# Egypt–SPIN Newsletter

**Issue 13, Jan. – Mar., 2006**

Sponsored by SECC


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

Welcome to our 13th 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, in **January 2006**, SECC became an **SEI Partner**. This authorizes **SECC** to **deliver "Introduction to CMMI Course"** and **provide SCAMPI appraisal services** commercially worldwide except in USA.

*This issue introduces some hot topics in three series and three independent articles as follows: experience with SCAMPI class A (1st article), real life experience in process improvement (2nd article), highlight the importance of the open source, information security requirements and discuss the different software lifecycle models (3rd, 4th and 5th articles respectively).*

***Eng. Sherine M. Murad** shares her experience in the **SCAMPI class A**. Her article discusses the appraisal's Pre-On Site and On-Site processes.*

***Dr. Ramiz Kameel** completes his **series to discuss** the **nature of the Egyptian software industry**. His article introduces a new **hypothetical model to improve the Egyptian software industry.***

***Eng. Ahmed Abd El Aziz** shares his experience –in form of a series - in **CMMI Implementation** journey according to the **IDEAL** model. His article describes the acting phase in the cycle.*

***Eng. Amr Shaltoot** discusses the importance of **open sources** to the strategic goals of the Arab countries. His article highlights the European initiative in the FLOSS project and other interesting aspects.*

***Eng. Ahmed Gad Al-Karim** completes his **series to discuss** the **Security in the Virtual Environment**. His article discusses the **information security requirements.***

***Dr. Mohamed El Zeweidy** compares the waterfall and iterative **lifecycle models.** His article focuses on the advantages of the iterative model in the changing environments.*

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 aselshikh@mcit.gov.eg

## Table of Contents

# ITWorx Experience with SCAMPI Class A

***By: Sherine M. Murad***

ITWorx would like to share its process improvement experience in achieving CMMI ML3 with other software companies to assist them in their process improvement journey. Different approaches and implementation best practices for process improvement have been covered by the intensive and rich SECC courses, the experiences of other companies represented in the SPIN newsletter, and the experiences of process group members across companies. We chose to present our experience from another perspective that might be more useful in this phase. We chose to tackle ITWorx Experience during the Appraisal.

In this article, we are using the SEI SCAMPI Class A Pre On-site activity, and On-site activity charts as a guideline and router for the article contents.

Although we tried to cover the entire ITWorx experience in the appraisal activities and the overall cycle, we need to highlight that the document will not cover all the appraisal activities' details; as it is well known that the appraisal activity take up to 2 to 3 days of training from the Lead Appraiser to the Appraisal Team members.

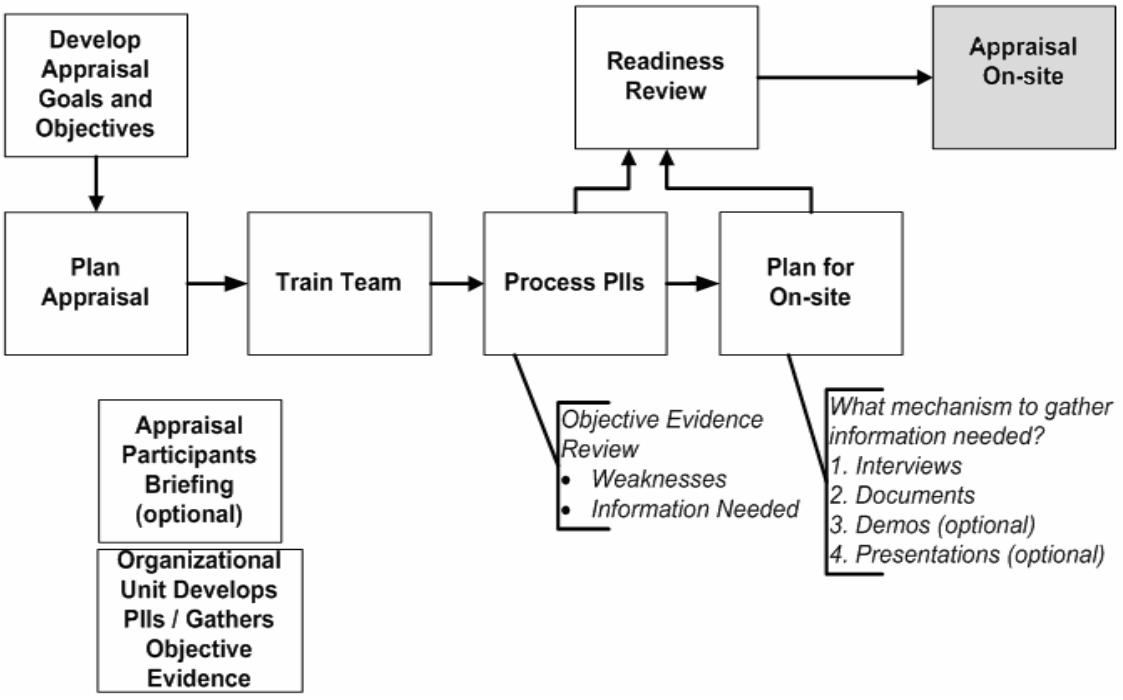



Figure 1: SCAMPI Class A Pre On-site Activity Chart

## DEVELOP APPRAISAL GOALS AND OBJECTIVES:

ITWorx appraisal goals and objectives were as follow:

1. Understand ITWorx current software engineering practices.
2. Identify ITWorx systems/software process's strengths and weaknesses.
3. Identify highest priority issues for software process improvement.
4. Rate the organizational characterization/ maturity level.
5. Facilitate the continuation of process improvement actions:

    a. Build ownership of results.
    b. Provide framework of actions.
    c. Establish commitment and sponsorship.

## PLAN APPRAISAL:

The Planning phase for the appraisal was performed as a part of the Organizational Process Improvement Plan that was created at the begging of 2005, based on ITWorx business objectives and needs. In this phase all the needed preparations to conduct the appraisal were carried as follow:

1. Prepare the appraisal plan to include the tailoring of appraisal method, activities to be performed, schedule estimate, resources, logistics and risks with associated mitigation plans. Team members had already reviewed the plan for the upcoming appraisal.

2. Send a list of ITWorx active projects list to the Lead Appraiser.

3. Work with the Lead Appraiser to select suitable projects for appraisal; this was done based on specific criteria which includes:

    a. Selected projects should cover all ITWorx divisions.
    b. Selected projects should cover all identified and different life cycles which are implemented at ITWorx.
    c. Selected projects must cover all defined project life cycles' phases.
    d. Number of projects must be reasonable compared to the appraisal period.
    [6 projects were selected]

4. Allocate **Appraisal Team Members** based on required qualifications. ITWorx assigned 8 ATMs which is the maximum number accepted by our Lead Appraiser this will help the organization to implement periodical internal audits across the organization to assure continuity of process improvement.

5. Allocate the **Project Leaders**. They are the interviewees, who provide data and review the preliminary findings.
[6 project leaders were allocated].

6. Allocate the **Functional Areas Representatives**. FARs are the practitioners interviewees, who provide data and review the preliminary findings.
[17 systems/software practitioners and a process engineering group of 7 persons were allocated]

7. Allocate the **Middle Managers**. They are in staff management positions, Interviewees, providing data and reviewing the preliminary findings.
[7 management representatives were allocated]

8. Prepare appraisal requirements for CMMI (ARC), V 1.1.

9. SCAMP method definition document (SMDD), V 1.1.

10. Complete Practice Implementation Indicator Descriptions for each process area on direct and indirect artifacts as applicable.

11. Agree with Lead Appraisal on the appraisal detailed agenda.

## TRAIN TEAM:

The training activity was considered as the most critical activity, not only because the team members must be well trained to perform their roles professionally as appraisal team members, and to recognize and identify the weaknesses and opportunities for improvements, but also because they will be responsible for handling the periodical internal audits across ITWorx.

After the training, ITWorx Appraisal Team Members were able to effectively and efficiently:

1. Perform document reviews.
2. Conduct interviews.
3. Determine practice characterization.
4. Develop preliminary and final findings.

## APPRAISAL PARTICIPANTS BRIEFING:

ITWorx personnel have enough experience with appraisals; therefore Appraisal Participants Briefing was combined with the Opening Briefing.

## OBJECTIVE EVIDENCE REVIEW AND INITIAL ANALYSIS:

ITWorx appraisal team members were divided into mini-teams based on the Categories of Process Areas. Thus the team worked in several time-consuming tasks in parallel.

Each mini-team reviewed available objective evidence for their assigned areas (and instantiations) to identify:

1. Areas for which objective evidence is:
    a. Not identified
    b. Not accessible
    c. Insufficient or sufficient
    d. Unclear or non-relevant

2. Potential strengths and weaknesses
3. Questions for interviews

## READINESS REVIEW

Only one readiness review is required per SCAMPI. It has to occur before the on-site appraisal. However, the readiness review need to be granted enough time before the actual on-site appraisal for the organization to fill identified gaps, fix logistics problems, and for the team lead to adequately address any team issues.

## OPENING BRIEFING

Main topics that were covered in the Opening Briefing are:

1. Senior management sponsorship
2. Goals of the appraisal
3. Sequence of the appraisal activities
4. On-site schedule
5. Roles of appraisal participants
6. Appraisal principles:
    a. Confidentiality and non-attribution
    b. Focus on process, not people
    c. Collaboration between the appraisal team and the members of the organization

## CONFIRMING PRACTICE IMPLEMENTATION

This phase was the longest one for ITWorx ATMs. The implemented activities were:

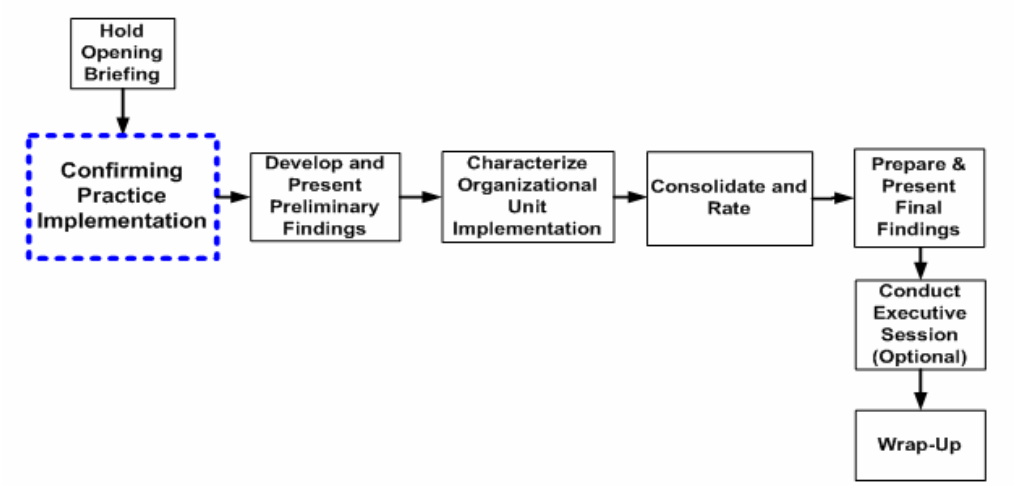



Figure 2: SCAMPI Class A On-site Activity Chart

1. **Review and adjust on-site schedule as needed.**

   As the appraisal proceeded, the data collection plans derived adjustments to the schedule to allow for additional data to be included, or sometimes to reduce planned activities if the data found was sufficient.

   ***Note*:** It is very important to understand that the activities are continuous and iterative.

2. **Review collected data by mini-teams**

   ITWorx ATM prepared scripting interview questions, and determined if the data is acceptable as objective evidence.

   ***TIP*:** Document review may provide clues that additional information is needed. And this is not a one time event. As the team progresses, document review can occur numerous times.

   Data Types are:

   1. Direct artifact: Tangible output(s) resulting directly from implementation of a specific or generic practice.

   2. Indirect artifact: Those are the consequences of performing a specific or generic practice, or substantiations of its implementation which are not the purpose for which the practice is performed.

   3. Affirmation: Oral (interviews) or written statements confirming or supporting implementation of a specific or generic practice.

      ***TIP*:** There is a thin line between **goodness** and **existence**. A document can exist but may not reasonably meet the intent of the CMMI practice. Teams must use their own judgment to ensure that the document reviewed supports the intent of the practice.

   ITWorx document library is a Web-based and shared document

library. This facilitated and organized the review process especially when more than one ATM needed to review the same document at the same time.

***Note*:** Know what you are looking for, and know when to stop looking

## 3. Conduct Interviews

SCAMP appraisal is not only paper-based; a minimum portion of Objective Evidence must be done through face-to-face interviews.

There are two affirmation coverage rules:

1. One row- One column Rule: for each goal, each related practice must at least have one affirmation objective evidence for at least one instantiation. AND every instance must supply at least one affirmation objective evidence for a related practice.

2. 50% Rule: for each goal at least 50% of the cells in the practice/instantiation matrix must have at least one affirmation data point

We implemented both rules; actually 50% rule is easier for implementation but in some specific Process Areas [OPF, OPD and OT] the 50% rule was difficult to be implemented and One row-One column was used instead.

Individual Interviews were conducted with project leaders. Group interviews were conducted for practitioners and management representatives.

Demos were held by Process Improvement Leader for Organizational Process Assets System which includes OSSP, Tailoring Guidelines, different Project's Life cycles, Proposals for Improvements and their analysis…etc. Demos were also held by Organizational Performance Manager for Measurements Repository and Internal tracking tools.

To handle an interview session you need to define different roles with different responsibilities. These roles are: a facilitator who conducts interview session, a timekeeper who tracks progress against plan through session, a librarian who records requests for documents if required, note takers who take notes recording answers to the questions (all team members except facilitator), and interviewees who provide answers.

Interview session can be summarized in few steps as follow:

1. Open interview.
2. Conduct Interview.
   a. Ask questions.
   b. Ask for documentation; if needed.
   c. Listen and take notes.
   d. Monitor Team and Interviewees.
3. Close interview.

Data Tagging should occur immediately AFTER the interview session. Data tagging means relate notes to the specific CMMI components; it is not an easy job and it requires CMMI knowledge.

## 4. Characterizing Practice Implementation on the Instantiation Level.

The practice implementation can be characterized as:

**FI: F**ully **I**mplemented,
**LI: L**argely **I**mplemented,
**PI: P**artially **I**mplemented,
**NI: N**ot **I**mplemented.

The rule to characterizing the practice implementation is as follow:

1. **FI:**
   a. Direct artifacts present and appropriate.
   b. Supported by indirect artifact and/or affirmation.
   c. No substantial weaknesses noted.
2. **LI:**
   a. Direct artifacts present and appropriate.
   b. Supported by indirect artifact and/or affirmation.
   c. One or more substantial weaknesses noted.
3. **PI:**
   a. Direct artifacts absent or judged inadequate.
   b. Artifacts or affirmations indicate some aspects of the practice are implemented.
   c. One or more substantial weaknesses noted.
4. **NI:**
   a. Any situation not covered above.

## 5. Characterizing Practice Implementation on the Organizational Level.

To characterize practice implementation on the organizational level simply you have to follow the following rule:

1. Combination of FI & LI results LI.
2. Any PI no NI results LI or PI [Team Judgment]
3. Any NI results NI, PI or LI [Team Judgment]
4. All X (the same characterization) results X

## Develop and Present Preliminary Findings

ITWorx ATMs extracted findings from the practice implementation worksheets, and from the interviews feedback. They used the standard template to develop the Preliminary Findings presentation. This activity considered as the last opportunity to get data, require additional evidences, and listen to FARs comments and suggestions.

## Consolidation and Rating:

A goal is rated **satisfies** if and only if:

1. All associated practices are categorized either Largely Implemented or Fully Implemented.

AND

2. The aggregation of weaknesses doesn't have a significant negative impact on goal achievement.

A Process Area is rated as follow;

- **Satisfied:** if and only if all relevant specific and generic goals are rated as satisfied.
- **Unsatisfied:** if and only if one or more relevant goals of the process area are rated unsatisfied.

- **Not Applicable:** Process area is determined to be outside the organization's scope.
- **Not Rated:** Insufficient is available to determine satisfaction or process area is outside the appraisal scope.

A Maturity Level is achieved if and only if all process areas at that level and lower levels are satisfied or determined to be not applicable.

### Prepare and Conduct the Final Findings Presentation:

The Final Findings presentation covered the appraisal objectives overview and scope, findings, rating and next steps.

### Wrap-up:

The lead appraiser collected the lessons learned, what went right and what went wrong, and completed the reporting templates to send required appraisal data back to the CMMI Steward (SEI). Follow-up activities were assigned to appraisal team members.

## Biography

**Sherine M. Murad** is the Process Improvement Team Leader in ITWorx. She had leaded the journey to achieve CMMI Maturity Level 3 in ITWorx. She has 5 years experience in a Process Improvement and leaded the process improvement journey in different companies. She played different managerial and technical roles since 1994. She has a Master Degree in Computer Science from Louisville University, Kentucky, USA and Certified in CMMI from SEI.

## Feedback Contacts

Feedback, comments and questions are appreciated by the author.

Email:
sherine.murad@itworx.com

# Toward Egyptian Software Industry: Methodologies for ESPC Model

***By: Ramiz Kameel***

## OBJECTIVE

This article "Methodologies for ESPC Model" is the fourth article of a series of articles "Toward Egyptian Software Industry" that concerns with methodologies that are responsible to maintain the efficiency of the ESPC model. This article will study choosing of proper methodologies for ESPC Process Model [1,2,3]. 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.

## INTRODUCTION

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 [1].

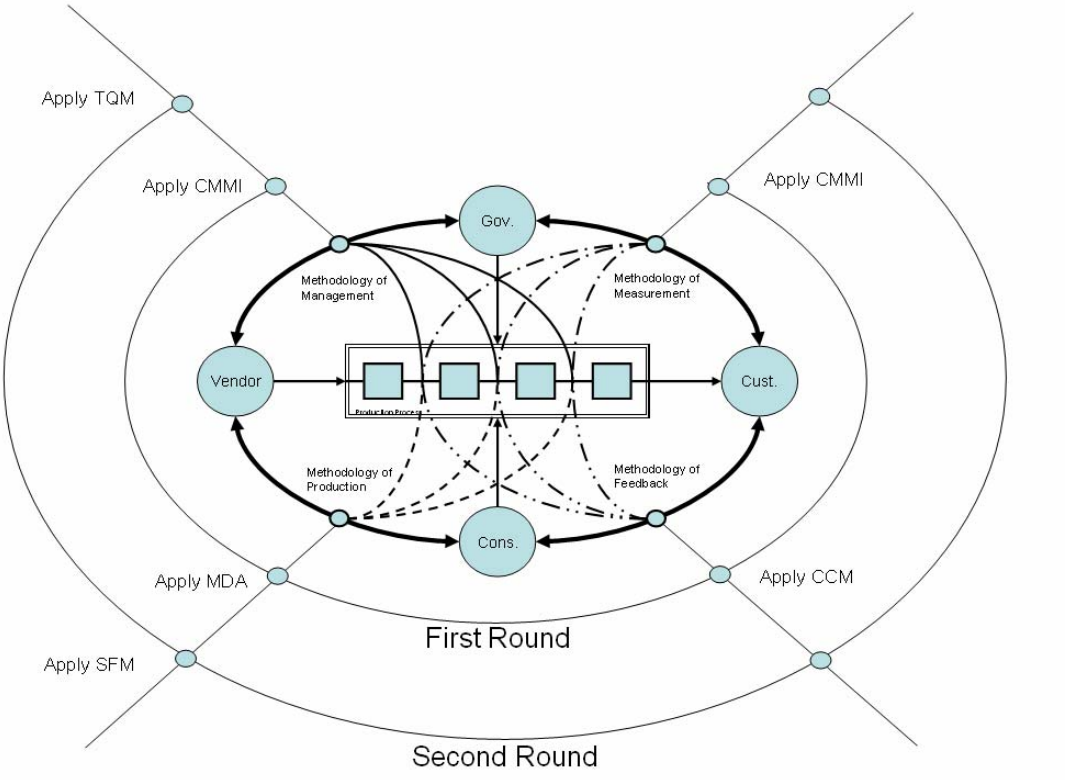



Figure (1): A Spiral Election Model

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.

## METHODOLOGIES ELECTION MODEL

A spiral election model can be followed to set the methodologies of the ESPC. The government pushed ESPC toward the starting. But still there is no a long plan to set the completed set of methodologies of the ESPC and there is no a corrective plan can be noticed during the last period of execution. The presented spiral model focuses on the all roles of ESPC. But this model is not completely spiral; in actual, it is a semi-spiral one. This model consists of several rounds, each one starts from the vendor role (software companies). There is no obligation to reach the end of the round to start another. So, the vendors can starts the higher round before closing the current executed one.

Software companies are invited to apply the CMMI as a first step in applying quality assurance in the ESPC, to prepare the ESPC for this new stage and to define the Egyptian customized experience that can be applied later on the whole ESPC. It is preferable to consider the earlier CMMI applying in software companies as a pilot stage. It is recommended to follow this experience in ESPC with considering all notes and comments and applying the proper modification without violating the CMMI methodology.

Software companies are invited to share the government role to select and define the proper methodology that should be followed prospectively for continual improving of the ESPC. The network model that was proposed previously [2] is proper to perform this task.

CMMI is not sufficient for improving the product quality of the software community. By definition, CMMI concerns with the quality of the producing process. There is no evidence till now that ensure producing a perfect product by applying the CMMI. That shortage can be supported by applying the Model Driven Architecture MDA. This will be another step in the first round progress. Back to the network model [2], the consultancy role can share the vendors in accomplishing this step. And the consultancy role is invited to interpret that to serve the customers and beneficiaries through a proper Customer-Consultancy Model CCM.

Historically, CMMI is developed to be a tool in the customer's hand to evaluate the vendors. CMMI is a proper tool for controlling and measuring the performance of vendors. The customers should be trained well on this tool. Government should evolve the customers and their IT staff to recognize the software producing phases and relationship between their requirements and the final product.

Not necessarily to close-up the first round to start the second one. Government can prepare for higher level of quality control and assurance that focuses on the product and its quality. This is the real challenge in the prospective future. No doubt, the starting by controlling the process is vital stage. But still no evidence, that the CMMI can produce a perfect industrial community regardless the product quality. By calling-back the experience of any industrial community, one can conclude that there is no separation between the

product quality and process quality. And from previous experience, the product quality is most important former step in improving the community.

Consequently, from the mentioned above, the Software Factorization Model SFM is required to follow in the next stage of the improvement. Software production automation is a recurring economic and organizational preoccupation. The leitmotiv is to rationalize production for improving productivity, quality and flexibility, in order to reduce costs and to increase profits, financial as well as technical [4].

## CONCLUSION

More focusing in the product quality is required and government is invited to support that. Professional involvement of the customers must be established. The customer role has to share by its input in the ESPC to examine the powerful of local vendors to fulfill, before depending on the foreign product [5].

## REFERENCES


[1] Kameel, R., 2005, Toward Egyptian Software Industry: Egyptian Software Industry Improvement Features, Egypt-Spin Newsletter, SECC, Issue 12, Oct. – Dec. 2005, Pages 6-11.

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

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

[4] Langlois, B., Exertier, D., 2005, MDSOFA: A Model-Driven Software Factory, THALES Research & Technology France, 2005.

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


## 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. 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

# CMMI Implementation Series (3)

***By: Ahmed Abd El Aziz***

In the previous two issues we went through the Initiating, Diagnosing, and Establishing phases of the IDEAL model in the trip of applying the CMMI and started talking about the Acting phase. In this issue we will continue the acting phase. In the last article we stopped at "Create Solution". In this article we will continue starting from "Pilot/Test Solution".

Before I go with the model, I would say thank you to all the people who gave me their feedback on the last two articles. I appreciate their criticism and I actually learn a lot from them.

## THE ACTING PHASE

### Pilot/Test Solution

Because of the short period of the project, we did not have time to test our solution. We took the risk to directly implement the solution after creating it. We thought that sharing and discussing the solution during the creation phase with the relevant professional staff in the company will mitigate this risk.

I am still convinced that we did the right thing within the time restrictions we faced. However I see this phase is very important and if you have time, you must pilot the solution before it is fully implemented. Whatever the effort you do for creating the solution and the consultancy you take from the professional staff, when it comes to implementation you will find many issues that are more theoretical than practical. Sometimes you have two possible solutions to the same problem and you can not judge at the solution creation time which one is better. You decide to implement one of them and see. During implementation you find that the other one is better. If you test your solution first with one or two projects, you get this feedback from these projects and refine your solution. If you implement directly, you get this feedback from many many projects and you waste the time for all of them.

One example from our experience is that we created an SRS (Software Requirements Specifications) document which contains a section for documenting the use cases. We discussed the template with the technical staff and they approved it. However when we implemented it we got the same feedback from all the projects that used it. They all said "We use Enterprise Architect and it generates the use cases document in a different format other than the one in the template and hence we do not want to follow the template". If we tested the solution first, we could have this feedback earlier from one project only, updated the template, and spared the discussion time with the other projects staff.

One important point regarding testing the solution is that how you test it and with whom. When you select a project to test the new process, select the project that has the staff that really wants to apply the process and wants to learn from it. Do not simply pick the project that its schedule is suitable or has a little pressure. If the staff is resisting the process, they will give you invaluable feedback and will depress you. You will spend time fighting for process implementation instead of testing the solution.

## Refine Solution

This step complements the previous step, testing the solution. Actually the testing step is meaningless and waste of time if it is not accompanied with the refinement step. Here we update our solution based on the feedback from the testing step to provide a better solution to the rest of the company and projects.

## Implement Solution

When we decided to start implementation, we divided it into two main parts, Organizational and Projects. In the organizational part we had no problems to start implementation as most of its activities were related to the EPG and QA team.

For the Projects part we decided to study each project separately in coordination with its Project Manager and decide together which processes it will implement and which it will not based on the project lifecycle and the current phase of the project. This applies for engineering activities only like system analysis, design, coding, and testing. However for project management activities, we decided it should be followed with all the projects regardless the current phase of the project is.

For example all projects must follow the planning process and monitoring and control process, but starting from now to the end of the project. We did not ask them to re-plan the project from its start and show evidence for past activities. They were asked to update their plans from now on to the end of the project according to the new planning process.

I have too many lessons I learned during the implementation which I would like to share with you.

## Do not take it personal

When implementation starts, you will face many unpleasant situations with the staff. For example I got the statement "You are not authorized to do …" in an email that is CCed to the company management. In another email I got the statement "If you are not satisfied with what I am doing, till the management". In these situations and similar ones I became very nervous. However I had to be quite and calm down before taking any action. People say that because they are already under stress from many directions. They have nothing against me or any of the process improvement staff. So in similar situations, take it easy and ignore such things. Always remember your main target and do not take things personal.

## Avoid excessive overload

A lot of effort is required for defining and implementing a good quality system. This effort is doubled in case of aggressive deadline – the case of the current SECC project. Do not forget to take suitable breaks in between so you can continue. If you work hard for continuous four or more months without any vacations or any change in your life, you may feel bored in very critical times and can not continue.

In the current project you feel like you are in a race. In the start you have a consultant visit every month and you want to show the consultant something. Then you have a spot check. Then the pre-appraisal. A series of continuous stress. You prefer to continue to do something more before each step of these. This will work at the beginning, but may be just before the final appraisal or the pre-appraisal you feel tired and bored and start looking for break.

I recommend that you take short break after each milestone to become active and can continue.

### Do not expect too much

One of the mistakes I felt in is that I expected that implementation is not going to be a problem as our staff is well prepared and have good experience and they will not resist the new system. When we started implementation, we faced a huge resistance, or in other words the normal resistance.

Regardless the awareness level of the staff and their wishes and promises regarding implementing the new system and learning from it, they will strongly resist the system either directly or indirectly. Over expectation may get you depressed when you find that the real world is not the same as you expected or as you wanted it to be.

### Avoid multiple views

This point is actually related to the definition, but we discovered it during implementation. In some of our templates we have many sections that show the same information but from different angles (views). For examples in the project plan template we have a section for stakeholder involvement that shows the roles and responsibilities of every stakeholder in the project in every lifecycle phase. We also we have another section for the documents to be communicated between the stakeholders. The information in the two sections is nearly the same.

For example who is responsible for preparing the SRS in the stakeholder involvement section, must be responsible for sending it in the communication section. Who reviews it in the stakeholder involvement section must receive it in the communication section. Here the communication section has added a small value. On the other hand it added a lot of effort on the project manager to write that section and make sure that the information in both sections is consistent. It also added effort in the review and the maintenance. So I strongly recommend avoiding such approach.

### Do it with them, not instead of them

Whatever the training you offer to the staff is, when they start implementation they do not do it as it is written. People hate to read. They can ask you ten times about the same process wasting hours of their and your time but they will not spend ten minutes to read the process and follow it. Even if they read the process, they may not follow it accordingly because of misunderstanding of the process as a whole or some points in the process.

One thing we tried and it was very effective is that we participated with the teams while they were doing the activities. This consumed reasonable time from the process improvement team, but it had many benefits. It helps the team do things better. The process improvement staff lived the implementation and saw what is good and what is bad in the system. We could easily identify the weaknesses of the process and fix them.

### Do not show your depression

For one or more of the reasons mentioned above, you may get depressed. This is very dangerous. The depression of the process improvement team member is automatically reflected on the other people in the company. Do your best to avoid showing others this feeling. If time permits, take a vacation for one or two days. If not, avoid contacting

people and delegate this to your colleagues in the process improvement team – who are not depressed at the moment. Do not say that I will contact them but I will not show them that I am depressed. This feeling can not be hidden. So it is better to avoid people till you feel that you are ok again and can continue.

In the next issue, I will share with you more about the lessons I learned from the implementation phase.

## Biography:

**Ahmed Abd El Aziz**, has about nine 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

# The Importance of being FLOSS!

***By: Amr Shaltoot***

When Oscar Wilde wrote his comedy "The importance of being Earnest, 1895" he meant to draw our attention that people, money and time are the great resource of human being, and should not be wasted nor badly managed.

For governments and large enterprises –whether public, governmental, or private, looking into meanwhile is important, because it drew the instant attention of people, and how they tend to benefit from the services/products these enterprises provide.

On the other hand, thinking of the future is of no-less importance, to the present. It has been always the job of visionaries and think tanks in every society, no matter, neither where they originally belong to nor what ideology they stick to. They have been always de-embroiling a thread among the past, the present and the future; trying to resolve the code of possible and impossible.

For the IT industry, Operating Systems (OS), Databases (DB), Security (SECU), and Hardware (HW) are the most important foundation pillars beside People (PPL), who must be focused upon.

As Egypt, not ready yet to establish a serious committee to discuss the future of such National Security issues, at least we must discuss the future of the huge investments in packages based on closed source OS, DB, and Security products/services!

What if any OS company, decided to charge any application that runs on its platform? What if they created –and I believe they will- a pricing metrics that charges applications per used service per session time for an example?

What if any government legalized a law that oblige search engines to deliver to the Intelligence a report of keywords, IP-s, and more breaching privacy components, which have been searched using these search engines. What if our Intelligence using these Search Engines!

What an Intelligence need more than analyzable data to form a validated and verified opinion or understanding or estimate?

The EU "European Union" has decided upon FLOSS OS, and now they are building a "second generation" search engine called Quaero, will cost Euro 1.2 to 2.4 Billions on five years!

Some voices will shout and say, are you out of your mind 2.4 billions of Euros! Yes, and they paid almost the double on the Galileo satellite positioning system as a replacement to the American GPS system. They spent and will continue to spend because it is about National Security!

Can Arabs pay such amounts, before it is too late, I don't know.

To dig down more to other aspects of the problem –in five years it may transform into a crisis, and in ten it will be the mother of the catastrophes, I'll pinpoint the economical aspect. The Egyptian Government currently spending hundred millions of pounds[1], in the infra-structure of the IT sector. No doubt, it made a large pace

[1] The largest portion is grants come from The UN, EU, and Microsoft.

towards a "digital society[2]"! But, do these investments are secure enough? The purchase which have been made, soon will obsolete and as it is a strategic agreement or partnering with theses three pillars (OS, DB, SECU) which are composing the huge infra-structure serving now 73M people, The Government will pursue an upgrade!

So, what are the conditions within the agreement that protects such investments?

Of course, we must be proud of what have been accomplished, but we must take the initiation now and think ahead a few steps. The most these closed source companies get wealthier, the most they become monopoly oriented ideologists. They will keep crunching the consumers, until they take the maximum benefit, and then form other companies which will start over and over again.

Licensing metrics are going to change soon, and their "Army of public relations infantries" will keep trying to convince the consumer that he is not in a need for installing the application to his computer, nor having anything at all! You pay, you get the right of using the application on-line, you don't have money seek a typewriter, or an ordinary calculator!

They will place a huge Application Servers in each region or country and consumers will find them selves paying to use such applications per session! And for the most sick minded, the ISP itself will be owned by the closed source company too!!

They will not stop; they will manufacture PCs which consumer pays for it too -and the closed source uses its processor and memory power free of charge, PDAs and every thing end-to-end. Is this against Fair Trade? Is this against humans? Dare to question? Don't know!

Subtle dealing with such high-sensitivity National Security issues will end up with a bad surprise to everyone, sooner or later. `It is very important to inspect how any OS, DB, and SECU-s encrypts/decrypts data, and how they transmits/receives them. These unique areas of the application must be available to the community. Also, closed source manufacturers must allow third-party encryptions to dock into their engine. It is no more accepted not to agree upon this minimum level of protecting ourselves.

There are more than this to discuss and elaborate on, but to me a serious-enough committee must be established to discuss, depict and come up with resolutions, and guidelines that to be published and availed to all Governmental, Public Private, and Education sectors.

Out there in the field, it is either you are ready or not. So, who can tell me if we are or not? You can tell…but don't forget to email me, please!

## Biography

**Amr Shaltoot**, Software Engineer, holds MBA in Strategic Marketing and Business Intelligence. Also he is a Doctoral student in Marketing Intelligence. He is interested in: Investment management, Strategic Marketing and Start Up-

## Feedback contacts

Feedback, comments and questions are appreciated by the author.

Email:
amr@shaltoot.net

[2] As to my knowledge, no clear definition to this term up till now.

# Information Security in the Virtual World Series: Information Security Requirements

***By: Ahmed Gad Al-karim***

As mentioned in the last article, the overarching security policy has three primary goals: to ensure the availability, integrity, and confidentiality of an organization's information assets. A fourth requirement, assurance, is mandatory to ensure that the measures taken to provide confidentiality, integrity, and availability include a mechanism — such as auditing logs — to prove that the mechanisms are working. These requirements are stated briefly below.

## Confidentiality

An organization's information assets must remain confidential. Information should not be disclosed to anyone, whether inside or outside the organization, who is not authorized to access it.

## Integrity

An organization's information assets must maintain their integrity. The system must not corrupt information or allow any unauthorized malicious or accidental changes to it. Information integrity also encompasses communications integrity — ensuring that network communications are transmitted accurately and are not forged or modified during transmission.

## Availability

An organization's information assets must be readily available to those people, processes, or agents (whether internal or external to the organization) that are authorized to use them. The opposite of availability is referred to as "denial of service," when authorized system users aren't able to get the resources because of system failures or deficiencies. "Denial of service" attacks are those that focus on occupying system resources so that this situation occurs — flooding an e-mail gateway with traffic until it crashes or occupying a file server with a batch-file that executes an endless loop of commands would be examples of such attacks.

## Assurance

An organization must have a means of ensuring that the mechanisms implemented to ensure the confidentiality, integrity, and availability of information assets are working. Auditing facilities, systems management tools, logs, alarms, and a variety of tools and procedures provide this measure of accountability. For example, In the NAC Security Framework it's presumed that each of the mechanisms that provides confidentiality and integrity includes an auditing capability. Auditing ensures that if security is compromised, it can be traced to the source.

## Functional Processes

The enterprise computing environment is heterogeneous, composed of network operating systems, distributed client-server applications, databases, and a range of network services, such as file and print, directory, and messaging, to name a few.

Two basic processes which ensure that only the appropriate users get onto the network, look at information in a database, open files on a network

drive, access a mailbox in their name, and so on, include authentication and authorization.

**Authentication** is a process which occurs in three different security contexts:

- verifying user identity
- verifying the origin of a message
- verifying message content

Verifying message origin and content will be discussed later when discussing the concept of "*Encryption*."

After a user's identity is validated, he or she can access the network object within the parameters specified for that user via the network object's *authorization*, or *access control*, mechanism.

**Note:**

throughout the discussion, the word "user" refers to any client process, whether initiated by a human user or non-human processes, programs, or agents and the word "message" refers to network packets, e-mail messages, inter-process communications, remote procedure calls — computer-related communications of any kind and at all layers of the OSI protocol model.

## Identification and Authentication

Verifying user identity prior to providing access is typically referred to as Identification and Authentication, or I&A. It's a two-step process that validates a user's purported identity (subject) prior to providing that user access to a resource (object). For example, when users wish to connect to a client-server database application, they enter a user name or ID, in conjunction with a password.

Identification and authentication processes can rely on combinations of one or more of the following:

- User logon plus password
- User logon plus third-party generated password or token (for example, Security Dynamics ACE/Server and SecureID token-card combination, or a cryptographic authentication token such as that created by Kerberos or another cryptographic authentication protocols)
- Biometric-based authentication (fingerprints, handprints, voice recognition, retina scans)

Each of the techniques listed is increasingly more secure. For example, a user logon and password combination, the most common authentication method, can be enhanced by the addition of a token-card mechanism. This adds another factor — the token-card — to the password: in order to prove identity, the user must not only know something (the password), he must also have something in his possession, specifically, the token-card. A two-factor authentication mechanism is stronger than a single factor mechanism, that is, a password alone.

A third way of proving identity is through biological information that is unique to each individual, such as the fingers, voice, or eye. Biometric-based authentication have historically been too costly for anything but environments demanding the highest degree of security, but these are coming down in price and being adapted for client-server environments. Desktop fingerprint-scanning devices with PC interfaces specifically for use with distributed computing environments are on the market.

In a distributed, client-server environment, the identification and authentication process occurs at several stages. First, a user must logon to the network and then logon and provide a password (or other mechanism) to access e-mail, database applications, groupware, and so forth.

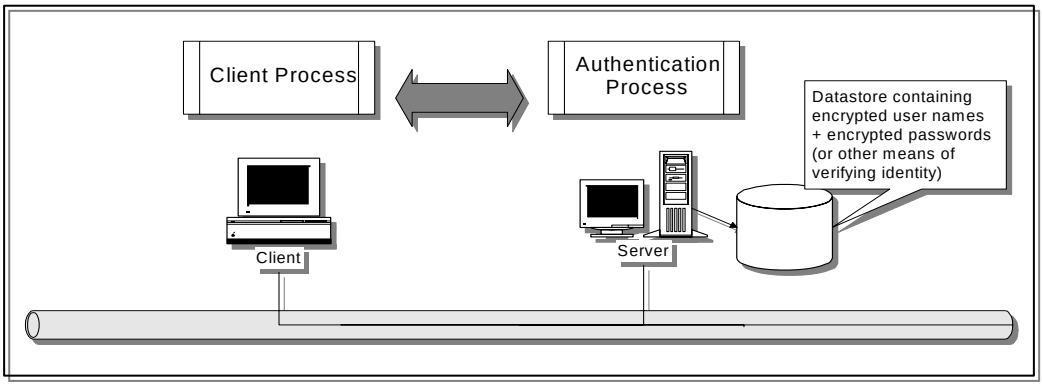
## Authentication Services

Rather than authenticate user identity on an application-by-application basis, an authentication service provides a focal point for authentication, ideally across all applications or what is so called Single Sign-On (SSO).

An authentication service has a database of user accounts, passwords, and information about services. When a user logs on, the request is sent to the authentication service, which issues a ticket or some other such "credential" after authenticating the user. The credential then authenticates the user to all services and applications for a specific period of time.

Kerberos, a cryptographic authentication protocol that was devised as part of MIT's project Athena, is a widely used authentication service. The protocol has been implemented by several vendors, including CyberSafe, and has also been adopted by the OSF for its DCE, although the DCE version of Kerberos is at this point not compatible with MIT Kerberos; implementations include both versions 4 and 5.

Other authentication services available include various vendor products, such as Bull Access Master Service, ICL/Access Manager, and SESAME. An authentication framework, X.509, is specified as part of the X.500 directory service recommendation. The NetWare 4.x directory service includes a ticket-based authentication function that is conceptually similar to Kerberos. The authentication process is as shown in the following fig.




| Client Process | Server Process |
|---|---|
| 1. User enters name and authentication token. | 3. Compares encrypted User name and authentication token to encrypted version of the token in authentication database. |
| 2. Client process sends User name and authentication token to server. Entry may be encrypted before sending over network. | 4. If entry in database matches entry sent from client, service grants access (based on ACLs (access control lists or other authorization mechanism specific to the service or application.)<br>— If no match, access is denied. |

The table below lists some of the identification and authentication mechanisms in use today. The list below is by no means complete, but is merely a representative example of the situation in the industry today.

| Product | Identification & Authentication |
|---|---|
| Network | |
| Banyan VINES | VINES Security Service (formerly Vanguard) |
| H-P Unix | HP/DCE Logon |
| IBM/SNA | RACF |
| Novell NetWare | NWDSLogin, NWDSAuthenticate |
| OSF/DCE | DCE/Kerberos |
| Unix | Unix Logon |
| Windows NTS | NT Logon + LANMAN Password |
| Database | |
| Oracle Oracle7 | O3LOGON (or optionally, subsystem OS, Kerberos, DCE/Kerberos, SESAME) |
| IBM DB2 | Subsystem + DB2 |

## Authorization

In the case of file share or file service, the authorization mechanism is referred to as an access control list (ACL) or access rights list (ARL); the list contains a list of users that can access the file share and specifies the level of access that they each have; for example, Read Only, Write, Execute, Delete, or a combination of whatever rights the file share supports.

The two major types of access control mechanisms, discretionary access controls and mandatory access controls, offer different degrees of security.

*Discretionary access controls* function at the discretion of the person who owns the entity. With discretionary access controls, security can be circumvented, albeit indirectly: a user who has rights to an object can transfer those rights to a third entity, who by rights may not necessarily have access.

For example, if User A has rights to File X on a file share but User B doesn't have rights, User A could conceivably reset the rights or copy the File X to a diskette and give it to User B.

On the other hand, mandatory access controls enforce security based on a system of labels and clearance levels. Using the example above, in a mandatory access control environment, User A and User B would each have security clearance levels and File X would have a security label — Top Secret, Secret, Confidential, Unclassified, for example — attached to it.

Presuming User A's clearance allows him to access File X but User B's security clearance isn't high enough, even if User A copies File X to disk and hands it to User B, User B won't be able to access File X.

## References:

[1] Enterprise-wide Security - A NAC Position Paper - First Edition - July 14, 1996.

[2] Enterprise-wide Security - Authentication and Single Sign-on - A NAC Position Paper - July 14, 1996

## Biography

**Ahmed Gad Al-Karim**, is a security and infrastructure consultant in the Egyptian e-Gov Program. He has 7 years of experience in the field of information technology. Currently, he holds Techno-MBA. Information security is his major interest. His interests include ISO 17799, BS 7799 security systems.

## Feedback Contacts

Feedback, comments and questions are appreciated by the author.

Email:
ahgad@mcit.gov.eg

# Waterfall - iterative development: A comparative study

***By: M. El Zeweidy***

## INTRODUCTION

Producing quality software on time is very challenging; you must overcome a large number of obstacles and problems. Problems are often first recognized by their symptoms. Dealing with changes is one of the most important issues that should be taken into account during s/w life cycle. The reality is that you can't stop change. Normally, we used to try to baseline requirements at the start of a project and then resist any changes to these baseline requirements. A better approach is to develop practices that allow you to manage changes when it occurs, with minimal impact to the development process.

This article compares the waterfall and the iterative models and explores the improvements that an "iterative" approach offers to the software development process over the traditional waterfall approach; which might encourage teams decide to move gradually from a waterfall-like approach to a more iterative one.

## WATERFALL MODEL

Software has been developed using the waterfall model for more than 30 years. The concepts are familiar to most developers and most project plans are developed with the waterfall in mind. One of the problems with complex system design is that you cannot foresee the requirements at the beginning of the project. In many cases, you think that you can start with a set of requirements that specifies completely the properties of your system but finally you end up with something the user does not need.

The basic concept of the waterfall model is that; the entire system goes through the phases linearly. First all the requirements are defined, and then the design is completed. Finally, code is written and tested. The key assumptions are that when design begins, requirements no longer change. When coding starts, the design ceases to change. These assumptions are, of course, incorrect. Hence problems result and software is late, over budget, and/or unacceptable.

### Waterfall Development Characteristics

- Delays confirmation of critical risk resolution
- Measures progress by assessing work-products that are poor predictors of time-to-completion
- Delays and aggregates integration and testing
- Precludes early deployment
- Frequently results in major unplanned iterations

Waterfall is conceptually straightforward because it produces a single deliverable. The fundamental problem of this approach is that it pushes risk forward in time, when it is costly to undo mistakes from earlier phases. An initial design will likely be inconsistent with respect to its key requirements; meanwhile, the late discovery of design defects tends to result in costly overruns and/or project cancellation. The main and great disadvantage of the waterfall approach is that it tends to mask the real risks

to a project until it is too late to do anything meaningful about them.

The waterfall model problems can be summarized as:

1. Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.

2. Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.

3. Few business systems have stable requirements.

4. Managers love waterfall models because it has :

    - Nice milestones
    - No need to look back (linear system), one activity at a time.
    - Easy to check progress : 90% coded, 20% tested

In practice, software development is not sequential. The development stages overlap. The tendency to freeze parts of the development leads to systems the client does not want and which are badly structured as design problems are circumvented by tricky coding

## FROM WATERFALL TO ITERATIVE DEVELOPMENT

Most software teams still use a waterfall process for development projects. Taking an extreme waterfall approach means that you complete a number of phases in a strictly ordered sequence. You also postpone testing until the end of the project lifecycle, when problems tend to be tough and expensive to resolve; these problems can also pose serious threats to release deadlines and leave key team members idle for extended periods of time.

## Definition of Iterative Development

Iterative development is defined as steering a project by using periodic objective assessments, and re-planning based on those assessments.

**Iteration** is a distinct sequence of activities based on an established plan and evaluation criteria, resulting in an executable release (internal or external).

Iterative process was developed in response to waterfall characteristics. With an iterative process, the waterfall steps are applied iteratively. Instead of developing the whole system in lock step, an increment (i.e. a subset of system functionality) is selected and developed, then another increment, and so on.

The selection of the first increment to be developed is based on risk, the highest priority risks first. To address the selected risk(s), choose a subset of use cases. Develop the minimal set of use cases that will allow objective verification (i.e., through a set of executable tests) of the risks that you have chosen. Then select the next increment to address the next highest risk, and so on. Thus you apply the waterfall within each iteration and the system evolves incrementally. The earliest iterations address greatest risks. Each iteration produces an executable release. Each iteration includes integration and test.

## Iterative Development Characteristics

- Resolves major risks before making large investments
- Enables early user feedback

- Makes testing and integration continuous
- Focuses project short-term objective milestones
- Makes possible deployment of partial implementations

In practice, most teams use a modified waterfall approach, breaking the project down into two or more parts, sometimes called phases or stages. This helps to simplify integration, get testers testing earlier, and provide an earlier reading on project status. However, that runs counter to the thinking behind the waterfall approach: Many design teams would view modifying the design after Stage-1 as a failure of their initial design or requirements process. And although a modified waterfall approach does not prevent the use of feedback, it does not facilitate, accommodate, or encourage it. And finally, the desire to minimize risk does not typically drive a waterfall project.

## Advantages of an iterative approach

Iterative approach involves a sequence of iterations. Each iteration includes some, or most, of the development activities (requirements, analysis, design, implementation, and so on), as you can see in Figure 1.

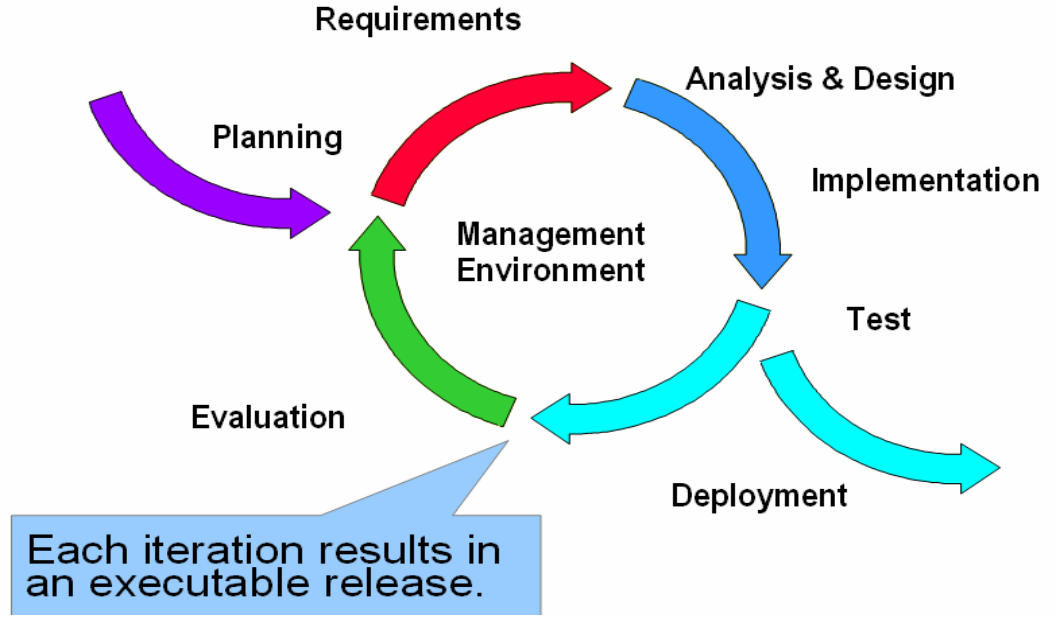



Figure 1: Iterative development.

The main advantages of the iterative approach are:

- Mitigates risk
- Iterative Development attacks highest project risks first
- Commits resources on a proven plan
- Improves software economics
- Provides a framework in which adjustments can be made as the project progresses.
- It accommodates changing requirements.
- Integration is not one "big bang" at the end of a project to avoid time-consuming rework.
- It facilitates better use of project personnel.
- Team members learn along the way.

## Biography

**Dr. Mohamed El Zeweidy** main areas of research include Software Engineering, Databases, Data warehousing and Data Mining, and DSS.

## Feedback contacts

Feedback, comments and questions are appreciated by the author.

Email:

melzeweidy1@yahoo.com