CIO's Beware: Very Large Scale Systems Projects

Kenneth C. Laudon

Department of Information, Operations and Management Sciences

Leonard N. Stern School of Business, New York University

44 West 4th Street, New York, NY 10012

January 1999

CIO's Beware: Very Large Scale Systems Projects

Kenneth C. Laudon

Stern School of Business

New York University

Center For Resesarch on Information Systems

Working Paper

January 1995

[This research was supported by the National Science Foundation, Information and Technology Program in the Division of Information, Robotics, and Intelligent Systems, Grant IRI-8619301, 1986-1995, with my colleague Prof. Alan F. Westin (Columbia University). National Information Systems Project, Working Paper 2]

Filename: VLSS.DOC

Abstract

Very Large Scale Systems (VLSS) play a powerful role in shaping what an organization does and can do in a practical sense. VLSS are deeply embedded in the organizational procedures, business plans, and strategies. These systems evolve over long periods of time, often not according to some rational plan, and for a limited time they provide a significant competitive advantage over other firms. In the long run, however, VLSS become *strategic liabilities* and must be rebuilt.

Many organizations experience great difficulty rebuilding VLSS. Indeed, most organizations attempt to avoid rebuilding VLSS until the last possible moment. Often, the organization is in a state of crisis, a strategic transition. Because of the complexity and size of VLSS, existing methodologies often are not helpful. To make matters worse, the typical management incentive structure discourages rebuilding VLSS.

In a typical VLSS effort, participants soon discover that they must rebuild the organization in order to take full advantage of new technologies. A major organizational engineering effort is often required. Senior management as well as systems management routinely underestimate the complexity of the task before them. Consequently, large errors are made in estimating costs and time.

Drawing on research in both the private and public sector, this paper examines why VLSS fail, why are VLSS so difficult to rebuild, what are the strategy options, and how can senior management guide the rebuilding processs.

CIOs Beware: Very Large Scale Systems Projects

In 1982 the Social Security Administration began a 500 million dollar, ten year program to rebuild its core information systems. A well known Big Eight accounting firm wrote the plan, and one of the country's most experienced systems integration company was hired to manage the process. The project was touted as a flagship Reagan era example of how to do things right with modern computer technology, experienced business designers, and a powerful integration manager. New software, hardware, databases and communications were promised.

By 1988 the project was cancelled with Congressional auditors claiming it was a failure, and SSA management claiming limited success. To be sure new hardware and telecommunications hardware was in place. But there was little progress in writing new software or developing a modern database. Employment at SSA had been squeezed down to 67,000 employees from 78,000 in 1982. But complaints from employees were rising rapidly as the agency tried to demonstrate productivity gains from its technology investment. Worse, the way you and your parents actually obtain social security benefits is largely unchanged---you still trudge down to the district office and fill out an application.

As a consultant to the Office of Technology Assessment, I worked from 1985 to 1986 trying to answer the 500 million dollar question on many Congressmen's mind: would SSA succeed in rebuilding its systems? After a year's work our team finally developed a consensus that, "No", SSA would most likely not succeed despite the massive investment in hardware and telecommunications.[1]

After many months of investigation it became clear that the Big Eight Accounting firm and the experienced systems integrator had developed a plan to automate the existing procedures of SSA and had failed entirely to re-think how SSA should be doing business in the 1990's. The basic procedures of the agency, the business procedures built by the Social Security Administration of 1936, remain fundamentally unchanged. Of course, now these 1936 procedures are implemented on very fast state of the art computers.

It was also clear that the 1982 senior management of the agency, management and systems personnel at the accounting firm and the systems integration firm, as well as Congressmen and their staffs, had seriously underestimated the complexity and cost of re-writing 12 million lines of computer code, and rearranging the basic files of SSA into a modern computer database.

No one seemed to understand in the period 1982-1987 that they were dealing with a relatively new problem: rebuilding very large scale systems (VLSS). No one seemed to understand that in order to take advantage of the new information technologies, one has to re-think completely the business procedures, rules, practices, and even values. Just as one has to design products for modern manufacturing, so to must one design business organizations for contemporary information technology.

After this experience we became more aware that many organizations in both the private and public sector were experiencing difficulties with very large scale systems rebuilding. Below are some examples of the difficulties which large systems pose for organizations.

Some Infamous Government VLSS

*The IRS has been rebuilding the tax transaction systems since 1978. Called by various names like the "new Tax Administration System," and the "Tax System Redesign," IRS has spent over a billion dollars trying to rebuild. A recent critical report by Congressional auditors found the new systems cannot last beyond the early 1990's and a U.S. Senate Finance Committee Chairman called the effort a "Train Wreck" about to happen.[2]

*The FBI has been trying to build a national electronic network to exchange criminal history records among the states for the last decade. After expending about one billion dollars, only ten states contribute records to the electronic files.

*In 1983 the U.S. Patent Office awarded a $293 million
19 year [sic] contract to Planning Research Corporation of McLean Virginia to rebuild the old patent storage and search system in which patent records are stored in wooden boxes. In 1989 the system was four years behind schedule, did not work at all, the completion cost estimate was $600 million, and the delivery date 1994.[3]

*The grand daddy of all system sink holes is of course the Department of Defense World Wide Military Command and Control System (WWMCCS--pronounced "Wimmics). Under the careful management of the Air Force. This system was begun in 1966 as a federation of 158 different computer systems using 30 different software systems at 81 different locations. After one billion dollars in expenditures, and literally hundreds of studies, there is still no agreement on the architecture or information requirements!

The system has been judged by the General Accounting Office to be a failure, with a management structure "so complex and fragmented that no one central organization or individual has a complete overview of the program."[4]

Some Infamous VLSS in the Private Sector

*A major insurance company started out in 1982 to re-build its policy issuance system from a batch system to an on-line system driven directly from the agents desktop computer. This would automate completely the process of issuing new policies, and permit the introduction of new types of products within one month as opposed to the traditional three years or so.

A well known systems integration firm was hired to develop software and install it on the firms hardware, as well as assume consulting and integration roles. The target date was 1987 and the price tag $8 million. The project is still not finished, the target date is 1993, and the price tag is $15 million.[5]

*A major mini-computer manufacturer was well known for its chaotic order processing and billing system. In 1981 the company hired a small but well known consulting firm to rebuild the order entry system in cooperation with the MIS department. The original contract called for an 18 month delivery and a price of 14 million dollars. In 1985 the project was finally cancelled and was costing 10 million a year in direct costs, plus indirect costs of customers lost due to faulty billing.

*A major American Bank attempted to re-design the Bank's Employee Benefits and Trust Services Division. After several years of effort, a cost to the Bank of 45 million dollars, the project ended in failure in 1988.[6]

*A major Midwestern health care insurance provider hired a systems integrator to re-build the firms seven different claims processing systems and three different membership systems for $200 million dollars. The project was completed on schedule 18 months later but it did not work. The system created $60 million in overpayments and lost 35,000 customers before being unplugged.

A Private Sector Example: VLSS: MultiChem's Experience

MultiChem is an international manufacturer of chemicals, pharmaceuticals, and dyestuffs, with 3 divisions and 20 operating units scattered throughout the U.S. It employs 20,000 employees of whom 7000 are unionized. Numerous acquisitions during the late 1970's could not be handled by the corporation's core Personnel information system, which had already been extensively modified and patched.

The corporation embarked on an ambitious Grand Design to build a new Human Resources Information System that would automate all of the firm's Human Resources activities. In addition to providing on-line real-time personnel processing in a state-of-the art technical environment, the proposed system would automate all of the firm's Benefits plans, plus track positions and organizational relationships for manpower planning.

The project never proceeded beyond the requirements-gathering stage because the organizational complexity was too overwhelming. There were 2 major compensation evaluation systems, 32 pension plans, 70 insurance plans, and an Investment Savings Plan with complex securities valuations to automate. The old Personnel system had been designed to be a major supplier of employee data to other systems--Payroll, Benefits and the firm's Investment Savings Plan. These systems were linked by elaborate and often heavily manual interfaces, that were poorly documented and understoood. Twelve separate interfaces had to be constructed for the firm's disparate Payroll systems alone. In many instances the COBOL code from the old system had to be utilized to reconstruct fine-grained requirements because end-users overlooked many important details.

Analysts from the project proposed streamlining the system by "rationalizing" some of the firm's plans and procedures--consolidating pension plans, reducing the number of Insurance plans, forcing all operating units to use the same Payroll system. But senior management refused, fearing "political" backlash from profitable acquisitions and

operating units that insisted on doing things their way. After four years and expenditures of over $5 million, the project had only produced a prototype for positions tracking, a pension processing system that didn't work, and three thousand pages of design specifications. The project was cancelled. Senior management insisted that Human Resources develop a strategic business plan before any new systems projects proceed.

The project never proceeded beyond the requirements-gathering stage because the organizational complexity was too overwhelming. There were 2 major compensation evaluation systems, 32 pension plans, 70 insurance plans, and an Investment Savings Plan with complex securities valuations to automate. The old Personnel system had been designed to be a major supplier of employee data to other systems--Payroll, Benefits and the firm's Investment Savings Plan. These systems were linked by elaborate and often heavily manual interfaces, that were poorly documented and understoood. Twelve separate interfaces had to be constructed for the firm's disparate Payroll systems alone. In many instances the COBOL code from the old system had to be utilized to reconstruct fine-grained requirements because end-users overlooked many important details.

Analysts from the project proposed streamlining the system by "rationalizing" some of the firm's plans and procedures--consolidating pension plans, reducing the number of Insurance plans, forcing all operating units to use the same Payroll system. But senior management refused, fearing "political" backlash from profitable acquisitions and operating units that insisted on doing things their way.

After four years and expenditures of over $5 million, the project had only produced a prototype for positions tracking, a pension processing system that didn't work, and three thousand pages of design specifications. The project was cancelled. Senior management insisted that Human Resources develop a strategic business plan before any new systems projects proceed.

What is it about very large scale systems that causes them to fail in the first place? Why can't VLSS systems keep operating forever? Why--in this day and age--are VLSS so difficult to rebuild? What are the strategic options to consider when rebuilding VLSS? How can senior management be sure that the right questions have been asked and the right direction chosen?

Back to Basics: What is a VLSS?

At the outset it is important to see that VLSS are fundamentally a different kind of system from the normal run-of-the-mill, nickel and dime computer system. Unfortunately, the existing academic literature makes little or no distinction between a VLSS behemoth involving 200-300 systems personnel, millions of lines of code, and a ten year development trajectory and a new sales management system with twenty systems personnel, 10,000 lines of code, and a one year development path.[7]

There are two kinds of definitions of VLSS. *Technical definitions* describe VLSS in terms of lines of code, the number of computer hours required to process the application, the number of daily transactions, the number of records, files, and so forth. While interesting and useful once other problems are solved, technical definitions generally fail to grasp the essence of VLSS.

Technical definitions tend to produce technical solutions such as CASE tools (Computer Aided System Engineering), programmer workbenches, purchasing more computer and communications hardware. Generally, these technical solutions are peripheral to the problem, but they are nevertheless extremely popular. Armed with slogans like "software engineering," managers are misled in the belief that they are attacking the problem with modern tools.[8]

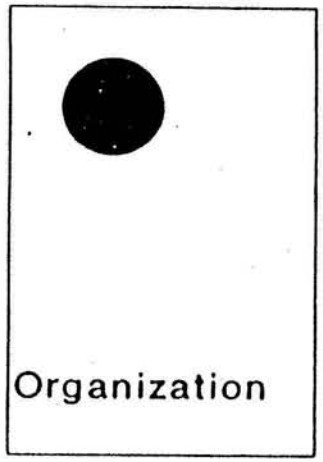
A second type of definition is organizational. In *organizational definitions* VLSS have the following characteristics:

*They play a dominant role in shaping the organizational metaphor of production and service delivery. They are rooted in the culture of the company.

*They are deeply embedded in the thousands of day-to-day organizational procedures. VLSS are **the** standard operating procedures. These systems are in every sense a strategic asset to the company, and play a strategic role by creating some comparative advantages that are hard to duplicate. At the same time, these systems--as we see below--VLSS become *strategic liabilities* as they age.

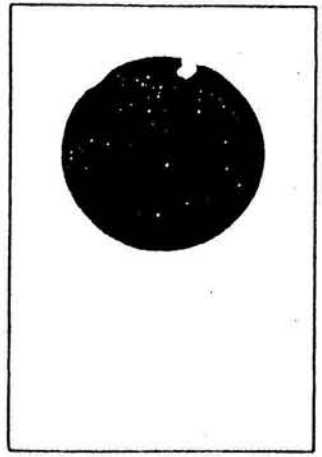
*VLSS directly control the information flow of central "core" activities in the organization. [See panel for illustration of "core" activities and VLSS]

VLSS and Organizations



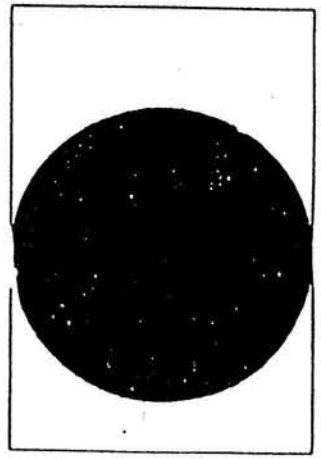
Technical

1950's-60's



Managerial

1970's



Institutional

Core

Center for Digital Economy Research
Stern School of Business
Working Paper IS-95-09

Picture: "Core" Organizational Activities, Spin-offs, and Systems"

Caption: At the heart of every large, contemporary organization is a database, or collection of data files, which have developed over decades. These files reflect the predominant business metaphor: what products are made and how, who is the customer, and how is product distributed.

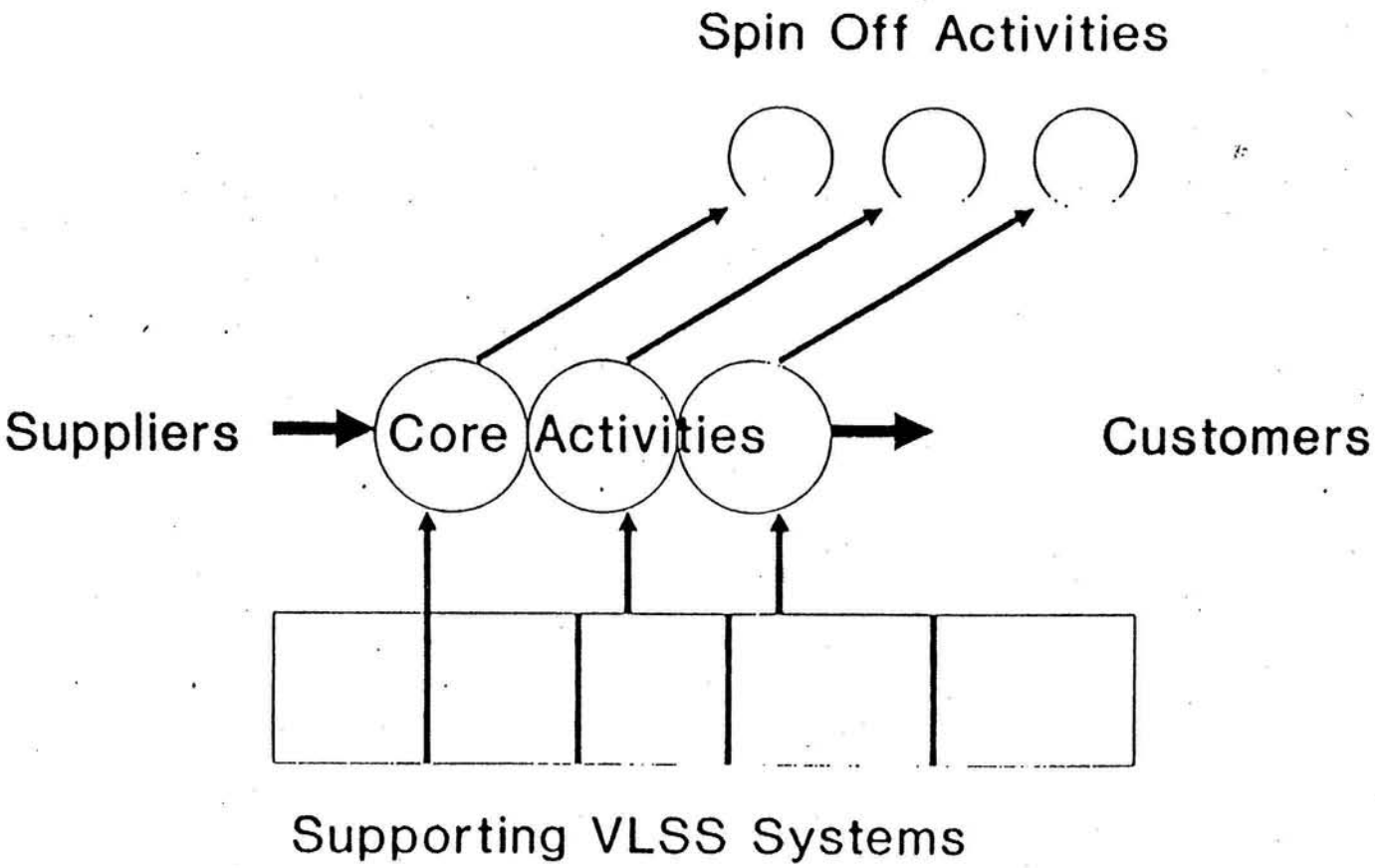
All organizations develop a central institutional "core" over time--a collection of a few products, activities, and services which are the predominant focus of organizational life. Supporting each core activity are Very Large Scale Systems.

Over long periods of time, organizations develop "spin offs" from the central core in blips, drabs, and major trickles. Supporting VLSS take on additional complexity as they support these spin-offs.

*They directly involve the vast majority of the employees.

*They involve powerful organizational units. Therefore they are bound in the politics of the organization.

*They are intellectually complex because of shear size and because no one individual, or small group of individuals, can adequately understand them. Knowledge of the system is widespread, but inherently not collective or shared.[9]



"Core" Activities, Spin-offs, and VLSS

An Example of VLSS Complexity

SSA operates with a Program Operations Manual of approximately 40,000 pages. One can think of this manual as the definitive standard operating procedures for SSA because it expresses all the formal, legal rules for arriving at SSA eligiblity. 2,000 pages are changed each month and 40,000 field employees attempt to keep up with these monthly changes.

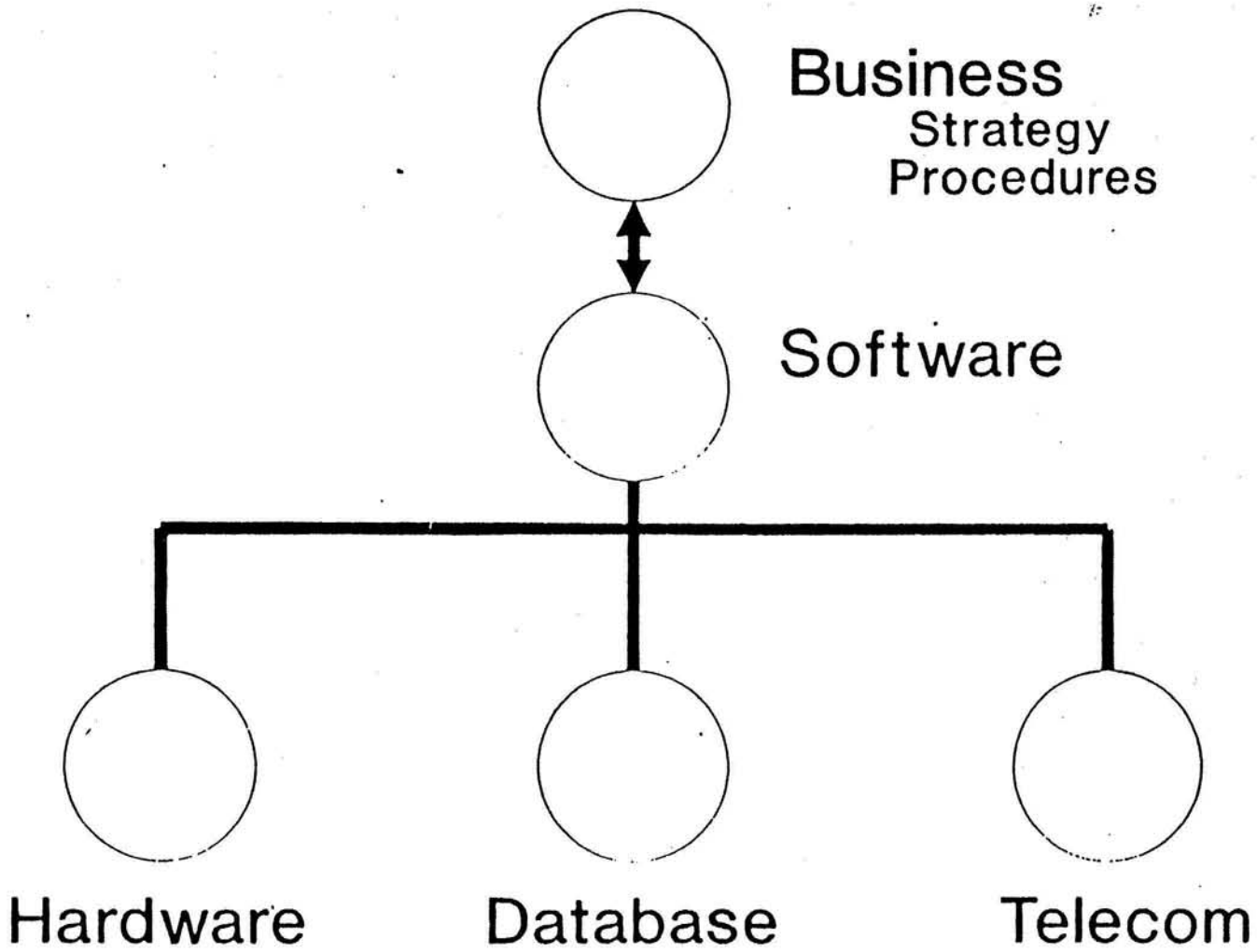
Most of the 40,000 pages in the Program Operations Manual is bound up in the 12 million lines of SSA software code which issues SSA numbers, tracks earnings, establishes eligibility, and retains the master beneficiary records.

In organizations like these, much of what is done and can be done on a daily basis is shaped by the capabilities and performance of their VLSS. In VLSS, there is a "tight couple" between business procedure and strategies, and the firms VLSS hardware, telecommunications, software, and database.

Picture: A "tightly coupled" relationship between and an organization's business rules, procedures, personnel and strategies with VLSS software, hardware, telecommunications, and database.

These social definitions of VLSS grasp the complexity of the problem but tend --as we see below-- not to lead to any clear cut, decisive, or clean "solutions." There are no

Very Large Scale Systems



quick "technical fixes" for the problems raised by re-building VLSS. Organizational definitions tend to support unpopular and difficult solutions to the problem of VLSS.

How Many VLSS Are There?

Virtually all Fortune 1000 firms more than 10 years old have at least one VLSS, many have a collection of such systems. Peat Marwick Mitchell and Co. has built a 35 million dollar a year business trying to re-build VLSS. It surveyed 600 of its clients in 1987 and found that most had at least one VLSS, and that 35% of the firms had at least one "runaway" VLSS rebuilding project in which a VLSS Project was years behind schedule and millions over budget.[10]

Where do VLSS Come From?

Most organizations do not live long enough to experience the pain of having to rebuild VLSS. Briefly, a VLSS which evolved in the 1960's, or even the 1950's, can live on and provide support for an organization indefinitely so as long as the environment of the organization does not materially change. Because the average life span of a business organization is less than ten years, the original VLSS are usually sufficient to last the lifetime of the corporation.[11]

Ironically, it is only the small number of very successful organizations who live longer than ten years who experience the pain of having to rebuild VLSS.

Almost all VLSS evolve over long periods of time and virtually no historically existing VLSS more than five years old were "built" in the sense of being the result of singular, conscious, intentional, organizational efforts to produce a "system". [As it turns out, this fact provides a critical insight into how to rebuild VLSS described below].

The airlines reservation systems, and virtually all actual "strategic" systems about which we know, evolve over long periods of time. VLSS evolve in spurts, and in patches,

as they respond to more and more unplanned demands to exapand so that new activities can be supported. Not until the 1980s are VLSS being rebuilt as a part of well structured plans.[12]

The evolutionary character of VLSS means that over time they accrue new unplanned functions; their internal structure becomes a lattice work of interdependencies, most completely undocumented. Early designers retire, and latecomes know only specific functions of the system. An overall perspective on the VLSS is lost in the dim past.

Why do Existing VLSS Fail: Strategic Transitions

Organizations usually rebuild VLSS when they are in a state of crisis. Because decision making in an organizational crisis is usually worse, and certainly no better, than in "normal times," the odds of managers making the right decisions during the crisis are slight.[13] Indeed, because the existing lot of managers was at the helm as the crisis developed, they are often the least capable of solving the problem.

The existing VLSS is usually not the cause of the crisis, but it is directly involved. Typically the organizational crisis occurs because the environment has shifted and no longer supports the organization.

Deregulation, new competition, the appearance of new technology pursued vigorously by competitors, changing public tastes, and so forth are typical of the kinds of environmental changes and discontinuities which make existing business procedures, products, services, and strategies irrelevant and no longer useful.

In this environment of discontinuity, the old VLSS is typically stretched to its limits and is being asked to do impossible things. Most VLSS can do "impossible" things, things never intended, by virtue of dedicated staff willing to work on week ends and nights.

But eventually not even these crisis tactics work the simple, basic functions performed by the system like cutting checks, tracking sales, inventory control, transaction process of orders becomes dubious.

Faced with this crisis, organizations must come up with new procedures, business rules, and indeed an entire new organizational metaphor of what they do. What is needed is a new vision of how to do business, what to produce, how to deliver the product or service, and how to use new information technology to support the new vision. Needless to say, the organization will have to rebuild its VLSS to support the new business. We can call this period a *strategic transition*.

Why Don't Managers Rebuild VLSS Before the Crisis?

In most VLSS failures, managers see the crisis develop long before it happens. There is almost always a long lead time from the first signs of trouble in a VLSS to its ultimate collapse. Researchers have found that it takes at least five years, often longer, to really kill off an organization.[14] This leads to the question, "Why don't managers and organizations do something before the storm hits?"

Because organizations are so dependent on VLSS, and because they are so immensely complex, expensive, and troublesome, most rational managers and most short term rational organizations will avoid rebuilding them in any serious way unless absolutely necessary.

In most organizations the managerial incentive structure is not supportive of rebuilding VLSS. In general, VLSS rebuild projects take from 3-10 years. In this time frame, managers who start the painful rebuild are rarely around to reap the rewards. Those who come on the project late are liable to be blamed for the likely failure. Hence most politically astute managers will avoid the assignment. In other words, VLSS have a different life cycle than managers.

The organizational incentive structure is also not supportive. At the organizational level, truly immense investments are required over long periods of unpredictable time and the rewards--if any-- usually cannot be calculated in any believable way. It's a roll of the dice. Most large rational organizations will avoid rolling the dice.

What happens when VLSS Fail?

Because VLSS are so entwined with the value chain of activities, and the value chains of suppliers and customers, rebuilding them is inherently risky. Small mistakes can amplify throughout the chain of related activities, drastically effecting suppliers, and customers, not to mention one's own organization.

Patches of existing systems rather than rebuilds are the typical response to the first signs of trouble. Patches are politically acceptable because they require no great organizational change.

Why the Old Systems Development Life Cycle Won't Work

One of the most dangerous pitfalls to avoid in this situation is to turn to the tried and true techniques developed in the 1950's for building systems. Called the "systems development life cycle", this traditional technique calls for systems people to study the old information system, investigate the "information requirements" of the organization, and design a rational solution.

This all sounds good until you get down into the trenches. For instance, there are no requirements statements on the old system, there is no book where you can find out how the old system works--its all in the heads of thousands of people because as the VLSS grew over the years no one really documented the system. This means you often have to look at 100,000 lines of code to figure out what is happening in the system.

If we assume that one skilled-programmer analyst can understand and document about 20 lines of code per day (a generous assessment and about the same as he could

write new code), and if we assume a VLSS with say 2 million lines of code, then it would a team of twenty programmers about 21 years just to figure out how the old system works! If we triple the productivity of programmer/analysts from current levels (something nobody even envisages actually happening with CASE tools), then the task would only take seven years. By that time of course, the organization will be dead.

For instance, SSA hired a very prestigious accounting firm to conduct a so-called "enterprise wide information analysis" of how SSA's existing 1982 systems worked, and what information requirements the old system fulfilled. the study itself took two years and three million dollars. It took SSA another year to develop a statement of what the requirements were for the old organization. After three years, no one could understand the information requirements statement however because it took up thirty feet on a book shelf! The study has remain unused as SSA later decided to develop entirely new systems based on contemporary views of how to conduct business. The old ways--and their information requirements--were essentially irrelevant.

Briefly, the intellectual complexity of the existing VLSS are so great, that several years will be required just to understand them in some formal sense. By that time it is often too late, and it is almost always irrelevant to know how ancient business practices were performed with outdated computer technology.

The idea that VLSS evolve overlong periods of time is a critical insight here because it means that just about any intentional or conscious plan to re-build such systems will be simplistic, even naive, and fail to capture the full functionality of the existing VLSS. Imagine designing a plant, say an orchid, given some DNA.

Second, there's a difference between "Doing something according to plan" and "creating the conditions for something to grow and flourish." In the traditional systems

development life cycle, we try to do something according to plan. This is a civil engineering technique suitable for bridging gaps across rivers with bridges.

The nice thing about land masses is that they don't change much in the short term of a few centuries. But the analogy may be inappropriate. Information systems have to be built for uncertain future environments which change in a few months, years.

Biological analogies--as opposed to civil engineering metaphors-- may provide a better to guide to choosing a proper VLSS rebuilding strategy. What we need is a systems development life cycle methodology which can evolve flexible robust systems over, say, ten year time frames.

What we need to start with is a vision of how to do business five or ten years from now, and build for that future environment. Where can we get that vision? Let's examine the realistic choices facing senior management.

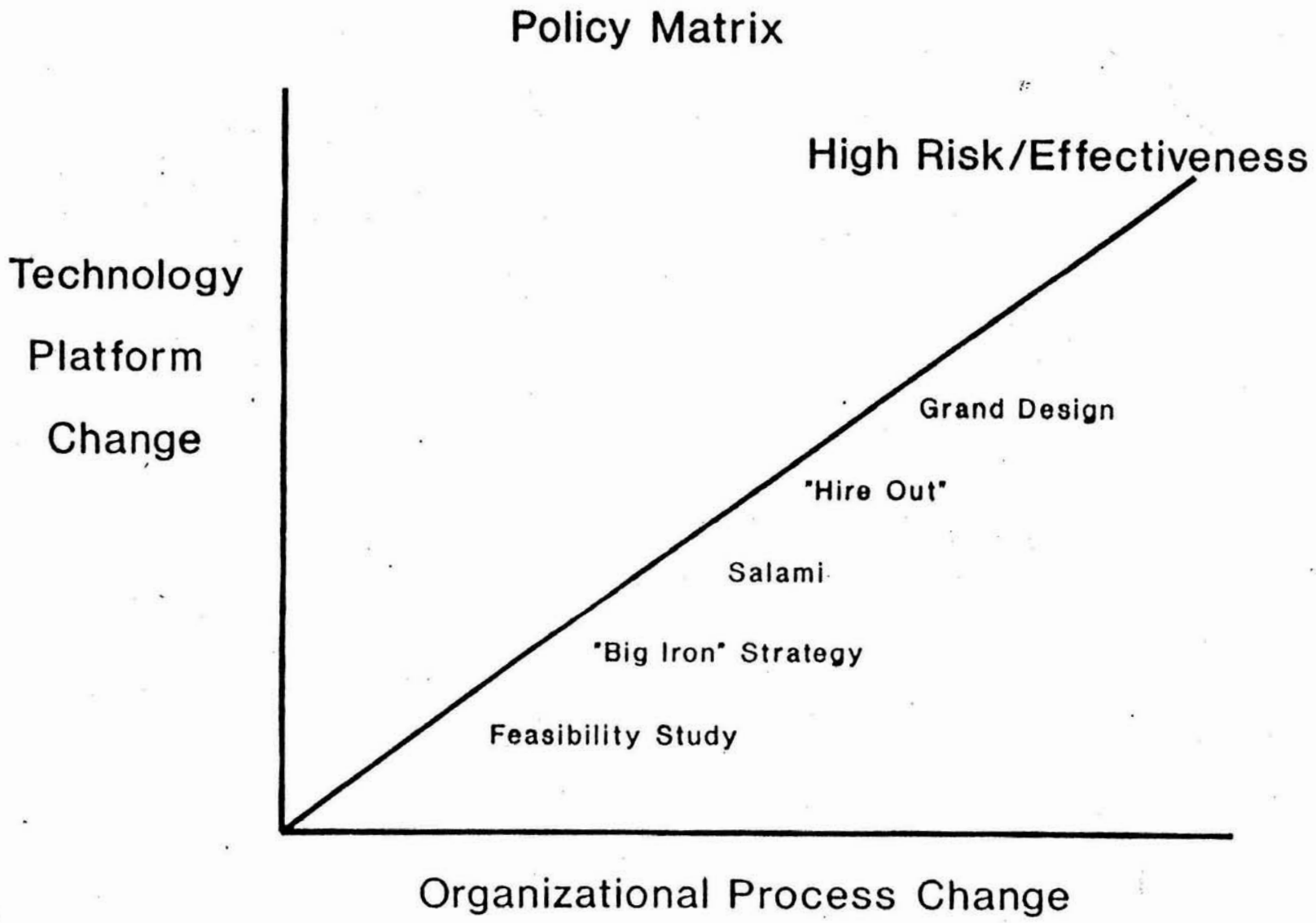
What are the re-building strategies?

We need to consider three dimensions: the organization, the technology, and strategy [see box].

Picture: Policy Matrix.

Caption: All VLSS rebuilding strategies can be analysed using a three dimension policy matrix.

We can understand any strategy chosen better by placing it in this three dimensional space. All VLSS rebuilding strategies involve some change in organizational structure and process--some more than others. All strategies involve some change in the technology platform--the collection of computers, telecommunications, and databases.



The strategies themselves have different characteristics of risk, reward, and resistance (popularity). In general, as we move out the strategy vector, risk increases but so does reward (effectiveness). Unfortunately, resistance increases as well. Lets examine six common VLSS strategies using this matrix.

Feasibility study

In the short term a low risk possibility is to spend six months to a year studying the problem. Sometimes inaction is the best action. Some problems do go away, others solve themselves, and others simply need a new crew of managers and professionals which takes time. If you can afford inaction for a year, you should try to identify the precise nature of the VLSS problem: people, technology, and organization.

Let's Go Shopping

The most common short term survival strategy is to go shopping for the biggest, fastest, and most powerful mainframe machines available. In the trade this is called the "unlimited MIPS" or "Big Iron" strategy and it is of course a favorite among systems people, all technology lovers, vendors of mainframes, as well as most system integrators. Generally, this strategy involves minimal organizational change, sometimes significant technology platform change, and moderate short term effectiveness with low risk.

On the positive side this strategy buys time for large "embedded base" systems in a state of crisis. As a survival strategy, bigger machines can process the old software and the old files as much as ten times faster, relieving backlogs, unclogging communications channels, and increasing service to customers. Moreover, this strategy fits in with a manager's life cycle: a "turnaround" of sorts can be produced in 18 months to 3 years.

On the down side this strategy involves a tremendous financial commitment to the old software, the old business procedures embedded in the software, and the entire business metaphor of the past. If you want to be doing business 5 to 7 years from now (the expected life of "new" systems) the same way you do business now, then this is the

strategy for you. Many years from now you will find yourself locked into a centralized, mainframe architecture, operating software now twenty years old, with little capability to accommodate changes in the market place.

Salami: one bite at a time.

One approach to large systems is to cut them up into smaller systems. The idea behind *salami* is to break up huge VLSS into component parts, and rebuild the parts one at a time. This strategy recognizes that "Big Iron" is not itself the answer and that some serious redesign of the business is in order. At the same, *salami* recognizes that most businesses do not have a comprehensive vision of how they want to do business in the future, and even if they did, it would be too risky to try to implement the entire vision all at once. Better to break up the problem into "natural" parts---like sales, marketing, production, etc.--and do the easy parts first.

Generally, *salami* strategies emphasize change in technology platforms. Within specific functional areas--like sales--salami tactics can produce important though limited organizational changes.

The advantage of *salami* is that it focuses scarce personnel and system resources on a small part of the problem for maximum effect. Most VLSS really are too big to attempt a single planned rebuild. With *salami*, interim deliverable targets can be met, resources can be minimized, and personnel have a chance to learn by doing as they explore new technology and business design principles in limited areas. The damage done by errors with *salami* is less than with more ambitious designs. While slow to deliver completed VLSS--it may take ten years--eventually the firm arrives.

The disadvantages of *salami* follow from its virtues--there is a distinct prospect that in five years you will end up with some high tech parts that cannot be tied together, which do not share data, and that are built according to different principles learned over time. The result is a garden of applications which do not fit together. However, fitting

things together can be someone else's problem, and *salami* is therefor a popular, reasonable sounding strategy, which produces results according to schedule.

Hire out: somebody knows how.

Increasingly organizations faced with rebuilding their own VLSS turn to outside consultants and system integrators. The fastest growing segment of the computer service industry--as major accounting firms have demonstrated--is systems integration. In large part this is due to the problems faced by the Fortune 1000 in rebuilding their VLSS.

Many CIOs have decided that the internal system staff is overwhelmed just operating the existing systems, or they simply lack the skills to be build contemporary systems appropriate to the future ten years. Many times the organization does not have a "grand vision" to implement, and no one in the organization has any idea about how to do business five or ten years from now. Therefore, proponents argue, turning to outsiders who are skilled, experienced, and creative is a wise move.

Generally, the most skilled programmer/analysts, systems analysts, and system designers end up working for the systems integrators simply because the salaries are so high. Moreover, these skilled persons have been exposed to a much wider variety of technologies and organizations than your own staff. On the surface, pursuing this strategy seems to lower the risk presented by internal development of systems, and seems to offer the best opportunity for developing state of the art systems suited for the future decade.

System integrators usually do not understand the specifics of your firm, and often times do not undertand the dynamics of your own industry especially if there experience is developed in other industries. For this reason, system integrators tend to be excellent at managing technical platform changes but they do poorly when it comes to organizational process change. As one CIO noted based on his experience with a well known system integrator, "They were fine when we told them to run the data center. The "took" the

Data Center like real Marines--which many of them were. But when we asked them to think, when we asked them to integrate closely the tens of sub-projects, or when we asked them to take a critical look at the direction we were moving, they did miserably."

Unfortunately, success with external system integrators and accounting firms as VLSS rebuilders is no higher than flipping a coin and there are as many disaster stories as success stories.

The problems are multiple and serious. The insight of system integrators into the future of your business and industry is usually no better than yours, and often worse. Hence, system integrators cannot be relied on to do your thinking. Integrators typically cannot transfer skills to your staff, so you will be dependent on them forever. Your staff tends to resent outside integrators and consultants, and miscommunication or non communication grows as the project proceeds. If you think managing a large project is difficult, imagine what it is like when the staff is fighting with outside consultants, and communication of any helpful sort is non-existent.

If you are dissatisfied with the results of system integrators, you tend to be locked in--afterall, they are the only one's who really "know your system." Experience has found that external system integrators are no better estimating ultimate costs or delivery schedules than your own staff--which is miserable.

Despite its growing popularity, reliance on external problem solvers exposes you to a great deal of risk, at a high cost, and diminishes the development of your own staff. You may get a system, but it will almost certainly be over budget and behind schedule, and you will not have created the conditions for growing powerful, robust systems in the future.

Grand Design: Let's do it right.

Grand design is the most serious VLSS rebuild strategy if only because it is the most ambitious, expensive, and risky of all the strategies. The rewards of *grand design* can be significant although a long time in coming (more than five years). These characteristics make *grand design* highly desirable to managers who appreciate taking risks, and to vendors of hardware and software equipment. Other organizational players tend to resist grand design because it promises to bring significant organizational change.

Grand design is the only strategy that correctly diagnoses a major part of the problem: there is an intimate connection between the basic metaphor of the business, the day to day procedures, and the business VLSS. You cannot change one without changing the others. Second, in order to seriously rebuild VLSS taking maximum advantage of contemporary technology, you will need to change the software, and databases, as well as the hardware and telecommunications. Third, a major redesign of the business strategy and procedures will be required both to drive the system rebuilding process as well as to simplify the traditional practices. Briefly, a new vision is needed of what the business is all about and how to do it. *Grand designs* imply a "frame breaking" change in the business.

Grand design exercises the talents of strategic planners, CEO's who like to make strategic plans, and organizational staffers who like carrying out detailed plans over long periods. More than any other strategy, *Grand design* promises to "do it right"--to build integrated, state of the art systems which are carefully crafted to support a future vision of the business. In the few cases where grand design has worked, the results have been spectacular.

Unfortunately, most *grand designs* are not successfully implemented. Redesigning the business procedures, rules, and strategies, while at the same time trying to redesign the support software, databases, hardware, and telecommunications, proves to be a totally overwhelming task for corporate staffs no matter how big and no matter what technique of planning or technology of planning is used. In the absence of clear-cut information

requirements (which cannot be given because the business design is not yet solidified in the minds of planners), systems people tend to overcompensate by buying more of whatever hardware is available just in case it might be needed.

The end result of *grand design* is usually a new collection of hardware--that afterall is the easy part of systems. When it comes to software and data, the end result is typically busted budgets and products not delivered as described in the early part of this paper. Because all that works is the new hardware running the old software, that tends to be what is accomplished.

A federal study of *grand designs* in government found that senior management often did not understand the commitment of time and effort required, that staff was often not available or not skilled to carry out the plan, and that it was impossible to coordinate all the organizational players. [see panel for top ten problems]

Leading Problems of Grand Design Strategies

Problem			Total*
1. Top management lack of understanding		34.3	
2. Staffing problems		32.8	
3. Lack of acquisition skills		29.0	
4. Protests from contractors		28.6	
5. Problems that occur when falling behind	28.3		
6. Coordination problems			28.1
7. Inappropriate organizational placement	27.6		
8. Audits by GAO			27.4
9. Problems with procurement regulations	27.4		
10. Unrealistic time schedules set by others	26.6		

*Survey of 21 senior federal managers involved in the design and installation of 10 Grand Design federal systems. Respondents were asked to score the problem in four areas (planning, procurement, implementation, and operation) on a 10 point scale (1=not very important, 10= extremely important). The scores were totalled across the four areas. The maximum possible score in each problem area is 40, the minimum possible is 4.

Source: Francis A. McDonough, "An Evaluation of the Grand Design Approach to Developing Computer Based Application Systems," United States General Services Administration, Information Resources Management Service, September 1988.

The problems of *grand design* are typically complicated by the on-going organizational crisis--as described earlier-- of which the VLSS is a significant part. Enormous hopes and considerable pressure is placed on system designers who are often seen as the technological messiahs who will deliver the firm into a healthy future. Drastic measures are sometimes implemented, like appointing a systems Czar (or in the Federal Government a "Trail Boss") to "take command" of the situation.

What Can Be Done?

Organizations typically go into and out of focus in terms of systems planning over long periods--say 20 years. They fluctuate back and forth from one strategy to another, some working, others not. [see box for history of SSA strategy].[15]

SSA's Strategy History

SSA went through a fascinating historical cycle of VLSS strategies. What emerged was never planned, although current plans now support what has emerged.

As the 1960's equipment and software broke down and deteriorated in the 1970's, in resulting from Congressional and executive branch failure to understand the need for modernization, SSA planners in 1978 proposed a grand design approach called *Future Process.*

Future Process sought to totally transform the paper based agency into a contemporary on-line data base organization. In this plan, regional processing centers would be radically changed into electronic storehouses, and 1300 district offices would offer telephone service to clients. However, this plan generated so much opposition from Operations Management and Field Staffs, who feared the impending changes, that it was dropped.

In 1979, a new Commissioner who was formerly a career officer in the Navy proposed a salami strategy: divide SSA into four main bundles of activity, and sub-contract each bundle separately in isolation from the others. This strategy was opposed by leading members of Congress, the Federal Data Processing establishment at General Services Administration, and both systems and operations people at SSA who knew the future required linkages among the various parts of SSA. The Commissioner soon left the agency.

In 1982, with machines literally breaking down and threats of interruptions to the stream of 40 million checks each month, SSA proposed a "technical solution" called the *Systems Modernization Plan.* The cost was 500 million dollars over five years and sought to buy new hardware and rebuild the software and data files.

While it promised new software and databases, the SMP plan resulted in 1987 after five years in new, faster, more powerful central mainframes in Baltimore and 25,000

desktop terminals connected to a new data communications system. As noted earlier, not changed in the way SSA conducted business with 40 million beneficiaries.

In 1986 a new Commissioner was appointed, Congress threatened to withhold funding for SMP because of lack of progress on software, and this forced a "refocussing" period on SSA. In response, SSA withdrew SMP and put forth a new plan called the *Agency Strategic Plan 2000*.

At first glance, this new plan appears to be in the best traditions of a real *Grand Design*: a new way of doing business is defined, one largely based on electronic interactions with citizens. A ten year, multi billion dollar effort is contemplated.

But oddly, much of the new plan is already operating although some parts require still more hardware. From 1986 to 1989 SSA refocused its software efforts recognizing that it had seriously underestimated in 1982 the time and cost of rewriting 12 million lines of code. SSA managers also realized, as outside consultants and Congressional researchers had been arguing, that a new vision of how to run the agency was needed, a vision based on new models of doing business--800 telephone service, ATM like cards, reach out programs to clients like private insurance companies, electronic interaction with clients, and so forth.

SSA management has successfully implemented a new on-line, interactive, telecommunications driven customer support service which processes much of the work involved in its largest program--RSI (Retirement and Survivors Insurance).

In the end, SSA managers pulled off one of the more remarkable transformations of a VLSS. A paper oriented, batch processing organization is now well on its way to becoming a contemporary on-line, telecommunications oriented service agency. The transformation required seven years, somewhere between 500 million to one billion dollars (depending on how one counts and where one stops counting), the efforts of a 2000 person systems staff, and a large group of Congressional critics and private consultants (sometimes estimated by SSA senior management to be equal in size to the systems staff).

Most important, the transformation was much more evolutionary than anticipated. Contemporary plans for the future are much more evolutionary than earlier plans. Real organizational learning has occurred.

Of course most organizations would have expired under this load. SSA benefitted from the fact that American society would not let SSA die regardless of conservative attempts to kill it off, or because of shear incompetence.

SSA's transformation did not proceed according to plan, but it did proceed, and it did achieve a desirable result. In many respects, the goals of the original rejected *Future Process* plan of 1978 have finally been achieved.

If we look back historically at VLSS rebuild efforts there seem to be two kinds of risks to avoid. One nasty risk to avoid is a technology shopping spree which locks the business into an information architecture for the next decade. This strategy offers a solution, but one which can be fatal in the long run. A second nasty risk is trying to do everything at once--a *grand design*. Here are two simple ways to avoid both pitfalls.

Make the problem big enough to solve

The best way to avoid succumbing to technical solutions is to ask you and your staff to spend some considerable time thinking about the problem. In most cases of VLSS failure, the failure is the result of a long term pattern of decline in technology, a growth in organizational complexity, and a decline in staff skills and morale. People and organizational problems are just as important as technology problems. You must discover what the people and organizational problems are before permitting technology solutions to run away with budgets and schedules.

A good place to look for organizational problems is in the very design of the production and service delivery process. Often what happens is that a cumbersome, complex, and error prone set of organizational procedures inherited from the distant past becomes entrenched in the business, and existing systems have attempted to automate these manual procedures.

For instance, at the Social Security Administration there was an initial failure in 1982 to rethink business procedures. Existing software was written to support a manual paper process invented in 1936. New machines were purchased in the 1982-1985 period in order to make the old software work faster. The result was that in 1986--four years and 400 million dollars after the project began--the 1936 business practices of SSA were indeed more efficient, but service to the public, productivity, and staff morale was not markedly higher.

From 1986 to 1989 SSA refocused its efforts using teleprocessing and telephone service as the vision of the future SSA. It wrote software to support this new vision and ultimately succeeded by 1989 in establishing a modern, interactive, telecommunications based business interface with the public.

Most private companies do not have the luxury of 400 million dollar budgets. In the private sector, a mistake in defining the problem as merely technological may very well mean the death of the firm. The historical lesson of VLSS is clear: if you throw MIPS and hardware at the existing procedures, if you fail to simplify the business procedures first before designing new VLSS, if you fail to develop a new vision of how to do business in the next decade, most of the potential efficiency gains of new technology will be lost.

The poor track record of information technology in the service sector to markedly increase productivity is one indication that many firms have thrown considerable computer resources at problems but not achieved the productivity gains that they anticipated.[16]

Opportunism and the Evolution of VLSS

You may not have the luxury to redesign or rethink the business in the short term. SSA and most large businesses cannot stop existing systems to rethink the business. There is an immediate crisis often which needs a solution. What can be done in this situation?

You should make use of the fact that most VLSS evolve over long periods of time, and you should prepare yourself for an immediate survival strategy. SSA's strategic history is instructive [see box]. SSA's new VLSS required seven years to evolve and followed a survival tactic which involved enormous hardware investment in the old software just to survive. But the same computer and telecommunications hardware needed to survive could be used to operate more sophisticated and much better software which supported an entire new way of doing business in the future. No doubt serendipity and plain luck played a role here as well as management insight.

A large telecommunications firm followed a similar strategy. Faced with ancient software operating on saturated, aging machines, this firm employed a survival tactic similar to SSA's: it isolated the old files and software in a "core" database, purchased enormous computing power to speed up transaction processing, and developed a user friendly front end which isolated applications developers and users from the core data and files [see box].

Picture: Building An Envelope to Survive

Caption: Building a survival envelope around existing files and software may buy enough time for the more difficult task of organization and system redesign.

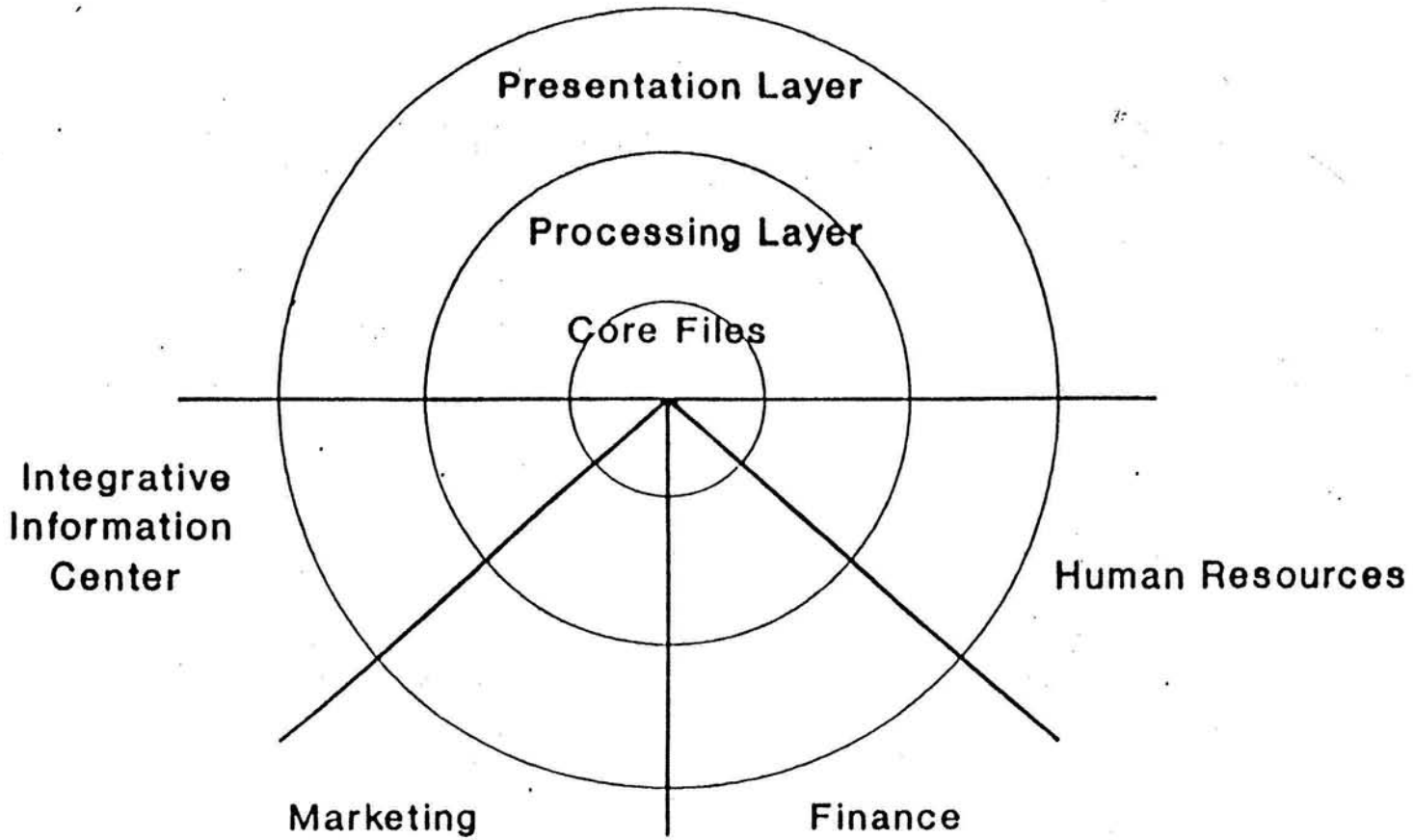
In other words, just to survive, this firm built an envelope around its traditional files and software.

Now with survival assured, it is possible to evolve new business procedures and the underlying files and software to support them. But the process is expected to take more than five years, to proceed in an evolutionary and adaptive manner, according to a flexible plan which optimizes locally, and exploits advantages as they become apparent.

While a great deal has been written about the strategic advantages available to firms who effectively use information technology, the historical evidence suggests as well that periodically systems become *strategic liabilities*. Huge embedded base systems developed to automate old business strategies and procedures become corporate loadstones. At moments of strategic transitions in an industry, these aging VLSS can contribute to further decline. All Fortune 1000 firms who have survived the last decade have such systems. In the late 90s we will have to learn how to rebuild these systems as we redesign our core business activities.

---

[1]. My colleague in this work was Professor Alan F. Westin, Columbia University. The results of this work are forthcoming. See Alan F. Westin and Kenneth C. Laudon, Information Technology at the Social Security Administration, 1935-1990.



Survival Envelope Based on Traditional Files

[2]. See "ADP Modernization. IRS' Tax System Redesign Progress and Plans for the Future," Comptroller General, General Accounting Office, April, 1988. See also Government Computer News, march 6, 1989.

[3]. See "Patent Files Vs. Computer Age," The New York Times, September 12, 1988.

[4]. General Accounting Office, "WWMMCS Continued Problems," Comptroller General of the United States, March 1989. For an earlier report which identifies the same problems see "The World Wide Military Command and control Information System--Problems in Information Resource Management," Comptroller General of the United States, General Accounting Office, October 19, 1981.

[5]. See "Strategic Systems Plans Gone Awry," Computerworld, March 14, 1988.

[6]. Ibid.

[7]. The only exception here is Robert W. Zmud, "Management of Large Software Development Efforts," MIS Quarterly, June 1980. And Robert W. Zmud, "Large Scale Interconnected Information Systems," Large Scale Systems, vol. 7, 1984. Both of these articles are prescient. Zmud points out the evolutionary nature of large scale systems, the need to develop "adaptable" systems, and systems which encourage organizational flexibility.

A few articles in the trade literature address system transformations. See for instance Michael Goldstein and John Hagel, "Systems Discontinuity: Roadblock to Strategic Change," Datamation, October 15, 1988. And a short piece by Frederic C. Withington, "Managing a Systems Transformation," Datamation, August 1, 1988.

Of course, Frederic P. Brooks' Jr. book The Mythical Man Month (Addison-Wesley, 1975) which describes the development of the operating system for the IBM 360 machiens in the 1960's stands out as a classic contribution to understanding VLSS. Nevertheless, Brooks' book dealt with building new systems. Our focus is on the still more complicated task of re-building systems *in situ*.

[8]. For instance, Software Productivity Research Inc. produced a study which claims that as program size grows, defects per thousand lines of code grows. And in a report they claim that for large scale projects (more than 60,000 lines of code):

*25% of the projects are cancelled before completion

*Less than 1% are finished on time, within budget, and meet all user needs.

*On average, large projects are finished a year late and costs twice as much as predicted.

The report ends up calling for reusable code and more programmer productivity even though there is no evidence tthat program errors, lack of productivity, and the like, were the principal cause of failure.

[9]. See Kenneth C. Laudon, "A General Model for Understanding the Relationship Between Information Systems and Organizations," National Science Foundation, Working Paper #1. Center For Research On Information Systems, NYU. 1989.
See also Kenneth C. Laudon and Jane P. Laudon, *Management Information Systems--A Contemporary Perspective.* New York: Macmillan, 1989.

[10]. See Businessweek, "It's Late, Costly, Incompetent--But Try Firing A Computer System," November 7, 1988.

[11]. On expected organizational life spans see William H. Starbuck, "Why Organizations Run into Crises . . .and Sometimes Survive Them," in Laudon and Turner (eds.) Information Technology and Strategic Management, Englewood Cliffs, New Jersey: Prentice Hall, 1989. See also Herbert Kaufman, Are Government Organizations Immortal?. Washington D.C.: the Brookings Institution, 1976.

[12]. On the historical evolution of the airline reservation systems see Duncan G. Copeland and James L. McKenney, "Airline Reservations Systems: Lessons From History," MIS Quarterly, September 1988. On the historical development of strategic plans in general, see Henry Mintzberg and Alexandra McHugh, "Strategy Formation in an Adhocracy," Administrative Science Quarterly, 30, June, 1985.

[13]. On manager misperception of risks and reward, see the review article by James G. March and Zur Shapira, "Managerial Perspectives on Risk and Risk Taking," Management Science, vol. 33, No. 11, 1987.

[14]. See for instance Donald C. Hambrick and Richard A. D'Aveni, "Large Corporation Failures as Downward Spirals," Administrative Science Quarterly, vol. 33, No. 1, March 1988

[15]. See for instance Henry Mintzberg and Alexandra McHugh, "Strategy Formation in an Adhocracy," Administrative Science Quarterly, 30 (1985).

[16]. Massive investments in information technology by the services sector of the U.S. economy simply has failed to raise the levels of productivity in that sector, the largest and fastest growing sector of the U.S. economy. This finding is true at the industry level as well. The finance industry, for instance, despite its technological investment, has actually experienced a negative productivity growth. See Stephen S. Roach, "Technology and the Services Sector: The Hidden Competitive Challenge," Technological Forecasting and Social Change, vol. 34, No., 4, 1988. Information technology investments in the goods or manufacturing sector have produced much better results.