Forecasting Support Costs

~ By Luc Richard

Did you know that maintenance accounts for 50% to 80% of the overall product cost? Well, it does! And while most project managers are fairly good at sizing new product features, many are terrible at estimating the effort required to support a product once it becomes generally available. As a result, maintenance projects are inadequately staffed, companies can’t respond to customer requests in a timely manner, and products never reach payback.

This article presents a methodology to help you guesstimate and therefore plan for the maintenance phase of generally available products. But first, let's define a few terms that are important to the comprehension of this article.

Maintenance

Maintenance is defined as the effort associated with fixing defects in a software system after general availability (GA). In other words, how many person-months will it take your organisation to fix bugs discovered by your customers in the field?

Maintenance can be subdivided in three sub-categories:

1. **Corrective maintenance** involves fixing bugs that are discovered in the system after it becomes generally available. An example of a corrective maintenance activity is a developer fixing a Java method that causes a compilation error.

2. **Adaptive maintenance** involves changing the system to work in a different environment such as a different network topology, platform, or operating system. An example of an adaptive maintenance activity is a developer fixing a Java method that works on BEA WebLogic but not on IBM Websphere.

3. **Perfective maintenance** involves changes that allow the software to meet the same requirements but in a more acceptable manner. For example, the designer might change some code simply to make the system more efficient or easier to maintain.

Enhancements

Enhancements, also known as change requests, are defined as the effort associated with adding new capability to a software system, or modifying a software system to meet newly defined non-functional requirements.

Imagine an application that requires the user to authenticate using a username and password. Pretty standard stuff, right? Maybe, but some customers might want to add a third credential to the password mechanism such as a domain. Others might want the username to adhere to an email address pattern. Finally, others might want the application to remember the user’s credentials over sessions, thereby authenticating the user automatically.
Support

Support is defined as the sum of the maintenance and enhancements efforts performed after the product is generally available. In other words, support includes all the activities that go on after a product is declared generally available.

Methodology

Early in my career, I realised that simple rules of thumb could be applied to estimating the support cost of certain projects. For example, the annual cost of supporting a static website after it goes live is more or less equivalent to the cost of developing it. In other words, if developing a static website costs £10,000, you can expect to spend £10,000 per year maintaining it.

Understanding such rules is very practical. Unfortunately, few of them are transferable. In other words, the same rule would not apply to an ecommerce enabled dynamic website distributed across 3 tiers.

Various models have been developed over the years to predict maintenance costs based on defect-density (e.g. Raleigh Curve, Weibull Analysis), KLOC and KDSI, and development efforts. Unfortunately, these models are not without any shortcomings either. Many of them are either highly inaccurate or too complex to bother learning them. As a matter of fact, some are so complex that you need to purchase an application worth thousands of pounds and enter 100+ parameters in order to have it compute the effort required to maintain your product.

After having studied over a dozen forecasting models, there is one methodology that I highly recommend to any beginner or seasoned project manager.

Boehm's Model

Boehm's model is widely accepted in the industry as a valid model for predicting maintenance costs. It's relatively simple to understand, and more importantly, it allows you to refine your forecast thanks to cost multipliers, which will be explained later in this article.

Boehm's formula is the following:

\[ AME = ACT \times SDT \]

- **AME** is the annual maintenance effort measured in person months.
- **ACT** is the annual change traffic, which represents a fraction of a software product's source instructions that undergo change during a typical year through addition or modification.
- **SDT** is the software development time in person months.

Say a software project required 100 person-months of development effort and it was estimated that 15% of the code would be modified in a typical year. The basic annual maintenance effort estimate (AME) is therefore:

\[ AME = 0.15 \times 100 = 15 \text{ person-months}. \]

In other words, you should plan to spend 15 person-months of effort per year to maintain this specific software project.

The basic annual maintenance cost estimate may be refined by judging the importance of each factor that affects the cost and selecting the appropriate cost multiplier. The basic maintenance cost is then multiplied by each cost multiplier to give the revised maintenance cost estimate.
Say in the previous system the factors having most effect on maintenance costs were Product Complexity (CPLX), which was very high, and the availability of support staff with application experience (AEXP), which was very low.

If CPLX = 1.30 and AEXP = 1.29, then:

\[
AEM = 15 \times 1.30 \times 1.29 = 25.2 \text{ person-months.}
\]

### Forecasting Enhancements

The revised maintenance cost does include the impact of the cost multipliers but does not include product enhancements, also known as change requests.

The bad news is that forecasting enhancements is extremely difficult because it requires you to know ahead of time what additional capabilities your future customers will request. The good news is that you can charge your customers for any enhancements they require. As a result, a good organisation does not consider enhancements to represent a cost but rather a source of incremental revenue.

### Conclusion

When forecasting the cost of maintaining a product that is generally available, follow this advice:

- Learn and use this (simplified) version of Boehm’s Model to forecast maintenance costs.
- Track your SDT.
- Measure your ACT.
- Define cost multipliers to refine your forecast.

Furthermore, make sure you have a professional services team to implement change requests required by your customers, but do not treat them as costs since they are in fact a source of revenue.

*Luc Richard holds an MBA with a major in high technology. For the past ten years, he’s been managing the development of software applications.*