

TITLE: "THE LANGUAGE OF SYSTEMS"

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"*

*"Systems will fail more for the lack of administrative procedures than well written computer procedures."
- Bryce's Law*

INTRODUCTION

A few years back, I was at a Comdex show exhibiting our "PRIDE" Methodologies for IRM and gave a brief overview to an inquisitive attendee. He listened to me patiently, but at the end asked me pointedly what language "PRIDE" was written in. He looked at me dumbfounded when I told him it was written in English. I guess he honestly thought "English" was some new programming language. I could have gone on with the charade and said that it was, but honesty got the better of me and I explained to him our corporate slogan, *"Software for the finest computer - the Mind."*

The language of systems is no different; No, it is not C++, Java, COBOL, etc., but rather simple English (or whatever your native language happens to be). In my last Bulletin (No. 37 - "Systems: What's in a Word?" - Aug 15, 2005), I went into length about the differences between Systems and Software, the two are simply not synonymous. Whereas systems include business processes implemented by human beings, computers and other office equipment, software is simply instructions for the computer to follow. Systems are for people who must also take an active role in its execution. In fact, systems will fail more for the lack of people procedures than they will for well-written computer software. There are more people procedures in a system (we refer to them as "administrative procedures") than most people imagine. Overlooking their role in a system is a serious error. Let me give you an example...

We had a large manufacturing customer who designed a new "state-of-the-art" shop-floor control system whereby they wanted to spot errors along the assembly line and then quickly react and correct the hiccup. From a software perspective, it was a well thought-out and elegant solution coupled with an integrated data base. There

was just one problem; it didn't work. Consequently, we were called in on a consulting basis to try and determine what was wrong with it. We carefully examined the architecture of the system overall, not just the software, and quickly found the problem; Whenever an error occurred on the shop-floor, an error message was displayed on a computer screen for the shop-floor supervisor to act on. Unfortunately, nobody told the supervisor about the computer screen, the messages, or procedurally how to respond to it. We wrote a simple administrative procedure for the supervisor who then read and responded to the errors properly and the system then ran perfectly.

As my example demonstrates, clearly written administrative procedures immeasurably improve the processes of system implementation and operation. Often, preparation of these procedures has been eliminated in order to expedite system start-up. Experience has shown this approach can cause considerable expense, frustration and problems.

WRITING FOR PEOPLE

Even when administrative procedures are considered, they are often sloppily written in an inconsistent manner. Unlike the computer who will do anything you instruct it, right or wrong, writing for the human being is actually more difficult. People are more emotional and can be lazy and uncooperative at times. Writing for people, therefore, can be an arduous task. Instituting writing standards can materially help in bringing about consistency to this task and should be encouraged.

Whenever writing administrative procedures, they should answer these basic questions for the end-user:

- What is the purpose of the procedure?
- Who should perform the procedure? When?
- How should the procedure be accomplished?
- What is needed to accomplish the procedure?
- What are some examples?
- What should be done after the processing is accomplished?

As any writer will tell you, you must write in terms your audience will understand. As such, you should consider the intelligence level of your audience. For example, most newspapers in the United States write for people at

(continued on page 2)



(continued from page 1)

the 6th grade level. You may possess a sophisticated vocabulary, but does your audience? When it comes to writing administrative procedures, write so your audience can understand the instructions and implement accordingly.

PLAYSCRIPT

One of the most effective techniques for the preparation of procedures, is the "Playscript" technique as developed by Leslie H. Matthies, the legendary "Dean of Systems." There are basically three parts to a Playscript procedure:

1. PURPOSE SECTION - Containing the Business Purpose of the procedure.
2. SETUP SECTION - Listing all of the inputs, outputs, and files that will be used during the execution of the procedure.
3. OPERATION SECTION - Listing all of the instructions required to perform the procedure. Each operation is described using action verbs and nouns. Use the following guidelines when preparing the instructions:

- Avoid needless complexity in word choice and sentence structure. Express an idea in the simplest possible way.
- Begin each Operation with a verb; samples include:

Accept	Establish	Perform
Analyze	Estimate	Place
Assign	Evaluate	Prepare
Assist	Execute	Print
Attach	File	Process
Authorize	Finish	Receive
Begin	Forward	Recommend
Check	Gather	Record
Change	Get	Report
Compare	Give	Retain
Complete	Identify	Return
Conduct	Indicate	Review
Correct	Initial	Schedule
Count	Insert	Select
Create	Install	Sign
Define	Interview	Sort
Delete	List	Start
Describe	Locate	Store
Design	Log	Submit
Determine	Maintain	Survey
Develop	Mark	Terminate
Direct	Move	Total

Discuss	Notify	Verify
Distribute	Obtain	Write
Enter	Order	

- DO NOT begin the first sentence of the operational step with a conditional clause, such as "if," "when" or "should." Begin the sentence with "compare" or "evaluate" as a verb, followed by sub-clauses; for example:

COMPARE THE VALUE OF DD-1 TO DD-2:

A. IF DD-1 IS GREATER THAN DD-2, GO TO STEP 4.

B. IF DD-2 IS GREATER THAN DD-1, GO TO STEP 16.

- To make the narrative more understandable to the reader, minimize the number of references to outside narratives.
- Avoid colloquial expressions (jargon-e-z-e).
- Use present tense. Avoid future tense (using "shall" or "will") unless you are actually referring to a future event, or you want to make a strong promise or threat.

Incorrect: Systems Engineering will then prepare a cost/benefit analysis.

Correct: Systems Engineering next prepares a cost/benefit analysis.

- Use functional titles rather than personal pronouns. For example:

Systems Engineering vs. Systems Engineer
Project Management vs. Project Manager
- Include references to numbered examples whenever possible.
- Avoid Standard Operating Procedures (SOP); you can make reference to them, but it is not necessary to go into detail over a commonly accepted practice.

(see example on page 3)

(text continued on page 4)

EXAMPLE

```
-----
OD-90162                PLAYSRIPT INSTRUCTIONS          "PRIDE"-ISEM
-----
NOV 17, 2004           RELEASE 1.2.0                 PAGE 2.5.1
```

```
*****
*   PROCEDURE OP-08-03 ORDER-PROC FILE UPDATE SESSION   *
*****
```

PURPOSE-

The purpose of this administrative procedure is to enter and validate data through a computer terminal using the Order Processing Window. Entries are made in accordance with the prompts and messages associated with the panel. HELP text is also available upon request.

SETUP-

- 1. "ORDER PROCESSING WINDOW" ID-00100
- 2. "ORDER PROCESSING WORK FILE (MANUAL)" FD-00234

OPERATIONS-

- ORDER-PR
- 1. Logon to computer using the standard logon procedure.
 - 2. Access "Order Processing System" screens by double-clicking on the "Order Processing" icon on the computer desktop.
 - 3. Select the "Order Processing Maintenance" icon to invoke the "Order Processing Screen."
 - A. Enter your PASSWORD when prompted by the computer.
 - 4. Select the type of data entry you wish to perform from the window's action-bar:
 - A. If you wish to enter a customer's order, select the "Orders" pull-down-choice from the "Customer" action-bar-choice. Go to step 6 for additional instructions.
 - B. If you wish to query the status of a customer's order, select the "Status" action-bar-choice from the "Customer" action-bar-choice. Go to step 8 for additional instructions.
 - C. For all other queries, proceed to the next step (5).

```

/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
                        * * * END OF OUTPUT * * *
```

(continued from page 2)

"HELP" TEXT

Help text is normally associated with interactive processing at the computer screen where the user requires instructions from the computer to guide them through processing. This is needed to answer both common and technical questions regarding processing. Development of help text is almost a prerequisite for all PC processing.

There is basically three areas requiring HELP text:

1. For Window/Screen Processing - providing tutorial describing the purpose of the screen, who it serves, and the basic processing action of the screen. In the industry, this is often referred to as "general help" or "extended help." System, sub-system and procedure narratives are useful for this purpose, as well as "Playscript" instructions.
2. The user will also require help in making field entries. This includes acceptable values, what the values mean, and the physical characteristics of the entries (e.g., length).
3. Special function keys - like field entries, HELP text should be provided to explain special keys (function keys or combination keys (e.g., CTRL + C)).

Help indices are also very useful for reference purposes, such as by subject, by field entry, by keys, etc. Fortunately, standards are emerging in the industry for writing help text which can be implemented using such things as MS Windows HLP files and Web files (HTML),

There is absolutely no incompatibility between "Help" text and "Playscript." In fact, "Help" text makes a convenient vehicle for accessing "Playscript" instructions.

WHEN SHOULD YOU WRITE ADMINISTRATIVE PROCEDURES?

If you are following a top-down design approach, such as the "PRIDE"-Information Systems Engineering Methodology (ISEM), administrative procedures can be written upon the completion of sub-system design (where business processes are defined). This also means the administrative procedures can be developed in parallel with software design. Some people have difficulty imagining this. I do not. I'm a firm believer of having your systems documentation completed before system startup. This greatly assists in educating the end-user and helps overcome problems during startup. Whether you believe in

pre-documentation or post-documentation, it has to be completed nevertheless.

CONCLUSION

In summary, administrative procedures should receive the same attention as procedures written for computers. Without these procedures, which represent the critical human interface to systems, a well designed and programmed system may be useless.

In reality, there is little difference between an administrative procedure and a computer procedure. The only difference is the "actor" assigned to perform the task. To appreciate this, one should understand how Les Matthies came about devising his "Playscript" technique. Les graduated from the University of California at Berkeley in the early 1930's with a journalism degree. This was during the midst of the Depression where work was hard to find. For a while, Les tried his hand at writing Broadway plays and became intimate with writing scripts (where actors enter, speak their lines, and exit). When World War II broke out, Les was too old for military service and, instead, was recruited by an aircraft manufacturer in the U.S. mid-west where he was charged with establishing procedures for the production of aircraft thereby expediting the development and delivery of planes to the war front. Using his writing skills, he devised "Playscript" with actors and actions which proved effective to procedurally produce aircraft.

Let's fast-forward to the 1950's and the advent of the UNIVAC I. Computer programming languages had moved from machine language to assembly languages, both of which were difficult to program in. Enter Grace Hopper who was looking for an easier and more intuitive approach to programming. As such, she invented an English language compiler called "Business Compiler Zero" (B0) which ultimately became the COBOL programming language. To do so, she modeled the language after a procedure language she was familiar with, "Playscript." Think about it. Playscript defined the environment, the files to be used and its use of verbs and nouns are easy to assimilate. What this ultimately means is that "Playscript" is the mother of all third generation procedural languages and that our premise, that there is little difference between an administrative procedure and a computer procedure, is true.

In the end, it all comes down to verbs and nouns - the Language of Systems.

END

(continued on page 5)

"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.

"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.