Halstead Complexity

Yegor Bugayenko

Lecture #4 out of 24
80 minutes

The slidedeck was presented by the author in this YouTube Video

"Any attempt to find a universal set of metrics that could be applied to any computer program, might at first glance appear destined to be unfruitful, if not merely difficult. But, without universal, measurable parameters, we would be in the position of trying to develop the science of thermodynamics before the advent of a temperature scale."

— Maurice H. Halstead. *Elements of Software Science (Operating and Programming Systems Series).* Elsevier Science Inc., 1977

Inputs

- η_1 — the number of distinct operators
- η_2 — the number of distinct operands
- N_1 — the total number of operators
- N_2 — the total number of operands

Example from Wikipedia

Example [edit]

Consider the following C program:

```
main()
{
  int a, b, c, avg;
  scanf("%d %d %d", &a, &b, &c);
  avg = (a+b+c)/3;
  printf("avg = %d", avg);
}
```

The distinct operators (η_1) are: `main` , `()` , `{}` , `int` , `scanf` , `&` , `=` , `+` , `/` , `printf` , `,` , `;`

The distinct operands (η_2) are: `a` , `b` , `c` , `avg` , `"%d %d %d"` , `3` , `"avg = %d"`

Operators and Operands

"When a program is translated from one language to another, as from FORTRAN to machine language for example, the actual operators and operands may indeed change, but both versions must still consist of combinations of operators and operands. No other category of entities need be present." — *Advances in Software Science*, Maurice Halstead, Advances in Computers, Volume 18, 1979

Length and Vocabulary

- $N_1 + N_2$ = Length
- $\eta_1 + \eta_2$ = Vocabulary
- $\eta_1 \times \log_2 \eta_1 + \eta_2 \times \log_2 \eta_2$ = Estimated Length

Length vs. Vocabulary

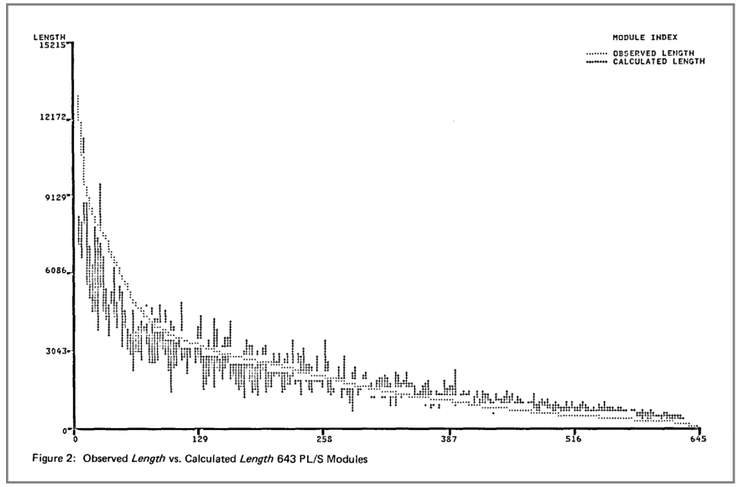

Figure 2: Observed *Length* vs. Calculated *Length* 643 PL/S Modules

"The size of a program, regardless of the metric used to measure size, is a function of the vocabulary of the program." — *A Software Science Analysis of Programming Size*, Charles P. Smith, IBM, 1980.

Volume, Difficulty, and Effort

- N = Length
- η = Vocabulary
- $N \times \log_2 \eta$ = Volume
- $\eta_1/2 + N_2/\eta_2$ = Difficulty
- $D \times V$ = Effort

Effort vs. Understandability

"We have independently tested the hypothesis that the mental effort required to create a program (measured by E) is related to a person's ability to understand a program or to find bugs in existing programs. The studies of Gould and Weissman as well as our work strongly support these hypotheses. " — *Review and Evaluation of Software Science*, Ann Fitzsimmons and Tom Love, Computing Surveys, Vol. 10, No. 1, 1978.

Time and Bugs Estimate

- E = Effort
- $E/18$ = Time (in seconds)
- $V/3000$ = Bugs

Bill Curtis

"In studying some error data provided us by Rome Air Development Center, Phil Milliman and I found Halstead's metric a remarkably accurate predictor of delivered bugs in a system developed with modern programming practices and tools"

— Bill Curtis. Program Complexity and Software Errors: A Front End for Reliability. In *Proceedings from the Fourth Summer Software Engineering Workshop*, pages 217–238, 1979

Halstead Complexity is supported by a few tools:

- multimetric for C++, Java, Python, and many others
- JHawk (not free) for Java
- Halstead Metrics Tool for Java
- PhpStorm for PHP

Read this:

Advances in Software Science, Maurice Halstead, Advances in Computers, Volume 18, 1979

References

Bill Curtis. Program Complexity and Software Errors: A Front End for Reliability. In *Proceedings from the Fourth Summer Software Engineering Workshop*, pages 217–238, 1979.

Maurice H. Halstead. *Elements of Software Science (Operating and Programming Systems Series)*. Elsevier Science Inc., 1977.