

## **Guidelines for Modeling Automatic Software Code Generation (Uncalibrated)**

Automatic software code generation is a software development methodology where software code is generated via a workstation. Normally very little, if any manual programming is accomplished when this process is utilized. Use of this methodology requires sophisticated software tools/workstations (when compared to the PRICE S CPLX1-software tool adjustment table). This methodology is generally used in business system software development and screen generation.

Benefits of automatic software code generation are: Reduced development costs, and shorter development schedules. The major disadvantage of code generation is that the number of software instructions generated may be twice the size of a manually programmed software program. This disadvantage prohibits the use of this methodology for many military programs due to size limitations.

Modeling code generation with PRICE S requires adjustments to CSCI and/or CSC project level inputs, language input parameters, and global input parameters. PRICE S CSCI/CSC and language input parameters that require adjustment or at least **evaluation** are:

- INTEGI
- INTEGE (If additional integration is required)
- CPLX1
- NEWC
- APPL
- PROFAC
- SLOC

Internal integration input parameter is typically very low (range 0.001 to 0.1). Software workstations typically provide the capability to define how the software will be linked and integrated to form the CSCI. In many cases the only test effort required for screen generated software is to validate input fields.

Guidelines for external integration are 0.1 to 0.5 (**Note:** This range does not mean acceptable values for INTEGE must be within this range). This parameter should be evaluated with respect to the type of software that is being generated. For screen generated software, INTEGE should be low. For other types of generated software this value could approach 0.5 (Note: Size and application values will drive system software integration costs). This input parameter should be evaluated with respect to the amount of effort required for system software integration.

CPLX1 - Current guideline tables are applicable for this parameter. When evaluating this parameter, users must be careful not to take credit for code generation twice. For example, if PROFAC, was increased then credit for tools should not be taken for CPLX1.

NEWC - The percentage entered for each functional mix new code input should range from 0.001 to 0.25. Where 0.001 indicates reflects effort for generating the software with no manual changes made to interfaces, protocols, etc. The typical input for new code will be around 0.10, which reflects a small amount of manual code changes.

APPL - The typical range for code generated software is 2.3 to 5. The lower application values reflect screen generated software, while the higher values reflect not only screens, but also data base storage, and some on-line communications.

PROFACs for this type of software development has ranged from 8 to 12. These values are from predominately business software applications, and screen generations.

SLOC - Values for this parameter can be either pseudo code or expected size of the generated software. **If pseudo code is entered for SLOC, then all above guidelines should be disregarded. Normal PRICE S guidelines are applicable for pseudo code and 4th Generation should be selected as Language.**

If the expected size of the code generated software is entered for SLOC, then above guidelines are applicable.

Global input parameters requiring adjustments are: Elimination of non-applicable software schedule phases, and adjustment/evaluation of Design Engineering, Systems Engineering, Quality Assurance, and Configuration Management.

Quality assurance and configuration management cost multipliers can be reduced to 0.50. Typically these functions are only five percent of the code generated software costs. Evaluation of the organizational accounting practices and functional personnel assigned to the task should be evaluated before eliminating or adjusting design engineering and/or system engineering. This is a murky area, in that many times the systems analyst/engineer, design engineer and programmer are performed by the same person. Extreme care should be taken to ensure that the cost multipliers correspond to the organization performing the code generation to avoid double counting the benefits of code generation.

**Note: In cases where the user has calibrated the PRICE S Model to code generation, calibration methods should override all of the above guidelines.**