

There Are No Silver Bullets!

Michael Harris
DMR Group Inc.

P.O.Box 24346 M/S 7M-48
Seattle, WA 98124-0346 USA

ABSTRACT

Many organizations have attempted to improve their process for delivering application systems to their end-users. Most have not succeeded because they attempted to implement the "quick-fix" of automated tools or new techniques. The only proven approach is through an overall program to change the software management process. This process can take software development from the craft shop era to the software factory era.

INTRODUCTION

In order to accomplish the goals of higher productivity and quality in the management of application software, it is important that an integrated, coherent process be created and managed. The answer does not lay in "silver bullets" such as CASE and project management tools or the newest analysis and design techniques. The answer lays in the CQI concept of a continuously improving process. This process can take software development from the craft shop era to the software factory era.

The Software Management Process (SMP) is defined as the means to provide automated business solutions through the development, implementation, and maintenance of application software. Key words in this definition are:

- o management
- o automated
- o business
- o application

Management vs. Development: The process must not only be capable of developing new systems but also integrating "off-the-shelf" packaged solutions and integrating with existing systems. The term "management" is therefore used instead of development.

Business vs. Computer: The process must deliver business solutions and not just computer solutions. It must define the integration of automation into the

business. Manual work procedures must be developed as an integral part of the process.

Automated vs. all: The software management process should deliver business solutions but only where computer automation is the most significant portion of the solution. The process is not intended to develop new work procedures where little or no computer automation exists.

Application vs. all: The process is limited to application systems and is not targeted to non-application software such as operating system software and utility software. Although many of the principles are applicable, the specific techniques and tools are not.

CURRENT SITUATION

In the information systems industry today there is little consistency, structure, or rigor in the development of application systems. Often when a rigorous process for software management is introduced it is viewed with disdain and as bureaucracy. Other professions such as architecture and engineering have well established disciplines for the execution of their tasks. The information systems profession has not yet matured to the point of using a complete and consistent approach to building systems.

Before a process can be automated an organization must be capable of doing the process manually. CASE has not been a major success throughout the IS industry for this very reason. CASE makes a good analyst better but it does not turn a programmer into an analyst. Project management tools in IS have suffered a similar fate. Millions have been spent on tools and training but the impact has been negligible, at best.

The implementation of tools, techniques, methods, and standards within organizations, for the most part, has been ad hoc and uncoordinated. Automated tools have been selected without regard to the techniques in use in the organization. The development environment has been created bottom-up with little overall planning. The goals are usually set for the component (such as a CASE tool) rather than the overall process. Errors of omission are common and usually result in project

overruns, reduced user functionality, and cancelled projects. Systems are rushed into production before adequate testing can occur in order to meet the original schedule or budget (which typically results in the search for better testing or estimating tools). Some projects succeed technically but are considered failures.

So what must be done to solve the problem? The following items are required:

- o integrated development environment,
- o overall implementation strategy, and
- o culture change within information systems.

DEVELOPMENT ENVIRONMENT

The development environment must integrate in a coherent manner:

- o methodologies (phases and deliverables)
- o techniques (modeling)
- o standards & procedures
- o automated tools
- o metrics
- o training material.

IMPLEMENTING THE NEW SOFTWARE MANAGEMENT PROCESS

The implementation of a new software management process in an organization is a complex task. It brings the major changes of a new methodology, techniques, standards, tools, and changed roles and responsibilities. For an organization whose information resource is characterized by well-established assets and culture, the introduction of a new software management process is a considerable challenge.

The implementation of the new software management process is a long-term program that requires a change in the culture of the information systems environment. This change will require the commitment of not only the information systems community but also the end-users. This change will affect the development teams, project managers, end users, technical support staff, and most importantly the management of the organization. Without the commitment and dedication of senior management, culture change is unlikely.

It is important that change be well planned and not occur faster than the organization can cope with it. It is very important that each component be validated before full implementation. Change is upsetting to any organization so it is important that it is perceived to be valuable and well managed. To ensure success it is recommended that a program be established to coordinate the implementation of the new software management process. The program's management team can then oversee all projects being undertaken

within the program to ensure their coherence, relevance, and quality. The program should be organized using the following key roles:

- o sponsor,
- o implementation manager,
- o development manager,
- o architect, and
- o steering committee.

The staffing of these roles with the appropriate people is the most important success factor.

CULTURE CHANGE

The implementation of the Software Management Process (SMP) requires the same discipline that end-users face in the installation of a new computer system. Many of the computer professionals who will be required to use the SMP will resist the change to the way they perform their jobs. Others will insist that there are better ways (different tools, techniques, etc.). To accomplish the goal of a process that can be continuously improved, a single process must be installed. The marginal gain of one tool or technique over another is insignificant compared to the gain attained by consistency.

IS professionals must move from viewing their role as technologists to that of implementors of improved business processes. Management must resist the pushing of project teams into coding before the analysis and design are complete. Coding should only represent 10-20% of the effort in modern software management.

CONCLUSIONS

The information system industry, being only forty years old, is only starting to mature. The advent of the disciples of the new methodologies, techniques, and automated tools are a sign of this maturing process. When an organization is ready to commit to better planning, it is now possible to implement an effective planning framework. Modern system development and project management methodologies such as DMR's Productivity Plus™ are now available and form the foundation for the implementation of an integrated development environment. The main question that each organization must ask is "Are we ready to demand this discipline of our IS staff?". Then and only then should they invest in an overall software management process. Otherwise it is best to forego the "silver bullets" and do nothing.

REFERENCES

- 1 Harris, Michael, "A Planning Framework for Systems Development Projects", Journal of Information Systems Management, vol. 8, Winter 1991