Milestones: Measuring Software Productivity

Marvin V. Zelkowitz

Institute for Advanced Computer Studies,
Department of Computer Science, University of Maryland,
College Park, Maryland

Our ability to measure the cost and development time for a software application is poor at best. Software is often delivered late and is generally over budget. The software engineering research community has been searching for effective ways to measure the progress of developing new software projects.

Currently, measures such as "lines of source code" are the most accurate, but one must have a completed program in order to be able to count such values. It is a poor estimator when used during design or maintenance activities. Various other measures have been proposed, but all suffer similar fates.[1] We need a new measure that is applicable across the software life cycle that better predicts the performance of developing computer programs.

Within other engineering fields, progress is often measured via "milestones." We informally use milestones in software development when we state progress on a project in terms of the completion of a specified number of milestones, the final milestone being that the user accepts the finished product. Other typical milestones are completion of specifications, completion of design, completion of source code, and completion of unit testing. A complex project may have many milestones, while a small project may have only one.

However, quantitatively, what is a milestone? What does it mean to be halfway there? When is a milestone 90% complete? On the basis of an earlier estimating algorithm,[2] we can adapt the milestone to the software development problem.

Software Development Milestones

A stone is a British unit of weight equivalent to 14 pounds. Since a mile is 5280 feet, we can compute the metric equivalent of the milestone in order to determine the amount of work it contains. A milestone is equivalent to 1 mile x 14 lb. Since the pound is actually used to measure mass, it can be converted to kilograms at 2.2 lb = 1 kg. We compute work of one milestone as

5280 ft x 14 lb =
73,920 ft lb =
10.1818 km kg

Using a value of 9.8007 m/sec^2 as the force of gravity, 10.1818 km kg converts to 100,161 joules as the metric equivalent to the milestone.

Since a staff-year of 52 weeks consists of 2080 hours or 7,488,000 seconds, a continuous power consumption of 13.4 mW is needed to achieve one milestone per staff-year. It is therefore a simple matter to measure the power consumption on a project to determine the effective progress in achieving these milestones.

However, accurately measuring such power consumption poses other risks, such as interference from overhead lighting or computer consoles near the programmer. Therefore we can convert such power dissipation to more meaningful and readily measurable quantities.

Consider the work necessary to produce one milestone per year. Assume further that a programmer weighs 150 pounds and works

However, quantitatively, what is a milestone? What does it mean to be halfway there?

for 2080 hours per year. (For consistency we are ignoring the effects of holidays, vacations, sick leave, and time around the coffee pot. Those can be factored in later.) Thus one milestone per staff year reduces to

1 milestone/staff year =
73,920 ft lb/(150 lb x 2080 hr =
0.237 ft/hr =
2.844 in./hr

A programmer must work at a rate of almost 3 inches per hour to achieve a milestone in a year.

If two programmers work at the same rate, the heavier programmer will accomplish more milestones in the same time period.

Consistency With Existing Standards

If we assume that a typical line of text used in a programming language (e.g., BASIC, FORTRAN, C) contains 20 characters and that each character is 0.1 in. long, then

2.844 in./hr =
2.844 in./hr x 10 char./in. x 1/20 line/char. =
1.422 line/hr

This figure of 1.422 lines per hour translates into 2958 lines of code per staff-year. If we further assume that a rate of one milestone per staff-year represents a reasonably complex program for a single programmer, then this figure of 1.4 lines/hour is comparable to industry figures of 1 instruction/hour on large complex systems,[3] and is within the range of 2000–8000 lines of code per year generated on other large-scale projects.[4] Thus this measure has the advantage of being quite easy to generate and consistent with other such measures.

Conclusions

In this paper we have shown that we can translate the old British measure of the *milestone* and use it to develop an accurate quantitative measure for software development. We reduced the measure to the continuous power consumption of 13.4 mW in order to achieve one milestone per year. Further, we can validate this measure by showing that it is equivalent to a programmer producing 1.422 lines of source program per hour, a figure that is well within industry guidelines.

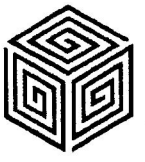
This model allows us to make further predictions. For example, *heavier programmers are more productive.* In computing productivity we used a figure of 150 pounds per programmer. The larger this figure, the fewer lines are needed to achieve one milestone per year. If two programmers work at the same rate, the heavier programmer will accomplish more milestones in the same time period (e.g., a 200-pound programmer needs to write only 1.066 lines per hour to achieve one milestone per year).

This report only indicates some of the potential for this measure. We expect to validate it further in the months ahead.

References

1. DeMillo, R., McCracken, W. M., Martin, R. J. and Passafiume, J. *Software Testing and Evaluation.* Benjamin Cummings, Menlo Park, Calif. 1987.
2. Melmed, A. J. and Bertocci, W. Counting the state joules, *NBS Standard 1978, 23(13)*:8.
3. Wolverton, R. W. The cost of developing large scale software. *IEEE Transactions on Computers, 1974, 23(6)*:615–636.
4. Brooks, F. *The Mythical Man Month.* Addison-Wesley, Reading, Mass., 1975.

Solution To Last Issue's Puzzler



Send it by boat.

(With an inflexible deadline to be met, and with the stated postal regulations being strictly enforced, and with only 24,000,000,000,000 rolls of packing tape available, there is no other way to mail Mount Fuji to New Zealand without exhausting the treasury.)