology is demonstrated. This level extends to operational environment prototype implementations where critical technical risk functionality is available for demonstration and a test in which the software technology is well integrated with operational hardware/software systems.
8. Actual system completed and mission qualified through test and demonstration in an operational environment	Level at which a software technology is fully integrated with operational hardware and software systems. Software development documentation is complete. All functionality tested in simulated and operational scenarios.
9. Actual system proven through successful mission-proven operational capabilities	Level at which a software technology is readily repeatable and reusable. The software based on the technology is fully integrated with operational hardware/software systems. All software documentation verified. Successful operational experience. Sustaining software engineering support in place. Actual system.

Software TRL Limitations

- Software differs from hardware in that taking an operational product and using it in a new context or system does not necessarily correlate to system success in performance or in achieving planned cost and schedule benefits
  - In some situations it may introduce more complications and problems than if the code was not reused

- TRLs inherently assume “good reuse”
  - Increased dependability
  - Reduce product and process risk
  - Accelerated development
- TRLs do not adequately address “bad reuse” or COTS/GOTS and OSS
  - Obsolescence
  - Breakage
  - Requirements and usage differences
  - Unfamiliarity

Software Reuse Root Cause Analysis
Six Sigma Project #1299



RRL Background
Software Engineering Institute (SEI)

- TRL for Non-Developmental Item Software (Smith 2004)
  - Requirements Satisfaction
    - Rates how well requirements, including functional (e.g., throughput, accuracy, latency) and non-functional (e.g., reliability, maintainability) are allocated to a given software product or technology to be satisfied by it.
    - Accounts for the number of requirements are satisfied as well as any provided functionality that is not required
  - Environmental Fidelity
    - Addresses how faithfully the development environment of the software asset has been demonstrated to operate in the target operational environment.
  - Product Criticality
    - The degree to which the target system is dependent upon, or inseparable from the product or technology.
  - Product Aging
    - The availability of the product over its lifespan relative to the requirements of the system under development
  - Product Maturity
    - Maturity of the software product or technology relative to three distinct modes/domains: COTS, GOTS, OSS

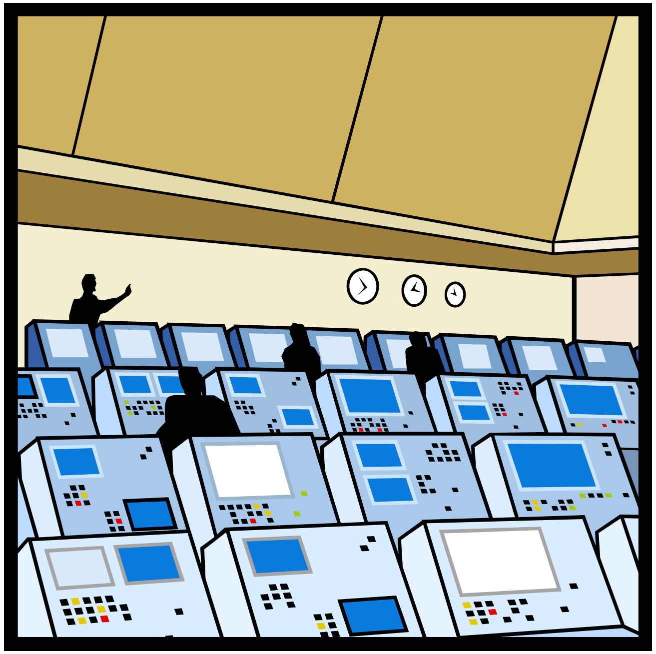
RRL Background
SEI (page 2)

- ImpACT Methodology for COTS (Smith 2005)
  - Importance
    - Criticality to the system; difficulty of effecting a work-around if the technology or product doesn't work (or isn't available)
  - Availability
    - The degree to which the product or technology is commercially available
  - Capability
    - The functional fit (or misfit) between the product or technology and the requirements of the system
  - Timeframe
    - A measure of how the lifecycle of the product or technology matches the lifecycle for the system. Will it be available when needed? Over the life of the system?

RRL Background
National Aeronautics and Space Agency (NASA)

NORTHROP GRUMMAN

- NASA Earth Science Data Systems (ESDS) Software Reuse Working Group (Wolfe, Marshall, 2007-2008)
  - Determine reuse maturity of software assets being prepared for reuse
  - Initially developed for the Earth science domain, applicable to general
  - Promote, facilitate, catalog and incentivize reuse
  - Reuse Enablement System
    - Web-based portal, Reuse metadata of an existing software asset
    - Aligned with familiar 1-9 scale TRL
- Topic Areas
  - Portability
  - Extensibility
  - Documentation
  - Support
  - Packaging
  - Intellectual Property
  - Standards Compliance
  - Verification and Testing
  - Modularity



NASA RRL Topic Areas and Rating Scale

NORTHROP GRUMMAN

	Portability	Extensibility	Documentation	Support	Packaging	Intellectual Property issues	Standards compliance	Verification & Testing	Modularity
Level 1	The software is not portable at any cost	No ability to extend or modify program behavior	Limited internal documentation available	No support available	Source code available	Potential owners and stakeholders of product have been identified.	Follows no particular standard	No testing performed	No designs for modularity or reuse
Level 2	Some parts of the software may be portable	Prohibitive costs and efforts need to modify or extend the system	Fully commented source code available	Known contact available		Relevant intellectual policies of potential owners and stakeholders have been reviewed.	Follows some parts of common standards and best practices	Software application formulated and unit testing performed	
Level 3	The software is only portable with significant costs	Can be extended with the input of considerable time and effort on par with recreating system separately	Basic external documentation available	Original developers provide proactive support	Detailed installation instructions available	Intellectual property agreements have been proposed to potential stakeholders.	Follows a company-wide standard for development and testing	Testing includes testing for error conditions and proof of handling input errors	Modularity at major system or subsystem level only
Level 4	The software may be portable at a reasonable cost	Can be modified and extended through configuration changes, minimal modification of source	Reference manual available	Latest updates or patches are available but not very frequently		Potential stakeholders have negotiated on intellectual property agreements and authorship issues.	Most components follow a complete, universal standard, but not validated	Software application demonstrated in a laboratory environment	
Level 5	The software is moderately portable	Consideration for future extensibility designed into system, extensibility approach somewhat defined	User manual available	Informal user community available	Software is easily configurable for different environments	Agreement and approval on authorship, attribution, and intellectual property issues has been obtained from stakeholders.	All components follow a universal standard, but only partially validated	Software application tested and validated in a laboratory environment	Partial segregation of generic and specific functionality
Level 6	The software is portable	Designed from the start to allow easy extensibility, provides many points of extensibility and a thorough and detailed extensibility plan	Tutorials available	Centralized support available		Authorship, attribution, and intellectual property statements have been drafted to reflect agreement among stakeholders on intellectual property and authorship.	Validated to follow a specific proprietary standard	Software application demonstrated in a relevant environment (Earth science related)	
Level 7	The software is highly portable	Proven to be extensible internally, code structured to provide loose coupling and high cohesion	Interface guide available	Organized/defined support by the original developer available	OS detect and auto-build for supported platforms	Authorship and intellectual property statements included in product prototype.	Validated to comply to a specific open standard	Software application tested and validated in a relevant environment (Earth science related)	Clear delineations of specific and reusable components
Level 8		Proven extensibility on a major external program, provides a clear plan for modifying and extending features	Extension guide and/or Design/Development guide available	Support by organization available		Manifestation of authorship, attribution, and intellectual property statements reviewed in product prototype before product release.	Proven by validation to comply with a "gold" standard	Software application "qualified" through test and demonstration (meets requirements) and successfully delivered to the Earth science environment	
Level 9	The software is completely portable	Proven extensibility in multiple scenarios, provides specific documentation and features to build extensions	Full software lifecycle engineering design documentation available	Large user community with well-defined support available	GUI installation environment provided	Reviewed authorship, attribution, and intellectual property statements packaged with product for release.	"Gold" standard compliance of entire system and development, independently validated	Actual software application tested and validated through successful use of application output	All functions and data encapsulated into objects or accessible through web service interfaces

Northrop Grumman (NGC) Reuse Readiness Level Framework

NORTHROP GRUMMAN

- NGC is developing *Reuse Readiness Levels* (RRL) as a decision framework to evaluate the technical viability of leveraging existing software
  - Merges the TRL concept with NGC's Decision Analysis and Resolution (DAR) process
    - Aligned with the 1-9 ascending TRL scale
    - DAR
      - Reduces subjectivity, increases rigor and consistency
      - Encourages disciplined objective thinking and stakeholder buy-in via evidence
      - Ensures best possible solutions for high risk decisions
      - Avoids premature commitment to a point design
  - Flexible – fits all situations
    - Multi-attribute / multivariate considered
    - DAR allows tailoring
    - Applicable to product line, non-product line, COTS, GOTS, NDI, OSS, etc.

Results in well-reasoned, timely software reuse decisions and better software estimates and plans

NGC Reuse Readiness Attributes

NORTHROP GRUMMAN

- Resources
  - Supporting processes and resources
  - Software familiarity
  - Developer experience
- Readability
  - Quantity and level of documentation
  - Accuracy and completeness of documentation
- Usability
  - Configurability, Openness and Modularity
  - Extensibility
  - Scalability
  - Well-defined and stable interfaces
- Maturity
  - Product life cycle stage
  - Maintenance
- Compatibility
  - Platform compatibility
  - Version compatibility
  - Language compatibility
- Tailoring / Rework
  - Restructuring / Re-factoring
  - Re-engineering
  - Re-implementation
  - Re-integration and Re-test
- Transportability
  - Architecture / design synchronization
  - Percentage of translation to new context
  - Index of new requirements incorporation

RRL NGC – Attributes (1 of 3)

#	Category	Attribute	Description
1	Resources	Supporting Processes and Resources	The consonance of the development methods and activities to the integration of the reuse software in the new context/system as well as the accessibility and availability of expertise related to the reuse software (either internal or external to the organization).
2	Resources	Software Familiarity	The level of understanding and practice that the development team has in working with the reuse software.
3	Resources	Developer Experience	The knowledge, skill, proficiency and expertise of the development team within the system domain.
4	Readability	Quantity and Level of Documentation	The amount and the detail of available descriptions of the software such as: annotation in the code, reference manuals, style guides, developer user guides, use cases, etc.
5	Readability	Accuracy and Completeness of Documentation	The degree to which the reuse software documentation is comprehensive, usable and reliably describes and explains the product.
6	Usability	Configurability, Openness and Modularity	The extent to which the reuse software may be added, upgraded and have its components replaced; as well as the efficient separation of system concerns realized through the logical boundaries between components.
7	Usability	Extensibility	The ability of the system to accommodate future growth either through the addition of new functionality or through the modification of existing functionality while minimizing the impact to other existing system functions or infrastructure.
8	Usability	Scalability	The degree to which the design of the reuse software handles increasing amounts of work, data, throughput, quantities, resources, etc. with graceful or no degradation in performance.

RRL NGC – Attributes (2 of 3)

#	Category	Attribute	Description
9	Usability	Well-defined and Stable Interfaces	The clarity, understandability and integrity of the reuse software (internal and external) interfaces as well as the robustness of the interfaces under changing, stressing or anomalous conditions
10	Maturity	Product Life Cycle Stage	The current point in the reuse software's evolution (ranging from "bleeding edge" new to obsolete) and the degree to which it has been tried, tested and proven in a working system. Factors to consider: usage and acceptance in the domain and the Industry
11	Maturity	Maintenance	The required resources to upkeep of the reuse software for correcting faults and keeping it operational. Factors to consider: Software Problem Report history, number and frequency of software patches, etc.
12	Compatibility	Platform Compatibility	The degree to which the original hardware architecture and software framework on which reuse software runs is similar or complimentary to the new context/system. Factors to consider: computer architecture, operating system, graphical user interface, etc.
13	Compatibility	Version Compatibility	The level at which the reuse software behaves in the intended and expected manner when it interacts with the other software components, products, tools, environments and platforms in the new context/system. Factors to consider: rate of change/upgrades of underlying products, frequency of synchronization points, etc.
14	Compatibility	Language Compatibility	The extent to which the programming set of instructions of the reuse software requires translation, reimplementation, or re-compilation in order to work in the new context/system

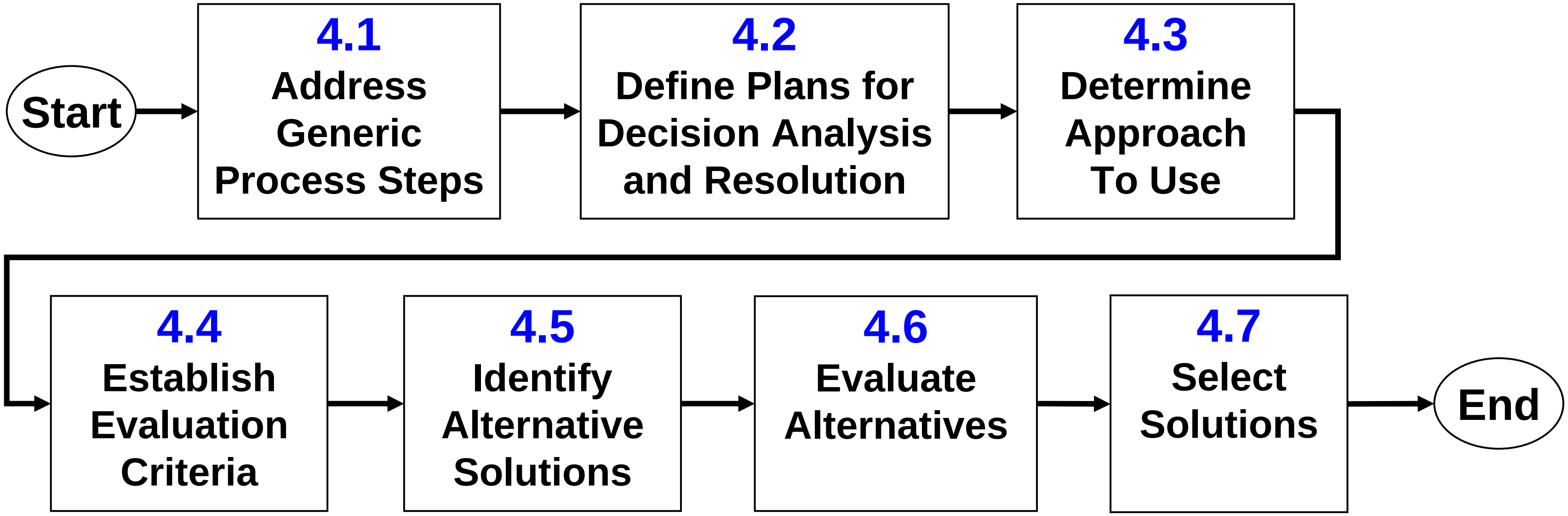
RRL NGC – Attributes (3 of 3)

#	Category	Attribute	Description
15	Tailoring / Rework	Restructuring / Re-factoring	The extent to which the existing software needs to be cleaned up - i.e.; improve its understandability; remove extraneous (dead) code, make the internal structure and design more efficient, maintainable and amenable to change, etc.
16	Tailoring / Rework	Re-engineering	The amount of reverse engineering or learning required to modify the design for integration in the new context/system.
17	Tailoring / Rework	Re-implementation	The amount of adaptation of the existing code and/or the addition of new code to meet the objectives and environment of the new context/system
18	Tailoring / Rework	Re-integration and Re-test	The effort to combine the existing software into the new context/system and verify that resulting product functions within performance, reliability, and other criteria in the new system/context
19	Transportability	Architecture/Design Synchronization	The degree of similarity of the structure in which the reusable software will interact in the new context/system. Factors to consider: reuse of an entire product or functional components; control mechanisms, data exchange, logical dependencies
20	Transportability	Percentage of Translation to New Context	The percentage change in the behavior, conditions and/or constraints in which the reuse software will operate in the new context/system. Factors to consider: operational scenarios, operational threads, use cases, etc.
21	Transportability	Index of New Requirements Incorporation	The ratio of component level requirements allocated to the reuse software that are new relative to a normalized measure of the requirements that are already fully and partially satisfied by the software

Comparison of Reuse Attributes

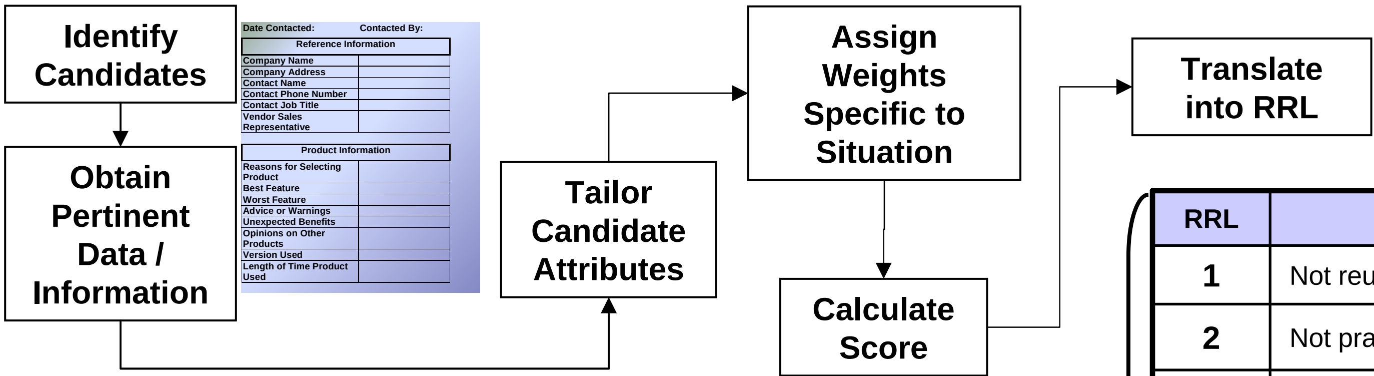
NGC RRL	NASA RRL	Army SW TRL	SEI NDI
Resources			
Supporting Processes and Resources	Support	Development Process	
Software Familiarity			
Developer Experience			
Readability			
Accuracy and Completeness of Documentation	Documentation	Previous System Documents / Code	
Level of Documentation			
Usability			
Open Architecture / Modularity	Modularity		
Configurability and Openness	Portability		
Extensibility across Platforms	Extensibility		
Scalability			
Well-defined and Stable Interfaces	Standards Compliance		
Compatibility			
Platform Compatibility	Packaging	Development Environment	Environment Fidelity
Version Compatibility		Test (Verify) Environment	
Language Compatibility			
Maturity			
Years in Operation	Verification and Testing	Technology Prototyped/ Used Existing System	
Maintenance		Open Problem Reports	Maturity
Upgrades / Technology Insertion			
Rework			
Restructuring/Refactoring		Change To Code	
Re-engineering			
Re-implementation			
Re-integration and Re-test			
Transportability			
Number of Contexts/Instantiations in which reused		Studies / Test Use Results	
Architecture/Design Synchronization			Technology Critical
Percentage of Translation to New Context			
Other			
Index of New Requirements Incorporation		Precision / Performance	Requirements (Functional and Non-Functional)
	Intellectual Property		Availability
		Safety / Security	

Standard Process Manual 944 – Decision Analysis and Resolution



- Sample Methodologies
  - Trade Study
  - Simple Multi-Attribute Rating Technique (SMART)
  - Analytical Hierarchy Process (AHP)
  - Pay-off table with application of an analysis technique (MiniMax, Expected Value, MaxiMax, Minimum Regret, etc.)
  - Decision Trees and Influence Diagrams
  - Simulation
  - Group Techniques (e.g., Delphi)

Reuse Readiness Levels – NGC Approach



Sensitivity analysis: Green = 1st highest score, Yellow = 2nd highest score, Red = 3rd highest score

No.	Evaluation Criteria	Evaluation Method	Scoring Function	Weight (%)	Reuse Asset 1 Score	Reuse Asset 1 Weighted Score	COTS A Score	COTS A Weighted Score
1	Supporting Processes and Resources			2.0%	2.0	0.04	4.2	0.08
2	Software Familiarity			2.0%	9.8	0.20	4.8	0.10
3	Developer Experience			10.0%	1.6	0.16	3.5	0.35
4	Quantity and Level of Documentation			1.0%	0.1	0.00	8.2	0.08
5	Accuracy and Completeness of Documentation			1.0%	1.3	0.01	3.0	0.03
6	Configurability, Openness and Modularity			17.0%	4.0	0.67	5.2	0.89
7	Extensibility			8.0%	7.8	0.62	5.5	0.44
8	Scalability			5.0%	8.1	0.41	6.5	0.33
9	Well-defined and Stable Interfaces			12.0%	1.8	0.22	1.0	0.12
10	Product Life Cycle Stage			5.0%	2.3	0.12	1.7	0.09
11	Maintenance			3.0%	7.6	0.23	9.0	0.27
12	Platform Compatibility			0.0%	0.0	0.00	0.0	0.00
13	Version Compatibility			7.0%	5.1	0.36	2.1	0.15
14	Language Compatibility			0.0%	4.3	0.00	2.0	0.00
15	Restructuring / Re-factoring			6.0%	2.2	0.13	3.0	0.18
16	Re-engineering			0.0%	0.0	0.00	0.0	0.00
17	Re-implementation			4.0%	9.1	0.36	3.0	0.12
18	Re-integration and Re-test			4.0%	2.0	0.08	2.0	0.08
19	Architecture/Design Synchronization			8.0%	1.8	0.15	1.5	0.12
20	Percentage of Translation to New Context			0.0%	4.0	0.00	3.5	0.00
21	Index of New Requirements Incorporation			5.0%	3.8	0.19	2.7	0.14
TOTAL:				100.0%		3.94		3.55

RRL	Description
1	Not reusable in the given context
2	Not practical to reuse in the given context
3	Conceptual reuse possible; significant risk to implementation
4	Reuse with 6 or more attributes of concern - assume substantial cost and risk
5	Reuse with 3 to 6 attributes of concern - assume a reasonable amount of cost and risk
6	Reuse with 1 to 3 attributes of concern - potential for some cost and risk
7	High reuse with minimum cost and risk
8	Demonstrated reuse by multiple adopters
9	Proven serial reuse by multiple skills and experience level

Outcomes

- Decisions / Assessments
  - Technical viability and rank order of reuse candidate software
  - Justification <u>not</u> to reuse
  - Investment in maturing a potential reuse asset
  - Use as a component to determine an overall for Software TRL of a critical technology
- Insight
  - Understanding of the level risk associated with incorporating software technologies into a system or solution
  - Sensitivity of driving factors that affect reuse success
  - Degree of modification (effort) required to reuse the product
    - Improved size and cost estimates

NORTHROP GRUMMAN

DEFINING THE FUTURE