Operating Environments (11)	
Avionics	Software developed for aircraft avionics systems like those found on aircraft, helicopters, etc.
Business	Software developed for systems that process business information like financial and management information systems.
Manned Space	Software developed for manned space applications like the space shuttle or station.
Manned Ground	Software developed for ground-based systems like artillery and tactical information centers.
Military Mobile	Software developed for ground-based mobile systems like mobile command stations (in trucks or vans), tanks and transports of any kind (trucks, trains, etc.).
Missile and Unmanned Airborne	Software developed for missiles or unmanned airborne systems like an air-to-air missile or remotely piloted vehicle.
Shipboard	Software developed for sea based systems like aircraft carriers, destroyers, freighters, submarines and tankers.
Telecommunications	Software developed for communications systems that transmit voice, data, video or other information.
Unmanned Ground	Software developed for unmanned ground-based systems like reconnaissance vehicles.
Unmanned Space	Software developed for unmanned space systems used for planetary exploration, imagery or weather prediction.
Web	Software developed for applications that run and utilize the Internet.

Copyright 2009 University of Southern California

Application Domains (24)	
Bus	Software used to manage infrastructure services (attitude and articulation control, commanding, etc.) for manned or unmanned space applications.
Command and Control	Software used for network monitoring, network control and switching, sensor control, signal/telemetry processing, message processing, data reduction/analysis, mission control and command processing in any of the operating environments described above.
Communications	Software used for transmission and receipt of voice, data, digital and video information within or to/from operating environments described above.
Controls and Displays	Software that provides the interface between the user and system within any of the operational environments described above. For example, software that controls the Heads-Up Display (HUD) in an aircraft provides the pilot with a virtual environment during engagement.
Database	Software used to collect, manage and search information stored as files or databases in any of the operating environments described above.
Electronics Warfare	Software used to deny the opponent an advantage in the electromagnetic spectrum through its attack, protection and support components.
Executive	Systems software used to control the hardware and operating environment within which software applications execute within any of the operating environments described above. Executive software is typically specially developed to control the environment because the operating system or standard platform software cannot accommodate specialized run-time requirements.
Information Assurance	Software used to protect and defend information systems to ensure confidentiality, integrity, authentication, availability and non-repudiation within any of the operating environments described above.
Logistics	Software used to manage the flow of goods, information and other resources involved in maintaining and sustaining systems operational in the field.
Maintenance and Diagnostics	Software used to perform maintenance functions including diagnostics and diagnosis of problems.
Mission Management	Software used to support mission management activities in any of the operating environments described above such as the command and control of systems aboard an aircraft.
Mission Planning	Software used to support mission planning activities in any of the operating environments described above including those for scenario generation, feasibility analysis, route planning, and image/map manipulation.
Payload	Software used to manage payload functions (experiment control, sensor management, etc.) for manned or unmanned space applications.
Platform	Software used to bind, load, schedule and run applications on the selected computer hardware. Typically such software includes the operating system, linker/loader, middleware and utilities required operationally for any of the operating environments described above. In some cases, such platforms provide services like database management and security as part of the platform offerings.

Copyright 2009 University of Southern California

Application Domains (24)	
Process Control	Software that provides closed-loop feedback controls for applications like attitude control, propulsion, etc. that run in real-time in industrial and military environments.
Radar	Software used to automate radar functions in any of the operating environments described above.
Signal Processing	Software used to enhance, transform, filter, convert or compress signal data typically in real-time in any of the operating environments described above.
Simulation and Modeling	Software used to evaluate scenarios and assess empirical relationships that exist between models of physical processes, complex systems or other phenomena that may have complex relationships in any of the operating environments described above.
Situation Awareness	Software used to provide decision-makers with a perception of their environment within a volume of time and space within any of the complex, dynamic operating environments described above.
Sonar	Software use to automate sonar functions in any of the operating environments described above.
Test and Evaluation	Software used to support the test and evaluation of software within software, hardware/software and pseudo-operational environment like test labs within any of the operating environments described above. Activities performed include test case generation, test data recording, test data reduction/analysis and test driver/tool development.
Tool and Tool Systems	Software packages and/or integrated development environment that are used for analysis, design, construction and test of operational software for the operating environments described above.
Training	Software used to support education and training of users within any of the operating environments described above. Activities performed include computer-based training and computer-aided instruction within the operational and/or dedicated training systems.
Weapons Delivery	Software used to target, control and command weapons within any of the operating environments described above.

Copyright 2009 University of Southern California

Table 1: Software Size Types

Size Type	Description
New	Software developed entirely from scratch using no existing software artifacts.
Adapted	Pre-existing software that is used as-is (Reused) or changed (Modified).
Reused	Pre-existing software that is not changed with the adaption parameter settings: • Design Modification % (DM) = 0% • Code Modification % (CM) = 0%
Modified	Pre-existing software that is modified for use by making design, code and/or test changes: • Code Modification % (CM) > 0%
Equivalent	A relative measure of the work done to produce software compared to the code-counted size of the delivered software. It adjusts the size of software relative to developing it all new. This is also sometimes called *Effective* size.
Generated	Software created with automated source code generators. The code to include for equivalent size may consist of the generator statements directly produced by the programmer, or the 3GL generated statements produced by the automated tools.
Converted	Software that is converted between languages using automated translators.
Rehosted	Software that is adapted for installation on different platform(s)

Table 2: Adaptation Adjustment Factor (AAF) Parameters

Adaptation Parameter	Description
DM (Design Modified %)	The percentage of the adapted software's design which is modified in order to adapt it to the new objectives and environment.
CM (Code Modified %)	The percentage of the adapted software's code which is modified in order to adapt it to the new objectives and environment.
IM (Integration % Required)	The percentage of effort required to integrate the adapted software into an overall product and to test the resulting product as compared to the normal amount of integration and test effort for software of comparable size.

Copyright 2009 University of Southern California

Table 3: Software Understanding (SU) Rating Scale

	Very Low	Low	Nominal	High	Very High
Structure	Very low cohesion, high coupling, spaghetti code.	Moderately low cohesion, high coupling.	Reasonably well-structured; some weak areas.	High cohesion, low coupling.	Strong modularity, information hiding in data / control structures.
Application Clarity	No match between program and application world-views.	Some correlation between program and application.	Moderate correlation between program and application.	Good correlation between program and application.	Clear match between program and application world-views.
Self-Descriptive -ness	Obscure code; documentation missing, obscure or obsolete	Some code commentary and headers; some useful documentation .	Moderate level of code commentary, headers, documentation.	Good code commentary and headers; useful documentation; some weak areas.	Self-descriptive code; documentation up-to-date, well-organized, with design rationale.
SU Increment to ESLOC	50	40	30	20	10

Table 4: Programmer Unfamiliarity (UNFM) Rating Scale

UNFM Increment	Level of Unfamiliarity
0.0	Completely familiar
0.2	Mostly familiar
0.4	Somewhat familiar
0.6	Considerably familiar
0.8	Mostly unfamiliar
1.0	Completely unfamiliar

Copyright 2009 University of Southern California

Sizing Guidance

Table 5: Expanded Size Type Descriptions

Size Type	Description
Modified	Pre-existing code that is changed.
Major Mod	More than 25% modified;
Minor Mod	Less than 25% modified
Reused	Pre-existing code that is not changed.
As-is	Code carried over from one build/release to another
Tailored	Reused code that is designed for reuse and not modified in any way. It is tailored (typically through parametric settings) and configured for reuse.
Generated	Software created with automated source code generators using different technologies
As-is	Generated code that is not changed.
Optimized	Generated code that is changed for optimization purposes.
Converted	Software that is converted between languages with automated translators
As-is	Converted code that is not changed.
Optimized	Converted code that is changed for optimization purposes.
Rehosted	Software that is adapted for installation on different platform(s)
Major	?
Minor	?

Copyright 2009 University of Southern California

Adapted Software Parameters: _____

	DM	CM	IM	SU	UNFM
Modified					
Major Mod					
Minor Mod					
Reused					
As-is					
Tailored					
Generated					
As-is					
Optimized					
Converted					
As-is					
Optimized					
Rehosted					
Major					
Minor					

Adapted Software Parameters: _____

	DM	CM	IM	SU	UNFM
Modified					
Major Mod					
Minor Mod					
Reused					
As-is					
Tailored					
Generated					
As-is					
Optimized					
Converted					
As-is					
Optimized					
Rehosted					
Major					
Minor					

Copyright 2009 University of Southern California