CHAPTER 4: IT Organizational Structure

All the 3 quality standards in discussion (ISO, SEI & SPICE) put very little emphasis on the organizational structure, though they do acknowledge the importance of it.

This chapter starts with discussing and analyzing the organizational structure impact on the software quality standards. But before the organizational structure is looked at the SEI definition for the organizational structure (adopted for this project) is defined. The weakness of the existing organization model also be analyzed which will follow by analyzing the organization style of company making the frame work for the proposed organization model. The justification, the implementation process and the limitation of the propose model will be discussed in details.

The proposed organizational structure model will be confined to the decision process and 'power play"[3] within the organization rather then the actual physical organizational structure.

Besides the definition and analysis of organization structure that is given by SEI and ISO, the other 3 good article that form the basic framework and understanding of organization structure are that of Dillon & Backhouse (1996), Grover & Segars (1996) and Currie (1996). Dhillon's studied the risk in using IT within the organization. Grover and Segars analyzed the organization characteristics and Information System Structure base on an international survey. Currie's study too based on an survey finding of private and public sector on the organization structure and the use of Information Technology. The organization style method was taken from the classic article of Allison (1969).

[3] *Power play – here refers to the flow of control within the organization.*

4.1 Roles of IT Organization

Grover and Seger (96) defined the variable that attributed to organization context as follows:

(a) organization size
(b) economic sector
(c) Structure: coordinating and distribution of decision making)
(d) Role of IT

The hardware policy was measured along the following

(a) Centralized
(b) De-centralized
(c) Distributed

Grover and Segar's (96) organization definition (which is adopted for this thesis) is as follows:

(a)	Small organization:	Employee between 0 and 251
(b)	Medium Organization:	Employee between 250 and 1001
(c)	large Organization:	Employee more than 1000

Although the SEI/CMM and ISO attempts to remain independent of specific organizational structures and models, both the 2 software standards have defined terminology relating to organizational structure and roles which may differ from that followed by any specific organization. In this section we will review the SEI/CMM definition of the organizational structure terminology.

A SEI/CMM definition of an organizational role is a unit of defined responsibility that may be assumed by one or more individuals.

The roles identified in SEI are listed below. They will not be discussed in detailed and will take the same definition as that defined by SEI.

- ❑ Manager role
- ❑ Senior manager role
- ❑ Project Manager

- ❑ Project Software Manager
- ❑ First-line software manager
- ❑ Software task leader
- ❑ Staff
- ❑ Software engineering staff
- ❑ individuals

A similar breakout of roles can be identified for other engineering groups such as a system engineering or a system test.

In a particular project or organization, there does not need to be a one-to-one correspondence between these roles and individuals. One person could perform in multiple roles, or each role could be performed by separate individuals.

For example, on a small, software-only project, one person might have as many as six roles: the system engineering first-line manager, the project system engineering manager, the software first-line manager, the project software manager, the project manager, and the software configuration management manager.

On a slightly larger project, one person might be the system engineering first-line manager, the project system engineering manager, and the project manager while another person might be both the first-line software manager and the project software manager. These two managers might be in the same second-line organization or in different second-line organizations.

On a large project, many roles, especially those of management, would likely be filled by separate individuals.

Two new propose "functions" will be defined in addition to the SEI defined position and they are:

a) RAD Function

b) IT R&D Function

RAD (Rapid Application Development) Function: This is a new position and the role will be to provide the business requirement from the IT perspective. The responsibility of the RAD manager is to provide a short term strategic software solution to the IT organization.

IT R&D (Research and Development) Function : This is also a new position that is introduced in this thesis and will be used in the proposed organization. The responsibility of this manager is to provide the business direction.

4.2 Organization, Group and Project

Similar to the organizational role the organizational structure definition for this thesis will be adopted from the SEI/CMM definition. The key terms in the organizational structure are: organization, project and group.

CMM defines an organization as a unit within a company or other entity (e.g., government agency or branch of service) within which many projects are managed as a whole. All projects within an organization share a common top-level manager and common policies.

A project is defined (by SEI/CMM) as an undertaking requiring concerted effort, which is focused on developing and/or maintaining a specific product. The product may include hardware, software, and other components. Typically a project has its own funding, cost accounting, and delivery schedule.

A group is the collection of functions, managers, and individuals who have responsibility for a set of tasks or activities. A group could vary from a single individual assigned part time, to several part-time individuals assigned from different functions, to several individuals dedicated full time.

Groups commonly referred to in the CMM are as follows:

- Software engineering group
- Software-related groups
- Software engineering process group
- System engineering group
- System test group
- Software quality assurance group
- Software configuration management group
- Training group

The software engineering group is the collection of individuals (both managers and technical staff) who have responsibility for software development and maintenance activities (i.e., requirements analysis, design, code, and test) for a project.

Groups performing software-related work, such as the software quality assurance group, the software configuration management group, and the software engineering process groups, are not included in the software engineering group. These groups are considered to be one of the "other software-related groups."

A software-related group is the collection of individuals (both managers and technical staff) representing a software engineering discipline that supports, but is not directly responsible for, software development and/or maintenance.

Examples of software engineering disciplines include software quality assurance and software configuration management.

The software engineering process group is the group of specialists who facilitate the definition, maintenance, and improvement of the software process used by the organization. In the key practices, this group is generically referred to as "the group responsible for the organization's software process activities."

The system engineering group is the collection of individuals (both managers and technical staff) who have responsibility for specifying the system requirements; allocating the system requirements to the hardware, software, and other components; specifying the interfaces between the hardware, software, and other components; and monitoring the design and development of these components to ensure conformance with their specifications.

The system test group is the collection of individuals (both managers and technical staff) who have responsibility for planning and performing the independent system and testing of the software to determine whether the software product satisfies its requirements.

The software quality assurance group is the collection of individuals (both managers and technical staff) who plan and implement the project's quality assurance activities to ensure the software process steps and standards are followed.

The software configuration management group is the collection of individuals (both managers and technical staff) who have responsibility for planning, coordinating, and implementing the formal configuration management activities for the software project.

The training group is the collection of individuals (both managers and staff) who are responsible for coordinating and arranging the training activities for an organization. This group typically prepares and conducts most of the training courses and coordinates use of other training vehicles.

4.3 Independence of the Organizational Structure

SEI/CMM and ISO calls for an independence of certain key practices within an organization. They call for independence when technical or organizational biases may affect the quality or risks associated with the project. The roles

that is requested to be independent in SEI/CMM are the SQA group and the test group.

SEI/CMM says there is a need for independence of the system and acceptance testing is based on technical considerations. This independence ensures that the testers are not inappropriately influenced by the design and implementation decisions made by the software developers or maintainers.

SEI/CMM says the independence of the SQA group is necessary so its members can perform their jobs without being influenced by project schedule and cost pressures. Ensuring effective operational independence without the organizational independence is difficult. For example, an employee reporting to the project manager may be reluctant to stop a test activity even though serious noncompliance issues exist.

Professional Judgment Application

To provide a complete set of valid principles that apply to a wide range of situations, some of the key practices are intentionally stated to allow for flexibility. Throughout the key practices, nonspecific phrases like "affected groups," "as appropriate," and "as necessary" are used. The use of such nonspecific terms is generally minimized in the key practices, with examples provided in many cases, at least for the first use of the term. These phrases may have different meanings for two different organizations, for two projects in a single organization, or for one project at different points in its life cycle. Each project or organization must clarify these phrases for its specific situation.

Clarifying these phrases requires the organization to consider the overall context in which they are used. The pertinent question is whether the specific interpretation of one of these phrases meets the goals of the key process area. Professional judgment must be used to determine whether the goals have been achieved. The glossary in Appendix B may provide guidance in interpreting these and other phrases in the key practices.

Professional judgment must also be used when interpreting the key practices and how they contribute to the goals of a key process area. In general, the key process areas describe a fundamental set of behaviors that all software organizations should exhibit, regardless of their size or their products. The key practices in the CMM, however, must be interpreted in the light of a project's or organization's business environment and specific circumstances. This interpretation should be based on an informed knowledge of both the CMM and the organization and its projects. The goals of the key process areas provide a means for structuring this interpretation. If an organization's implementation of a key process area satisfies the goals, but differs significantly from the key practices, the rationale for the interpretation should be documented. A documented rationale will help assessment and evaluation teams understand why certain practices are implemented the way they are.

Applying professional judgment leads to the issue of the "goodness" of the software process. The CMM does not place "goodness" requirements on the software process, although it does establish minimal criteria for a "reasonable" process in many software environments. The objective of process management is to establish processes that are used and can act as a foundation for systematic improvement based on the organization's business needs.

What are the criteria for a "reasonable" software process? A reasonable software process is one that is effective in building the organizational capability and satisfies most of the requirements of a defined process. Specifically, it is practiced, documented, enforced, trained, measured, and able to improve.

If an organization established a software process for estimating that consisted of rolling the dice, would that constitute a reasonable process? It could certainly be documented and consistently followed. Some might even argue that it would be as realistic as many estimating techniques. "Rolling the dice" would, however, not be judged a reasonable estimating process by most

software professionals. Since it responds only to the laws of probability, it cannot be improved.

How far is it from "rolling the dice" to documenting a process to "go ask George?" This could be a very good method for estimating. As long as George is around, it could even be consistent and repeatable. It would not, however, satisfy our criteria since it cannot be trained to other individuals. It is a person-centered process that cannot be repeated without George. It does not build an ongoing organizational capability.

Using some variant of a Delphi method (a method where experts in a subject review the issues under consideration and come to a consensus on the recommendations related to the issue) for estimating would usually be judged a reasonable software process. A size estimating approach based on a Delphi method satisfies the criteria for a reasonable and effective process, even though the Delphi method is a person-centered process. An organizational capability can be based on a structured technique such as a Delphi method.

In a fundamental sense, professional judgment is necessary to make such distinctions. The difficulty lies in discriminating between compliance and goodness. The goals summarize the key practices, which, in turn, describe a reasonable software process. Complying with a reasonable process, however, does not mean that the process is efficient in achieving its purpose. There may be many factors influencing both organization and project success. For example, a successful project that builds a product that no one buys is a failure in the commercial world.

"Goodness" attributes can only be interpreted in the context of the business environment and specific circumstances of the project and the organization. Such "goodness" judgments can be made only by the organization as part of its continuous process improvement cycle. Perfection is never achieved, and continuous process improvement never ends.

4.4 The Firm's Organizational Style

There are many definitions of the organization style of a firm. This thesis will adopt the definition that is defined by Allison (1969). The reason is it tends to take mostly all the factors involved and because of its general acceptance.

The famous organizational model (Allison, 1969) is still applicable to today's organization in defining an organization. Allison's study looks at an organization from the decision making process and basically defines the organization as one of the following 3 models, namely: rational policies model, organizational model and bureaucratic/politics model. The table below briefly defines the 3 models:

Model	Decision Process
Rational Policy	These organizational uses policies to make decision. Here the choice is base on value maximizing
Organizational Process	Here the decision process is based on a group of people with common vision. organizations uses a strategic team to make decision
Bureaucratic Politic	Here the decision maker is a number of individual player with the most power or influence. These organization are controlled by whoever make the most powerful

Most of the Quality standards ignore the influencing factor of the firm's organizational factor and assumes them to be in the 'organizational category' whereas in reality many firms fall in the 'bureaucratic or political category.

This concept is important to note because in most cases in the Bureaucratic Politic model organization the software quality model is never implemented unless there is an immediate buy back.

4.5 Limitation of the Current Organizational model

To start with, the software quality standards do not define an organizational model to accommodate their quality requirement but rather identifies certain position/functions within the organizational structure. In general the current structure of the IT organization are quite synchronize among all IT function (changing very little). There is no data available as to the type of organizational structure exist. For the sake of argument the term 'current' organizational model (referred to in this project) will refer to the common organization structures that exist within ABC-Malaysia, DEF-Corp and GHI-Corp. These organization structures are shown in appendix (C).

The current organizational models do not define or address the software quality and business requirement as integral part of the organization structure. They completely ignore the new technology and the continuous improvement functions within the organizational model. The current model tends to be too ambiguous and does not take into the quality model requirements.. The organizational style of the organization is not considered at all in most organizational structure.

Most of the fundamental quality base should start with the organizational structure and then move to the IT policies & procedures.

A study (Yourdon, 96) suggested that Client/server systems programmers could be striving for too much quality in their systems which was supported by a survey which reports that the typical client/server project is 50 percent over budget and 50 percent behind schedule. This study suggests that a developer should not always focus on a superb end product but instead a practical alternative: good-enough software. The writer explains why a programmer's next client/server application might be better off if it is just good enough. This argument was supported by another study (Knutson, 96) that says, "the distributed development of distributed software can be a nightmare

when the goal of creating high-quality software under tight schedules is also factored in".

One of the greatest weakness of the SEI/CMM and the ISO requirement of the organization structure is that they look at more of the physical organization structure rather than the organizational decision structure .

4.6 The Proposed Organizational Structure: STARO Modal

There are many debates as to whether the software quality standards must fit into the *organizational structure*[4] or whether the organizational structure should be altered to meet the software quality standards.

The organizational structure of the IT functions within the organization plays a very important part in the Software Quality Management. Both SEI and ISO records an independent Software Quality Assurance Person to exist and he/she must not be part of the IT functions.

This section does not intend to propose the physical location of the characteristics and does not intend to provide the designation titles of the organizational structure, but looks at the structure from the functionality point of view.

[4] *The word organizational structure in this report refers to the organizational structure of the decision process and power play rather than the physical structure.*

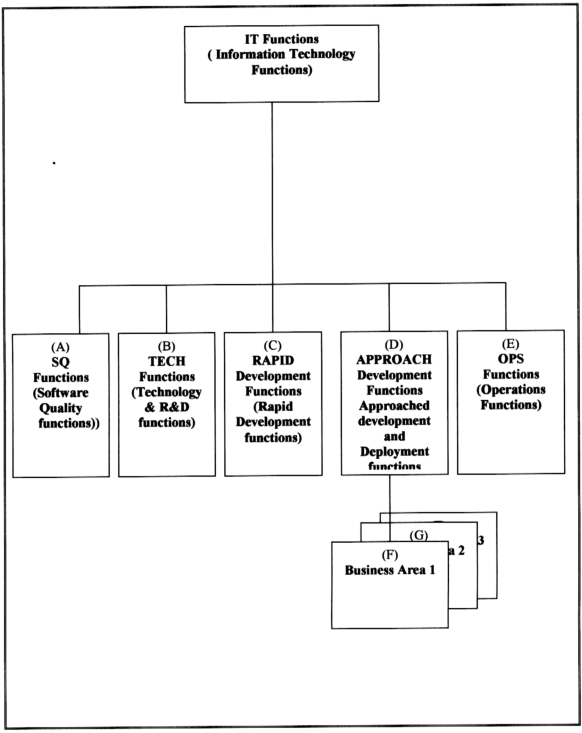


Figure 4.1: The STARO[5] Model (Organizational Structure)

[3] ***STARO***: ***S****oftware Quality – **T**echnology – **A**pplication Development and Deployment –**R**apid Development – **O**perational*

The STARO model is an empirical Organizational structure model proposed in this project. The model proposes the following general requirements:

- A ***leveled organization*** -
- Eliminates or reduces bureaucracy with IT
- the division of IT sub-functions with IT (based on functions of each dept
- the people's skill required to manage these functions (or function)
- the flexibility in term of company effort, resources and size variation of each functions and focus of the company.
- The primary objective of this model is to produce quality systems, meeting business requirement, customize as per the companies requirement
- The STARO model proposes that the sub-functions function for the software development and deployment be aligned with the firm's business functions rather than the IT process functions. The reason for this is to produce a better integration of the work flow process and at the same time reduce the bureaucracy between the IT functional functions.
- The position within the firm must be at the top most. IT functions is placed at the closest level at the top most level of the company organizational chart as IT is (growingly becoming the key functions (next to the marketing functions that will determine the companies' competitiveness in these competitive world.
- Most of the IT position as per defined by SEI can be used with the STARO model, with addition to 2 new positions (which were already discussed in the previous section) namely the RAD Manager and the Technology Manager.
- STARO model proposes one quality standards for all, but with flexible quality controls process and audits. The Rapid Functions is suppose to have a loose controls standards.
- Within each business application unit is just 2 levels; *Project Manager* and the *project team players*. The flatter organizational approach is applied here to reduce conflict of inter-functional conflict and to remove inter-functional gray areas within a business unit. The project manager's role and responsibility is as that defined by SEI and the team players consist as that defined by SEI too.

- Outsourcing consideration.

The names of the section are just symbolic names given to identify a particular functional group with IT.

The key essence of the STARO model is the skill set required. For this project 2 type of skills set are identified and they are namely: ***accumulated skills*** sets and ***individual skill*** sets. The accumulated skills set are combinations of individual skills put together, whereas the individual skills set refers to the individual staff skills.

Software Quality functions

All the firms that have gone through the ISO9001 or SEI CMM certification will have an independent quality function. In reality the quality functions are at the same level as that as an IT function. The responsibility of the quality functions is to define and ensure the quality standards defined are adhered to. The skills set required for the staff for functions is: software quality standard knowledge (in depth), software auditing skill, review skill, software security, disasters recovery and business skill. They will be involved in software review and testing process and are ensured at the relevant control points.

Technology Functions

The Technology function is the Research & Development section. It's prime focus will be to do R&D and to improvise on a new technological product or process to ensure productivity, quality and are in line with the latest technology available. In some cases it will experiment with new technological and innovation method. One other function of this functions will be to ensure the software productivity, quality and other IT matrix are collected and analyzed. It will be the responsible of these functions to improve the IT matrix figures. The person who should head these functions must be researched and academically qualified. The key goal of these functions will be to continuously improve current processes.

Rapid Development/Delivery Functions

The Rapid Functions can be loosely define as ad-hoc IT functions but executed in a controlled environment. The most important requirement of these functions is the people's skill sets. These functions are people-depended rather than process dependent. For this reason the skill set of staff in these functions is a pre-requisite for these functions to exist. The staffs in these functions must have at least the following skills set: a very high knowledge of software quality standards and requirement, very well versed with the firm's business process and requirement, very good in software development skills. These people will likely to have been with the company for more than 2-5 years[6]. Hence the RAPID functions is something that cannot be developed overnight and will require between 2-5 years (depending on the complexity of the firm) to be formed as per the requirements stated These skills set applies to individuals – unlike that of technology is 'accumulated skills'. In addition he should be self disciplined, loyal and people with principles and ethics and a team player and gives assurance that his work is of high quality. Since these functions will have very little (or minimal) controls and security standards in place and the staff is entrusted to it is further proposed that there should have some 'code of conduct standards' (some of it concerning with ethical issues) to be in place for these functions. The danger of these functions is that that software product normally produced by skilled individuals tend to be very difficult to be 'handed over' to other groups for continuity in work. To reduce this risk the work of this development will be handed over to the "APROACH development" functions immediately after development. The "APROACH development" will do a reverse engineering to document all the development effort so that future enhancement can be done on the software. In some cases the software product produced by the ***Rapid Development*** functions will be used for short terms solution. The main goal of these functions will be to provide immediate (permanent or temporary) software solution that will immediately add value to the company's business.

[6] *Assuming it takes that long to grasp al l the firm business process and requirement.*

Software Deployment and Development Section

This is what that normally exists in most organizations. The one key highlight of these functions is to segregate the software development, deployment and testing function.

Operational Functions

These functions takes care of the operation of the firm. For instance the computer hardware, the communication circuit, the telephone system and the helpdesk process. The main skills set will be software and system and operating system skills, system security skills.

The above functions characteristics can be summarized in the table below. In this table the STARO Model's proposed functions are tabulated against the skills set required from the functional stuffs. The skills set are divided into the following:

- SQS Skill Software Quality Standard & Process skills. This refers to skills in the existing software quality standards and process skills. The High mean must know all standards and medium means in the standard process.
- TECH skills Technology skills which will include the new software and hardware (including communication) technology
- ANL Skills Analytical Skills. This refers to the analytical skill of the existing analyzing tool and also the new tools proposed in researches.
- CMP Skills Competitor skills. This refers to the staff skills in the external competitive force (e.g. the rival company)
- BUS Skills Business Skills. This refers to the staff skills in the firm's Business Process and requirements.
- OTH Skills Personal Skills. This refers to the individual character attributes.

Functions	Quality Standards/ Process Skill	Technology	Analysis Skills	Education Level	Competitors market	Business Skills	Key Personnel Attributes Required
RAPID Development	**VH**	**L**	**VH**	**M**	**VH**	**VH**	Self Disciplined Loyal Ethical/ Profession
Approached Development	**VH**	**M**	**VH**	**H**	**L**	**M**	Team Player Quality aware
Operational	**M**	**L**	**L**	**L**	**L**	**L**	Team Player
SQ	**VH**	**L**	**L**	**M**	**L**	**L**	Quality
Technology	**VH**	**VH**	**VH**	**VH**	**VH**	**VH**	Research Quality Academic Analytical

Functions	Goal
QA	• Ensure data & software security • Ensure quality controls in place & implemented
Technology & R&D	• Define Q standards • Define Infrastructure • Define process improvement methods • Define new technology into IT • Define tools for process improvement • Conduct Research and Development activities to improve product quality – such as response time.
Rapid Development	• Provide immediate short term solution • Provide quick system solution • Provide ad-hoc "critical" system
Approached	• Provide long term high quality system

Development	• Provide strategic software development • Ensure quality controls are adhere too
Operation	• Provide hardware and operation support from the operating system point of view

Major difference of STARO Model than existing organization structure.

The major difference of the STARO model as compared to the organizational structure are as follows:

- Unlike the existing organizational model the STARA model is flexible and most organizations can easily fit into the model by changing and re-aligning the focus and putting the right skill level as the specific functions
- This organizational structure always have quality software and the same time business requirements as objective. The current organizations either tend to be too much business focused (and hence creating wastage of IT resources) or too focused on the quality on the expense of business focus. A few major companies were able to balance the two and create a good IT functions.
- The key difference in the STARO model is the new ***'Rapid Development functions'.*** This proposed functions will not be accepted by any of the existing software quality standards like SEI or ISO for the simple reason that it's focus is on the people's skill rather than the process skill. Most of the ad-hoc IT organizations will argue that they follow this requirement. That is not true. Because most of current organization will not meet pre-requisite of the people skill set forth by the RAPID development functions.

The above empirical deduced characteristics can be supported with its compliance with existing major quality standards (ISO and SEI), the firms strategic management style and the business needs from IT.

Object-oriented technology developer (Mattison & Sipolt, 95) are telling that the real payoff will not be achieved until the IS organization has a restructure the IT organization to fit this new model. The above characteristic list aims to do that.

It is an undoubted fact that 'appropriate' software development tools can enhance software quality significantly in each phase of the development process, from requirements analysis to final testing and integration. But there is the danger of choosing the wrong tool that can actually hinder software development Kavi and Nahouraii (96).

Giant software houses such as Microsoft has emphasizes on time instead of quality and attempts to deliver mission-critical, enterprise-quality systems to users in a short time, but on the expense of quality. They can effort it since they are the currently have the monopoly of the software industry, but it will becoming increasing difficult to maintain this practice and stay ahead. Microsoft uses technology innovation and coupled with as TQM's attribute of forging strong partnerships with leading system vendors, system integrators, and independent software vendors to give customers integrated technology and service offerings.

Both SEI and ISO do not really support technology innovation in-spite of their preference for automated process. The tools recommended by these quality standards are not checked but are accepted to improve the process.

Speed is the new customer requirement. The differentiation between a good quality software and a bad quality software is increasingly becoming narrower with the automated tool available and the cost to increase the slight software quality level has becoming increasingly more expensive (one propose study to do survey on the cost versus the amount of quality). Customers are demanding new architecture.

In addition the dependency of the software on hardware is becoming increasingly smaller and the rate of the software is growing is not proportion to

the rate of the communication lines are growing. The response time and uninterrupted system uptime is an increasing area of focus. Lots of software programs are becoming the integral part of the hardware – one such example is the Windows 95 replacing DOS.

One of the strongest point of SEI is its 'reusability objective". 'Reusability is an undoubtably an accepted requirement for quality and also for optimization. Frakes and Fox (95) explained the definition of re-useable is pretty clearly stated in their study which is supported by a survey data. He defined 'reuse' as "the use of existing software knowledge or artifacts to build new software artifacts". His article defines the questions which refer to the reuse of in the following areas:

- Reuse of common assets,
- effects of programming language on reuse,
- influence of CASE tools,
- perceived economic feasibility,
- reuse education,
- software engineering experience,
- recognition rewards,
- reuse measurement,
- promotion of reuse by a common software process,
- developers' preferences for building from scratch or reusing,
- inhibition of reuse due to legal problems or quality issues,
- use of a reuse repository,
- prevalence of reuse in particular industries,
- predictors of organizational reuse, and
- measurement of reuse, quality, and productivity.

Another study (Griss, Wosser, 95) in support of reuse and with a smaller scope discloses Hewlett-Packard's experiences with software reusability. This study focuses on the (3) factors that are required for a successful reuse programs. These factors (according to the study) are management leadership and support, organizational change, and the creation of a reuse mindset. It

focuses on a systematic reuse of the IT processes. RAD approach to development in general has 'reuse' as an integral part of the it methodology. Part of a special section on a rapid application development (RAD). Reuse reduces cost and cycle time (Henry & Faller, 95) .

There is no Universal Systems Development Methodology. The methodology used depends on the type of hardware platform the programming language and the management IT innovation strategy. A universal methodology is unlikely to satisfy all needs and tends to be rigid. Integrated approaches, the current trend in SDM development, attempts to cover all phases of the systems development process. Jaakkola and Drake (Jeaakkola, Drake, 91) advanced SDM (ASDM) approach is proposed that combines the coverage of the scope of the development life cycle offered by traditional methodologies and the methods and techniques offered by an integrated methodology. The ASDM likely would comprise a predefined set of methods based on each of the traditional structured, automated, and prototyping approaches.

A formal tailoring process is required at the start of each project.

The are many legacy systems sitting in many larger firms such as Motorola. These systems have a large maintenance bill and too costly to be replaced. On top of it they have been enhanced (with patch on) to include all the "collection" of requirement through the years.

With a new leading edge of technology emerging there will be a need to integrate business and technology strategies. The importance of a skilled and informed workforce and it knowing the greatest ideas are worthless without the resources to support them. Industry leaders offer some predictions as to what the technology challenges insurers will have to overcome to achieve success in the 21st century.

TQM appears to be heading toward the right direction. It will not die so soon. More and more concepts will continue to build on the TQM methodology. We

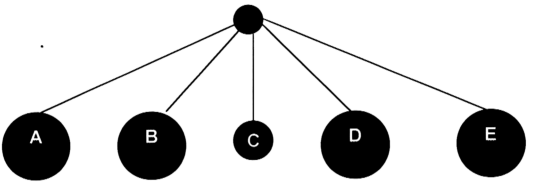
will probably witness the TQM concept moving from the awareness stage to the commitment stage in an organization.

Independence of functions

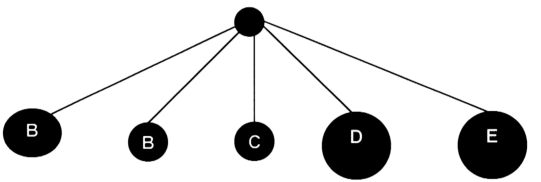
The STARO model follows the SEI and ISO quality model in defining the independent function within the organization structure. The independence function in STARA model is the Software Quality Function.

4.7 Application of the STARO Model

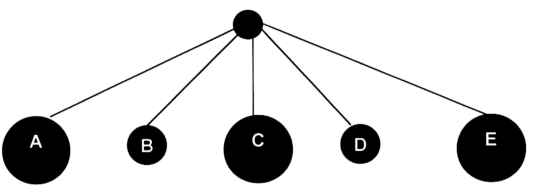
The diagram below show the possible application of STARO model (in size)



Orientation (1)



Orientation (2)



Orientation (3)

Red Bubble signifies a smaller or no emphasis on this function

Blue Bubble signifies a medium emphasis on this function

Green Bubble signifies a large emphasis on this function

The following symbol where used in the application diagram in the earlier page.

Symbol	STARO Model Function name
A	Quality Function
B	Technology and R&D functions
C	Rapid Development Function
D	Approach Development Function
E	Operational Function

The STARO model does not require the firm to have all the functions divided by department but rather by function. The model also is flexible enough to allow the activity to be of low key or usage within its organization structure depending on it size. The only requirement is that the minimum function that must exist within IT organization are Quality, Approach Development and Operational function.

The figure above shows some of the orientation in terms of a firm's resources and effort focus. Orientation (A) is the firm that is putting it's emphasis in all the functions except for RAPID development function. A typical of this organization model for a software house. Orientation (B) probably shows that

the company that do not require much technological, quality and RAPID development functions. Typically these will be organization that have organization model are that which are not much into technology and not in need of electronic information much – probably labor orientation, such as company in the cement or construction industry. Orientation (C) is a good example of an organization that is heavily dependent of electronic information but yet IT is not their main business. That is the reason their focus is to get quick software solutions to meet their business needs. Example of firm that fall into this category are air express company such as DHL.

4.8 Justifications for the STARO Model

The STARO model is built on the TQM model and links technology, people skill and business needs. In addition it considers the company's organizational styles. The STARO model addresses some of the current disadvantages of the organizational model

One of the greatest advantages of this proposed organizational structure is that it is above all to address the immediate requirement of the business community. This is a very important key objective as it proposes to provide a quick and immediate solution to the customer with very little (or no) consideration of the 'existing' software quality standards. This does not mean the product created is of a low quality – but just that the product was created with very little controls in place to meet the user's requirement of speed. For this reason the people in the RAD functions are highly qualified and experienced programmer, designer, analysis and managers. The STARO model take into the human knowledge/skill as a competitive tools. This is something SEI and ISO do not support and they are into moving this human skill into documented processes.

The proposed organizational structure is able to balance the software quality requirement and the business requirements and move the software

organization and hence the software quality standards closer to the real world requirement instead of just merely meeting the software quality requirements.

This organizational structure can also put a focus according to the need of the external environment completion . For instance if the external competitors are focused in creating a rapid development of a product by foregoing some of the quality checks that will slow the development process. It will fit any structure and decision process styles.

Hence considering the business needs, the IT organization must have the following characteristics:

- Able to provide software solution to reduce cost
- Able to improve software solution to improve productivity
- Able to provide 100% uptime of the existing application
- Able to provide ad-hoc IT request with the short cycle time
- Able to provide accurate financial statement.
- Able to provide trend or forecast estimates (of say sales)
-

How the above is done the IT organization must have the following characteristics:

- By providing a short software development time
- Able to provide an ad-hoc software solution
- Abel to provide long term software solutions
- Able to ensure 100 % uptime and availability of existing software systems

One of the strongest point of this model is that it takes the firm business objective, the firm decision process into consideration and it is very flexible to change with time the firm business and at the same time does not loses the focus on quality process requirement.

Finally this model propose to reduce conflicts between business needs and software quality requirement.

4.8 STARO Model Implementation Consideration

As Michael Hammer and James Champy (Hammer, 93) stated in their book, the best way to make changes and re-engineer the corporation is by rapid and fast transition. It is recommended that the STARO model be implemented in this fashion to get the best and effective result. To implement the STARO model there are a few factors to be considered and they are: the cost factor, the moral factor, the refuse to change factor, the training factor and, the quality factor.

The cost factor relates to the cost involve in implementation the STARO model. The cost can be substantial depending on the size of the company. The bigger the organization the bigger the cost. It will also involve hiring new set of skills that never existed before in the organization. There may be cost to pay off redundant positions/staff (though is discourage). The cost of training is another cost to consider.

The moral factor relates to the moral of the staff that may speculate that their current role may be redundant. To overcome this factor the organization must increase awareness among the staff to them the company's goal and objective. Layoff should be avoided to the last resort but instead there is a position for all in the company, through training.

The refuse to change factors relates to the reluctance of staff to change. Most staffs that are comfortable in a particular position will be reluctant to change and fight against these new changes implementation. Some of these staff may be senior managers. To situation must be tackle very tactfully.

Finally the training factor relates to the training of the IT staff. The training of the staff must include both the business skill and IT skill (and other individual skills that will be required for the specific function in the STARO model.

Finally the last consideration is the quality factor. The STARO model will not work alone. It requires a proven software quality model to be in place. Let it be

SEI CMM (the preferred model propose by this research project) or the ISO 9001 the quality model must be in place. This STARO model will be able to form a good foundation for both the ISO 9001 and SEI CMM quality model.

4.9 Limitation of the STARO model

The STARO model focus more on the top line functions of the IT organization structure rather than going down to the function where the firm products are align within the IT organization structure. The STARO model support the idea that at one point or other the IT function will link with the business Area function (and it is stated by the business Area function) but does not goes into detail as how this link is to happen.

Probably one of the major weakness of the STARO model is the requirement stipulated to create some functionality's with the STARO model. For instance it is very difficult to find a person with very good technical, software quality and business skills (all 3 in one). Hence at most time the true Rapid function will take at least 5 year to be created (provided that the person trained for the position does not leave after training).

The other limitation of the STARO model is that it does not show the relationship between the user community and the IT organization.

As define in the project scope, the STARO model is an organizational decision structure and does not propose a link to the IT physical organization structure. The virtual office concept, where the office is physically located away from the physical site is not analyzed.

One limitation is it requires to a certain about the re-engineering of the organizational structure for most organization. In some cases it may require changes

The STARO model form the foundation required for a quality model. This research project focused of 3 software quality models namely: SEI-CMM, ISO 9001 and SPICE. All these 3 quality will work along with the STARO model but other quality model will require some analysis before the STARO model is adopted.