HOW TO LOSE PRODUCTIVITY WITH PRODUCTIVITY TOOLS

E. J. Chikofsky

Index Technology Corporation, One Main Street, Cambridge, MA 02142, USA

Abstract: Organizations introduce software tools, methods, and Computer-Aided Systems Engineering (CASE) environments to improve productivity and to enhance their ability to deliver meaningful, consistent, and complete projects in a timely manner. But many organizations have found that these tools require more effective management approach to achieve these objectives. Without reviewing their management practices, organizations unknowingly lose the very productivity benefits they strove to achieve. This paper covers four key topics: misconceptions and pitfall in the use of tools, the problems of tool introduction, failing the learning curve, and management strategies to gain and retain productivity from tools.

Keywords: Software Tools, Computer-aided Software Engineering, Software Management, Software Productivity, Productivity Tools

Over the last 15 years, the development of software engineering tools has grown and matured. This progress has resulted in the recognition of the CASE field, which is primarily the industrialization of reasonably well-understood software technologies. The period of time it has taken to reach this stage is not unreasonable for such a transition to the widespread practical application of software tools. Studies of technology transfer in a number of different industries generally support the notion that it takes 15 to 20 years for a new technology to reach sufficient maturity for general use.

But in the meanwhile, many otherwise excellent software tools have not successfully been transfered from the laboratory to industry. They have not made it beyond the pilot project stage into real productive use in organizations. Clearly, one reason for this is that many of these tools were inherently unable to scale up to the reality of large projects. But that is not sufficient to explain the wide range of tool failures. In re-examining the last 15 years of software tool introduction, and reviewing my own experience as a tool developer and technology advocate, I contend that most tools fail because organizations do not recognize the symptoms of their own mis-management of tool acquisition, tool introduction, and continuing tool use.

Problems Begin in Tool Acquisition

Many of the problems of tool use are direct results of an organization's approach to the introduction of tools. There are a number of key symptoms which signal mismanagement in the acquisition and introduction of productivity tools:

- **Shopping Without Real Goals.**
  Organizations often shop for tools without well-defined goals. If you walk around the tools fair of one of the many CASE marketing shows and pick up a list of clients from each vendor of a brand new tool, what do you see when you compare the lists? The same major company names appear over and over again on the long lists of well-known corporate clients. These companies buy at least one or five of everything. Many such organizations have "Tool Finder" as an implicit or explicit job role -- someone who is charged with finding technology. But they fail to assign responsibility for successful transition of the tools acquired into user projects.

- **Throwing Tools Over the Transom.**
  Many organizations isolate the real users from the acquisition of tools. The user project teams are not consulted regarding the real needs to be satisfied for real projects. Instead, they receive new tools that looked good to the tool finder and to upper management but may have little direct value to solving the problems at hand. Tools thrown over the transom often become shelfware on the other side.

- **Conducting Pilot Projects Without Support.**
  For a pilot project of a software tool to be most useful, there should be some up-front agreement on the criteria for success. The parties to this agreement should include management, prospective user from other projects, and the pilot team. Too many organizations go into pilot projects without having sufficient management support and without determining how the project will be judged.

In contrast with these first three symptoms, the transition-successful tool finder relies on a well-defined set of objectives, often the result of corporate strategic or project management tactical planning efforts involving multiple projects or departments. Pilot projects are most successful when planned with the informed and interested participation of users and management. Through the use of well-planned pilot projects, organizations can identify the correct scope in which new methods and tools can succeed, and where they need to be engineered to the appropriate scale.

- **Managing Solely by Edict.**
  Some organizations try to introduce new technology solely by executive fiat or management edict. Without grassroots support among project teams, mandated tools may be followed without fervor or embraced in name only.

- **Implementing Revolution Without Evolution.**
  Organizations and people do not like drastic change, making it hard to succeed by introducing revolution. Tools and methods which require analysts or project groups to work in a manner which departs significantly from their experience and understanding have been harder to introduce and get accepted. This implies that it will be quite difficult to change to better tools and methods which are more demanding, recognizing the uphill effort needed to gain acceptance.

  It is easier to garner support with an evolutionary approach to tool introduction. It is often best to encourage use by introducing tools or features which are supportive of the way the organization works now. Building on such a

foundation, it can be easier for the organization to accept improvements and advanced featured in the future. By customizing new tools to better capitalize on familiar aspects and procedures of the potential audience, it should be possible to improve the implementation process and aid their acceptance.

- **Adopting Instead of Adapting.**
  For a tool or method to be accepted in an organization, it must adapt to the corporate culture. This adaptation occurs in several ways. The project management procedures of the organization provide a framework into which the products of the tool must be fit. The contractual obligations of the organization constrain, and often mandate, the deliverables of the development process which the tool must meet.

  Many organizations looking to acquire methods have not been satisfied with implementing the methods in off-the-shelf form. Instead, they have selected what they consider to be the best and most applicable aspects of several methods, often covering different life cycle phases or different modeling perspectives, and have welded them together into a new corporate method. (For an example of a corporate integration of methods, see Kathleen Mendes' article "Structured Systems Analysis: A Technique to Define Business Requirements" in the Summer 1980 edition of the Sloan Management Review.)

  The adaptation of methods means that tools to support methods must be able to adapt to be successful. The best tools are able to be customized and evolve with the organization's needs without extensive intervention by the tool developer.

- **Allowing "Acquisition Deadlock".**
  With so many alternative approaches in the marketplace regarding software engineering tools and methods, some managers are at a loss to find reasonable criteria to make informed decisions. This is further compounded by the organization's lack of identified goals. The drive to examine each and every tool on the market for the "best" solution has caused many organizations to go into a form of "Acquisition Deadlock". Another new tool entering the marketplace causes a new round of deliberation.

  Organizations which have broken the cycle early and have chosen some initial tool set, knowing that it might not be a perfect choice, have better handled the technology introduction hurdle. These organizations are now better prepared to receive and utilize second- and later-generation tools.

Forgetting to Manage Tool Use

Watching how many organizations introduce tools, often with a lot of attention to detail because of the up-front investment of resources, one would expect to find a healthy management interest continue into monitoring - or at least occasionally reviewing - the tools in regular use. However, most organizations never revisit their use of tools. Once a tool gets beyond initial introduction, many organizations seem to forget that the technology still needs to be managed. They allow implicit management decisions and inattention to become the norm.

By not watching what is happening, management encourages the rise of many pitfalls and misconceptions which impede the effectiveness of tools and methods in practice. Without periodic re-examination, many user practices become the accepted or "proper" way to use the tool, without regard for their overall impact on productivity. Together, these pitfalls form a strategy for losing the productivity gains that the organization thinks it has achieved:

- **Underestimate the Effort Required**
  Tools tend to take a lot more time and effort than we initially think they do. There are many examples of organizations failing to comprehend and allocate the resources required for successfully extending the introduction of tools into new projects teams and helping them to scale the learning curve. This is compounded by allowing people to take short cuts which, in many cases, are not savings in the long run.

  New users undergo a progression through stages of tool familiarity: introduction, first education, initial experience, re-education, then finally productive expertise. By presuming expertise too early, and by failing to account for re-education as part of the process, we artificially short our estimates of the effort required.

- **Have the First Student Train the Rest**
  Training of users is often treated as overhead or holiday boondoggles. We give it insufficient support. In many organizations, the first person to attend a training class is expected to come back to the office and teach everyone else. This shortchanges everyone in the process. You don't learn a tool in the classroom. You learn a tool by using it. The classroom exposes you to the capabilities of the tool, application possibilities, and what features are where. By actually using the tool on a project of your own is where you really learn it. When the first student tries to train the rest, that student doesn't get the opportunity to really learn the tool - and the rest of the students have a teacher who doesn't really know the subject.

- **Introduce a Savior**
  Tools have often been brought in as the salvation of projects in crisis - the worst possible scenario for tool introduction. Projects which are already projected to be overdue, over budget, understaffed, and under-supported are unsuitable candidates for new technology. This fact is all too often ignored.

  We tend to forget that saviors usually come with their own rule sets to be followed. To get the benefit, you have to be willing to adhere to the rules. Yet many organizations try to use tools as magic wands to get them out of trouble without being willing to adopt and follow the rules they require.

- **Assume Organizational Stasis**
  Having chosen the right tool for the job, an organization may have failed to notice that the job has changed. Because of how long it sometimes takes to gain acceptance for a tool as a regular part of the development process, we may not notice differences in the environment. In fact, the very introduction of the tool may have changed the organization's balance of power between project leaders or managers. One department has gotten upper management support for new technology, while another has not. By not recognizing changes to the organization and its culture, we can miss opportunities or fail to recognize impediments.

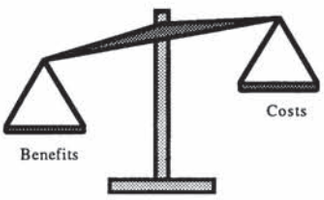
- **Presume One Size Fits All**
  We often try to apply the same tool in the same way to all projects. Tools in place are often overused - without examining whether they are really appropriate solutions. The adage "if you have a hammer in hand, every problem begins to look like a nail" is particularly true of the recent history of software tools.

  There is also a tendency to ignore the tool's real purpose. The reality of the intended job in the organization may match the apparent purpose of the tool, but may not match the developer's intended purpose of the tool. Many copies of early analysis and design tools have been put to use as over-qualified word processors, with little regard to their analytic facilities and true potential.

  We also fail to customize the tool to the need. Tools are often introduced without adequately tailoring them to the organizational environment. The existing facilities of the tool may never have been examined to choose appropriate defaults for the project at hand, and to select project management options. Further, the tool may not have been constructed for adaptation. There may, in fact, be another model or variation which would be more appropriate.

- **Blame the Tool**
  No matter what happens, it's the tool's fault. Tools are easy scapegoats and get blamed for a lot of management mistakes and inattention.

- **Fail to Recognize Economics**
  It is important to keep in mind the economic incentives for tool use: the profit motive. A productivity tool's purpose is to allow the user to do more with the same or less resources. More might be measured in volume, quality, or shortened time.



  In deciding to introduce the tool, someone concluded that the benefits outweighed the costs. The costs for introducing tools are most often very visible: software, training, computer facilities. Benefits are less visible, less tangible. They may be in terms of savings; improvement in quality of products; improvement in productivity; the ability to tackle otherwise unmanageable problems. Benefits are usually harder to measure than costs, and clearly take much more time to accumulate.

  Organizations often have only vague notions of the benefits to be achieved. In examining the use of existing tools, it is useful to ask some key questions: Is management still waiting for the benefits to be realized? Who is expecting the benefits: specific users, first-level management, top-level management? Are those benefits still applicable to the continued use of the tool?

- **Fail to Define Productivity**
  Most tools are promoted and acquired based upon objectives dealing with productivity, yet most organizations have not considered what they mean by productivity. In 1981 IEEE articles, Tony Wasserman pointed out that for many organizations "teaching all developers the skills of touch typing ... might have a greater impact upon productivity than would the introduction of new software tools or design techniques."

  We often relate productivity to efficiency -- building it faster -- when what our real goal should be is effectiveness -- building the right thing better. This leads us to mistakenly follow form instead of content in our use of tools and methods.

  Without a viable definition of the organization's productivity objective, effective measures of progress are not possible. As the Cheshire Cat said to Alice, if you don't know where you want to go, which direction you go doesn't matter.

- **Use Methods Half Way**
  Organizations claim to follow structured methods but many follow the form but not the content. They implement the visual attributes of a method - the diagram types and symbols or the notation of data dictionary - but not the rules of the method or the analytic tests for good quality designs. Larry Constantine, a founder of structured design who recently returned to the software engineering after a ten year hiatus, observes that "it's almost as if no one read past chapter three in any of the texts."

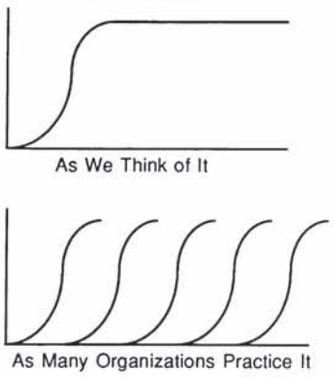
  When users of automated dataflow diagrams don't push the analysis button for DFD Balancing, and when users of structured design don't recognize and understand the key terms such as "cohesion" and "coupling", how can their organizations hope to reach the potential of their investment in software tool technology?

Failing the Learning Curve

Most people view the learning curve as something the organization goes through once to achieve productive efficiency. This is not true for many organizations. They keep repeating the learning curve. By not having mechanisms in place to preserve and exchange expertise between project teams or departments, they implicitly encourage projects to cover the same ground again. In such cases, the organization never quite attains the productivity benefits it should expect from the effort put in.

The learning curve is not a rise to a plateau. Organizations can lose the productivity they've gained by allowing what they've learned to decay over time.

The Learning Curve



Winning Back Productivity

There are, however, various techniques which organizations have used effectively to manage the Learning Curve and stop reinventing the wheel:

- Provide meaningful support to new projects and users, regardless of political boundaries in the organization.
- Make training a valuable and respected part of the tool introduction process. Avoid the presumption that someone who has been to one class can train the rest of the new users.
- Tailor training to the user audience by including meaningful problems and examples which capitalize on the users' prior experience and knowledge.
- Explain standards and conventions and provide "how to" documentation tailored to the organization and its type of projects.
- Make the materials from prior projects accessible.
- Apply the tool in levels with new users by matching features of the tool to needs and levels of expertise.
- Revise conventions and standards to reflect real project experience.

Conclusion

Productivity tools are only part of the solution to achieving productivity in system development. Management practices of the user organization regarding the use of tools have to be as effective at fostering productivity as the tools themselves.

There are many ways in which an organization can unknowingly fail to realize the benefits expected from productivity tools. The tool acquisition process, operational issues, and the treatment of the learning curve each contribute to a potential loss of productivity. By periodically re-examining our explicit and implicit management decisions regarding tools, we can be sure that we are getting the value we need out of tool technology.

Making software tools more successful requires that we make our approach to managing them more successful. The best tool is no substitute for poor management.

Biography

Mr. Elliot Chikofsky is Director of Research & Technology at Index Technology Corporation, maker of the Excelerator line of CASE products. He also teaches graduate courses in Software Engineering and Database Management Systems at Northeastern University in Boston.

He is Associate Editor-in-Chief of IEEE Software and editor of its Software Manager department. He is co-editor of the upcoming special issue on Software Maintenance and Reverse Engineering (Jan. 1990), and was editor of the March 1988 issue on CASE. He is also serving as Vice Chair for Membership of the IEEE Computer Society's Technical Committee on Software Engineering.

Mr. Chikofsky is President of the International Workshop on CASE, and was the General Chair of its 2nd workshop (in cooperation with the IEEE Computer Society). He was a keynote speaker on CASE at the 4th technical symposium of Japan's Sigma Project in June 1989, and has presented tutorials on CASE for both ACM and IEEE-CS. His recent book on CASE is published by IEEE Computer Society Press in the Technology Series.

Elliot J. Chikofsky, Index Technology Corporation, One Main Street, Cambridge, MA 02142; (617) 494-8200 x552