Egypt–SPIN Newsletter

Issue 12, Oct. – Dec., 2005

Sponsored by SECC

From the Editor (**Ahmed S. El-Shikh**)

Welcome to our 12th issue of Egypt –SPIN newsletter. In each issue we are trying to put together relevant information in the form of articles and recaps from the previous 6 months events hoping to provide our members of Egypt – SPIN with information to support their current interests.

> **SECC** is pleased to announce that **ITWorx**, one of the greatest Egyptian software companies, has achieved **CMMI Accreditation Maturity level 3 (Staged Representation)** by means of **SCAMPI Class "A" v1.1** on 29th December, 2005.

This issue introduces some hot topics in three series and two independent articles as follows: applying CMMI up to maturity level 5 (1st article), discuss the software industry in Egypt (2nd article), share real life experience in the field of software test automation, supplier management and real life experience in process improvement (3rd, 4th and 5th articles respectively).

***Eng. Ahmed Nabil and Eng. Mona Arishi** share their experience in the **journey to achieve CMMI maturity level 5**. The article provides some insight on the people and teamwork as the major factor of success.*

***Dr. Ramiz Kameel** completes his **series to discuss** the **nature of the Egyptian software industry**. His article identifying the **standards that can control the relation between different roles in industry,** especially the role of the government toward the vendors.*

***Eng. Omar Kamal** interrupts his series about software testing with an article to brief an approach for developing **infrastructure for automatic validation** and its relation with CMMI.*

***Eng. Mohamed Abo Zied** continues the **series for explaining the CMMI** version 1.1 process areas as presented in the "**Intermediate Concepts of CMMI**" course. His article describes the **Supplier Agreement Management** process area.*

***Eng. Ahmed Adb El Aziz** shares his experience –2nd article in the series- in **CMMI Implementation** journey according to the **IDEAL** model. His article describes the establish phases and a part of the acting phase.*

We hope we succeed to give you an idea about what is going in our community. Please write to the editor your comments about our progress. We always ask you to submit short articles for publication that deal with your experience in defining, developing and managing software efforts as well as process improvement experience. Remember that our goal is to encourage an interchange between our readers. You can email spin@secc.org.eg or jaselshikh@yahoo.com

Table of Contents

SPI Journey in TDC and Achieving CMMI L5

By: Ahmed Nabil & Mona Arishi

Software Process Improvement is a subject that has been well presented in previous editions of this SPIN. Details of success stories of how organizations reached their Quality Objective were also discussed. In this article, we will provide some insight on the people: the team that led the journey that transformed a small unit of software developers at maturity Level 1 to a strong, aggressive and highly competitive software development organization. The first organization in Egypt, Middle East and Africa to reach the challenging quality target of Capability Maturity Model Integrated Level 5.

The Technology Development Center (TDC) inspired by a strong commitment from the Senior Manager started the SPI journey by establishing a strong quality framework. This framework was the key factor for success in the improvement journey.

The improvement strategy and directions are identified by the highest management levels. The Software Process Improvement plan then materializes through rigorous planning sessions chaired by the Quality Process manager and the SPI team.

The SPI highly qualified and dedicated team has the ownership of all processes at the organization level. The team facilitates the activities of Software Process Improvement by incorporating the required input of the processes from relevant Subject Matter Experts (SMEs) inside the TDC. The wealth of knowledge the various SMEs provide is well discussed and challenged among them to ensure best practices are the ones that finally get captured in the organization's knowledge repository. The SPI team and SMEs consolidate the main body of the SEPG (Software Engineering Process Group) in addition to the Software Quality Assurance and Measurements team.

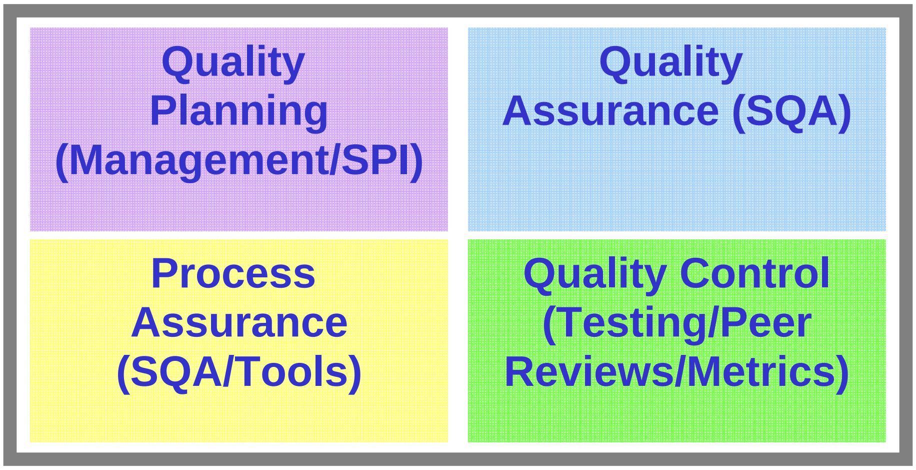
The second quadrant of the Quality Framework is the Software Quality and Processes Control. The adherence to, and actual implementation of the well-established TDC processes is the business objective of the Software Quality Assurance (SQA) team. The SQA team starts hand in hand with the projects to support the correct implementation of the TDC processes throughout the whole journey of its lifecycle: on all aspects: be that technical or Project Management related. Special project training is conducted for the project's team members to ensure all newcomers to the organization are in-line with the latest practices. The team also conducts Quality and Processes reviews both at the project and the organization levels. Keeping a close eye on areas for improvements that are immediately fed back to the SPI team to be taken into consideration either in the current or following SPI cycle.

The Measurements team's mode of engagement resembles the SQA in its early engagement with the project. Team members support the different projects in the various measurements activities such as collection and analysis of data as entailed by the organization. The team also supplies the organization with the consolidated performance measures and controls to monitor the different process activities. These measures are then used to feedback in the Organization's

Objectives for improvement, which are closely monitored and revisited by the Senior Management.

Once the processes become well established and mature, the need for seamless control and monitoring increases. Here comes the significant role of the Process Tools team, responsible for providing Process Automation; applications; another means of Process Improvement that cuts down on the processing time required for the flow of the processes with appropriate control. The team handles the requirements for automation through close interaction with the SPI team that feeds back all suggestions for innovations, and root causes of problems to be handled as an organization-level project.

The highly professional Testing Team engages from day zero with the projects to ensure that technical requirements are fulfilled. Testing activities are handled as a shadow project parallel to the overall development project providing continuous feedback on all the Software Engineering phases. Together with the golden rule of Peer Reviews for all activities this forms the required Technical Quality Control.



Quality Planning (Management/SPI)

Quality Assurance (SQA)

Process Assurance (SQA/Tools)

Quality Control (Testing/Peer Reviews/Metrics)

Figure 1: Quality Framework

This SPI journey had started three years ago. The SW-CMM was selected as the model of choice for Software Process Improvement. The decision was made based on the SW Industry recommendations and models, and was in line with the strategy and direction of the country. Mainly to increase, encourage and facilitate software development exports.

The SEPG virtual team was established representing the different TDC functions. The group started by developing the SW CMM L 2&3 processes. The SQA function was founded with the mission to monitor the processes implementation.

An SPI dedicated team was recruited along with the Process Automation team to hold the ownership of the process improvement maintaining the drive and thrust of this task force all over the Organization.

The institutionalization of the new processes was the of the utmost importance to the TDC Accordingly, proper focus was given to the training of TDC members on then relevant and fairly new processes. The Quality Team: SPI, SQA and SEPG offered consultancy and assistance to the projects. The implementation was a new experience to all parties involved. A whole SPI cycle - one whole year -

was dedicated to establish the strong foundation and infrastructure that was the root cause for the upcoming success.

The enhancement and improvement of the now widely deployed processes ensured the Organization and projects' needs were met without affecting the spirit of the CMM. Processes this automatically lead to a successful appraisal of the SW CMM model at maturity level 3.

Being SW-CMM L3 was a major milestone towards the ultimate goal of the highest maturity level in the industry: The goal was then enhanced to target CMMI L5 with more focus on process improvement, the SPI and SQA teams grew to accommodate the-now aggressively growing TDC. The TDC population tripled so had the Quality team. A whole new dedicated measurements team was founded to cater for the now extensive need to implement the TDC Measurement Program. This team of qualified measurements analysts shared in the quest for quality, establishing baselines, and providing consultancy and awareness to the projects to support them on the new and higher maturity measures all had to follow.

CMMI L5 was the natural result of all the previous activities that were delivered with the utmost dedicated and enthusiasm.

The journey that has just ended has just begun. Staying on top is the most challenging part. So, another cycle of SPI activities is planned. Higher levels of quality levels are targeted. More growth for the TDC and the Quality Team is in progress. The TDC is continuously improving. Our mission is ongoing to share in the prosperity soon to come over this country, God willing.

Biography:

Ahmed Nabil is the "lead software process improvement engineer" in IBM-TDC. He had leaded the journey to achieve CMMI maturity level 5 in IBM-TDC. He has BSc in computer and information systems, Cairo University. He is Certified Software Quality Engineer (CSQE) from American Society for Quality, Certified Quality Management Expert by TŰV Germany and Certified ISO 9001:2000 Internal Auditor.

Mona Arishi is the Software Quality Process Manager at Cairo Technology Development Center - IBM Egypt from Mar. 2002 until now. Mona worked as senior project manager of the National Language Support and Translation Center (1986-2002), program manger for Arabic Competency Center (ACC) Marketing Support & Press Relations (1995-1999), e-business Center Manager (1999– 2002).

Feedback contacts:

Feedback, comments and questions are appreciated by the author.

Email:
anabil@eg.ibm.com
mona@eg.ibm.com

Toward Egyptian Software Industry Series:
Egyptian Software Industry Improvement Features

By: Ramiz Kameel

OBJECTIVE

This article "Egyptian Software Industry Improvement Features" is the third article of a series of articles "Toward Egyptian Software Industry" that concerns with identifying the standards that should control the relation between different roles, especially the standards among vendors, consultants, and customers. This article will study the role of government toward the vendors. This article will analyze the presented standard methodologies and their feasibility for the Egyptian community. This article introduces the required procedure features that should be followed among ESPC roles [1, 2] to improve the Egyptian software industry from process prospective.

INTRODUCTION

Production process is the corner stone of the software industry in ESPC. The improvement of the software industry starts by improving the mechanism and controlling of the production process; and this specific process improvement is carried out by means of process mapping.

Prior to discussing process mapping, it is important to understand what a process is; the process can be defined as being a "continuous and regular action or succession of actions, taking place or carried on in a definite manner, and leading to the accomplishment of some result; a continuous operation or series of operations". Another definition provides a description of a process in its simplest form, as being "a combination of inputs, actions and outputs". Anjard [3] further defines it as being "a series of activities that takes an input, adds value to it and produces an output for a customer", Figure (1).

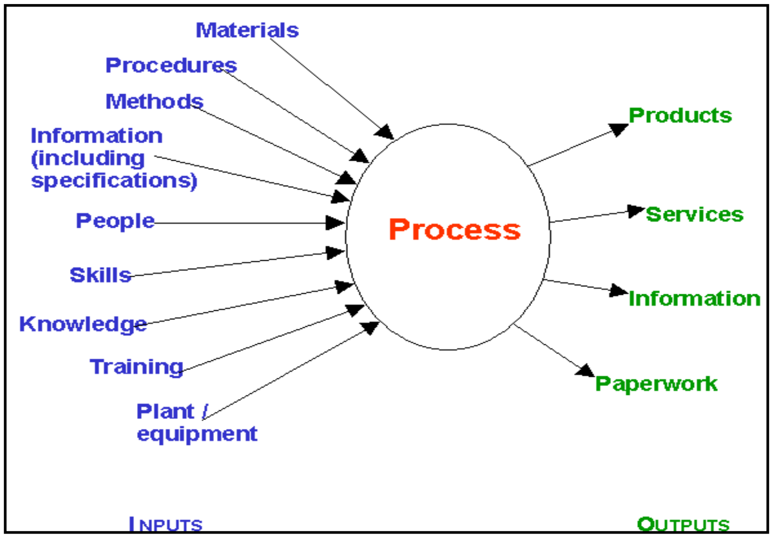


Figure 1: A diagram of a process

PROCESS MAPPING

The IDEF∅ (International DEFinition) is a method designed to model the decisions, actions and activities of an organization or system. The US Department of Defense developed it, mainly for the use of the US Air Force during the 1970s. Although developed over thirty years ago, the Computer Systems Laboratory of the National Institute of Standards and Technology (NIST) released IDEF∅ as a standard for Function Modeling in FIPS Publication 183, December 1993. Peppard et al (1995) describe as it having started life as a software development tool, although it is now an accepted process mapping tool within manufacturing and service organizations. Computer packages (for example, AI∅ WIN™) have been developed to aid software development by automatically translating relational diagrams into code.

An IDEF∅ diagram consists of boxes and arrows. This shows the function as a box and the interfaces to or from the function as arrows entering or leaving the box. Functions are expressed by boxes operating simultaneously with other boxes, with the interface arrows "constraining" when and how operations are triggered and controlled. The basic process mapping according the IDEF∅ model is shown in Figure (2).

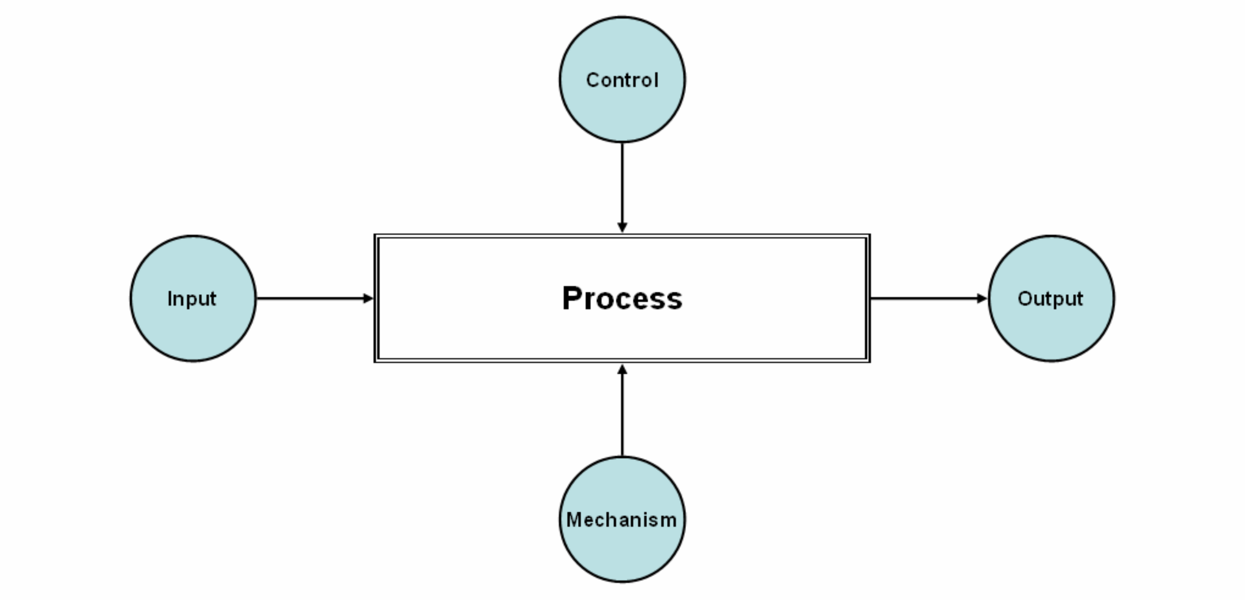


Figure 2: IDEF∅ Process Map

Mapping using this standard generally involves two levels. The first level, the high level map, identifies the major processes by which the company operates, Peppard et al. [4], for example, direction setting, winning the customer, delivery to the customer, support to the customer, support to the organization. The second level map breaks each of these processes into a sequence of steps, and then breaks those steps down again until the appropriate level is reached.

ESPC PROCESS MODEL

By the aid of this model, the ESPC process mapping is achieved. Figure (3) represents the modeling of the global industrial process of the software production in Egypt, based on the suggested model and roles [1, 2]. This model represents the main four methodologies that should be followed in the ESPC to improve the Egyptian software industry.

According to the process mapping technique, the vendor role can be invited as the input line to the production process of the software industry according to the pervious definition of the role [1]. Consequently, the customer will be the output receiver of the production process. On the other hand, based on the definition of the government role, the government can play the controlling role in the production process. The mechanism of the process progress can be guided by the consultancy role in the ESPC model.

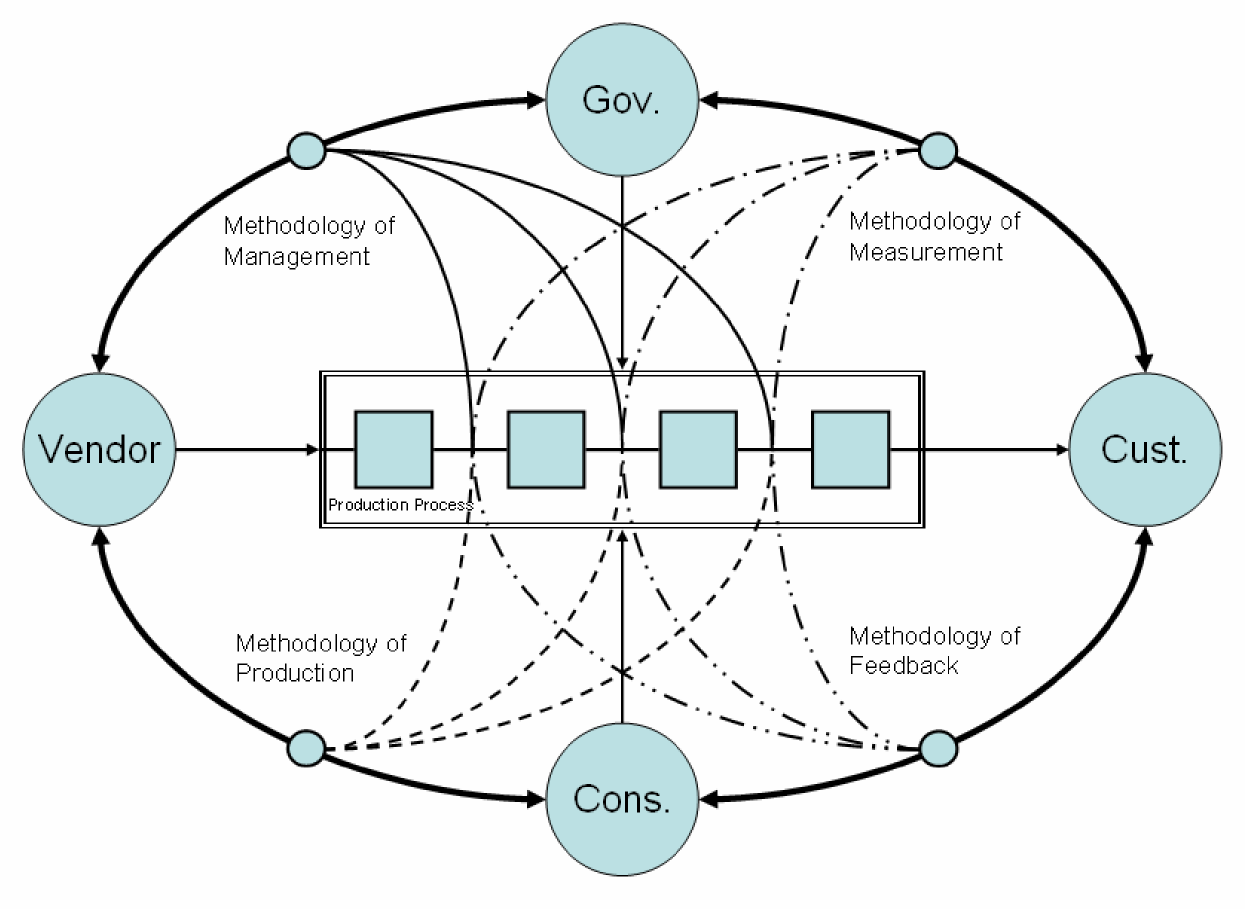

Gov.
Methodology of Management
Methodology of Measurement
Vendor
Production Process
Cust.
Methodology of Production
Methodology of Feedback
Cons.

Figure 3: ESPC Process Model

The direct interaction relationship among various roles results different methodologies of technical directing and controlling the process. Each methodology has several features gives the stakeholders the capability of the directing or controlling the process. Table (1) represents the features of methodologies and common features among them [5].

There is an interaction relationship between the government and consultants. This relation is necessary to obtain the corrective actions between management and production. The consultants provide the latest modifications and updates on the production mechanism of the vendor-consultant-customer path line, which enables the government to re-develop the technical standard of production.

The government provides the latest modifications and updates on the management control of the vendor-government-customer path line. This relation between government and consultants does not appear on the Figure 3 due to the plane representation of the relations.

	Vendor	Customer
Government	**Methodology of Management** Establishing optimum program for management Providing management organizational change Providing execution actions for correction	**Methodology of Measurement** Providing the system of evaluation for management correction Providing continuous customer educating and upgrading
Consultant	**Methodology of Production** Developing proper strategy for production based on technology Providing process modeling and engineering for improvement Supporting in selecting the optimum technology for production	**Methodology of Feedback** Establishing assessment procedure based on technology Supporting in defining the proper production technology

Table 1: Features of Different Methodologies among Roles

All features of different methodologies should be covered using different models or standards to provide the optimum execution for the software production process. These models should be selected to achieve the prospective performance of each role [2]. All models should be also having the common links to interact together as one system.

All of these features are proposed based on considering that all roles (except vendor or manufactures) are in local. In the case of foreign customers and consequently foreign government and consultants, the vendor may have to follow the international features. Actually, the aim finally is that the vendor must target these foreign markets. Maybe, the relations in this case are more adaptive than regulatory, but that should need focusing from modeling simulation phase to prepare for that. And ideally, the free trade global mechanism will have strong impact on all the methodologies [5].

Conclusion

In local market, the regulations of ESPC process model are required to follow certain methodologies that are responsible to maintain the efficiency of the model. Ideally, the foreign markets should be in focus during preparing that model, without ignoring the local market nature. The different methodologies that control the relation among different roles must be integrated in optimum way to ensure the harmony in the model and in the process performance.

FUTURE WORK

In prospective articles, the choosing of proper methodologies for ESPC Process Model will be investigated. This investigation will be derived on the base of different outcome methodologies that are resulted from the different role reactions. The relation among these roles that are ensuring the harmony will be investigated.

APPENDIX 1

A Process Map, a definition

"A process map is considered to be a visual aid for picturing work processes which show how inputs, outputs and tasks are linked" [3]. Soliman [6] also describes it as being the "most important and fundamental elements of business process re-engineering".

Process maps, as well as prompting new thinking, are also one of the most effective ways of gaining an understanding of existing processes by drawing them onto a map [4]. Peppard et al [4] have defined the two advantages and three disadvantages to process mapping.

The advantages:

- They are deemed to be usable, insofar as they give a clearer explanation of a process than words ('a picture paints a thousand words').
- The mere fact that individuals are working on maps means that a great understanding is gained of the tasks and problems that are faced within the organization.

The disadvantages:

- Process Maps can prove to be too distracting. The reason being that some companies have reported to have slowed down actions to ensure the integrity of the maps.
- Process maps can also take on a life of their own and lose relevance to those working on the process.
- It is also suggested that the maps do not always make good means of communications between layers of management.

Reasons for Process Mapping

One of the most efficient ways to understand an existing process is to draw them diagrammatically, to map them. Process maps are intended to represent a process in such a way that is easy to read and understand, Peppard et al [4]. There are a number of reasons why organizations choose to conduct process mapping, as following:

- Business Process Reengineering
- Process Management
- Software Process Modeling

REFERENCES

[1] Kameel, R., 2005, Toward Egyptian Software Industry: Egyptian Software Production Community, Egypt-Spin Newsletter, SECC, Issue 10, Apr. – Jun. 2005, Pages 10-13.

[2] Kameel, R., 2005, Toward Egyptian Software Industry: Egyptian Software Industry Improvement, Egypt-Spin Newsletter, SECC, Issue 11, Jul. – Sep. 2005, Pages 4-9.

[3] Anjard, R. P., 1998, Process Mapping: a valuable tool for construction management and other professionals. MCB University Press. Vol. 16, No 3/4, pp 79-81.

[4] Peppard, J., and Rowland, P., 1995, The essence of business process re-engineering. Hemel Hempstead: Prentice Hall Europe

[5] Abd El-Hady, A., 2005, Private Communication.

[6] Soliman, F., 1998, Optimum level of process mapping and least cost business process re-engineering. International Journal of Operations & Production Management, vol. 18, No 9/10, pp 810-816 MCB University Press.

ACKNOWLEDGEMENT

This work is supported by Research Activities in Quality Dept. of Prima Soft. The author is grateful to, Eng. Ahmed A. Hady, for his valuable recommendations.

BIOGRAPHY

Ramiz Kameel is SPI Consultant of Egyptian Software and Systems; Prima Soft. Author holds a Ph.D and IT.D. in Engineering. Author is SPI Consultant of Information Technology Institute - ITI.

FEEDBACK CONTACTS

Feedback, comments and questions are appreciated by the author.

Email:
rekameel@primasoft.com.eg

Automatic Validation Low Cost Infratructure

By: Omar Kamal

Introduction

The initial intention for this article was to continue introducing basic knowledge about software testing engineering techniques as previous published SECC articles [3][4] tried to do. However, I decided to interrupt the series with an article that briefs an approach for developing an infrastructure for what we can call "Automatic Validation".

"Validation" as a term can have multiple meaning based on the context of the speech. Therefore, we will first establish a common understanding of the term. The Software Engineering Institute (SEI) states in its technical report "Capability Maturity Model® Integration (CMMI^{SM}), Version 1.1" the following definition for Validation: "*Validation confirms that the product, as provided, will fulfill its intended use. In other words, validation ensures that "you built the right thing*."

Accordingly, the validation activities should be triggered as soon as the requirements analysis is finished and a requirement document is baselined. A number of problems appear in most of the phases in the validation cycle. The following list highlights some but not all of those problems:

- Validation execution efforts seem to appear at the top of most time consuming activity list.

- Requirement traceability can be used as a measure for validation incompleteness and hence failure but it is a very week criteria when considered as a validation success criteria. In other words, mapping each labeled requirement to a corresponding test case ignores the complexity of the problem and over simplifies the needs for requirement coverage. Inspecting the code coverage reports after executing all validation test suites minimize the perceived risk that the validation test cases are incomplete, however the approach still have a number of limitations. The assumption that "*Covering all (statements, branches, decisions, or predicates, etc …) inside the code with enough functionally driven system test cases, will converge to a corresponding strong coverage criterion at the system level*" isn't true. The source code represents the developer understanding for the "designer view" to the stated requirements, which is a long chain of different representation that is subject to human misconception. Doing so, is more toward verification than to validation.

- Peer review is an important process that uncovers as much defects as possible in the document under review. Validation test cases can reach thousands for a medium business application. Reviewing each test case can be time-consuming for any peer if he/she carried out the peer review the standard way. On the other side, it is unacceptable for the quality assurance to accept the risk of baselining such test cases with inappropriate review. Hence, the problem is to find a smart way that allow peers to review a huge number of test cases without consuming too much time, and also without trading off quality.

- Organizations that choose to adopt manual test execution are facing hard time with test regression. It is extremely inefficient to manually carry out regression testing and in certain cases; it can be a project killer if manual regression is too frequent.

The previous mentioned problems motivate software engineers to address the validation process differently.

This article suggests adopting an automated approach for designing, executing, and tracking the validation activities. Executing validation test cases has become so popular using capture/replay tools or through commercial testing execution frameworks. Similarly, if the test execution is carried out automatically, a chance for populating test results in detailed or summarized reports is possible.

The problem appears to be automating the test design phase, which although possible is not that popular. Few commercial test case design tools exist but the following factors limit its popularity:

- Most of the tools available on the market have a high cost relative to other software support tools.

- The majority of those vendors do not have any kind of local support in the Middle East Region, which is reflected in the total cost of ownership including the cost for training, customization and technical support.

- Some tool does not integrate easily with other software support tools that are part of the organization infrastructure.

This article proposes an automatic validation infrastructure constructed by either an open source framework tools/libraries or easy in-house developed scripts. The proposed infrastructure cannot be explained within introducing an automation model that shows dependencies and interfaces between all of the composed tools.

Validation Automation Model

The model can be divided into five major set of activities that follow in sequence. However, change in any of the model components has a relatively very low impact on cost and schedule when compared to the traditional validation process. Figure 1 (*in the next page*) illustrates the overall model and the following sections will explain its components in brief.

Inputs (Phase 1)

As soon as an initial requirements specification document is baselined, the validation team starts its work by understanding the requirements from the validation point of view. If requirement analysis models exist, it will certainly make the process of understanding easier and can even facilitate constructing corresponding validation models. The validation team can face a situation where a certain piece of requirement is very clear but unverifiable or will cost the organization too much in order to validate it. If the situation exists, the project manager may choose to do any of the following:

- Agree with the customer to wavier the dynamic validation for that specific requirement and depend only on static validation (inspection, walkthrough, etc …).

- Modify unverifiable requirements to become verifiable ones, and agree with the customer that such modification may cost more than the agreed upon cost.

- If an unverifiable requirement has a lower business priority, the customer may agree with the development organization to ignore implementing it.

2. Validation Models Development (Phase 2)

Developing validation models is the most creative set of activities in that proposed model. It represents a good opportunity for managers that used to look for justifications to convince talented software engineers to join their validation teams. Validation models abstract the defined problem by focusing on certain system characteristics important to the validation context. A model can focus on any of the following:

- The required flow of the control and/or data

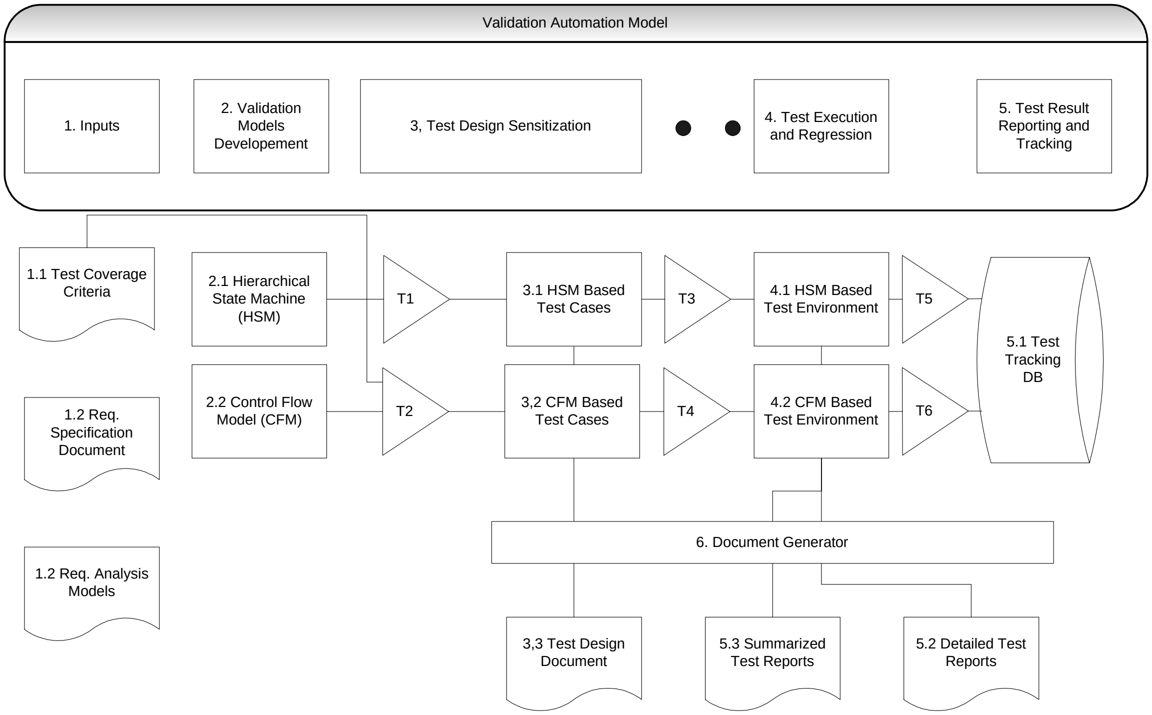

Validation Automation Model
1. Inputs
2. Validation Models Developement
3, Test Design Sensitization
4. Test Execution and Regression
5. Test Result Reporting and Tracking
1.1 Test Coverage Criteria
1.2 Req. Specification Document
1.2 Req. Analysis Models
2.1 Hierarchical State Machine (HSM)
2.2 Control Flow Model (CFM)
T1
T2
3.1 HSM Based Test Cases
3,2 CFM Based Test Cases
T3
T4
4.1 HSM Based Test Environment
4.2 CFM Based Test Environment
T5
T6
5.1 Test Tracking DB
6. Document Generator
3,3 Test Design Document
5.3 Summarized Test Reports
5.2 Detailed Test Reports

Figure 1: Validation Automation Model

- The representation of problem entities or objects
- The system response for certain inputs at certain point of time
- The performance characteristics of the system

In addition, the domain of requirements or the project context may necessitate developing other models that are not listed here like (transaction flow models, security models, or Petri-nets models etc ...). Validation models are then used for automating the process of generating system test cases and their associated test design documents. In my previous published articles [3] [4], I introduced (Control Flow Models) based on the work of the testing guru (Boriez Bizer) [6] [7]. The articles are available at the SECC website[1] for readers interested to know more about the model. The scope of this article will be limited to the Finite State Machine/Hierarchical State Machine models that can be used in a broad range of business and engineering applications.

Finite State Machine/Hierarchical State Machine Models (Block 2.1)

Event Driven Systems

For any event driven system, the outer world (boundary of the system) triggers the system with events (actions). The system interprets those events and reacts according to its type:

- **System without memory:** System of this type reacts only based on the event type. There isn't a need to store any kind of data for future use. A system that computes a sin(x) function doesn't need to store any data for the next event excitation. The reaction of the system is only dependant on the instantaneous event. Such systems are not likely found in the business life.

- **System with memory:** System of this type reacts based on both the event type and the system's current state. The system current state can be presented as a single simple variable or can be a set of variables known as "state configuration" variables. Number of states for the system isn't linearly correlated to the number of variables. Adding more variables may complex the system in a multiplicative way. The system then interprets the incoming event and takes any of the following action based on its current state:

  - Stay on its current state
  - Stay on its current state and produce a specific output
  - Change its current state to another state
  - Change its current state to another state and produce a specific output.

The whole set of events, states, outputs, and transitions can be modeled in what is known as Finite State Machine (FSM) model. Figure 2 illustrates a sample one.

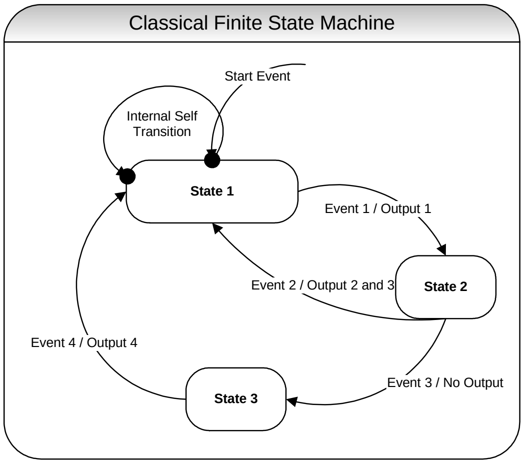


Figure 2: Sample classical Finite State Machine

Abstract State Machines (ASMs)

Using Finite State Machines (FSMs) in system modeling is a very powerful concept because it allows modeling to be done at any level of abstraction. A modeler may be interested in some system characteristic and isn't interested in the rest of them. That is

[1] You can find those articles at the SPIN website: http://www.secc.org.eg/SPIN%20Newsletter.asp

why it is also called Abstract State Machine (ASM).

Hierarchical State Machines (HSMs)

Hierarchical State Machines (HSMs) even beat the classical (FSMs) by offering modularity, behavior inheritance and a layer-based representation for state machines. Benefits gained from modeling requirements in (FSM/HSM) influenced standard bodies to include them in the Unified Modeling Language (UML). State Charts[2] are part of the UML diagrams that illustrate (HSMs) graphically. A sample State Chart example is shown in figure 3.

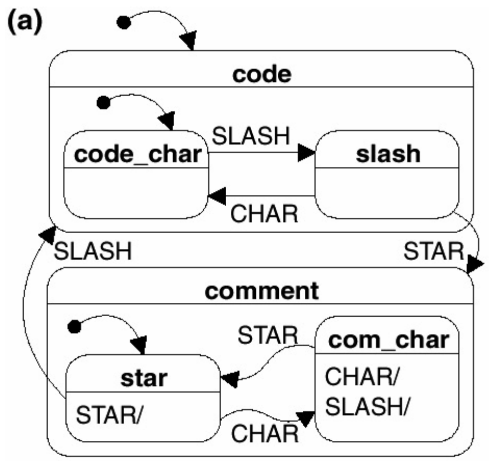


Figure 3: A Hierarchical State Machine (HSM) represented as a state chart.

Abstract State Machine Language (ASML)

Microsoft Research Lab developed a very powerful, useful, and freely distributed beta version tool for modeling FSMs called ASML which stands for Abstract State Machine Language [1]. ASML is easy to learn and with a minimum awareness of the language, a software test modeler can model any problem as (FSM).

The tool is installed over Microsoft Word, allowing the word processor to interpret ASML pseudo code (*Figure 4 in next page*) and detect any ASML syntax errors. Additionally, the test modeler can include embedded ASML code inside any of the requirements specification document text sections and can still get it compiled.

ASML Test Tool (T1)

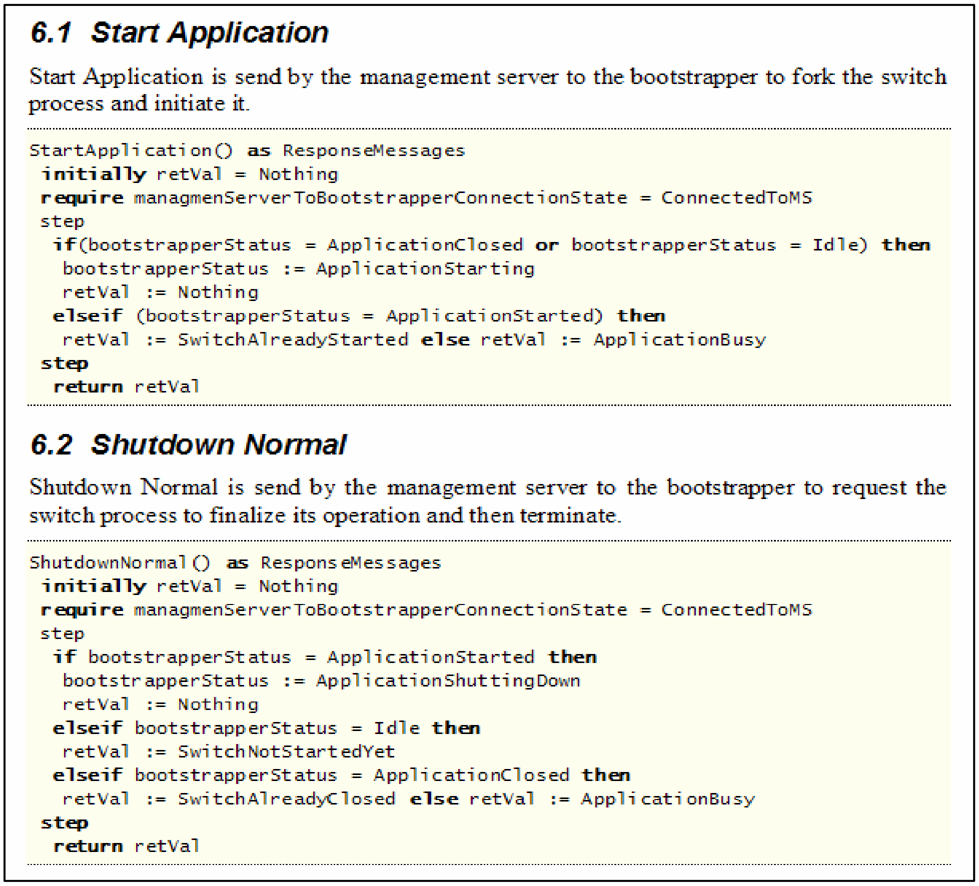
The test modeler generates the (FSM) model using the ASML Test Tool that comes with the ASML package (*Figure 5 in next page*). FSMs are represented in both tabular and graphical representation. Multiple FSMs can be generated for the same specification by identifying which system properties to include and which to ignore[3].

HSM Based Test Cases (Phase 3, Block 3.1)

The best feature that is offered by the ASML test tool which any test designer can find it extremely helpful is the *ASML test generation feature*. Simply, configure the required test coverage criteria for the active FSM and click on "*generate test suite from FSM*" and a whole set of test cases are automatically generated (*figure 6 in page 19*). Generating test cases that cover small FSMs can be hard to develop manually, can you imagine how much effort needed to cover a FSM with hundreds of states. Furthermore, the manual approach can lead to many redundant test cases. What is more, changing requirements can be handled easily, by altering the FSM and regenerating test cases again.

[2] Developed by David Harel

[3] For example, we may be interested in testing the system start up states only and ignoring other operational states.



6.1 Start Application

Start Application is send by the management server to the bootstrapper to fork the switch process and initiate it.

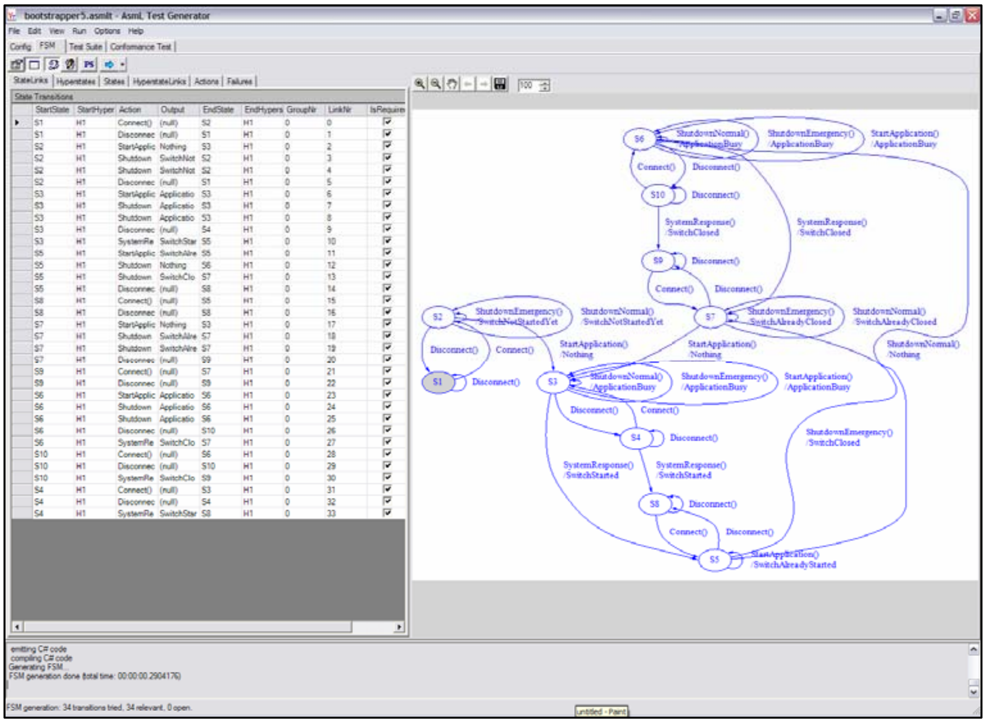
```
StartApplication() as ResponseMessages
 initially retVal = Nothing
 require managmenServerToBootstrapperConnectionState = ConnectedToMS
 step
  if(bootstrapperStatus = ApplicationClosed or bootstrapperStatus = Idle) then
   bootstrapperStatus := ApplicationStarting
   retVal := Nothing
  elseif (bootstrapperStatus = ApplicationStarted) then
   retVal := SwitchAlreadyStarted else retVal := ApplicationBusy
 step
  return retVal
```

6.2 Shutdown Normal

Shutdown Normal is send by the management server to the bootstrapper to request the switch process to finalize its operation and then terminate.

```
ShutdownNormal() as ResponseMessages
 initially retVal = Nothing
 require managmenServerToBootstrapperConnectionState = ConnectedToMS
 step
  if bootstrapperStatus = ApplicationStarted then
   bootstrapperStatus := ApplicationShuttingDown
   retVal := Nothing
  elseif bootstrapperStatus = Idle then
   retVal := SwitchNotStartedYet
  elseif bootstrapperStatus = ApplicationClosed then
   retVal := SwitchAlreadyClosed else retVal := ApplicationBusy
 step
  return retVal
```

Figure 4: Embedded ASML code in text sections



bootstrapper5.asmlt - AsmL Test Generator
File Edit View Run Options Help
Config FSM Test Suite Conformance Test
StateLinks Hyperstates States HyperstateLinks Actions Failures
State Transitions
100

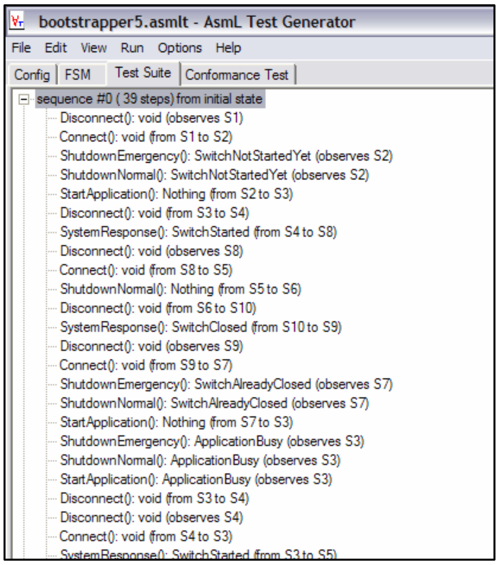
StartState	StartHyper	Action	Output	EndState	EndHypers	GroupNr	LinkNr	IsRequire
S1	H1	Connect()	(null)	S2	H1	0	0	☑
S1	H1	Disconnec	(null)	S1	H1	0	1	☑
S2	H1	StartApplic	Nothing	S3	H1	0	2	☑
S2	H1	Shutdown	SwitchNot	S2	H1	0	3	☑
S2	H1	Shutdown	SwitchNot	S2	H1	0	4	☑
S2	H1	Disconnec	(null)	S1	H1	0	5	☑
S3	H1	StartApplic	Applicatio	S3	H1	0	6	☑
S3	H1	Shutdown	Applicatio	S3	H1	0	7	☑
S3	H1	Shutdown	Applicatio	S3	H1	0	8	☑
S3	H1	Disconnec	(null)	S4	H1	0	9	☑
S3	H1	SystemRe	SwitchStar	S5	H1	0	10	☑
S5	H1	StartApplic	SwitchAlre	S5	H1	0	11	☑
S5	H1	Shutdown	Nothing	S6	H1	0	12	☑
S5	H1	Shutdown	SwitchClo	S7	H1	0	13	☑
S5	H1	Disconnec	(null)	S8	H1	0	14	☑
S8	H1	Connect()	(null)	S5	H1	0	15	☑
S8	H1	Disconnec	(null)	S8	H1	0	16	☑
S7	H1	StartApplic	Nothing	S3	H1	0	17	☑
S7	H1	Shutdown	SwitchAlre	S7	H1	0	18	☑
S7	H1	Shutdown	SwitchAlre	S7	H1	0	19	☑
S7	H1	Disconnec	(null)	S9	H1	0	20	☑
S9	H1	Connect()	(null)	S7	H1	0	21	☑
S9	H1	Disconnec	(null)	S9	H1	0	22	☑
S6	H1	StartApplic	Applicatio	S6	H1	0	23	☑
S6	H1	Shutdown	Applicatio	S6	H1	0	24	☑
S6	H1	Shutdown	Applicatio	S6	H1	0	25	☑
S6	H1	Disconnec	(null)	S10	H1	0	26	☑
S6	H1	SystemRe	SwitchClo	S7	H1	0	27	☑
S10	H1	Connect()	(null)	S6	H1	0	28	☑
S10	H1	Disconnec	(null)	S10	H1	0	29	☑
S10	H1	SystemRe	SwitchClo	S9	H1	0	30	☑
S4	H1	Connect()	(null)	S3	H1	0	31	☑
S4	H1	Disconnec	(null)	S4	H1	0	32	☑
S4	H1	SystemRe	SwitchStar	S8	H1	0	33	☑

S6
ShutdownNormal() /ApplicationBusy
ShutdownEmergency() /ApplicationBusy
StartApplication() /ApplicationBusy
Connect()
Disconnect()
S10
Disconnect()
SystemResponse() /SwitchClosed
SystemResponse() /SwitchClosed
S9
Disconnect()
Connect()
Disconnect()
S2
ShutdownEmergency() /SwitchNotStartedYet
ShutdownNormal() /SwitchNotStartedYet
S7
ShutdownEmergency() /SwitchAlreadyClosed
ShutdownNormal() /SwitchAlreadyClosed
Disconnect()
Connect()
StartApplication() /Nothing
StartApplication() /Nothing
ShutdownNormal() /Nothing
S1
Disconnect()
S3
ShutdownNormal() /ApplicationBusy
ShutdownEmergency() /ApplicationBusy
StartApplication() /ApplicationBusy
Disconnect()
Connect()
S4
Disconnect()
ShutdownEmergency() /SwitchClosed
SystemResponse() /SwitchStarted
SystemResponse() /SwitchStarted
S8
Disconnect()
Connect()
Disconnect()
S5
StartApplication() /SwitchAlreadyStarted
untitled - Paint

```
emitting C# code
compiling C# code
Generating FSM...
FSM generation done (total time: 00:00:00.2904176)
```

```
FSM generation: 34 transitions tried, 34 relevant, 0 open.
```

Figure 5: ASML Test Tool generates FSM from ASML document



```
sequence #0 ( 39 steps) from initial state
    Disconnect(): void (observes S1)
    Connect(): void (from S1 to S2)
    ShutdownEmergency(): SwitchNotStartedYet (observes S2)
    ShutdownNormal(): SwitchNotStartedYet (observes S2)
    StartApplication(): Nothing (from S2 to S3)
    Disconnect(): void (from S3 to S4)
    SystemResponse(): SwitchStarted (from S4 to S8)
    Disconnect(): void (observes S8)
    Connect(): void (from S8 to S5)
    ShutdownNormal(): Nothing (from S5 to S6)
    Disconnect(): void (from S6 to S10)
    SystemResponse(): SwitchClosed (from S10 to S9)
    Disconnect(): void (observes S9)
    Connect(): void (from S9 to S7)
    ShutdownEmergency(): SwitchAlreadyClosed (observes S7)
    ShutdownNormal(): SwitchAlreadyClosed (observes S7)
    StartApplication(): Nothing (from S7 to S3)
    ShutdownEmergency(): ApplicationBusy (observes S3)
    ShutdownNormal(): ApplicationBusy (observes S3)
    StartApplication(): ApplicationBusy (observes S3)
    Disconnect(): void (from S3 to S4)
    Disconnect(): void (observes S4)
    Connect(): void (from S4 to S3)
    SystemResponse(): SwitchStarted (from S3 to S5)
```

Figure 6: ASML generated test cases

Finally, ASML test tool comes with a limited feature[4] to execute test cases through binding generated test cases with production code. Conformance tests can be carried out and test results are presented for the test executer or stored as XML files. If the production source code isn't written in C sharp, the validation team is left with two options. The first is to execute test cases manually the first time and each time a regression is needed. The other option is to find a way to automate test execution and result reporting. The following section will introduce a low cost solution that realizes the second option.

Automatic Test Execution (Phase 4)

Although, a great deal of the cost is saved by automatically generating test cases, the cost associated with manually executing test cases is still painful. Projects characterized by frequent regression testing can't make it with a manually test execution approach. We can categorize automatic test execution approach into three categories.

First Approach

This approach is used heavily in GUI based application and web applications. The idea is to execute the test cases once while capturing the key strokes, mouse clicks, dialog box interactions, etc …. The set of tools that offer such functionality is called "Capture/Replay Tools"[5]. Some tools offer an intermediate high level scripting language that can control mouse clicks, simulate keyboard strokes, and even compare screen output as images stored in the test case. The main disadvantage of such tools is that it requires a lot of re-programming test case execution scripts when the requirements change.

Second Approach (T3, Block 4.1)

This approach is suitable for automating execution for either volatile or relatively stable requirements, however it requires an initial investment. Executing any test case involves exciting the system under testing by the input test vector and compares the actual output test vector with the expected one. ASML generated test cases are usually written in a format that is not understood by the system under testing. The idea is to build a conversion tool that is capable for converting the ASML format to the system format and vice versa. The development of such conversion tool can be easily done using any forth generation languages like (Perl, Python, Tcl/Tk, or Ruby). Conversion tools once developed and tested become part of the organization assets that significantly boost the validation

[4] C Sharp Only

[5] Rational Robot is an example of such tool

execution. Contrary to the first approach that depend on re-programming the test execution each time the requirements change, this approach doesn't depend on recording the test case execution at all. The following actions are automatically triggered in case the validation model changes:

- The FSM model automatically changes.
- Test cases are generated automatically.
- The conversion tool converts test cases and builds the test execution environment.

Third Approach

This approach is used for robustness testing and isn't based on the ASML generated test cases; however it depends on the HSM model developed in the earlier phase. The idea is to generate random inputs at a random order at an enormous amount with the intention to break the software. The conversion tool is still used to translate random input vectors into the system format. Expected outputs are generated using a system oracle, which is an executable HSM prototype. Expected outputs are converted and compared with the actual output and results are stored.

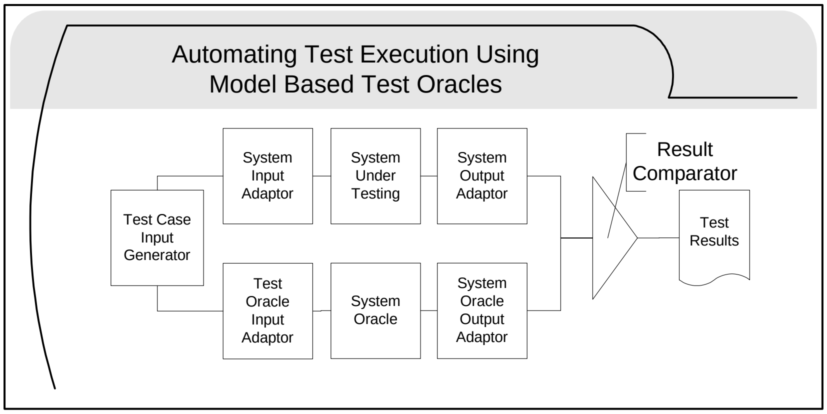


Figure 2: Automating test execution using model based test oracles

HSM Based Oracles

Developing an oracle shouldn't take so much time relative to the effort spent for building the original system; otherwise we are wasting our time building the system twice. The system oracle is just a functional prototype that relaxes non-functional requirements and other quality constraints. Although, ASML itself can by used as a system oracle I suggest another open source alternative. Miro Samek[6] developed a freely distributed C/C++ framework that can be used to develop HSM based systems [2] [5]. The framework is very efficient, modular, and easy to understand. For more details about the framework, visit http://www.quantum-leaps.com/

Miro introduced to the community a new paradigm that treats state charts as a way of design rather than use of particular tool. His Quantum Framework allows rapid hand-code working systems in C or C++ directly from UML state charts.

[6] Miro Samek is the lead software architect at IntegriNautics Corporation.

Document Generator (Block 6)

During test execution activities, reports are produced in textual format that mayn't be compliant with the organization's standard test tracking report template. Detailed test result may be important to developers or the quality assurance team but for project or senior managers a summarized version is more important. So, to automate the whole cycle we need to develop tools to convert those textual files to their corresponding Microsoft Word/Excel detailed and summarized standard reports. This can be achieved using any or all of the following:

- Visual Basic for Application (VBA).
- Capture/Replay Tools.
- Perl, Python or Ruby scripts.
- Recorded Macros in Microsoft Word or Excel.

Conclusion

Many of the problems attributed to validation can be solved by using validation models to automate the validation activities. Requirements frequent changes won't cause us trouble any more. Changing requirements will cost us changing the validation model only and everything else will be automatically generated. No more peer review on thousands of test cases, the review will be limited to validation models that generate those test cases only. Finally, the proposed automation infrastructure reduces significantly the time needed for test case design, development, execution, and tracking.

References

[1] http://research.microsoft.com/fse/asml/

[2] http://www.quantum-leaps.com/

[3] Omar Kamal, "Software Testing Techniques Series: Control Flow Testing Tutorial. (Part 1)," Egypt–SPIN Newsletter Issue 8, Oct. – Dec. 2004.

[4] Omar Kamal, "Software Testing Techniques Series: Control Flow Testing Tutorial. (Part 2)," Egypt–SPIN Newsletter Issue 9, Jan. – Mar. 2005.

[5] Miro Samek, "Practical Statecharts in C/C++: Quantum Programming for Embedded Systems," CMP Books, 2002.

[6] Beizer, B., "Black Box Testing," Katherine Schowalter, 1995

[7] Beizer, B., "Software Testing Techniques," second Edition, Van NostrandReinhold, 1990

Biography

Omar Kamal, 9 years of experience in wireless telecommunications, software development, training and software quality management. Working as a senior software engineer in QuickTel Research and Development, used to work with Lucent Technologies, Hewlett Packard, and Etisalat. He holds a bachelor's degree in telecommunications engineering from Cairo University, and master's degree in business administration from City University. In addition, he is a "Certified Quality Manager" by the American Society for Quality.

Feedback

Feedback, comments and questions are appreciated by the author.

Email:
omkamal@yahoo.com

CMMI Process Areas Explanation Series: Supplier Agreement Managemnt (SAM)

By: Mohamed Abo-Zied

Introduction

SAM is the Supplier agreement Management process area which is considered one of the important process areas in CMMI level 2, and in the same time is categorized under the project management processes.

SAM is the process area that helps you manage the acquisition of products from suppliers for which there exists a formal agreement. In more practical way we can say that SAM is the contract/subcontract management involving selecting a qualified supplier, mutually agreeing on contract requirements, maintaining technical and administrative communication, and tracking the supplier's actual performance against contract requirements.

SAM main areas:

SAM Applies to the acquisition of products and product components that are delivered to the project's customer. It could be also applied to the acquisition of significant products and product components not delivered to the project's customer but delivered to any organization entity like development tool, test tool or test enviorement.

The SAM process addresses the acquisition of such products in a project.

Acquisition types as listed below:

- Parts of the product to be developed as per the organization specifications. This is termed a subcontracting

- Ready-to-use products that need to be purchased from the market for the project – these are called COTS (commercial of-the-shelf products)

- Ready-to-use products that need to be acquired into the project from other parts of the organization (e.g., Systems Administration, Infrastructure)

- Products to be supplied by customers

The SAM process addresses the formal agreement for acquiring products or product components in the organization. The formal agreement may be of many types:

- Contract
- License
- Memorandum
- Service request
- Agreement

SAM Benefits

- Establish organizational acquisition strategy.
- Plan the acquisition activities within your organization.
- Provide support and minimize risks to the projects.
- Establish formal agreement between organization and suppliers.
- Maintain categorized Supplier list.
- Effectively Select Qualified Supplier.

- Track the actual supplier's performance against agreement requirements.
- Verify services or products delivered by supplier.

SAM relation with other process areas

- Overall planning, monitoring and control of projects are done through the Project Planning process and the Project Monitoring and Control process

- Activities related to maintaining the integrity of products and baselines, with respect to all products, data and plan related to a project, are performed using the Configuration Management process

- Quantitative Project Management in defining and negotiating the quality and process-performance objectives for suppliers

- Technical and non-technical requirements related to a project are managed using the Requirements Management process, Requirement Development and Technical Solution Process.

- Decision Analysis & Resolution for performing formal evaluation and selecting suppliers.

- Verification in performing technical reviews with supplier and in ensuring the acquired product meets its requirements.

- Senior management reviews the status of processes and projects using the Management Review process, which includes review of projects and review of operations

SAM Practices

- Plan for Product Acquisition
- Evaluate and Select Supplier
- Sign and execute supplier agreement
- Monitor Subcontracts
- Perform Acceptance of Subcontractor Deliverables
- Acquire Ready-to-use Products
- Track Acquisition and Transition Acquired Products

Plan for Project Acquisition

The Acquisition department / project Manager prepare the Acquisition Plan that documents the overall approach for acquisition. The plan addresses significant products that are not built by the project team, and

- may delivered to the customer, or
- may be used to create products to be delivered to the customer

Acquisition can be of many types. The acquisition types addressed through this process are as mentioned under the SAM main areas section.

- Parts of the product to be developed as per the organization specifications. This is termed a subcontracting

- Ready-to-use products that need to be purchased from the market for the project – these are called COTS (commercial of-the-shelf products)

- Ready-to-use products that need to be acquired into the project from other parts of the organization (e.g., Systems Administration, Infrastructure)

- Products to be supplied by customers

Evaluate and Select Supplier

Based on the subcontracting requirements specified in the Acquisition Plan, the Acquisition department /Project manager identifies the potential suppliers for each subcontract. Suppliers who are considered for this identification include:

- Suppliers used in the past/ currently executing subcontracts for the organization.

- Suppliers identified through professional magazines/ industry contacts

- Suppliers who have sent their details and profile and asked to be considered for such work.

- Suppliers who respond to any advertisement put out

- Recommendations of various members of the organization.

The Acquisition department /Project team starts to establish the evaluation criteria to be used for selecting the qualified supplier. The evaluation criteria will focus on both technical and commercial aspects.

The focus of the *technical evaluation* is the ability of the supplier to technically deliver the subcontracted work and includes evaluation of technical risks of subcontracting to a particular supplier. It includes an evaluation of the supplier's ability to perform.

The focus of the *commercial evaluation* is the commercial aspects of the supplier, such as price, payment terms, penalty clauses, etc., and the commercial risks associated with the supplier. The supplier’s financial stability is also evaluated.

Organization may depend on request for proposal way to get proposals from different suppliers, and then the evaluation of the corresponding suppliers based on the predefined criteria start by doing comparison of the technical capabilities of the potential suppliers. Aspects for comparison could include:

- Staff strength
- Availability of staff for the project
- Domain/ application familiarity
- Technology familiarity
- Facilities (computers/ communications/ infrastructure)
- Geographical proximity to the organization or client
- Past experience with in the acquired product
- Past experience with the organization
- Reputation in the market
- Process maturity
- Certifications and awards
- Reusable software available
- Client comfort with the supplier
- Ranked list of (up to three) potential subcontractors who qualify technically

Recommendations for the commercial evaluation

Sign and execute Supplier Agreement:

After selecting the qualified supplier the organization and an appropriate senior representative of supplier formally accept this supplier agreement.

Where the supplier’s proposal is complete in all respects, the supplier’s proposal is reviewed by the organization and if found acceptable, a formal acceptance of the proposal by the organization along with the supplier's proposal is treated as a subcontract agreement.

The Acquisition department / project Manager prepares a Supplier Monitoring Plan that addresses the aspects relevant for monitoring a specific subcontract. The plan includes or refers to the acceptance criteria and the acceptance test plan that will be used for acceptance of the supplier product. It also describes how the PM will monitor supplier execution and the processes being used at take corrective action. This includes process evaluation, project reviews, management reviews and technical reviews. This is consistent with the subcontract agreement.

The supplier's Project Manager also should prepare a project management plan to describe how the supplier will execute the subcontract. This plan should be reviewed by the organization the acquisition department / project Manager and approved.

Monitor Subcontracts

Acquisition department / project Manager will monitor the progress of the supplier plan execution.
The supplier's has to provide a periodic status reports to the organization. The periodicity and the contents of this report are as decided in the supplier agreement and the supplier's project management plan.

Acquisition department / project manager conducts a periodic status/ coordination reviews with the supplier's and the minutes of such meetings are recorded and distributed to all relevant stakeholders.

Acquisition department / project manager holds phase-end/ milestone reviews with the supplier team. These address critical dependencies, project risks (involving the subcontractor), schedules and budgets. Feedback from these is made available to the subcontractor to improve performance.
QA has to objectively evaluate the supplier's project management, QA activities and configuration management activities.
Based on all the above activities, Acquisition department / project Manager prepares periodic reports on the status and recommendations related to each supplier.

Perform Acceptance of Supplier Deliverables

Acquisition department / project Manager prepares an acceptance test plan and data for each supplier based on the previously agreed acceptance criteria, as per the subcontract agreement.

Acquisition department / project Manager performs the planned acceptance of the supplier's deliverables as per the acceptance plans and the supplier is informed about any issues (if any) that need to be rectified and discusses the actions to be taken.

Acquire Ready-to-use Products

The organization may decide to go to ready to use products in this case it will follow the following activities to get the acquired product:

Acquisition department / project Manager in consultation with the project team identifies candidate COTS vendors for each acquisition of COTS type. Vendors who are considered may include:

- COTS vendors used in the past/ currently in use

- COTS vendors identified through professional magazines/ industry contacts
- COTS vendors who have sent in their details and profile and asked to be considered for such work. This information is available with the F&AGH

- COTS vendors who respond to any advertisement put out (this may be necessary if suitable subcontractors cannot be identified in other ways)

Then the Evaluation and select supplier procedures will be followed to evaluate the COTS products and its vendor to select the acquire COTS.

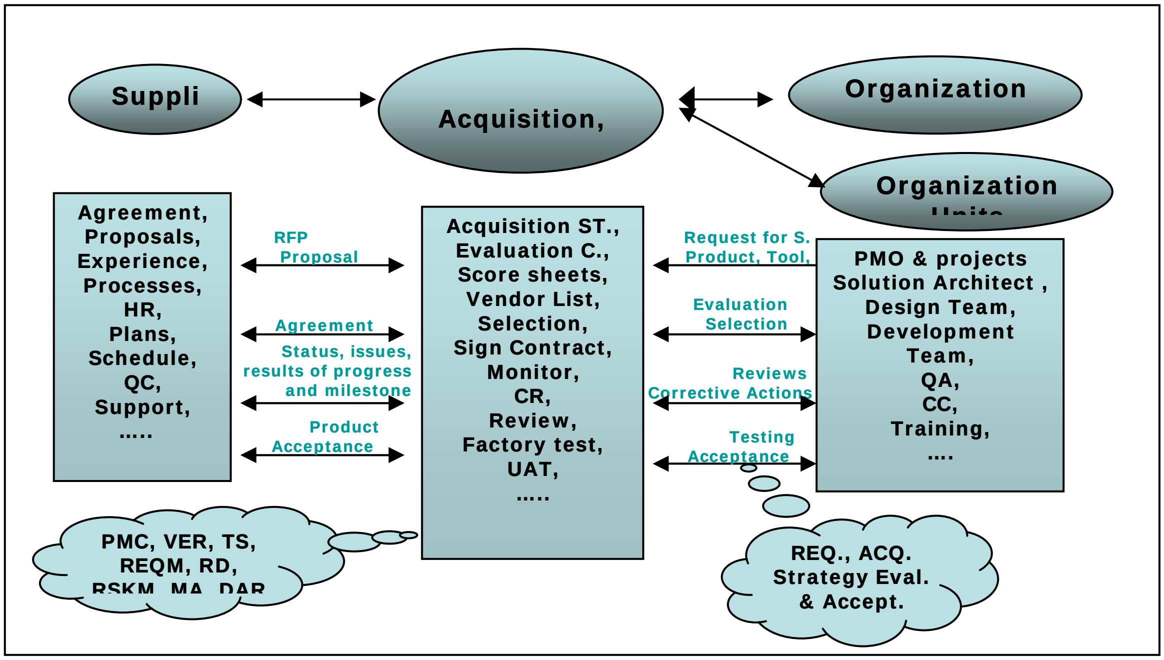
Track Acquisition and Transition Acquired Products

Acquisition department / project Manager stores the accepted acquired products as defined in the agreement and start to implement the activities required for transitioning the accepted products.
They will ensure that the use of the acquired products is as per the terms of the agreements and within any license agreements signed with the suppliers.

The supplier will provide the needed support for acquired products as per the terms of the agreements/ purchase orders.

They will report about the status of transitioning the acquired product to clarify any problems related to issue of subcontracts, purchase orders on COTS, internal requests for products, customer-supplied products, resource availability for acquisition activities, execution of responsibilities for acquisition, training related to acquisition, record keeping of acquisition records, stakeholder involvement in acquisition, etc.

The following diagram will illustrate the full cycle for SAM.


Suppli
Acquisition,
Organization
Organization Units
Agreement, Proposals, Experience, Processes, HR, Plans, Schedule, QC, Support, ...
RFP
Proposal
Agreement
Status, issues, results of progress and milestone
Product Acceptance
Acquisition ST., Evaluation C., Score sheets, Vendor List, Selection, Sign Contract, Monitor, CR, Review, Factory test, UAT, ...
Request for S. Product, Tool,
Evaluation Selection
Reviews Corrective Actions
Testing Acceptance
PMO & projects Solution Architect , Design Team, Development Team, QA, CC, Training, ...
PMC, VER, TS, REQM, RD, RSKM, MA, DAR
REQ., ACQ. Strategy Eval. & Accept.

Biography

Mohamed Abo-Zied is a veteran SW quality expert that has more than 19 years in the field. Mohamed has passed through all the levels of SW engineering industry and quality related matters. His expertise enabled ITSoft to apply SW quality assurance and to define needed processes and implementing them in the best way possible. He was one of ITSoft core team responsible of achieving CMM 2, 3 then CMMI 4 accreditation. He is currently the head of SW quality assurance department as well as customer care office.

Feedback Contacts

Feedback, comments and questions are appreciated by the author.

Email:
mazeid@itsoft.com.eg

CMMI Implementation Series (Part 2)

By: Ahmed Abd El Aziz

In the previous issue we talked about the Initiating and Diagnosing phases of the IDEAL model in the trip of applying the CMMI. In this issue we will continue with the Establishing phase and part of the Acting phase.

The Establishing Phase

Set Priorities

When the time is short – actually it is too short in the current SECC project – to implement CMMI, this step gets more and more importance. When a company wants to implement CMMI or any other process improvement model, management, process improvement, and the technical staff hold many brain storming meeting to discuss many ideas of process improvement. Most of these ideas are terrific and if implemented they will add a great value to the company. However you do not have enough time to implement all these ideas. You have a deadline you want to meet. Even if you do not have time restrictions, still you can not implement everything at the same time. You have to set priorities of what to start with and what to postpone.

Identifying priorities is not an easy task. If we think in process improvement as a project, we find that we have many stakeholders that may have many interests which some of them may be conflicting with each other. Project managers have commitments within their projects and they want the simplest processes to follow and the simplest documents to fill – actually the first two comments I here when I introduce any new document to the project managers is that it is too long and very complicated. Management may be concerned more about enhancing some processes that affect a specific business line or unit in the company regardless the effect of this on even the certificate time itself. The process improvement team is looking for self actualization and is always looking for the certification and does not want to consume time in issues that will not affect the certification.

Here comes the role of the Process Improvement Team to negotiate the priorities with the different stakeholders and try to reach a compromise between all parties.

Develop Approach

From my little experience in the Process Improvement field, I saw two main approaches. The first starts with the model and tries to answer the question "what should I do to satisfy the model?" and assumes that when we satisfy the model our process will be improved as a consequence. The second approach looks first to the company business and tries to answer the question "What should I do to improve my business?". After answering this question, it is time to answer another question "Is what I am doing complies with the model and will enable me to get certified? If not, what I am missing?"

First I was biased to the first approach. However when I worked with people taking the second approach, I have to admit that it is much better. Think first in the business improvement needs. If you take it logically, you will end up with something that will cover at least 80% of the model requirements. Then you will have to do little more effort to satisfy the model.

In developing the approach there are many other issues that must be taken into consideration. Study the culture of the company and the staff very carefully. Some people are willing to learn more and more and want to try the new process even before it is formally published. Some people always complain from time shortage and always claim that they do not have and will not have time to follow any process. Some people want to make everything automated and try to look for a tool for everything.

The process improvement team has to deal with all these kinds of people and has to be patient, good listener, and be able to absorb the varying ideas – and attacks in some times – from the company staff. The personalities of the process improvement team members play a great role here. Also the trust of the company staff in the capabilities of the process improvement team has a great influence. If the process improvement team loses this trust, it will lose a lot.

Plan Actions

Like any project, we need to plan for Process Improvement. However we face a real problem here. How to identify the detailed activities required to implement the CMMI? How to estimate effort for these activities? CMMI is new for us and we do not have history or previous experience that we can rely on. So how to plan? I recommend creating an initial plan based on bench marking, expectations, or whatever then periodically monitoring and updating the plan by analyzing the performance and progress of the work. This is typical for any project, however here it gets more importance because many things are unknown.

This makes us pay attention to the Risks associated with the process improvement. The top high impact risk I think they may affect the project are people resistance, loosing management interest, and leakage of process improvement experience. However I think that the leakage of experience is mitigated by SECC and QAI support. Loosing management interest could be mitigated by always reminding the management by the fees they paid to join the SECC project and the possible loss of market share if the company is not certified. The staff resistance needs a lot of work to convince them with the importance and the added value of the new system for both the company and themselves on the personal level.

Another important issue to think about during the planning is the lifecycle of the implementation. Waterfall is wonderful, but you may have not enough time for implementation before assessment as most of the time may be consumed in defining the system. Fast tracking – paralyzing some activities like starting implementation after definition starts with considerable amount of time and before definition ends – will make the company start implementation faster and get implementation feedback much earlier, but however there is a risk here that the process improvement team will not have enough time to support people during implementation and continue in the definition at the same time. In the beginning of implementation people need a lot of support from the process improvement team to understand and implement the system.

The Acting Phase

Create Solution

There is no standard way to create the solution. Companies differ and people differ. Otherwise SEI would have documented the solution creation within the CMMI manual or somewhere else. However I can share with you some of the lessons I learned during the solution creation.

Think in-group of related processes, not one by one

For example, think in project management processes (planning, risk management, estimation, schedule, monitoring … etc) together. Also think in the software engineering processes (analysis, design, coding, integration … etc) and organizational processes together. There are many interactions between these processes. If you think in them separately, you may have two risks; redundant information appear in more than one location which makes the integration harder. The other risk is that you may loose some data because when you think in each process separately you think that these data will be covered in the other process and hence it gets lost from the two processes and makes integration harder.

Do not use many sources of information

A lot of processes are available in the net. However none of them can fit as it is in your company. In addition, we do not want to reinvent the wheel. It is good to look for samples made by others, but do not depend on many source and spent your time in searching the web instead of doing your job. Processes could be written in many ways. I recommend identifying two or three maximum sources to use as a guide during the implementation.

Write your process, do not copy it

I realized that junior process improvement engineer usually do this mistake. They do a lot of effort in searching and selecting a process or any other asset then copy it and change the header and the footer. In most of the cases this does not work. First it wastes a lot of time. Second this process is written for another company and it cannot fit within your company. Use it as a guide only if you want, but do not simply copy and paste.

Start with what you have

If the company has some processes either partially or completely followed and people are familiar with them, start with them and enhance them with time, but do not drop them at the start even if they are not the best. Still you have many many new things and there is a long learning curve required. This will reduce the resistance of the staff as part of the system they already know and follow.

In the next series I will share with you my experience with the rest of the IDEAL model, which could be simply entitled Implementation. Till that time I wish you all the best of luck.

Biography

Ahmed Abd El Aziz, has more than eight years of experience in the Software Development field. He worked as a programmer, a project manager, and Internet Department Manager. He is now member in the process improvement team in HARF Information Technology. He is certified as Project Management Professional (PMP). He passed the "Intermediate Concepts of CMMI" course. He has a B.Sc. Of Engineering from Cairo University and a Diploma in Computer Science and Information from Cairo University, Institute of Statistical Studies and Research (ISSR).

Feedback Contacts

Feedback, comments and questions are appreciated by the author

Email:
aaz@harf.com