TITLE: "DEFINING INFORMATION REQUIREMENTS"

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"*

"If an information requirement is stated improperly to begin with, then everything else that follows will be incorrect."
- Bryce's Law

BACKGROUND

As our Bryce's Law above suggests, defining information requirements is perhaps the most critical aspect of systems development, yet it is the Achilles' heel of most developers today and an area where they typically spend the least amount of time. Consequently, considerable effort and money is lost developing an elegant system for the wrong problem and, as such, developers spend in inordinate amount of time re-writing systems until they get it right.

In terms of specifying information requirements, are the problems we are having today any different than what they were in yesteryear? Hardly. Let me take you back to an interesting report produced by the *EDP ANALYZER* in July 1977 (Vol. 15, No. 7) entitled, *"Getting the Requirements Right."* The report highlighted development failures based on poor requirements definition. An analysis was provided of the types of problems when specifying requirements:

	% of Total Errors
Incorrect Requirements	34%
Missing/incomplete/inadequate	24%
Unclear/ambiguous	22%
Inconsistent/incompatible	9%
New/change	3%
Outside scope of project	4%
Typographical errors	4%

It may be 27 years later but I would wager you most of today's projects suffer from the same type of problems. Many methodologies and university professors are vague on this subject and think defining information requirements is something we should intuitively know, consequently confusion and inconsistencies arise. Some see information requirements as:

* A set of data requirements for a system.
* Screen/report layouts.
* Programming specs.
* Machine requirements.

All of these items have their place, but they are most definitely not information requirements. Instead, they are by-products resulting from carefully written information requirements.

Such inconsistencies in interpreting "information requirements" leads to the communications gap between developers and end-users, along with the alienation between the two parties.

"They (the users) don't know what they want," is a common lament among developers.

"Why can't they give me what I want?" counters the end-user.

CONCEPTUAL FOUNDATION

The inconsistencies described above are due to the lack of a strong conceptual foundation regarding the nature of information. In "PRIDE" Special Subject Bulletin #2, we discussed the inherent nature of "data," here we will focus on "information." The two are most definitely not synonymous. Data is simply the digital representation of a fact or an event and includes primary values (e.g., "Customer Number," "Name," "Unit Price," etc.) as well as generated values (e.g., "Net Pay," "Total Ordered," "Percent Complete," etc.).

Information, on the other hand, is defined as *"the understanding or insight gained from the processing and/or analysis of data."* From this, we arrive at the "PRIDE" formula:

Information = Data + Processing

This means the two variables of information are "data" and "processing." If either of these variables change, the information will change as well.

Information is a consumable commodity used by human beings to support specific actions and/or decisions of the business. Information is not stored, it is produced. Data, on the other hand, is collected, stored, and processed in

(continued on page 2)

(continued from page 1)

a specific manner to produce information. The difference between the two are subtle but significant. An end-user has information when he/she can "act" on it. If they cannot act on it, they have nothing more than data. In this business, there is a tendency to produce too much data and not enough information. This hints at where to begin in Information Requirement (hereinafter "IR") definition: understanding the consumer.

UNDERSTANDING THE CONSUMER

Before we can specify information requirements, we must first understand the nature of our business. All enterprises, be it a commercial company or a non-profit organization, consist of business functions that describe the mission and operation of the business. Business functions are scopes of duties and responsibilities, each consisting of specific skill sets required to implement the function. Functions can be organized into a simple three level hierarchy depicting the Policy, Control, and Operational functions of the business (more on this later). Each function deals with one or more "objects" of the business. An "object" represents the facts and events of the business to be operated or controlled. They can be as tangible as a Customer, a Product and an Employee, or as intangible as an Order, a Shipment, or a Purchase.

Business functions are implemented by Organizational Entities representing jobs as implemented by human beings and machines. Organizational charts are used to represent superior and subordinate staff/line positions. As an aside, both functional and organizational charts make a convenient roadmap for specifying information requirements. Defining the enterprise model is the forte of the "PRIDE"-Enterprise Engineering Methodology (EEM).

Only when the systems engineer can walk in the moccasins of the user, does the engineer have the right to build a system for the user. After the developer knows the business, he/she is now ready to specify the information requirements of the user. So where do we begin? Let's start at the "Actions and/or Business Decisions" to be supported. Ask the user, *"When I give you the information you want (regardless of its form), what will you do with it?"* This will inevitably lead to an interesting dialog between the user and the analyst. Here, the "receivers" of the information are specified and their respective actions and/or decisions are detailed. At this time, it is not necessary to ask how the end-user wants the system implemented (that's actually none of his/her business; you, the developer, are charged with this responsibility). Stay on target, concentrate on the "Actions and/or Decisions" to be supported and when they have to be made (timing).

This leads to an interesting attribute of information: it is a perishable commodity that only has value to the end-user at a given moment in time. This is because the "actions and/or decisions" of the business must be performed at specific moments in time. Timing has three distinct attributes:

Frequency - specifying how often information is needed, such as:

1D - Once a day
6D - Six times daily
1M - Once a month
1Q - Once a quarter
R - Upon Request ("any time I feel like it")

(And No, not everything has to be "upon request" - such things as corporate earnings are reported periodically to the government, as is payroll, sales trends, shipping assignments, student grades, etc., etc.)

Offset - specifying the beginning of the cycle. For example, for a Monthly Sales Report, it will most likely begin on the last day of the month (as would payroll calculations). For a frequency of "Upon Request" there is no schedulable offset.

Response Time - specifying the maximum elapsed time from start to finish. (Note: this is not a measure a machine throughput).

The Systems Engineer must be wary of trying to specify too much (or too little) in a single requirement definition. As a general guideline, requirements should be separated and sorted according to:

* Timing (Frequency, Offset, Response Time)

* Common "Receivers" of the information (e.g., separate the sales function from the manufacturing function).

* Common purpose - here, information can be conveniently sorted based on the types of actions and/or business decisions; for example:

Policy information - supports executive actions/ decisions

Control Information - implements policy and oversees operations

(continued on page 3)

(continued from page 2)

Operational Information - represents the daily routine affairs of the business.

By this time, the Systems Engineer should be able to piece together the various Information Requirements into separate descriptions consisting of:

NAME (with some sort of control number to uniquely identify it, e.g., IR-12345)

DESCRIPTION (text)

- Business Purpose (specifying the knowledge needed; and "why" it is needed)

- Actions and/or Business Decisions (specifying "who" must perform the action and "when")

- Benefits (defining the tangible and/or intangible benefits of having this knowledge)

NOTE: This textual description validates the need for the information.

TIMING - specifying Frequency, Offset, Response Time

TYPE INFORMATION - Policy, Control, Operational

RECEIVERS (representing the "consumers" of the information)

Are we done? Nope. But we are getting close. After we have the IR Definition in this state, we can now determine:

OUTPUTS - Since information is ultimately conveyed through some form of output, be it a screen, paper, audio, or other, now we can begin to determine what outputs (one or more) will be needed to satisfy the information requirement (be it a new or existing output).

DATA REQUIREMENTS - Specifying all of the primary and generated data elements required to support each information requirement. Pay particular attention to generated values, as they have to be traced back to all of the primary values needed to compute the generated items (even if they will not ultimately appear on an output).

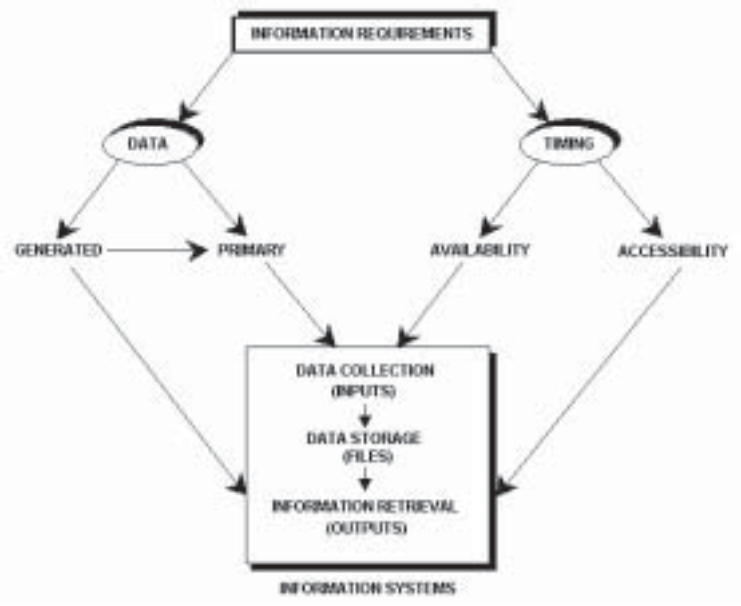
SYSTEMS - Specifying either a new system to implement the requirement or an existing system (which may require modification).

With the Information Requirement now fully documented, now is the time to carefully review it with the end-user for clarity. It is very important the end-user agrees to the textual description of the requirement (representing the business case for the information), the timing, and the required data elements (including the computation of generated values). Corrections should be implemented as required before proceeding.

INFORMATION DRIVEN DESIGN

After the Information Requirements have been defined in this manner, it becomes a relatively simple manner of designing the system and supporting software. The data and timing requirements of the IR Definition will ultimately dictate how data is collected, stored, and retrieved (processing).

(continued on page 4)



(continued from page 3)

We refer to this approach as "Information Driven Design" because it hinges on the requirement definition. If the information requirement is correct, the resulting design will be correct. If the requirement is wrong, no amount of elegant programming will be able to solve the problem.

Under "Information Driven Design" the emphasis is to work backwards to define the necessary business processes (sub-systems), complete with inputs, logical files and outputs. Following this, each business process is detailed in terms of its procedural work flow, from start to finish (forward design). By the time we get to software engineering, the programs should be fully specified in terms of data and processing requirements.

SUMMARY:

Let's review the basic attributes of information:

* Information represents the intelligence or knowledge needed to support the actions and/or decisions of the business; it is not synonymous with data.

* Information is a "consumable" commodity used by human beings.

* Information is a "perishable" commodity; it only has value to the end-user at a specific moment in time.

* Information Requirements dictate data and processing specifications.

* If the information requirement is incorrect, that everthing that follows will be incorrect.

Devising a standard approach for defining Information Requirements means we can begin to analyze them for consistency and quality. It also means we can start to communicate on a common level, not just between developers, but with end-users as well. Our approach to defining information requirements has helped many "PRIDE" users over the years. Where they had been floundering in the early stages of a project due to the lack of a consistent approach to IR definition, now they can accurately define business problems and opportunities and carry on a constructive dialog with the end-user. Even better, they are addressing the "right" requirements in systems development.

Within the "PRIDE" Methodologies for IRM, there is considerably more narrative describing information requirement definition and "Information Driven Design"; for additional information, see:

"PRIDE"-ISEM, "Information Driven Design Concept"
http://www.phmainstreet.com/mba/pride/ismeth.htm#design

"PRIDE"-ISEM, Phase 1 - "Defining Requirements"
http://www.phmainstreet.com/mba/pride/is10.htm#requirements

Information Requirement Worksheet
http://www.phmainstreet.com/mba/pride/iw007.jpg

Let us never forget that...

"No amount of elegant programming or technology will solve a problem if it is improperly specified or understood to begin with."

- Bryce's Law

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