

CHOICE OF SOFTWARE LANGUAGE

ERIC GARRIGUE VESELY
Vesely & Associates
Principal

Hundreds of computer languages and dialects are available to users for programming their application. A good choice saves money, time, and frustration; a bad choice can mean the complete failure of the computer installation.

The major languages include ALGOL, APL, BASIC, COBOL, FORTRAN, P/L I, RPG, and each computer's assembly language.

Each language has strengths and weaknesses. These must be carefully analyzed in the full spectrum of program design through program maintenance to arrive at the total cost of using that language. The language costing the least is the best choice.

Analysis criteria includes computer configuration, application, portability, conversion, compiler speed and aids, debugging tools, execution speed, self-documentation features, maintainability, programmer availability, other languages and software interfaces.

The following text provides tabular evaluation techniques and explanations.

CHOIX DU LANGUAGE DE SOFTWARE

Des centaines de langages de calculateur sont à la disposition des utilisateurs pour programmer leur application. Un bon choix économise de l'argent, du temps, et de la frustration. Un mauvais choix peut entraîner l'échec complet de l'installation de calculateur.

Les langages les plus importants comprennent ALGOL, APL, BASIC, P/L I, RPG et le langage d'assemblage de chaque ordinateur.

Chaque langage a ses points forts et ses points faibles. Ceux-ci doivent être examinés soigneusement en considérant toutes les phases de programmation depuis le projet du programme jusqu'au programme de maintien pour déterminer le coût total de l'utilisation de ce langage. Le meilleur choix est le langage le moins coûteux.

Les critères de l'examen comprennent la configuration, l'application, la mobilité, la conversion, la vitesse du compilateur et de ses aides, les outils de mise en point, la vitesse d'exécution, les dispositifs d'auto-documentation, la maintenance, la disponibilité du programmeur, d'autres langages et les interfaces du software.

Le texte suivant présente sous forme de tableau les explications et les techniques d'évaluation.

AUSWAHL EINER SOFTWARE-SPRACHE

Es gibt hunderte von Computersprachen und -dialekte, die die Benutzer zum Programmieren ihrer Anwendung gebrauchen können. Eine gute Wahl spart Geld, Zeit und Enttäuschungen; eine schlechte Wahl kann den vollkommenen Ausfall des Computers herbeiführen.

Unter den wichtigsten Sprachen sind ALGOL, APL, BASIC, COBOL, FORTRAN, P/L I, RPG und die Montagesprache jedes Computers.

Jede Sprache hat ihre Stärken und Schwächen. Diese müssen genau untersucht werden, und zwar vom Gesichtspunkt des Programmmentwurfes bis zur Programminstandhaltung um die Totalkosten der Anwendung der Sprache zu erhalten. Die Sprache, die am wenigsten kostet, ist die beste Wahl.

Zu den Untersuchungskriterien gehören die Computerform, die Anwendung, Beweglichkeit, Umwandlung, Compiliergeschwindigkeit und Hilfsmittel, Fehlerbeseitigungswerkzeuge, Ausführungsgeschwindigkeit, Selbstdokumentationseigenschaften, Instandhaltbarkeit, Programmierer-Verfügbarkeit, andere Sprachen und Software Übergangsflächen.

Der nachstehende Text beinhaltet tabellarisch dargestellte Berechnungstechniken und Erklärungen.

1. COMPUTER CONFIGURATION

The first choice consideration is the host computer configuration. Obviously, if a language is not available on your selected computer then it is eliminated. The tabular form is:

Computer Configuration Cost	1.3	Cobol ^{1.1}	SPL ^{1.2}
Compiler Price	1.4	_____	_____
Annual Maintenance	1.5	_____	_____
Price X _____ years	1.6	_____	_____
Total Cost		_____	_____
Est. No. of Compiles	1.7	_____	_____
Cost per Compile		_____	_____

1.1. The languages are selected for illustration; they could easily be APL, BASIC, FORTRAN, SPL, etc.

1.2. Hewlett-Packard 3000 assembler

1.3. The cost of the minimum configuration to run compiler

1.4. Purchase price of compiler if any (note: many unbundled manufacturers charge for compilers)

1.5. Annual maintenance cost if any. (If language is purchased from a third party then include estimated internal programming cost of providing support)

1.6. Estimated life of application

1.7. The first major guess! Each language for different thresholds for achieving syntactically correct compilers. Use internal knowledge, Auerbach, Data-pro, consultants, etc.

2. APPLICATION

The application characteristics provide a difficulty factor for each language

APPLICATION	CHAR ⁸	COBOL	SPL
BUS/SCI		2.1	
BATCH/INTER	_____	_____	_____

2.1. Enter basic characteristic of application (B=business/S=science; B=batch/I=interactive)

2.2. Enter 1 if language manuals devote chapters to processing your application character

Enter 2 if there is an appendix

Enter 3 if there is a reference

Enter 4 if there is no mention

3. PORTABILITY

Portability is the capability of moving an application programmed in a specific language from your current computer to another manufacturer or even to another model of your current manufacturer with a minimum¹⁰ of reprogramming. The table presents both a difficulty factor and a need factor (ie) your need for portability.

PORTABILITY	NEED ^{3.1}	COBOL	SPL
		3.2	

3.1. Minimum is defined as 99% of compiler are clean

3.2. If you have no need for portability then skip or else

Enter 5 if you expect to upgrade in 1 year

Enter 4 if you expect to upgrade in 2 years

Enter 3 if you expect to upgrade in 3 years

Enter 2 if you expect to upgrade in 4 years

Enter 1 if you expect to upgrade in 5 years

3.3. Enter 1 if your language is a "standard" without sub or super sets

Enter 2 if a standard language sub or super set

Enter 4 if not a standard language

4. CONVERSION

Conversion in this paper is defined as converting your program specifications, flowcharts, etc., into the actual program. It is assumed that some form of structured programming is required.

CONVERSION	COBOL	SPL
	4.1	

4.1. Enter 1 if language manuals devote chapter to structured programming

Enter 2 if there is appendix describing techniques

Enter 4 if there is a reference

Enter 8 if there is no mention

5. COMPILER SPEED

The actual language compiler must be analyzed to determine its relative speed. Again, this is a guess and it is recommended that internal studies or outside references be used

COMPILER SPEED	COBOL	SPL
	5.1	

5.1. Enter 1 if fast one pass

Enter 2 if slow one pass or fast two pass

Enter 4 if slow two pass

Enter 8 if anything else

6. COMPILER AIDS

The number of aids available to the programmer can substantially reduce the programming effort. The table below represents some available aids.

AIDS	COBOL	SPL
EDITOR ^{6.1}	_____	_____
COPYLIB ^{6.2}	_____	_____
SEGMENTER ^{6.3}	_____	_____
TOTAL AIDS	_____	_____

6.1. Enter 1 if language or host computer has an excellent editor

Enter 2 if good

Enter 4 if fair

Enter 8 if none

6.2. COPYLIB is a COBOL facility that enables standard data divisions and/or procedure divisions to be incorporated into a program with one line of code

Enter 1 if available

Enter 8 if not available

6.3. Segmenter provide assistance in dividing program into logical segments for better program utilization and faster execution

Enter 1 if excellent
Enter 2 if good
Enter 4 if fair
Enter 8 if none

7. DEBUGGING

Languages provide various forms of debugging ranging from providing hex dumps to step by step traces.

DEBUGGING COBOL SPL
 7.1 _____

7.1. Enter 1 if language reference devotes chapters to debugging
Enter 2 if there is appendix
Enter 8 otherwise

8. SELF-DOCUMENTATION

Self-documentation is somewhat of a myth. A specific language may provide the facility for self-documentation but without consistent enforced coding standards then any program can be obtuse. The table below rates only one variable, the naming of variables

DOCUMENTATION COBOL SPL
 8.1 _____

8.1. Enter 1 if variable names can be actual words
Enter 2 if variable names can be reasonable abbreviations
Enter 4 if otherwise

9. PROGRAMMER AVAILABILITY

The popularity of a language has dictated the available supply of competent programmers in that language. A simple gauge of availability is to determine from want ads and/or an employment agency the average salary for a programmer with three years of experience in each language you are considering.

PROGRAMMER COBOL SPL
 9.1 _____

9.1. Enter 1 for lowest salary
Enter 2 for salary 10% higher
Enter 4 for salary 20% higher
Enter 8 for salary 30% higher
Enter 16 if a salary is quoted at 40% or higher
Enter 32 if there is no quote

10. INTERFACES

Screen handlers, database, graphics, telecommunications, etc., packages are becoming increasingly necessary in computer installations. The ease of a language in interfacing with these packages is vital in any determination

INTERFACE COBOL SPL
SCREEN 10.1 _____
DATABASE _____
GRAPHICS _____

10.1. Enter zero if your application does not require else
Enter 1 if language manual devotes chapter
Enter 2 if appendix
Enter 4 if reference
Enter 8 if no mention

10.2. Skip if total is zero

11. SUMMARY

Fill in the table and choose your language

	COBOL	SPL
COST PER COMPILE	_____	_____
APPLICATION 11.1	_____	_____
PORTABILITY	_____	_____
CONVERSION	_____	_____
COMPILER SPEEDS	_____	_____
COMPILER AIDS	_____	_____
DEBUGGING	_____	_____
DOCUMENTATION	_____	_____
PROGRAMMER	_____	_____
INTERFACES	_____	_____
TOTAL OF FACTORS	_____	_____
LANGUAGE COST 11.2	_____	_____

11.1 The difficulty factors are added together to obtain "Total of Factors"

11.2. Language cost = cost per compiles * total of factors

12. POSTSCRIPT

The above is a simplified analysis of the factors for selecting a language. Other factors should be considered and different weights can be determined. However, the basic approach yields valuable insights and usually some interesting surprises!