



TenStep Supplemental Paper

15 May 2002

Evaluate the Use and Cost of Development Tools

If you are a senior developer, team leader, or development manager, you may have a sense for the yearly maintenance costs that your company pays for development tools. However, it would surprise most developers to see the license and maintenance fees being paid by your company.

The purpose of development tools

There are many tools available to support all aspects of the development lifecycle. There are tools that help in planning, analysis, process and data modeling, design, programming, testing and deployment. There are also many other tools that exploit development niches. These tools serve two main purposes. First, they help automate development processes that would be almost impossible to control manually. For instance, if you are developing a complex application with hundreds of requirements, a tool can help keep track of everything, including the dependencies between the requirements and the different types of data that are required. If your business applications must undergo maniacal testing for bugs and compatibility with prior versions, you will need automated testing tools with regression testing capabilities.

The second benefit of development tools is that they help establish and maintain consistent development processes across the organization. For instance, source code management software enforces a consistent library structure, naming conventions, check in / check out process and program integrity. If you find that you have various departments or projects doing development in different ways, the use of common tools and processes will help standardize the environment.

Large development shops need more tools

Since development tools help manage higher levels of complexity and enforce standard processes, it should not surprise you that the larger your development organization is, the more tools they will need. Some large companies have dozens of development tools some that are utilized in all areas, and some that are acquired to help only a small number of projects with specific characteristics. A smaller company may not have many tools other than for programming and debugging. If it does not perform heavy development, it will have no need for sophisticated analysis and design tools. It is much more important to look at the cost of the tool in terms of licensing cost, and the incremental cost to utilize the tool, versus the time required to do it manually. In a large organization, many of these tools are absolutely vital. In a smaller company, it is not always so clear. Given the high cost of many of these tools, they can be prohibitively expensive for smaller shops.

Cutting the cost of development tools

1. **Reduce tool overlaps.** Make sure the tools are rationalized from an architectural perspective. Each one must solve one or more development requirements and you should have only one tool in every functional space. With acquisitions and mergers,



TenStep Supplemental Paper

or a lack of internal control, it's possible that some of your development tools overlap. For instance, if you have two tools for data modeling that perform substantially the same functions, standardize onto one to simplify the environment and reduce costs.

2. **Perform a cost benefit analysis.** Make sure that the tools you pay for are actually being used. If your group has a lot of development tools, you may well find that some of them are used all the time, and some are hardly used at all. They may save time when they are used, but the time savings may be so infrequent that you would want to get rid of the tool and do the function manually.
3. **Look for enhanced capabilities.** Many tools are enhanced over time and end up having capabilities that did not exist when they were first purchased. You may find that using enhanced features and functions of one tool may eliminate the need for another. This simplifies the environment and reduces costs.
4. **Check on the number of seats you are buying.** This is related to item number 2 above. If a tool is being utilized infrequently, you may have the option of paying a reduced maintenance fee based on less people using the tool. For instance, five years ago, everyone may have used the tool, but now only a few people use it. Make sure that your maintenance fee gets reduced to recognize this lower level of use.
5. **Renegotiate the maintenance agreement.** Typically you can negotiate the list price of original software license down substantially, but you might feel that maintenance is something you are stuck with. After all, you signed a license – right? Well maybe right, but maybe you can renegotiate the maintenance as well. Your negotiating power comes from being willing to walk or generate new business with the vendor. If you are prepared to walk and uninstall the product, you may find that the vendor is very willing to work with you. If you are looking for new tools from an existing vendor, you may be able to renegotiate the maintenance on older tools as part of the new deal.

Summary

There are many tools available to help automate and standardize all aspects of the development life cycle. Tools help manage large and complex development activities, and help to enforce standard development processes. Usually, the larger and more complex your development organization is, the more tools you will need.

Your organization may be paying the licenses and maintenance fees for many development tools. It is important to understand the types of tools you have, their cost and whether they are being fully utilized. If two or more tools overlap in the same space, standardize on one. This will result in a simplification of the environment, more consistent standard processes and a reduced cost to the organization. Although it may seem you are stuck paying the maintenance on existing development tools, there are ways you may be able to reduce your charges substantially.