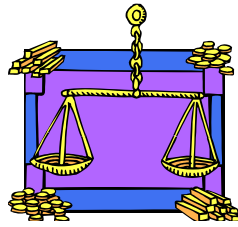


**The Positive Side of Gold Plating—
Introducing the Concept of Scope Interpretation Bandwidth**

*By Sharyn A. Brotz, PMP, Managing Director
Pragmatic Systems International Corporation
Key Biscayne, Florida
(305) 365.9400 (Phone); (305) 365.2659 (fax)
Sharyn@pragmatic-systems.com
www.pragmatic-systems.com*

Sharyn A. Brotz is Managing Director of Pragmatic Systems, a consulting firm that specializes in technical project management particularly for software development projects. The address is 77 Crandon Blvd., Suite 6-D, Key Biscayne, Florida 33149



ABSTRACT: Software Development Projects should often encourage the creative thinking associated with interpreting the business requirements to develop a more generic and typically more comprehensive technical solution. Rather than considering this increase in functionality, gold-plating, from a totally negative perspective, there ought to be some level of encouragement for it, up to an amount pre-determined by a new metric introduced in this document, that of Scope Interpretation Bandwidth.

Key Words: Gold-plating, Scope Management, Scope Control, Scope Interpretation Bandwidth

So much of the principles of Project Management as prescribed by the Project Management Institute through its key documents of Project Management Book of Knowledge (PMBOK) and the approved materials really provide excellent guidelines to follow for effective project management. However, the arbitrary guiding principle that suggests that “Gold plating” is a bad idea is one that this article will dispute, especially with regard to software development projects. Rather than considering the negative connotation of gold-plating, consider the value and probable inevitability of gold-starring in a properly managed software development project.

The term “gold plating” was first introduced to me in Rita Mulcahy’s PMP Exam Prep Guide, on page 95. Rita did an excellent job summarizing the key components of the PMBOK documentation and I totally thank her for assisting me through the study efforts. The paragraph that I wish to quote follows:

“Gold plating: PMI does not recommend giving the customer extras (e.g. extra functionality, higher-quality components, extra scope of work or better performance). Gold-plating adds no value to the project. Often such additions are included based on the project team’s impression of what the customer would like. This impression may not be accurate. Considering that only 26% of all projects succeed, project managers would be better off spending their time conforming to

requirements.” (Rita Mulcahy, PMP Exam Prep Guide, 2nd Edition, Beaver’s Pond Press, 2000, page 95)

There are so many objectionable and arguable phrases within this paragraph, especially in the context of a software development project. What makes REQUIREMENTS indicators of project success? Who wrote the requirements? In most software development projects that I have managed, requirements are written at a very high level by the business community. If we are lucky, the requirements may provide additional detail when more analytical business analysts are assigned this critical task. But in any event, requirements typically portray a functional need, addressing WHAT is desired and not HOW these needs will be provided.

In the DESIGN phase of software development, the details begin to be sorted out. A strong software developer often finds requirements to be too high a level to actually code from. Or, the requirements are so detailed that the developer may find that to satisfy the requirement literally, he/she would be compelled to develop a one-off system that could not be re-used at all. Hence, we establish the need to either develop a prototype or to drill down to design-requirements. As the design phase progresses, the phrase “the devil is in the details” begins to unfold.

Most people recognize and agree that custom-developed software is often the most expensive of software options. As a result, the movement over the past ten years has been toward developing re-usable code, object oriented technology that enables re-use on a more generic level. Well trained developers are geared toward developing solutions or finding obtainable plug-in components that are as generic as possible to gain greater economies for potential re-use by other projects. Therefore we have an inherent conflict between following the Requirements verbatim or providing additional value by providing a more generic solution.

In the PMBOK, there appears to be a softening of the anti-gold-plating principle in the recognition of the need for the scope statement to be revised or refined.

“The scope statement provides a documented basis for making future project decisions and for confirming or developing common understanding of project scope among the stakeholders. As the project progresses, the scope statement may need to be revised or refined to reflect approved changes to the scope of the project.” (PMBOK, 2000 Edition, Project Management Institute, 2000, Page 56)

The problem, however, with this statement, is that there is an inferred administration of changes to be revised upon a change control process. Where in the entire project management process is good sound judgment of EXCEEDING expectations and DELIVERING more than what was requested?



The statistic that Rita Mulcahy sites that only 26% of projects succeed leads me to ask the question—of the 74% of failed projects, how many failed because of providing additional value and additional capabilities? How many actually failed due to poorly written requirements and insufficient utilization of creative energies on the part of a strong technical project team?

The software development process is more of an art than a science. The goal associated with not “gold-plating” a project may be applicable in fields that have very definable, repeatable processes. For example, in a construction project, it may be entirely possible to accurately establish building requirements and estimates and manage exactly to them. But if the project is one of software development, and the processes are INDEED repeatable as they have been done exactly this way before, then WHY DO THE PROJECT AT ALL? Just buy the existing application!

Conversely, if the application being developed is a new concept and will be based on new technology, then there is no assured guarantee of the initially produced estimates. There are unlikely to be any true “experts” in a field where the technology has just been introduced, and not mature enough to have garnered sufficient experience in its use. Further, with the decision to endeavor to develop something new and different, the more generic the code is, the more

likely it is to be beneficial not only to this project but to other development efforts that would follow.

Shall we constrain our creative developers to follow the requirements so closely and literally that all possible additional peripheral benefits are suppressed for consideration? Suppose the requirement calls for 5 clearly specified hard-copy reports; shall we provide just those reports and ensure that no additional data fields are available for further analysis? Shall we not even advise our customer of the potential of drill-down online reporting (albeit with an associated cost or time increase). This decision about which way to proceed in terms of honoring the scope is to introduce a new concept—**Scope Interpretation Bandwidth** which should be set forward during the scope definition phase and added to the Contingency dollars in terms of project cost management. Similar to Risk Tolerance, the Scope Interpretation Bandwidth should establish a percentage increase ceiling by which the project funds cannot be exceeded. If the project sponsor above clearly wants 5 hard copy reports and nothing more, then providing the business user with the option of more of an on-line system may be met with anger rather than contentment. However, if the Scope Interpretation Bandwidth as set forth by the project sponsor is more tolerant of interpretations for potential increases in the scope, then this provides the green light to consider options.



While this article is not suggesting that scope should not be controlled, the concept is to provide for more of a carefully coordinated choreography between the creative capabilities of the technical project team and the sponsors of the project. By setting up an upper limit in the form of Scope Interpretation Bandwidth, the expectations the boundaries for interpretation can be more clearly delineated. My contention is that by avoiding “gold-plating” without any consideration of alternatives, we may also be preventing the chance for gold-starring, and providing our customers with projects that not only MEET but indeed EXCEED expectations.