Data-Oriented Quality Solutions
BEST PRACTICE
GUIDELINE SERIES

Clarifying Project Complexity

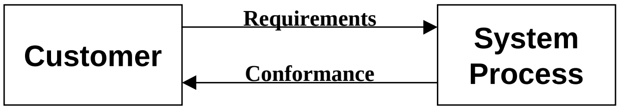
Introduction

Many information technology projects carry complexity that often goes unrecognized during the early stages of project initiation and requirements analysis. This complexity is inherent in the business system being described by the project requirements. Because such complexity has the greatest impact on the project during technical design, it is often overlooked during earlier project planning and activity. However, the complexity is a significant contributor to the effort required to build and test the system, and to the size and cost of the system. As a result, projects are more likely to conform to their schedules and budgets if these business complexities are built into the project charter and plan from the beginning.

Below are four quality-based patterns built on the idea that any system is a mechanism for turning customer requirements into satisfactory conformance. They allow simple quality management principles to be used to quickly pinpoint areas within the project scope that will experience increased complexity, and therefore higher need for project resources. These areas can then receive increased scrutiny during project planning.

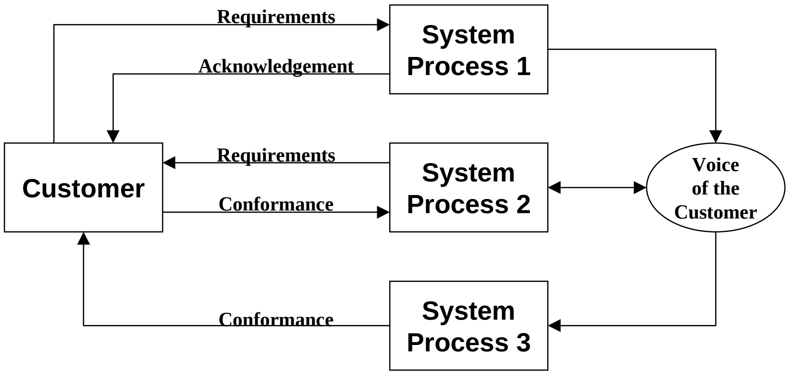
Voice of the Customer Pattern

The simplest business systems are those that directly interact with the customer, and that are able to completely conform to the customer's requirements during that interaction.



Such systems are simple because the interaction is complete when the basic transaction ends, with no need for additional controls and tracking. Such simple systems are typically only found in applications where information is being provided without charge, such as with catalogs, brochures, and reports. The requirement for information is satisfied as soon as the transaction is completed.

Most applications require much more complexity. The initial system processes are generally responsible for internalizing the customer's requirements, as in order entry or shopping cart functions. Rather than conforming to the requirements, the most such processes can do is acknowledge those requirements have been heard. Those requirements are then captured as the voice-of-the-customer in the system.



Eventually there will be a system process that takes that voice-of-the-customer and delivers conformance to the customer, as in production and shipping of an order. In between these initial and final system processes there will usually be one or more interim processes that continue to interact with the customer, further clarifying the voice-of-the-customer and working toward future conformance. What makes these interim processes much more complex is that they often place the customer in the *role* of supplier to the process - needing to conform to the requirements of the process in order to proceed. Such activity often includes processes such as account setup, and checkout or payment processing.

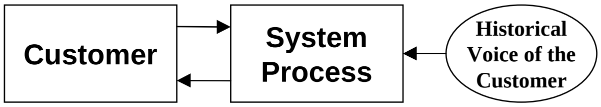
For clarifying project complexity, five questions need to be answered:

1. To what extent will the system need to internalize, track, and control the voice-of-the-customer?
2. How will each system process trigger the action of the following processes?
3. Can all of the needed steps be accomplished by this system, or will interfaces to other systems be needed?
4. If there are time delays, how will the customers identify themselves and their transactions to the system?
5. What happens to the voice-of-the-customer if conformance isn't possible or never happens?

Customer Profiling Pattern

Most systems are designed to interact repeatedly with the same customers over an extended period of time. Such systems usually include business requirements related to making the system as easy-to-use and user-friendly as possible, while increasing customer intimacy. Such requirements generally translate into some form of capability for creating and using customer profiles within the system.

The use of such a profile allows the system to know certain values that then need not be entered or provided by the customer during transactions, unless the customer chooses to override values in the profile.



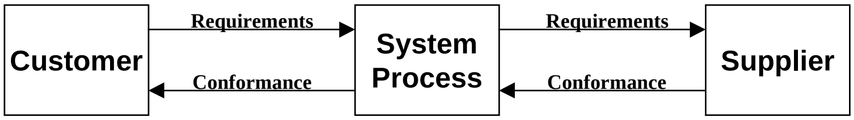
Data in a customer profile often includes contact information such as addresses and telephone numbers, account and credit information, and system feature and usage preferences. Demand for stronger support of customer relationship management in systems has increased the need for profile capabilities in system designs.

For clarifying project complexity, three questions need to be answered:

1. Will customers return to the system often enough to require profiling capability to be included in its design?
2. Where are the sources of information needed to build the profile, and does the system have needed access?
3. Will the customers need to be able to maintain their own profiles, and will privacy issues be addressed?

Process Added-Value Pattern

In order for any process to add value, it must be able to conform to the requirements of its customer using additional value obtained from its own suppliers. Usually that value must be obtained from multiple suppliers, thus increasing the complexity of the project significantly. For example, an order fulfillment system receiving requirements from customers must also receive customer information from sales, product information and prices from marketing, inventory from production, shipping information from logistics, and credit information from accounting. Even a simple brochure or catalog system must receive its content from somewhere.



The result is that supplier complexity often outweighs customer complexity when planning a system project. The number of suppliers needed by the system, and the potential system interfaces to receive the necessary data, becomes a multiplier in the estimating of resources for requirements, design, build, and test activities.

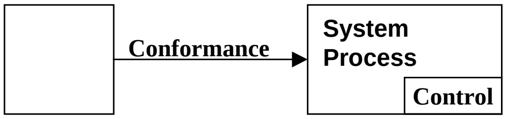
For clarifying project complexity, two questions need to be answered:

1. Are the suppliers of all the needed data, information, and materials identified and included in the project?
2. Is it generally known how each supplier will interact with the system in delivering its conformance?

Control of Nonconformance Pattern

System processes receive conformance from their suppliers, and sometimes from their customer when they've placed the customer in the *role* of supplier. As a result, the process is dependent upon those suppliers actually delivering conformance in order to add value.

For each conformance expected, the process must include a control to manage the supplier's nonconformance, either through missing or defective results. Such controls increase the complexity of the system, including such processes as error handling, back-order processing, aging of receivables, requests for pricing, etc.



For clarifying project complexity, three questions need to be answered:

1. What conformance inputs is the system dependent upon to deliver customer value?
2. How often must the system be able to continue functioning when such input is missing or incomplete?
3. What additional controls will be needed in the system to prevent or mitigate such failures?