Managing the impact of continuous change on the organization.

What light shines in yonder tunnel?

Point of View by
W. Thomas Hamlin

Table of Contents

Executive Summary 2
Operational Challenges 2
Overview 2
Indicators for Failure 3
Silos of information 3
Data redundancy and inaccuracy 3
Data Quality 4
Privacy and security 4
Customer attitudes toward service providers 4
Organizational Impact 5
Economics 5
People 5
Viability 6
Recommendations 6
Two Basic Alternatives 7
Ad Hoc Development 7
Purposeful Engineering 8
Conclusion 9
Bibliography 10

Executive Summary

Complex products, such as airplanes, ships, spacecraft, automobiles and computers offer a working metaphor for complex organizations. Both complex organizations and products are expensive to engineer and manufacture. They require maintenance and updates and on occasion decisions need to be made about them under pressure and within a compressed time frame. If the information is not readily available or is of poor quality the resulting decisions may place the organization in imminent danger. Its very survival may be a risk.

To manage this reality, models are used to simplify the complexity of the product/organization. Engineering models can be used to resolve architectural issues, as Work Breakdown Structures (WBS), parts lists, and are the basis for work scheduling. The identical principles can be used to engineer an organization for success.

Basically there are two kinds of models, implicit (assumed) and explicit (formalized). Every model contains information about the basic questions of what, how, where, who, when and why by who has the need to know. Models vary widely in their completeness and usability.

This *Point of View* reviews how to engineer a complex organization, discusses the negative impact of implicit models and offer recommendations to increase organizational survival rates in an environment with an ever-increasing rate of change.

To paraphrase an old proverb *business is always hard work success is optional*.

Operational Challenges

Overview

Models are a useful way to manage the complexity of real world products and organizations. A model retains only the data and information required for understanding and they are much less expensive to create than the real life version. For example, a blueprint offers an engineer a workspace to resolve design issues before a product is manufactured. Later, if questions are raised about the product, the model (blueprint) can be used as a valuable source of information for problem solving and resolution.

Modeling organizations is also very useful. For example, questions such as who manages department X can be resolved by reviewing an Organizational Chart.

Basic building blocks of complex models are data about things of interest to the organization and/or engineer. To be useful, the data must reflect the real world situation and must be in sufficient detail. For example, it is not good enough to show on a blueprint that ‘fasteners go here’, but it should indicate the diameter of the fastener, pitch, length and such. The same holds true for a blueprint for an organization. To use the

organizational chart example again, it should show the name of the position in the box, its place in the hierarchy, how it relates to other units and more. Managers know that anything less is a waste of paper.

To expand further, if a good customer wishes to buy a company's products, what information would a manager wish for the customer service agent to have available? Would it be sufficient to tell the customer, 'I think we have something like that, let me get back to you if I can find out.' Sound impossible, but it occurs everyday in organizations. To perform a QA on a company's Help Desk or other customer response systems, give it a call.

Comment: *A manager may have to accept this level of service within their organization, but customers don't.*

In contrast, when Apollo 13 encountered an explosion while on a mission to the Moon, NASA engineers had available to them very detailed drawings, parts lists, diagrams, and even a duplicate module to query for answers to their questions. Apollo 13 returned safely.

NASA engineers are not superhuman or super intelligent. However, NASA did make some very wise choices when architecting the organization. For example, all changes to any model had to be made through a central repository. In this way, the persons making the changes, and many changes were frequently made, were responsible for ensuring that everyone else was on the same page with the same information.

Indicators for Failure

If there is a better way, then there must be a worse way. The indicators of an organization that is bound to encounter lower than expected revenues, higher rates of customer/employee dissatisfaction, and declining market share are listed below along with explanations for each.

Silos of information

An information silo contains only information related to a individual department. Typically it is found on separate hardware platforms on a different database management system and written in a different computer programming language. For example, the marketing system may not contain any information about operations. And worse, the invoice system may not share information with the marketing system.

Data redundancy and inaccuracy

This is related to the silo situation. Many business applications may contain some customer information. With silos of information, each system must separately capture and maintain its own version of customer data. Inaccuracy and redundancy are

engineered into this kind of system. This results in confusion about which system contains the correct value.

Comment: *The resolution of the problem is not which department has the most senior executive who says their system is right or else! That is pure politics and nothing more.*

Data Quality

Building on the customer information example, the name Mr. J. Smith isn't inaccurate by itself. If customer names occur repeatedly in multiple databases (redundant data, see above) and if the business manager wishes to combine the information to meet an informational requirement, how can the manager resolve the difference between Mr. John Smith or Mr. James Smith in one system and Mr. J. Smith in another?

Comment: *A purist in data management might say that all the failure indicators, i.e., silos, redundancy, inaccuracy, quality, privacy and security are all ultimately quality issues. The author couldn't agree more. However, since the reader is assumed not to be familiar with the principles of Data Management, the author has chosen to break down quality into sub-components.*

Privacy and security

If every system maintains its own version of information, then the information is not private or secure because you can't be sure if all the access points and methods are accounted for. This point alone should drive a business re-engineering project.

Information scope

Some business managers might suggest that all information about a topic (for example customer information) should be combined and available to everyone. Worse, the offending manager's system requires information that is not legally or morally relevant. The customer does not want everyone in the organization to know their Social Security Number or other personal information. Implicitly the customer has released their privileged information on a need to know basis for a particular purpose, this is the fundamental question of data and information ownership. Abuse customer data and privacy at your own peril.

Customer attitudes toward service providers

Both in the public and private sector, customers are expecting more from their service providers. They do not expect to fill out multiple forms that require the same information at different locations or departments. The customer desires one stop shopping and recognition of their importance to the organization. The organization needs the customer's money and their good will.

Organizational Impact

This section describes the impact on the effectiveness and viability of the organization by the Organizational Change Issues listed above.

Economics

In the United States of America, one reason many organizations are created is to add value to a process in order to create a monetary differential large enough over inflation to reward the business owners with wealth. These are usually private sector organizations. An example is an automotive manufacturing firm such as Ford Motor Company or General Motors. The obvious product is an automobile. You can get to the store by foot, but the automobile makes it much easier and more efficient. Customers will exchange money for convenience.

Combining the efforts and resources of many persons to accomplish public goals creates public sector organizations. To tie in the automobile example, the obvious products are roads and highways. If a private company constructed a road they would charge a toll for each use. If a government hires contractors to build a road, the government uses taxes to pay for the construction and allows anyone with a driver's license and a licensed automobile to use the road or highway at anytime.

Both public and private organizations rely upon the individual to ultimately finance their products and services. This creates the efficacy of the individual that serves our country and citizens so well. Our laws reflect this sanctified reality for the individual and their political rights. The same individual is a citizen, a consumer, an employee and an investor. They expect the same respect, rights and privileges in all areas of their lives regardless of the nature of the organization.

It is this reliance upon the consumer that is the single most common point of failure for both the private and public sector organizations. That is, if the customer chooses another competitive product or service, an organization's economic survival is in jeopardy. In a commercial business, it is bankruptcy or being absorbed into another business that is the penalty for not understanding this principal. In the public sector, if a government does not fulfill the citizen's needs, the citizen votes for new officials who will.

People

The most confounding factor is this milieu of organizational survival is the ever-increasing rate of changes in the individual consumer's personal life. The consumer has more demands made on their time, many more sources of information, an unlimited range of choices in products and services, and there are many more consumers for an organization to deal with. They travel farther, faster and more frequently. Their family unit is in a constant state of flux including marriage, divorce, relocation, fashion, educational trends, and employment. They may speak languages other than English.

The successful organization has the ability to understand, internalize and transform the consumer's complex life by making it simpler or at least appear to be simpler. For example, once consumers have made themselves known to an organization, everywhere that consumer meets the organization, the organization knows who they are, what their preferences are, and what accounts are theirs.

If an organization does not meet these basic informational requirements they are increasing the complexity for the consumer, not reducing it. The organization may be driving the customer into the hands of a competitor or into the voting booth.

Viability

For the organization to remain viable in the future it must be well engineered. This includes its processes whether a process is manual, automated or a mixture of the two. Far too much reliance has been placed upon automating processes with computer based information systems without regard to understanding the principals of engineering.

Computer information systems are the marvel of the modern age. However, what passes for improvements in customer service or employee productivity is solely based on increases in processor speed not a fundamental improvement in the process of doing business better.

Computers are binary. They answer questions of Yes or No and only process in 1 and 0. This basic fact is hidden from the computer user because of the graphical nature of the computer interface and the depth of capabilities of the computer language. A computer cannot resolve a question with many to many relationships. For example, *many people own many cars*. For processing, a computer needs to restate this to read, *a person owns many cars*. A human being can understand and resolve many to many relationships, vague requirements and incomplete instructions, however a computer cannot.

Therefore, robust business processes that are automated (or semi-automated) must be engineered to meet the standards for computer processing in order to be viable in the long run.

Recommendations

Many readers may complain that it is too late to engineer their systems properly since they are already in place and functioning. Further new systems are already under development and it is too late to change the project plan. Other readers may be concerned about their project budgets paying for enterprise wide solutions when they are only chartered for building a departmental system.

All these objections are perfectly valid. Everyone understands that many cows are out of the barn. However, we also all understand that new systems are coming online all the time and more are in the planning stages. If everyone does not take the time to think and plan for the future, the organization may not survive long enough to complain about retirement.

Two Basic Alternatives

The best of all worlds is to carefully engineer an information system from a green field to be robust, flexible, and oriented toward continuous change in the environment. However, this may not be realistic since the system may already be in place. Existing systems can be re-engineered although it is more expensive and time consuming to do so. In the long run, it may be well worth the effort.

Purposeful engineering retains the most effective approach. It is being used in this country and around the world to build very complex critical systems. To use the Space Shuttle as an example, it has the same basic design (reusable) for over twenty years. In addition, many new innovations have been incorporated easily because it was so well engineered and documented. NASA does not have to guess about the design elements of the shuttle. There are no ad hoc systems contained in it. The flexibility stems from the detailed data and information that is available to current and future engineers.

All systems can meet these high standards.

The second alternative is to develop in an ad hoc manner, Ad hoc meaning 'as needed' or only with current system requirements in mind.

Ad Hoc Development

This is the 'Fire, Ready, Aim' method that has been such a failure throughout the years. This method begins with the programmer coding a new module or rewriting an existing program without proper requirements definition, analysis, or documentation. This method focuses on using the most current technologies and programming language features because they are interesting to the development staff.

Ultimately, the programmer is making business decisions for the organization. Most likely, they are using a loosely defined implicit model to meet the time requirements for development rather than a well thought out explicit model that meets the organizations strategic plan. The creation of the explicit models is the responsibility of the whole organization. Implicit models are used to fill the vacuum in order to meet tight project plans. They are a sin of omission.

The ad hoc method offers only two choices for developing systems: Scrap and Rework or Destroy and Recreate. Neither of these options is easy, cheap or efficient. Each will be explained is some detail below.

A. Scrap and Rework

This method takes existing information systems modifies them to meet current project requirements and redeploys. However, there are many dangers within this method since the documentation of the existing system may be inadequate for understanding how the changes will affect other parts (interfaces and existing pre-defined database queries are a prime example) of this system or other systems that relate to it. Thorough testing is

required to determine if the changes have unintended affects. This is an expensive and time-consuming task.

B. Destroy and Recreate
This method removes the existing system and builds a new system using outdated development methods. That is, it recreates the problem in a new replacement system. This may be a good time to use purposeful engineering for a better solution.

At this point, the business manager may be inclined to purchase a prepackaged integrated solution such as SAP, BAAN, Peoplesoft or Oracle. Depending on the organizational strategic vision this may be the best choice. The business manager must realize however that he is using an explicit model of a business that someone else engineered, understands and controls. If the business has a competitive advantage in their business processes, it will be lost or modified beyond recognition. Prepackaged integrated solutions are best when they are accepted as is.

Purposeful Engineering

The basis for this alternative is to think about, define, engineer, document, construct, implement and maintain the entire business in a great deal of detail.

The thoughtful reader may ask if there is only one model of a business. The answer is no. There are different models for different purposes and audiences. Every model contains information about the basic questions of what, how, where, who, when and why by who has the need to know. Below is a matrix reflecting this idea, including definitions of each cell and column/row labels.

	What/Data	How/Function	**Where/Network**	Who/Organization	When/Schedule	Why/Strategy
Planner						
Owner						
Designer						
Builder						
Sub-Contractor						

The first column lists job roles of the persons who have a need to know. The deliverables fill in the cells in a row from left to right. The job roles and the associated deliverables are:

A. Planner. This person(s) is responsible for the context of the models and develops the objectives and scope. The deliverables in this row could be a list of important items, processes, locations, organizations, events, and finally the goals and strategic plan for the organization.

B. Owner. This person(s) generates the conceptual enterprise models. The deliverables in this row could be the semantic models of the business, the business process, logistics, and workflow models, master schedule, and a business plan.

C. Designer. This person(s) creates the system or logical models of the organization. The deliverables in this row could be a logical data model, an application, distributed system and workflow interface architectures, the processing flow, and the associated business rules.

D. Builder. This person(s) develops the physical models. The deliverables in this row could be physical data models, system designs, technology and presentation architectures, the control structures, and the rule design based on the business rules.

E. Sub-Contractor. This person(s) could assist the Builder in transforming all the above models into detailed documents and programs. They are the data definition (meta data), the actual executable program, the detailed network and security architectures, the timing definition, and the final rule specification.

This matrix contains all the models necessary to analyze, design, construct, test, implement and maintain the organization. This holds true whether the organization is entirely manual, semi-automated or completely automated. These elemental primitives are the building blocks for an organization that can withstand the rigors of an environment in a state of continuous change.

The reason is deceptively simple. If this matrix is completely filled in with detailed deliverables, the managers of the organization will know everything about the organization. It will be in a form they can understand and the models should be readily available.

What is the alternative? Guess work, assumptions, mistakes and many 'blame game' meetings to attribute responsibility to someone else.

Conclusion

This Point of View has attempted in a few short pages to build a business case for engineering the enterprise with the same discipline, care and precision that goes into manufacturing very complex products. Certainly many more pages can be added to this document. However, be cautioned that the cells in the matrix are elemental primitives and as such cannot be further refined or developed. They are like atoms. They are scalar.

Readers may ask themselves if there is a good place to start the process and then continue to build upon their success? The answer is yes. The absolutely essential columns to initially focus on are the What/Data and Where/Network columns (in bold).

The data column is focused on the important things to an organization, usually things that have numbers associated with them such as an employee id or product serial number.

You do need to start at the top and work downward for a very sound reason. You will need the support and input of senior organization executives for this process to succeed. For example, in the cell at the intersection of Owner and What/Data the executive defines the meaning and purpose of things like a product. If this step is skipped or deferred only confusion will follow. The Owner may eliminate some items from the cell above at the intersection of Planner and What/Data as irrelevant. This provides focus for all the related models. Essentially, if the data is right many other models become much, much easier to develop and much more robust. Now the organization can face the future with the confidence that it will be able to meet the challenges of an organization in an environment of constant change.

Bibliography

Alvin Toffler, Powershift. 1990.

Zackman, J.A. and Sowa, J.F. Extending and Formalizing the Framework for Information Systems Architecture. IBM systems Journal. 31(3):76-292. 1992. IBM Publication G321-5488.

Zackman, J.A. Concepts of the Framework for Enterprise Architecture. Los Angles. Zackman International. 1996.