

**TITLE: "METHODOLOGIES VERSUS
TECHNIQUES AND TOOLS"**

by Tim Bryce
Managing Director
M. Bryce & Associates (MBA)
P.O. Box 1637
Palm Harbor, FL 34682-1637
United States
Tel: 727/786-4567
E-Mail: timb001@attglobal.net
WWW: <http://www.phmainstreet.com/mba/>
Since 1971: *"Software for the finest computer - the Mind"*

*"Having a Project Management system without a methodology is like attaching a speedometer to an orange crate; it measures nothing."
- Bryce's Law*

INTRODUCTION

The term "methodology" is being bandied about by just about every software development vendor and consultant imaginable. You would be hard pressed to find a vendor who, in addition to their usual tool offering, doesn't promise a methodology to solve all of your development problems. But like many things in this industry, the terminology is getting sloppy and it is becoming apparent the true definition of "methodology" is being bastardized.

IN THE BEGINNING

The term "methodology" became popular in information systems in the early 1970's, initially as a response to the question, *"What is it?"* Milt Bryce first applied the term to systems development in 1971, to describe his Information Systems Engineering process. Bryce referred to "methodology" as a process which ends with the delivery of a product or a completely defined result.

Later on, during the structured programming movement, a different interpretation of the word emerged from software gurus such as Yourdon, Gane/Sarson, Orr, Finklestein, Martin, Warnier/Orr, etc. Instead of describing the overall process by which development occurs, the structured programming people began to use the term "methodology" to describe their techniques for designing software (e., functional decomposition, data driven design, object oriented design, etc.). Consequently, software development tools, which represent automated extensions of these techniques, began to tout their products as "methodology" enablers.

This division in the use of the term "methodology" is a

major source of confusion to the industry. Not all "methodologies" are created equally. There are fundamentally two interpretations: as a term referring to the "process" by which work is performed, and; as a term referring to a particular design technique. To truly understand "methodologies" you must know the difference.

METHODOLOGIES AS "PROCESS MANAGEMENT"

We at MBA define a methodology as, *"a process which ends with the delivery of a product or a completely defined result."* Under this perspective, a methodology defines the "5-W's"; it defines WHO, is to perform WHAT work, WHEN, WHERE, and WHY. If this sounds like an engineering/manufacturing process, it is. MBA contends information resources can be designed and developed in the same manner as any other product. Here, a methodology defines the division of labor and synchronization of work effort. With this approach, the development effort is divided into smaller more manageable pieces just as in an assembly line process. Construction projects represent another example (e.g., shipbuilding, office/home construction, etc.), where the work is carefully divided into stages with precedent relationships.

METHODOLOGY AS A DESIGN TECHNIQUE

As opposed to the "5-W's" interpretation by MBA, a methodology supported by the software design people defines HOW a particular task is to be performed. For example, the forte of design techniques such as "object oriented programming," "structured programming," or "information engineering" is on HOW to accomplish specific activities of work. From this context, the term "methodology" is a misnomer which should be replaced by the term "technique," a more apt description.

Techniques may differ from company to company, and there is not always a single way to perform a task. For example, in the automotive industry, fenders have always been a part of the car, but they have not always been attached the same way. Originally, fenders were bolted to the body of the car. Years later, an automotive worker welded the fender to the car. Today, welding robotics perform the task. The task, attaching the fender to the car, hasn't changed, but the techniques to do it have. Improved techniques can mean realizing the same result with savings in time and money.

The same is true in the information systems world. Whereas there are generic stages of work for designing and developing a system, there are a multitude of tech0

(continued on page 2)

(continued from page 1)

niques for performing the work. For example, there are significant differences between "structured programming" and "object oriented programming," yet the result is fundamentally the same, the development of an executable program. The difference is the chosen approach of implementation (there are pros and cons for both techniques). Whereas "Software Engineering" represents a phase of work in a development project, "structured programming" and "object oriented programming" represent techniques that can be used to perform the phase.

Does this mean there are overlaps or conflicts in the use of the different types of "methodologies"? Not quite. But to appreciate the difference, one must understand the concept of "Productivity" (as we have discussed in other "PRIDE" Special Subject Bulletins).

PRODUCTIVITY = EFFECTIVENESS X EFFICIENCY

Productivity is not simply a matter of how fast a task can be performed, it's a matter of performing the right task at the right time. This is what underlies the concept of productivity. Whereas "efficiency" concentrates on speed of delivery, "effectiveness" is concerned with doing the right thing at the right time; the two are not synonymous. For example, performing a weld using robotics may be a far more efficient means than performing the task manually, but it is useless if you are welding the wrong thing. There is nothing more unproductive than to build something efficiently that should never have been built in the first place. Zero percent effectiveness times 1000% efficiency equals zero productivity.

A true methodology addresses the effectiveness side of the equation (Who, What, When, Where, Why), and a technique addresses the efficiency side (How to). Whereas a methodology defines the work environment, the technique defines how the work is to be performed. The two are obviously complementary and one does not eliminate the need for the other. But comparing one with another is like comparing apples with oranges, they are simply not the same.

FACTORY CONCEPT

As described in "PRIDE" Special Subject Bulletin Number 11 ("*Creating an Information Factory*") - Feb 14, 2005), within an engineering/manufacturing facility you will typically find:

1. An Assembly Line where products are developed in

stages.

2. Production Control monitoring the assembly line for delays or accelerations in production.
3. Techniques for performing work.
4. Tools providing mechanical leverage.

These elements can be found in any development environment, including the IT world. What is interesting is the relationship between the elements:

ASSEMBLY LINE - at the heart of the factory is the Assembly Line process where products are developed in stages by workers with different skills for the different stages of work. In IT terminology, this is the "methodology."

PRODUCTION CONTROL monitors the assembly line using dials and gauges. Production Control is not an entity by itself; it is totally dependent on the existence of the Assembly Line in order to measure performance. In IT terminology, this is Project Management. However, this brings up an important point; without a defined methodology, Project Management is an exercise in futility. It measures nothing. Only if a defined mode of operation exists can dials and gauges be effectively applied.

TECHNIQUES, as mentioned, represent ways for performing specific tasks ("how to"). A variety of techniques may be used on the Assembly Line. Obviously, it would be counter-productive to use a technique at the wrong time on the Assembly Line. This means the effective use of techniques is dependent upon a defined Assembly Line.

TOOLS implement techniques. Tools provide mechanical leverage for performing a specific task. In this sense, it is an extension of a technique, and like the technique, tools must be deployed at the proper locations along the Assembly Line. This is the reason why many software engineering tools are failing; not because they are bad tools, but simply because companies have not defined their Assembly Lines (methodologies) and haven't specified when the techniques and tools are to be used.

What this highlights is that a methodology is the focal point within a development environment. Without a defined methodology, Project Management will be ineffective, and design techniques and software development tools will be misapplied. Productivity will be low.

(continued on page 3)

(continued from page 2)

METHODOLOGY CRITERIA

Since a methodology is critical to the success or failure of a development environment, it is important to be able to differentiate between a methodology, technique and tool. The generic properties of a methodology include:

1. DEFINES THE STAGES OF WORK (a work breakdown structure normally consisting of phases, activities and tasks). The stages of work defines the "5-W's" (Who, What, When, Where, Why). The synchronization of work is needed to define direction and is provided by the precedent relationships between the various steps in the methodology. Defined duties and responsibilities provides insight for performing the work and methodology standardization improves communications between workers.

2. MEASURABLE - The stages of work can be evaluated in terms of how long it takes to perform them and how much they cost to perform. Further, criteria is provided to substantiate completion of deliverables thereby assuring the development of a quality product.

3. TECHNIQUE AND TOOL INDEPENDENT - various techniques and tools can be deployed as required.

4. PROJECT MANAGEMENT INDEPENDENT - can work with or without a Project Management system. For example, an Assembly Line can still function without Production Control, but not vice versa.

If the methodology you are evaluating does not match this simple criteria, it is not a methodology and probably some form of technique.

TYPES OF METHODOLOGIES

Of the "process management" methodologies, there are fundamentally three types:

LINEAR "WATERFALL" METHODOLOGY (sometimes referred to as "Life Cycle") - this is perhaps the best known of the methodologies. Various interpretations of this approach have been published for several years, both commercially and public domain. Fundamentally, it a sequential process where the design of an application moves from the general to the specific; for example:

1. FEASIBILITY STUDY
2. DESIGN
3. PROGRAMMING
4. TESTING

5. REVIEW

The problem with this approach has been its orientation towards computer software and not on total systems. But the biggest pitfall has been its sequential orientation which tends to prohibit parallel development.

SPIRAL DEVELOPMENT - this approach is based on the premise the development process is evolutionary in nature (which, in fact, it is). The concept is to initially design a program, then add additional phases of work to constantly revise the program to enhance its features. From a Project Management perspective, the problem with this approach is that the project never ends.

PRODUCT DEVELOPMENT - as proposed by MBA, this approach uses elements of the other two methodologies, with the added nuance of using a product orientation as the basis for the development process. Under this approach, a system is viewed as a product. Consequently, it can be designed in the same manner as any other product. For example, when a product is being designed (such as an automobile), the overall assemblies are first designed (such as the body, chassis, engine, etc.). After this phase, each assembly is designed by teams of engineers who refine the design of each assembly into sub-assemblies and parts. All of this occurs as parallel phases. MBA advocates the same approach for systems development. An initial phase is used to design the architecture of the system, followed by succeeding parallel phases to refine the design. This is the best approach for parallel development.

INDUSTRIAL ENGINEERING

In an engineering/manufacturing environment, the responsibility for defining the work environment is normally delegated to an "Industrial Engineer." It is the Industrial Engineer's responsibility to define the Assembly Line, the types of people and skill sets required to perform the work, and the deployment of techniques and tools to be used on the Assembly Lines. Industrial Engineering is a recognized profession in the engineering/manufacturing world. A comparable position is required in the information systems world.

Unfortunately, most development methodologies purchased today are evaluated by the wrong people. Quite often, the evaluation of a methodology is delegated to programmers or technicians who are more enamored with the latest software design technique or tool than in defining a managed development environment. This is like Henry Ford allowing the UAW to invent the concept of

(continued on page 4)

(continued from page 3)

the Assembly Line. They simply have the wrong perspective. Someone who specializes in installing headlights doesn't necessarily have the expertise to develop Assembly Lines. True, their input can be helpful when evaluating a technique or a tool, but not for an overall development environment. This is one area where American businesses have abdicated complete control.

CONCLUSION

There are essentially two interpretations for the term "methodology" in the IT industry. One interpretation is as a disciplined process for developing information resources, from inception to conclusion. Another is as a technique for performing a specific task of work. These are subtle but significant differences, particularly if a company is analyzing their development environment. As companies have learned, it is not simply a matter of purchasing the latest software engineering tool to overcome their productivity problems. Studies show such tools are failing to have an effect in this area, primarily because they are being misapplied by the users. People looking for programming tools to bring order out of chaos are going to be sorely disappointed. This is not their forte. Rather, they represent an efficient approach for implementing design techniques. The intent of a true methodology is to define the work environment, thereby providing the ability to effectively deploy tools and techniques. To implement a methodology, a development organization needs to reorient themselves into an "Information Factory" environment, where systems and software (products) are developed in the same manner as any other engineering/manufacturing facility.

END

"PRIDE" Special Subject Bulletins can be found at the "PRIDE Methodologies for IRM Discussion Group" at:

<http://groups.yahoo.com/group/mbapride/>

You are welcome to join this group if you are so inclined.

"PRIDE" is the registered trademark of M. Bryce & Associates (MBA) and can be found on the Internet at:

<http://www.phmainstreet.com/mba/pride/pride.htm>

Copyright © MBA 2005. All rights reserved.