

**TITLE: "THE DICHOTOMY OF SYSTEMS DEVELOPMENT MANAGEMENT"**

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@phmainstreet.com](mailto:timb001@phmainstreet.com)

Yahoo! IM: littleleaguerng

WWW: <http://www.phmainstreet.com/mba/>

Since 1971: *"Software for the finest computer - the Mind"*

*"A Systems Development Manager speaks with a forked tongue."*

*- Bryce's Law*

**INTRODUCTION**

In order to be a Systems Development Manager, you have to be a lot of things: front man, educator, mentor, sage, politician, etc. Oh yea, something else, you also have to be a hypocrite. In order for you to survive in today's corporate world you have to say one thing to your superiors and staff, but then do something entirely different in practice. Let me give you some examples:

- On the one hand, managers know it is important to do the upfront work in systems design, e.g., current systems analysis, information requirements definition, establish the proper systems architecture, etc., but on the other hand, they encourage their staff to rush to coding without first thinking the problem through. This is because programming is a much more tangible task than systems analysis, thus providing demonstrative evidence to the end-user that the project is progressing. Managers rationalize this by claiming they work in a pressure cooker and, as such, "We don't have time to do it right."
- On the one hand, managers claim they want standardization in their work effort (to get everyone communicating and working on a common level), but on the other hand, standards are thrown out the window the moment push comes to shove.
- On the one hand, managers want interchangeable workers who can easily pick up where another worker leaves off, but on the other hand, they are unwilling to train the workers to a uniform and consistent skill level.
- On the one hand, managers understand the virtues of sharing and reusing information resources, e.g., integrate systems and eliminate duplication, but on the other hand,

bo mechanism is implemented to check for redundancy. Consequently, systems lack integration, data integrity is questionable at best, and systems are routinely rewritten over and over again, representing redundant work effort.

- On the one hand, managers know their systems and software should be properly documented in order to expedite maintenance and future modifications/improvements, but on the other hand, documentation is one of the first things sacrificed when a project is delayed. It is assumed the system will be documented afterwards; unfortunately, it never is. Instead of documentation being viewed as a vital working tool and a byproduct of design, it is viewed as an inconsequential and burdensome task.
- On the one hand, managers claim they all want quality workmanship, but on the other hand, they are unwilling to impose the required discipline, organization, and accountability to implement a quality environment.
- On the one hand, managers promise to implement projects on time and within budget, but on the other hand, this seldom occurs as project management is superficially implemented in their organizations.
- On the one hand, managers want their systems to be portable, thereby making them independent of their machine environment, but on the other hand they fall prey to the latest technical promise and develop systems tailored to a particular physical device.

**THE "PILL" APPROACH**

Obviously you cannot have it both ways. You must take a position and implement accordingly. Basically, there are two alternatives: a tool-oriented approach or a management-oriented approach. On the surface, the tool-oriented approach appears to be the least painful as it doesn't require any political maneuvering or management chutzpah. I refer to this as the "pill" approach for problem solving. Let me explain. Years ago, comedian George Carlin talked about how America's drug culture came about. It was his contention that we are taught to pop a pill at an early age such as with children's vitamins. As we get older, it thereby becomes natural for us to pop a pill for whatever woes we experience. It may not be the right treatment, but we believe it is the most expeditious approach for satisfying our problem. Ask any doctor, and they'll tell you placebos can work wonders in certain situations, but they also know they have limitations and are no substitutes for suitable medical treatment.

*(continued on page 2)*

*(continued from page 1)*

This "pill" phenomenon is no different than purchasing a new development tool that claims to solve all of your problems. You know what? There is no such tool. It doesn't exist, it is a myth that rates up there with the Easter Bunny and the Tooth Fairy. Nor will it ever exist. The reality is that we will always need a variety of tools that address different aspects of the development process. And understand this, in software alone, there are hundreds of ways to skin a cat; thanks to different programming languages, design and data base techniques, etc. As much as we hate to admit it, systems development can be a lengthy process and anytime we try to short stroke it with the latest tool du jour, we only cause headaches later on. You cannot keep applying Band-Aids when major surgery is required.

On the other hand, there is the management-oriented approach. This requires structure, discipline, and responsibility; three ugly words in today's systems development landscape. But before we tackle anything of substance, it is essential that such an environment be created. Can you imagine designing a bridge or a building without such disciplines in place? Hardly. Why should systems be any different? What is needed is the establishment of a professional attitude among the staff; whereby a system is viewed as a product that can be engineered and manufactured like any other product. Once we have the proper perspective, we can organize the staff accordingly and create a concerted development effort. True, we will use pertinent tools in the development process, but we have to recognize that tools will come and go, and are dynamically applied. It is the process of building systems that should be regarded as a precursor to the application of tools, our methodologies. Only when we can reshape our homogeneous development environment into a homogeneous environment will we be able to act as true professionals. Unfortunately, this requires some management fortitude, something that is in short supply these days. A lot of people, throw up their hands and say this is not possible due to the management realities of today and resign themselves to doing small insignificant applications, hence the dichotomy mentioned earlier.

But let's consider what we have done over the last thirty years. We have tried CASE tools, 4GL's, program generators, prototyping aids, report writers, BPR tools, DBMS packages, programmer workbenches, etc. True, we have some great application development tools, but if they are so good why are we still experiencing problems? The answer is obvious; we have abdicated management control over our systems development environment.

Now is the time for systems development managers to stand up for their departments, their profession, and themselves, and act like managers. All of the things you claim to want and support are within your grasp, as long as you start behaving more like a manager as opposed to a pawn for the latest programming gizmo. Face it, you have been seduced and abandoned by your tool vendors. You can talk the talk, but can you really walk the walk?

## **CONCLUSION**

Managing a systems development environment requires someone skilled in the fundamentals of management, is not intimidated by technology, and has a more global view of systems. Some of the best systems development managers I have met over the years were people who didn't have a computer background, but, instead, came from a user area and were not intimidated by the latest technical gobbledeygook. They were pragmatists who were results oriented and implemented a management environment where development terminology and concepts were standardized and consistently applied. Frankly, some of the best candidates for the position of systems development manager, are the sharpest critics of the department. Companies then said, "*Okay, put up or shut.*"

Unfortunately, most of today's development managers are the antithesis of what I have just described. If the choice is between quality and speed, they will always take speed. The point is, you can have both without sacrificing either, it just requires some proficiency in management.

All systems development managers know what the cure is, they are just not willing to take it. But understand this, you cannot have your cake and eat it too.

**END**

### *About the Author*

*Tim Bryce is the Managing Director of M. Bryce & Associates (MBA) of Palm Harbor, Florida and has 30 years of experience in the field of Information Resource Management (IRM). He is available for training and consulting on an international basis.*

*"PRIDE" Special Subject Bulletins can be found at:*

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

*They are also available through 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.*

*The "Management Visions" Internet audio broadcast is available at:*

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

*Also, be sure to read Tim's Blog at:*

<http://blogs.ittoolbox.com/pm/irm/>

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

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

Copyright © MBA 2006. All rights reserved.



Since 1971: *"Software for the finest computer - the Mind"*