CONTRAVIEW

Transforming IT departments to professional solution providers
A new paradigm on adding value

In the IT sector solutions are hot. It has become the new mantra for selling the same and new technology and more of it. Solution selling has infected business managers who are demanding higher quality service from their internal IT departments. IT departments are struggling to keep pace with fast changing business demands and innovative technology, being outpaced by vendors left and right. To keep on adding value IT departments will have to become more mature in adopting new technology and mapping to business needs. The Capability Maturity Models (CMM) has been defined as the implementation of standard and formalized processes. These processes have to assist in quality and speed of delivery as well as the maintenance of technology. The fundamental assumption behind CMM is that process control will lead to quality products. However no evidence exists to support this claim or causality. Standardization and formalization is making organizations more rigid whereas there is a principle need to be more flexible and adaptable. This article describes a new way of thinking about how IT-organizations can be organized for innovation.

Traditional thinking

Traditional models, supported by TQM-related movements including Capability Maturity Models, have their roots in scientific management thinking. More popular this is called machine thinking. Standard processes, fixed functions and roles, controls, division of labor and statistical measures all form part of the machine metaphor. The machine is the epiphany of control and predictability (see figure 1). Known and measurable input is taken through a production process to be transformed into a known output. This is applicable for known situations that are recurring like mass production of physical goods. Software development does not follow these principles, at least not all development. Almost every product is unique. Quality of the product is part engineering and part human ingenuity and improvisation. Software is a knowledge product following knowledge principles. This is work of professionals. Just imagine telling an artist how to hold the brush, the amount and type of paint to use and how to use the brush. So quality of process and result depends primarily on the maturity of the professional.

Organic thinking

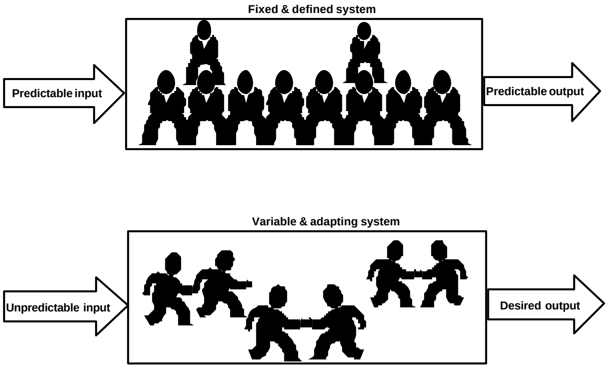

Figure 1: Machine and organic systems

A lot of misunderstanding exists of what organic thinking is. In part this is because no distinction is made between the machine part and the adaptive part of an organism. The machine part refers to the fixed functions of organs and the fixed interrelations between organs. Most of these organs form a closed system and are therefore not influenced by environmental changes. The adaptive part is the brain and environment facing organs like skin and limbs. They are part of the open system being influenced by environmental changes. Using organism as a metaphor implies the combination of open and closed, fixed and variable (see table

1). The open adaptable system is project and collaboration oriented due to the interdependency between skills to get the desired results (see figure 1). Key question is which type of thinking should be applied to which process within an IT department?

Machine perspective	Organism perspective
Closed to environment Inflexible Fixed and Mono functional Standard Recurring Robust	All machine attributes + Open to environment Flexible Situational and multi functional Adaptable Non-recurring

Table 1: Key differences between the two perspectives

Approach to work

Complexity science is giving us new insights into the level of predictability of processes. It also helps us to understand the level of repeatability of processes. To understand this better a distinction must be made between process and resource. When considering knowledge work it is neither the process that needs to be standardized nor the resources. Knowledge work is by its very nature conceptual activity of creativity and improvisation. It is a question of context and relevancy. Focus is on the things that need to be done instead of the way to do it. Standardizing processes generalizes context beyond a level of pragmatism, reducing complexity to a single and simple situation. By doing so the relevancy issue is made irrelevant. The world is reduced to elementary black and white.

System and cybernetic theory teach us that organisms are composed of three key elements; structure, process and patterns. So far in organizations structure has been viewed as mono-functional and fixed resulting in power hierarchies, processes as standard and inflexible. This is not a realistic representation of what happens in reality. The wrong attention is given to structure and process, little to none to patterns. Through the rigorous adoption of intranets and email these informal and shifting patterns within organizations are becoming more important. They are becoming a determinant in what gets done and how. Some important differences between standard and structured are given in table 2.

Standard	Structured
Uniform Context generic Control of behavior Linear, sequential Independent	Customized Context specific Control of state Cyclic, iterative Collaborative

Table 2: Essential differences between standard and structured processes

Knowledge work should be structured and not standardized. It is like playing sports. Set boundaries, agree on the rules but never interfere with the game. Off the field managers can coach and train, but on the field it is up to the professional. Not the formal but informal patterns within organizations determine speed and quality of result. So how can we organize for knowledge work?

Solution Development Cycle

Operational work within an IT or R&D department can be represented by the Solution Development Cycle (see figure 2). Solution is the fulfillment of a need in terms of people, process and technology services and products. These three levels form an integral part of the cycle and must be regarded as one interdependent system. The level of people is the user community or the business, including management. The level of process is the business process. The technology level can be anything technical, from a telephone to advanced CAD/CAM workstations and fully integrated systems. Within the technology layer the same three layers apply to produce and maintain the technology. In this light a new phenomenon is dawning called Professional Services Automation (PSA). PSA is a transaction based IT platform for professional service organizations (PSO) like R&D, consultancy and IT. PSA is the Enterprise Resource Planning system for the PSO sector.

Need phase	Maintenance phase
Multiple-Single skills 80:20	Multiple-Single skills 20:80
Cycle per solution long	Cycle per solution short
Long timeframe (weeks-months)	Short timeframe (hours-days)
Few projects (jobs)	Many projects (jobs)
Result oriented	Effort oriented
Effectiveness focus	Efficiency focus
Many interdependencies	Few interdependencies
Lose organic planning	Controlled planning

Table 3: Sliding scale between the phases of the Solution Development Cycle

The cycle consists of four main phases, from need to maintenance. The key differences between the types of work in these phases are highlighted in table 3. The phase of defining the need can be viewed in past, present and future timeframes. For past and present timeframes problems or issues that need resolving is normally used. For future timeframe need is regarded as opportunities or desires. This phase is about creating awareness of possible value IT can bring to the business. Problem analysis, visioning and strategizing are part of this phase.

The solution phase is about defining the answer to the need in qualitative and quantitative generic terms. Solutions are regarded from their technology component although solutions exist without changing the technology! Depending on the complexity of the need more than one scenario - from conservative to aggressive - can and possibly should be described. The solution description is in layman's terms devoid of jargon. The level of detail to start developing is reached in the next phase. The solution should have enough detail to assess feasibility and impact on the business and IT architectures. Part of this phase is the realization strategy highlighting approach, organization, skills and timelines of the realization project. The result of this phase is a value proposition in quantitative and qualitative terms.

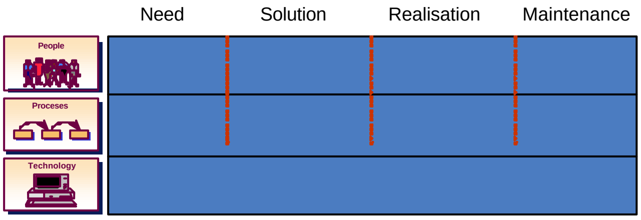

Figure 2: Solution Development Cycle

The phase of realization is the phase of further defining, development and implementing the solution. This phase is synonymous to standard development methods. It ends in acceptance of the solution and the taking into production.
The last phase of the cycle is keeping the solution operational. This phase includes the incident, problem and change management aspects. Significant changes must at all times go through the same development cycle to ascertain value and impact on architectures and other projects.

The cycle may give the impression of a linear and sequential way to create solutions. This is the wrong impression. During all phases continuous feedback to earlier content and decisions, certainly in the early phases, is critical. The cycle only works when an iterative, evolutionary view is adopted. How to prevent infinite and vicious loops? The simplest way is to define minimal threshold parameters to push the solution forward in the cycle. The second success factor is a continuation of key people involved in the solution cycle to ensure preservation of knowledge of both process and content. This will prevent unnecessary coming back on earlier decisions and consistency in terminology.

The critical issue of this cycle is twofold. By the nature of the way companies work and are structured an integrated system or synthetic approach is paramount for adding value. This implies multi-disciplinary teams collaborating. To do so effectively a common frame of reference must exist. This is not the case in practice. A further complicating factor is the use of external consultants causing discontinuity in knowledge and understanding. The continuation of the concept as defined in the solution phase is difficult due to its business orientation (see table 4). Furthermore, in practice people, process and technology development require different approaches and disciplines. Solutions with significant changes in the three layers must by definition be managed under a program, integrating and aligning the three aspects continuously. Again, in practice the discontinuity arises at the beginning of the realization phase.

Aspect	Business orientation	IT orientation
Scope & size	Large, complete functional areas	Limited, often 1 process
Detail	Highlights	In depth
Approach	Top-down from total system	Bottom-up from module
Language	Business jargon	IT jargon
Models	(work)Flow, generic steps	Technical models (ERD/DFD)
Purpose	Understanding and decision	Completeness
Objective	New business system	System development
Transforming factor	Product through process	Information flows and stocks

Table 4: Differences between Business and IT orientation

The Solution Development Cycle is a learning and self-generating system. This implies that it feeds other processes with information, that results including templates and documentation are made reusable. Key to a learning system is performance measurement, both of process and solution. Part of the solution development cycle thinking is another way of looking at business planning cycles. With the fast changes in technology it makes little sense to pretend to predict desired projects in the coming year, let alone give an overview of the coming years. Desired maintenance on existing systems can be analyzed but innovations should be calculated and budgeted on an opportunity basis.

This cycle is applicable to two categories of solutions. Those related to present systems or technology and those related to new technology. New in this context can be absolute or relative to the IT department. The type of work is fundamentally different for these two categories. For new technology the first two phases ending in a business case are not analytical. The past can be analyzed; the future must be designed. Synthetic skills and general knowledge transcending disciplines is needed. These skills are not common and often not present in IT departments. Coming back to the key question which processes should be structured in which way. Hierarchy and functional skill decomposition of the whole development cycle will reduce it to a bureaucratic system with multiple hand-offs. This should be avoided at all cost. Most IT departments are still hierarchically structured, which might fit well for the latter phases of the development cycle. The first two are best structured as one open system without skill centers, offices, entities or service lines. It is possible to create one pool of experts to draw from depending on the type of solution to be developed.

Churning the machine

Regular surveys on the quality of IT and related projects, like those held by the Standish group and other analysts that show no improvement in the quality of delivered work. On average only some 25% of the projects succeed in terms of fitness for need and being on time and within budget. Approximately 50% are deemed unsuccessful on one or more of these points. The remainder is stopped or completely unsuccessful. This track record has been relatively consistent over the last 40 years. Even process standardization has not changed this.

CMM is firmly rooted in the Total Quality Measurement concept of quality of the output depending on the quality of the process used to produce it. Process capability is defined as the statistical understanding of expected performance. The CMM design principles are repeatability, measurement, control and standardization. All of this points to mass production and bureaucratic design principles. Despite tenacious fact-finding no firm empirical facts exist to support the causality of product quality coming forth out of process quality.

In a recent workshop[1] held among high maturity IT departments a significant comment was raised about the uncertainty of the meaning of maturity. In fact the concept of high maturity is vague and ambiguous. Maturity as far as it is defined lies in the level of adoption of standard, bureaucratic and fixed processes, all neatly documented. This very narrow definition of maturity has been a driver for the misunderstanding of quality service from IT departments. Maturity of an IT organization depends on the level of value it adds to the business, irrespective of how this is done.

Part of the legacy we have inherited from the ISO-9000 hype in the early 90's is auditing by third parties. The only simple way to ascertain process control is by verifying whether written down procedures are being followed. This independent evaluation process has been a significant factor in how TQM has been interpreted and implemented. It has influenced the level of detail and the extent of documenting and standardizing processes. This interpretation is totally contra the present management thinking of breaking down bureaucracy and moving away from formalization. It in facts fuels the hunger of the bureaucratic monster, after all formalization leads to more formalization.

According to Russell Ackoff[2] it is mostly worse to do the wrong thing the right way than to do the right thing the wrong way. Systems' thinking has shown that doing the right things determines quality of results. Of course this depends on the definition of quality. So is there any sense in documenting *standard* processes and assigning *fixed* roles and responsibilities to people? This only makes sense when processes and results are repetitive in a stable environment. In reality very few processes possess this property. Customization and heterogeneous environments are in part cause of this.

What maturity is all about!

Maturity of an organization is the consistent supply of goods and services that meet the desired requirements. Key in this definition is what the requirements are. Besides functional and performance specification of technology there is the question of adding value to the business. Although commonly measured in terms of Net Value or Return on Investment the issue of reliable measurement has been haunting IT projects for decades. The increase in productivity due to the application of technology has been food for many debates and research without indisputable proof. Intangible benefits are left outside the scope of measurement to simplify matters. Despite being debatable from a practical application perspective maturity is regarded as consistently adding value. This does not imply doing the same thing in a standard way!

Adding value in itself, other than through financial measurements, is too vague and obscure. So the concept of value adding is decomposed into seven dimensions:

- Consistency
- Proactivity
- Sensitivity
- Relevancy
- Satisfaction
- Synergy
- Choice

Consistency

Consistency of service and solution implies process and technology control. The agreed level of service needs to be met over a period of time. Service agreements cover the fitness for need and purpose in business terms as well as the response to incidents, problems, changes and support. A mature organization gives transparent access to service status and regularly reports. Quick and appropriate escalation takes place when needed in collaboration with the user community. A common frame of reference between business and IT as well as within an IT department is more relevant for quality of interpretation. Software development is knowledge work and therefore susceptible to misperception and incorrect interpretation. A lot of effort and result is lost in the misalignment of people.

Proactivity

Most IT departments are reactive. They wait for the customer to tell them what they want. Mature departments are more forward-looking, both short and longer terms. They proactively search the market for new innovations. They analyze application and adoption of technology by competitors. They form interdisciplinary teams with the business in the phases of Solution

Development Cycle. For innovative solutions this means new work for which improvisation is a critical skill. Simulating and experimenting, by improvising and trying, learning new things. Another often-ignored aspect is motivation. Motivation can be a key factor in determining quality of work.

Sensitivity

In general people with a technical background are analytical and solution oriented. With solution orientation the problem and why it exists is left untouched. This orientation is action driven, quickly jumping to conclusions and designing possible answers to a question not asked. This sign of immaturity can be overcome but the change will be hard. It goes to the roots of thinking patterns and the road of unlearning can be long and cumbersome. A second aspect of sensitivity is business understanding. Detailed understanding of business process and organizational impact can in the long run lead to added value services like process modeling and change management.

Relevance

The issue of relevance is to do with context for rules, models and theory. Simply said, not all solutions are equal so don't treat them as such. Source, size, novelty, interdependence and interfacing are just some dimensions determining the complexity of a solution. Furthermore what needs to be done depends on external factors from the business and market. At each phase the approach, skills and desired details are determined based on what the solution under development is, thereby customizing the cycle according to relevance. Flexibility of role allocation and optimal deployment of scarce skills are signs of maturity. Another aspect of relevance is the active decommissioning of solutions when economical and/or technical reasons warrant this.

Satisfaction

Contra to general engineering principles applied in software development and other technical discipline's customers do not know their requirement. This reactive *modus operandi,* despite the attention it has been given, is still the general practice. With the pace of change the advisor role for internal departments is becoming more critical by the day. To cross the enormous gorge between translating requirements and assisting in defining them implies a fundamental change in thinking. The level of involvement with the business predominantly determines the value added. For this co-makership and co-engineering capabilities are needed. It has to do with the capability of forward looking, advising and assisting on how to make optimal use of new technology.

Synergy

To reduce solution costs to a minimum reusability of method, tools as well as IT components must be sought after. This is not the same as standardization, i.e. rationalizing similar problems to exploit the same technology. It is more about harmonization i.e. exploit same technology in multiple ways or apply in different contexts. Many centralization operations from local to regional development have failed because of lack of local flavors. Designing and technology selection should be done with reusability in the future in mind. Dedicated point solutions that are not reusable are best to be avoided. This implies that the architecture choice is critical and that choosing for open standards leaves the most degrees of freedom for later decisions.

Choice

As previously described most technical people are solution oriented. This implicitly implies that requirements are mapped to previous experiences making the solution predetermined or a self-fulfilling concept. In very few cases are scenarios with pro's and contra's presently to decisionmakers. If this is the case then usually investments are high and external consultants are driving the value proposition. In even fewer cases is a scenario included what happens if nothing is done. A business case with strong arguments not to develop a solution any further, basically saying no to the business, is part of the mature relationship IT departments need to develop with the business.

Solution Life cycle

From a quality standpoint solutions or just plain technology should be written off, both economically and technically. Very view companies actively and continuously decommission information systems. The process of information planning in which technical and economical considerations for decommissioning or upgrading are made is still in its infancy. Lifecycle and Development cycle are often confused. Although understandable the two are very different. Much depends on the definition of solution. Two of the most common definitions are a suite of technology (for example covering all the layers of the OSI-model) or a (group) of business processes. If the technology definition is chosen then usually the length in service is taken as indicator for maturity. Large organization like multinationals often have different version of the same solution in different regions or locations. These different operational versions form part of the solution lifecycle.

Making the change

Figure 3 shows the key aspects of a purposeful complex adaptive system like an IT department. Complex systems grow and sustain without central coordination and control but by applying rule-based purpose. Key is finding simple rules for people within the system to self-organize and to create complex and innovative solutions.

Purpose and principles give meaning and direction to the system. It allows people to identify with the system and contribute to it from mutual and individual interests. If no shared interest exists between individual and system interest a crisis occurs. Often purpose is vague and ambiguous causing misalignment and conflict. Purpose and rules are commonly interlocked and reinforce each other. Rules and behaviors form a sub cycle continuously reinforcing and sustaining old and undesired behavior. Rules are mostly the unconscious patterns in people's mind and determine how they act and react. People need to want to break with preconception, otherwise no change will occur. The only way to change such a system is by new understanding supported by theory, metaphor and novel formulation. A clear new world needs to be formulated and effectuated, continuously reinforced by senior people in the organization.

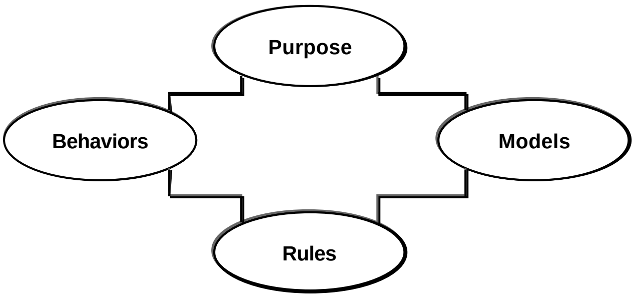

Figure 3: System aspects

[1] Mark Paulk and Mary Beth Chrissis, *November 1999 High Maturity Workshop*, March 2000, Special report CMU/SEI-2000-SR-003

[2] Russell Ackoff, *Re-creating the corporation. A design of organizations for the 21ste century*, 1999, Oxford University Press