



## Egypt–SPIN Newsletter

Issue 15, Jan. – Jun., 2007

Sponsored by SECC

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

Welcome to our 15<sup>th</sup> 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.

We are used to announce the major events happened in Egypt in the field of process improvement. At the same time, it is a pleasure to announce many success stories and great achievements that happened during the previous period of time.

Actually, we will not be able to mention in details, due to size limitation, all of the events that we can consider, with high level of confidence, as major milestones in our process improvement journey. Please, refer to SECC website, [www.secc.org.eg](http://www.secc.org.eg), for more information. Here, you can find a summary list of the most important events:

- **1<sup>st</sup> Quarter of 2007**
  - SECC held a “**TSP/PSP Overview**” session to introduce the SEI-TSP/PSP service to the Egyptian community.
  - SEI course entitled “**Introduction to CMMI (Staged and Continuous)**” was conducted in 2 rounds by an Egyptian SEI-Authorized CMMI Instructor.
- **2<sup>nd</sup> Quarter of 2007**
  - SEI courses entitled “**PSP for Engineers I & II**” were conducted by 2 Egyptian SEI-Authorized PSP Instructors.
  - SEI course entitled “**Managing TSP Teams**” was conducted by 2 Egyptian SEI-Authorized PSP Instructors.
  - **Six Egyptian companies** achieved **CMMI ML 2** and another **one** achieved **ML 3** as a result of the SECC/IMC-1 Project extension.
  - SECC started 3 initiatives for providing “**Microsoft VS Team Edition**”, “**IBM Rational**” and “**Oracle Servers & Development Suite**” tools for 50 companies in each initiative.
  - SECC participated in **GITEX EI-Riyadh 2007**.
  - SECC successfully completed the authorization processes for one “**SEI-Authorized SCAMPI A Lead Appraisal**” and two “**SEI-Authorized TSP Coaches**”.
  - SECC held a celebrating ceremony for the **CMMI-Appraised Companies** during 2006 and 2007.
  - SECC started the “**CMMI 2007**” project with 11 Egyptian companies to achieve CMMI ML 2 and 3.
  - SECC started the “**SPI for SMEs 2007**” project with 20 SME companies in Egypt to apply the “**SPIG Product Suite v1.1**”.
  - SECC started a new industry support services to S/W companies targeting **High CMMI Maturity Levels 4 & 5**.
  - SEI “**Implementing Goal-Driven Measurement**” and “**Managing Software Projects with Metrics**” courses were conducted in Cairo.

*This issue introduces some hot topics in one series and four independent articles as follows: SECC public events (1<sup>st</sup> article), minimum expectations of CMMI ML2 and ML3 (2<sup>nd</sup> article), black-box testing, software and system measurements and software process improvement (3<sup>rd</sup>, 4<sup>th</sup> and 5<sup>th</sup> articles respectively).*

**Ms. Sherine FaragAllah** summarizes the important **SECC public events** conducted during the first half of year 2007.

**Eng. Ahmed Abd El Aziz** raises a warring flag to the software companies that are ambitious to achieve the **transition from CMMI ML2 to ML3 using traditional approaches** in only one year. In addition, he summarizes the minimum level of **expectations in ML2 and ML3**.

**Eng. Omar Kamal** completes his **tutorial series about software testing**. His article discusses the concept of **pairwise testing** as one of the black-box testing techniques to validate that the product fulfill its requirements.

**Eng. Manal Helmy** shares her experiences in the field of “**Software and Systems Measurements**”. She gives an overview on the **Practical Software and Systems Measurement (PSM)** as an objective process for analyzing project issues, risks, and financial management.

**Eng. Ahmed M. Hammad** summarizes his experiences in the field of **Software Process Improvement** Including the importance of building a knowledge base and the **IDEAL** approach.

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](mailto:spin@secc.org.eg) or [aselshikh@mcit.gov.eg](mailto:aselshikh@mcit.gov.eg)

---

## Table of Contents

Extending SECC-SEI Partnership to Provide TSP/PSP Services .....	4
Important SECC Events.....	5
From CMMI ML2 to ML3 in One Year!	
ML2 Process Areas Expectations .....	7
Software Testing Techniques Series:	
Pairwise testing technique .....	11
Practical Software and Systems Measurement (PSM).....	14
Software Process Improvement.....	18

## Extending SECC-SEI Partnership to Provide TSP/PSP Services

*By the editor*

In **January 2006**, SECC became the only strategic partner in the region to the **Software Engineering Institute (SEI), Carnegie Mellon University (CMU), USA**. This partnership gave SECC the permission to deliver **SEI CMMI Services** world wide, except USA, including SEI official "Introduction to CMMI" Course and SCAMPI appraisals. To deliver these services SECC completed the authorization process for two **SEI-Authorized CMMI Instructors** by **September 2006** and one **SEI-Authorized SCAMPI A Lead Appraisal** by **May 2007**. In addition, the authorization processes for another three SCAMPI A Lead Appraisal are in progress.

As a part of **SECC mission** to promote and support the development of the software industry by improving the software engineering practices to higher maturity level and to achieve strong presence in the international market, SECC took the initiative to introduce the **SEI Team Software Process (TSP)** and **Personal Software Process (PSP)** to the Egyptian community. To be able to deliver **SEI TSP/PSP Services**, SECC extended the partnership with SEI to include the delivery of five **TSP/PSP Courses** and **TSP Coaching Service** world wide except USA. By **July 2006**, SECC completed the authorization process for two **SEI-Authorized PSP Instructors** and during **May 2007** SECC successfully completed the authorization process for two **SEI-Authorized TSP Coaches**.

The new partnership gives SECC the permission to deliver the following courses:

- TSP Executive Seminar (*1-day seminar for senior managers and executives*)
- Managing TSP Teams (*3-day course for project managers and team leaders*)
- PSP for Engineers I (*5-day course for software engineers*)
- PSP for Engineers II (*5-day course for software engineers*)
- Introduction to PSP (*2-day course for non-software engineers*)

These courses provide a **complete image** to a full software development team including project managers, team leaders, software engineers, hardware engineers, quality assurance representatives, testers and technical writers. The executive seminar helps the senior managers and executives understand the **benefits of applying TSP/PSP** in their organizations.

After passing the training track, each team receives a TSP preparation workshop to introduce the whole team together to the TSP Launch process followed by direct implementation of TSP/PSP concepts in a real life projects. **Utilizing TSP/PSP** can help organizations, at any maturity level, **accelerate CMMI-based process improvement** to **less than half the average time** usually required. The full and tailored implementation of TSP/PSP is guided by an SEI-Authorized TSP Coach.



## Important SECC Events

*By: Sherine FaragAllah*

The first half of year 2007 witnessed a series of activities and events organized by SECC in line with its mission of supporting and promoting the software industry in Egypt as follows:

### **SEI- Team Software Process (TSP) / Personal Software Process (PSP) Orientation Session**

On **January 29<sup>th</sup>, 2007** SECC held an orientation session on the "SEI- Team Software Process (TSP) / Personal Software Process (PSP)" at SECC premises located at the Smart Village on Monday 29th January, 2007.

The event targeted the executives and general managers of the IT companies in Egypt. SECC consultants gave an overview on the TSP/ PSP, their relation to CMMI and the return on investment.

### **Microsoft Visual Studio 2005 Program Launching Event**

**April 11<sup>th</sup>, 2007** witnessed the Launching Event of SECC collaborative project with Microsoft for providing Microsoft Visual Studio 2005 for 50 SMEs. During the event, Dr. Hossam Osman, SECC Project Manager, presented the "Training & Implementation Plans" to the audience and which are supposed to start on May 6th, 2007. The event was held at SECC premises at the Smart Village and was attended by 31 specialists belonging to the participating companies.

### **GITEX Riyadh 2007**

In **April 2007**, SECC participated for the first time in GITEX Riyadh 2007 as part of its new approach to export its expertise and services to the Middle East and Africa.

Dr. Gamal Aly, SECC Chairman headed the delegation which consisted of SECC & ITIDA staff and 22 Egyptian companies which were sponsored by ITIDA.

### **CMMI Program Kick Off Meeting**

SECC organized a kick- off meeting on **May 13<sup>th</sup>, 2007** attended by nine software companies. These companies participated with SECC last year in the first round of the "SPI for SMEs" project and are currently seeking to be appraised to CMMI.

The project phases, plan, expected commitments of involved parties and the assigned consultants were presented.

### **CMMI Appraised Companies Ceremony**

**May 22<sup>nd</sup>, 2007** witnessed a ceremony organized by SECC in recognition of the companies that have been appraised to different maturity levels of the Capability Maturity Model Integration (CMMI).

The ceremony was attended by H.E. Dr. Tarek Kamel, Minister of Communications & Information Technology, a number of prominent

figures in the IT field, as well as MCIT, SECC and ITIDA staff.

## **SPI for SMEs 2007 Program**

A kickoff meeting for the "SPI for SMEs 2007 Program" was held on **May 29<sup>th</sup>, 2007** at SECC premises. The program targets 20 SMEs offering a wide range of products and services and aims at providing the best practices pertaining to the software process improvement for 20 small & medium software enterprises.

The Terms of Reference (TOR) of the project was published on SECC Website. Around 30 Software companies submitted their "Application Forms" , whereby 20 of which are to be selected based on the selection criteria mentioned in the TOR.

The 20 beneficiary companies will receive the following during the 7-month period of the project:

- SPIG Handbook in Hardcopy and Softcopy format
- SPIG Process Map in Hardcopy and Softcopy format
- SPIG Interactive Process (Softcopy)
- Training Track on each process
- Onsite Consultation & Customization Services

## **Biography**

**Sherine FaragAllah** is a Business Development Specialist at SECC.

## **Feedback Contacts**

Feedback, comments and questions are appreciated by the author.

Email:

[sfaragallah@mcit.gov.eg](mailto:sfaragallah@mcit.gov.eg)

# **From CMMI ML2 to ML3 in One Year!**

## **ML2 Process Areas Expectations**

*By: Ahmed Abd El Aziz*

Many companies are ambitious to move from CMMI ML1 to CMMI ML3 in one year. They think that it is a matter of more effort they have to do in this year to obtain the required level. Ambition is good, but when it is unrealistic, it becomes a real problem. People either get depressed when they fail to achieve what they planned to, or if they did a "hard work" and achieved their target, most probably they will not be able to continue with this "hard work" attitude after the appraisal and they will go back soon to ML1 – ad hoc.

This article and the next articles are trying to highlight some of the implications of ML3 to help management take the right decision. This article will focus on how the results of applying ML2 process areas are expected to be more mature in case of appraising against ML3. The coming articles will focus on the additional work required in ML3.

### **Using Standard Process**

In ML3 is the process must be Defined while ML2 is the process must be only Managed. Defined means that the organization has a standard process and every project tailors this process according to tailoring criteria defined in the organization. This concept applies to all process areas even those of ML2.

For example it is accepted in ML2 that if there are many projects, every project manager may use a different tool to manage his project. Even the same project manager may use a different tool to manage each of his projects as far as every project has his

own process or his own plan that permits this. In ML3 it is expected that all projects use the same project management tool as they all belong to the same organization.

Another example is the WBS and schedules of the projects. In ML2 it is expected that every project has a WBS and a schedule. In ML3, it is expected that all the WBSs and schedules of all projects show the same phases, tasks, and work product names. If a project calls the planning phase "Planning" and another project calls it "Project Planning", there is no problem with ML2 but there is a problem with ML3.

### **Using Historical Data in Estimation**

When appraising a company in ML2 or ML3, the estimation must exist. However in ML2 appraisal, the estimation could be based on expert judgment only and is not based on historical data. In ML3 it is expected that the organization has built a historical database from the past projects that shows the actual performance and productivity rates inside the organization and these data is used in estimating the new projects.

Building this historical database is not difficult in itself. The problem is that it needs time to be built. Not every project could be used in building the historical database; the project must have followed the process in place. To clarify assume that an organization had never made design before. They used to move directly from requirements to coding. Their productivity rate is X. Now they are getting more mature and their process

says that they must design the solution before coding. How can they use X as a productivity factor? A complete phase is not there.

Another example is when an organization is not used to formally review its work products. Now after getting more mature, they formally review the plan, requirements, analysis, design, code, and test cases documents. Once again how can they use the historical productivity rates where these activities were missing?

### **Monitor and Control the Projects in a Consistent Way**

In ML2 organizations, there is usually great variation in the way the project managers monitor and control their projects. Even the same project manager uses different approaches with the different projects he manages. The most clear example is the frequency of the periodic monitoring activities. Sometimes it is every week, sometimes it is biweekly, and sometimes it is monthly.

When the project manager is asked about the frequency and why it is monthly instead of weekly, he says "that what fits with my project" without having a clear criteria that explains why the frequency is as he chosen. This is accepted with ML2. However when appraising an ML3 organization, it is expected that the maturity is higher and there a standard period or clear criteria defined, understood, and followed by the project managers in identifying the monitoring and control period.

### **Obtain Measurements In a Consistent Way**

In ML2, projects obtain measures. Even if all projects obtain the same measures, the mechanism they collect

and analyze the measure usually differs from one project to another. Let's take the schedule variance as an example. One project considers this as the largest variance between the planned finish date and the actual finish date of the tasks that already finished. Another project considers this as the difference between the planned finish date of the project and the prospected finish date of the project. Both are called schedule variance and both are valid from CMMI point of view.

In ML2 there is no problem. In ML3 it is expected that all projects collect the measures the same way. In ML3 we can easily compare the schedule variance of different projects to each other. To do this they must follow the same technique to collect, report, and analyze the measures. Otherwise we are comparing orange to apple.

### **Obtain Organizational Measures**

In ML3 it is expected that the organization has some measures collected on the organizational level, not only on the project level as in ML2. There is no specific requirements about what to be measured, but as it is looked to ML3 organization as a more mature organization, it is expected that the organization has something to measure and analyze on the organizational level even it was the average of the project measures.

### **Consistent Change Management**

All ML2 organizations have a change management system. In the early days of using that system, there is always some sort of inconsistency between the different projects within the same organization in following the change management system.

One clear example is the approval on the change requests. Sometimes the

change request is approved by the CCB (Change Control Board) and sometimes it is accepted by only the project manager. What is considered as a major change in one project and needs the CCB approval is considered a minor change in another project and could be approved only by the project manager.

In ML2 it is accepted that every project has its own criteria in identifying what is a major change request and what is minor. In ML3 it is expected that the organization has a standard definition of the criteria of ranking the change requests and all the projects use this criteria in ranking their change requests.

### **Configuration Items Identification**

In ML2 organization there is a common problem that there is no standard file structure and file naming conventions in the organization. What is called "Project Plan" in one project is called "PMP" in another project. Even within the same project, the series of status report files usually have different naming styles. For example some reports names follow the convention

"PSR-dd-mm-yyyy" and some others follow the convention "dd-mm-yy-Project Status Report".

The weight of such miss is minimal in ML2. However in ML3, it is expected that the organization is more mature and there is a standard naming conventions and a file structure that is defined on the organizational level and followed in all the projects within the organization.

### **Conclusion**

The magic word in this article is CONSISTENCY. All ML2 practices are expected to be consistent across all the projects in the organization when the organization is appraised against ML3. This consistency needs time to be established. People resist the change. In ML2 they are enforced to follow a process – their own process. In ML3 they are enforced the organization process. This is much harder and needs more time and a lot of patience to be achieved.

The following table summarizes the expectations mentioned in this article when appraised against ML2 and ML3.

	<b>ML2 Expectations</b>	<b>ML3 Expectations</b>
<b>Process Definition</b>	Every project can have its own process and its own implementation.	All projects follow the same process in a consistent way.
<b>Estimation</b>	Most probably historical data is not used as an input to the estimation.	Estimation is based on historical data.
<b>Project Monitoring and Control</b>	Every project has its own mechanism and frequency in monitoring and controlling the project.	All <u>similar</u> projects are monitored and controlled with the same technique and in the same frequency.
<b>Measurements and Analysis</b>	Every project can have a different set of measures that are collected and analyzed in different techniques. No organizational measures are collected.	All <u>similar</u> projects collect the same measures and analyze them with the same the same technique. Some organizational measures are collected and analyzed.

<b>Change Management</b>	Every project has its own criteria in ranking the change request and approving it.	Change request ranking and approval mechanism is defined on the organizational level and all projects follow it.
<b>Configuration Items Identification</b>	Every project has its own file structure and its own file naming convention schema.	All <u>similar</u> projects have the same file structure and naming convention schema.

## Biography

**Ahmed Abd El Aziz** has over 11 years of experience in the software industry. During that period he worked as developer, project manager, department manager, and process improvement leader. He joined SECC in the last year as a Senior Software Process Improvement Engineer. During that period he offered consultation on CMMI to many software development and implementation companies to maturity levels two and three. He also participated in many SCAMPI A appraisals beside many pre appraisals and gap analysis activities. Ahmed is certified PMP since May 2005 and is Candidate SCAMPI A Lead Appraiser and Candidate SCAMPI & C Team Leader since April 2007 and is expected to be Authorized SCAMPI A Lead Appraiser by the first quarter of 2008.

## Feedback Contacts

Feedback, comments and questions are appreciated by the author.

Email:

[aaziz@mcit.gov.eg](mailto:aaziz@mcit.gov.eg)

# Software Testing Techniques Series: Pairwise Testing Technique

By: Omar Kamal

## Abstract

**Keywords:** Testing, black-box testing, testing tools

In this article, a simple black-box testing technique called pairwise testing will be discussed. In general when the black box testing approach is used for validating that the software is meeting its stated requirements we don't care how the software is constructed.

The focus is examining the behavior of the system against its intended functionalities. Accordingly, most of the effort is directed to analyze inputs to the system and outputs from the system.

## Introduction

Usually, the system behaves according to each of the system inputs in a certain way. However, defects are not manifested in the typical combination of inputs that the developer or the user usually thinks of. Accordingly, the test designer job is to figure out those test cases and to implement them. The optimum testing is to exhaustively test all combination of inputs.

In figure 1, we will consider a trivial system for the sack of illustrating this black-box technique. The system takes 4 binary inputs (A, B, C, and D), processes them and produces a certain output O1 based on those inputs.

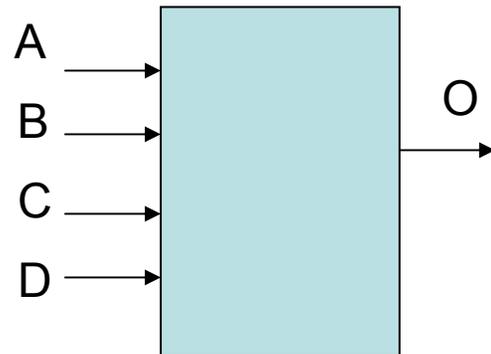


Figure 1: Sample System

Trying all combinations of inputs is the most comprehensive testing technique that may jump into the test designer mind. In our example, we have 16 combination of inputs stated in figure 2 and it is widely known as the "Truth Table".

Test Case ID	A	B	C	D
1	0	1	1	0
2	0	1	0	1
3	0	0	0	0
4	0	0	0	1
5	1	1	1	1
6	1	1	1	0
7	1	0	0	1
8	1	1	0	0
9	0	0	1	0
10	0	1	0	0
11	0	0	1	1
12	1	0	0	0
13	1	0	1	1
14	0	1	1	1
15	1	1	0	1
16	1	0	1	0

Figure 2: Exhaustive Testing for 4 binary inputs

Except for simple toy problems, exhaustive black-box testing won't be practical. For example, if system inputs grows to become 10 inputs that takes 4 logical values instead of two, the number of exhaustive test cases will grow in a nonlinear way to reach 410 (1048576) test case.

If the whole software testing life cycle (design, implementation, and execution) is automated then exhaustive may still be practical for small-to-medium problems. But it is certainly impractical to exhaustively test large system using such approach.

## Pairwise Testing

All-pairs testing or pairwise testing is a combinatorial software testing method that, for each pair of input parameters to a system (typically, a software algorithm) tests all possible discrete combinations of those parameters. Notice the difference between "all the combination of all inputs" and "all the combinations of pair of inputs". Bugs involving interactions between two parameters can be caught with all-pairs testing. The following table states 5 test cases that satisfy the criteria for pairwise testing instead of 16 test cases that exhaustively mentioned earlier.

Test Case ID	A	B	C	D
1	1	0	0	0
2	1	1	1	1
3	0	1	0	1
4	0	0	1	1
5	0	1	1	0

Figure 3: Pairwise testing for 4 binary inputs

If you pick for example (variable A), and fix it to zero and start looping over other variables you can see the following:

- For variable B, test case 4 covers the "0" case and test case 5 covers the "1" case.
- For variable C, test case 3 covers the "0" case and test case 5 covers the "1" case.
- For variable D, test case 4 covers the "1" case and test case 5 covers the "0" case.
- Then if we fix (variable A) to "1" and start looping over other variables you can see the following:
- For variable B, test case 1 covers the "0" case and test case 2 covers the "1" case.
- For variable C, test case 1 covers the "0" case and test case 2 covers the "1" case.
- For variable D, test case 1 covers the "0" case and test case 2 covers the "1" case.

Accordingly, you can see that all combinations of variables values were examined with relation to variable "A". Similarly, it can be demonstrated that variable B, C, and D is covered the same way.

The advantage of pairwise testing is significant for more inputs with multi-logical values. Previously, we mentioned that for [10 inputs with 4 logical representations] the number of exhaustive test cases was (410 = 1048576) test case. If we apply the pairwise testing, we can reach 31 test cases achieving a 99.997 test case reduction.

## Test case development

The algorithm for identifying the minimum number of test cases that achieve the pairwise criteria is beyond the scope of this article. However, there exist a number of tools that compute pairwise test cases and make the life of the software test designer easier. [www.pairwise.org](http://www.pairwise.org) [1] is a very resourceful source of information on

both commercial and free tools that exists on the market.

## References

[1] <http://www.pairwise.org/>.

## Biography

**Omar Kamal**, 11 years of experience in wireless telecommunications, software development, training and software quality management. Working as a Quality Assurance Manager for Mentor Graphics, used to work with Lucent Technologies, Hewlett Packard, IBM, QuickTel 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 Contacts

Feedback, comments and questions are appreciated by the author.

Email:

[omkamal@yahoo.com](mailto:omkamal@yahoo.com)

# Practical Software and Systems Measurement (PSM)

*By: Manal Helmy*

Practical Software and Systems Measurement (PSM) is an objective process for analyzing project issues, risks, and financial management, focusing on software project management. PSM provides government and industry managers with objective information for making sound decisions and for meeting cost, schedule, and technical objectives.

It is based on actual DoD, government, and industry experience, representing the best practices in the software and engineering communities. While the PSM process includes common issue areas and recommended measures, the process is completely flexible, to adapt to project-specific requirements and objectives.

## PSM Insight

PSM Insight is a PC-based software tool that automates the Practical Software and Systems Measurement (PSM) process. PSM Insight includes tailoring, data entry, and analysis functions to develop a project-specific software measurement tool. While PSM Insight provides templates of commonly-used issues and measures, it is also completely flexible to customize analysis to project-specific needs.

PSM Insight provides many benefits:

- Customization to project-specific needs
- Templates of commonly-used issues, measures, and indicators from best practices
- Insight into key software issues
- Objective data for informed decision making

- Identification of potential problems and solutions
- Assistance in meeting cost and schedule objectives
- Flexible data definitions and analysis tools
- Presentation-quality graphs and reports
- Support for risk management of software projects
- Ease of use in tracking complex projects

PSM comes with a comprehensive sample project data set that can be placed on disk drive when you install PSM Insight. The sample database shows how to tailor a project, navigate the data entry grid, create indicators, and analyze data. You can explore at your own pace without accidentally altering live data.

The sample project includes fully-tailored issues, measurement categories, and measures, as well as several hundred data records. You can work with the project, modify the data, import data, view graphs, and specify indicators.

## Tailoring

Tailoring is the first step in the Practical Software and Systems Measurement process, when you define your specific project tracking needs. Tailoring activities include:

- Identifying and prioritizing project issues
- Selecting and specifying software measures

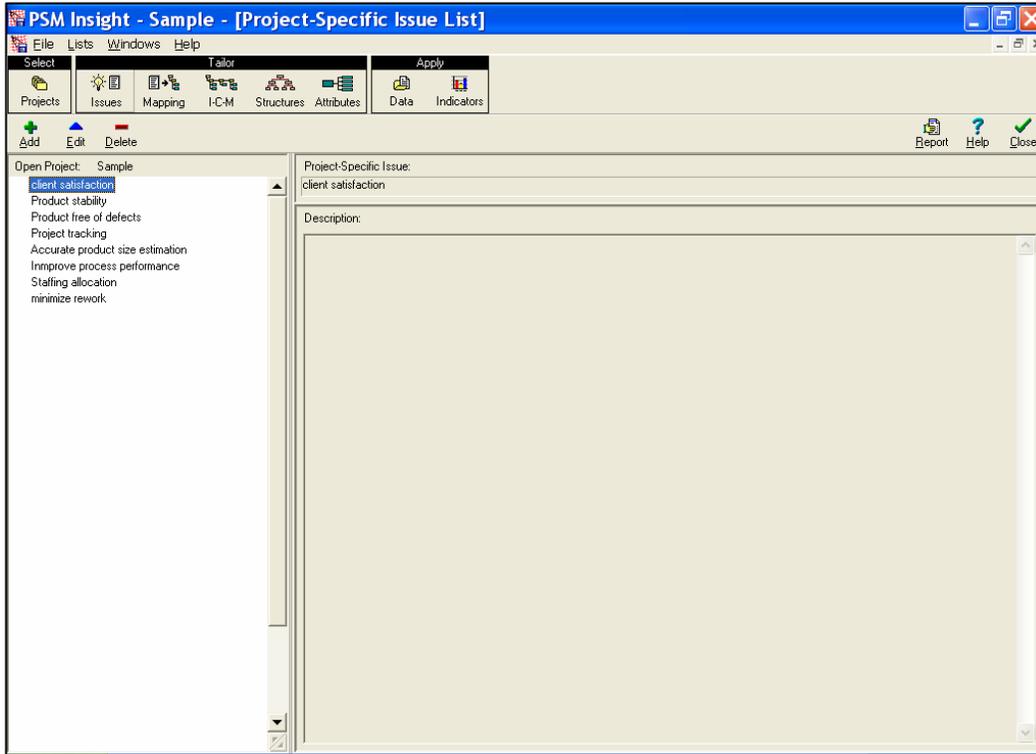


Figure 1: Sample project specific issues

## Issue

An issue is a critical aspect of a project that should be tracked and monitored. An issue may be a risk, potential problem, constraint, concern, or objective of a project. Issues usually describe project resources, progress, quality, or performance.

## Common Issue Areas

A common issue area is a PSM issue that is tracked in most AIS or weapons systems projects. A common issue area is also one of the most important or critical aspects of a development project. PSM Insight includes seven common issue areas that are pre-defined for your selection:

- Schedule and Progress
- Resources and Cost
- Product Size and Stability
- Product Quality
- Process Performance

- Technology Effectiveness
- Customer Satisfaction

These seven common issue areas are used as a starting point for your tailoring work.

## Measurement Category

Each issue in PSM Insight includes one or more measurement categories, further clarifying the information that you monitor. A measurement category is a set of related measures that can be tracked in your project, such as cost-related or schedule-specific measures. Each category answers different types of software-related questions. Many commonly used categories are pre-defined for your use.

## Template

A template is a pre-defined element in PSM Insight. When you tailor or define

your development project, PSM Insight offers templates for the 7 most common issue areas, together with their measurement categories, measures, structures, and attributes. Templates are developed by the PSM Team, reflecting the best practices of DoD and industry managers.

Templates are handy tools because they reduce the time it takes to tailor a project, and they provide guidance on the most important items to track in your project.

The screenshot shows the 'PSM Insight - Sample - [Problem Report Status]' window. It features a menu bar (File, Lists, Windows, Help), a toolbar with 'Select' and 'Apply' tabs, and a data table. The table has columns for Date, Configuration Item, Planned/Actual, PR Priority, PR Category, PR Status Code, Number of Problem Reports, Number of Problem Reports Resolved, Average Age of Problem Reports, and Average Time to Resolve Problem Reports. The data rows are as follows:

Date	Configuration Item	Planned/Actual	PR Priority	PR Category	PR Status Code	Number of Problem Reports	Number of Problem Reports Resolved	Average Age of Problem Reports	Average Time to Resolve Problem Reports
1/1/2004				Interface	Closed	100	25		
1/1/2004				Performance	Open	5	1		
1/2/2004				Performance	Open	10	2		

The status bar at the bottom indicates 'Name of a highest level component.' and 'Records: 3'.

Figure 2: Sample problem report status

## Tailoring Report

Tailoring reports are designed to help you in the tailoring process and to help you share information about your project with other staff. PSM Insight provides customized project-specific reports, and several PSM Guide Template reports. Each report can be generated as a summary report or a detailed report with lengthy descriptions.

## Indicator

An indicator is a measure or a group of measures that provides insight into a

project issue. Indicators frequently make comparisons between two values, such as planned and actual values. An indicator may be called a metric in other software measurement tools.

An indicator, as shown in the following figures, is usually a graph or table. Graphs may depict comparative data, trends, or a snapshot of a project's current status, based on your specific tailoring data. All calculations are done by PSM Insight without user intervention.

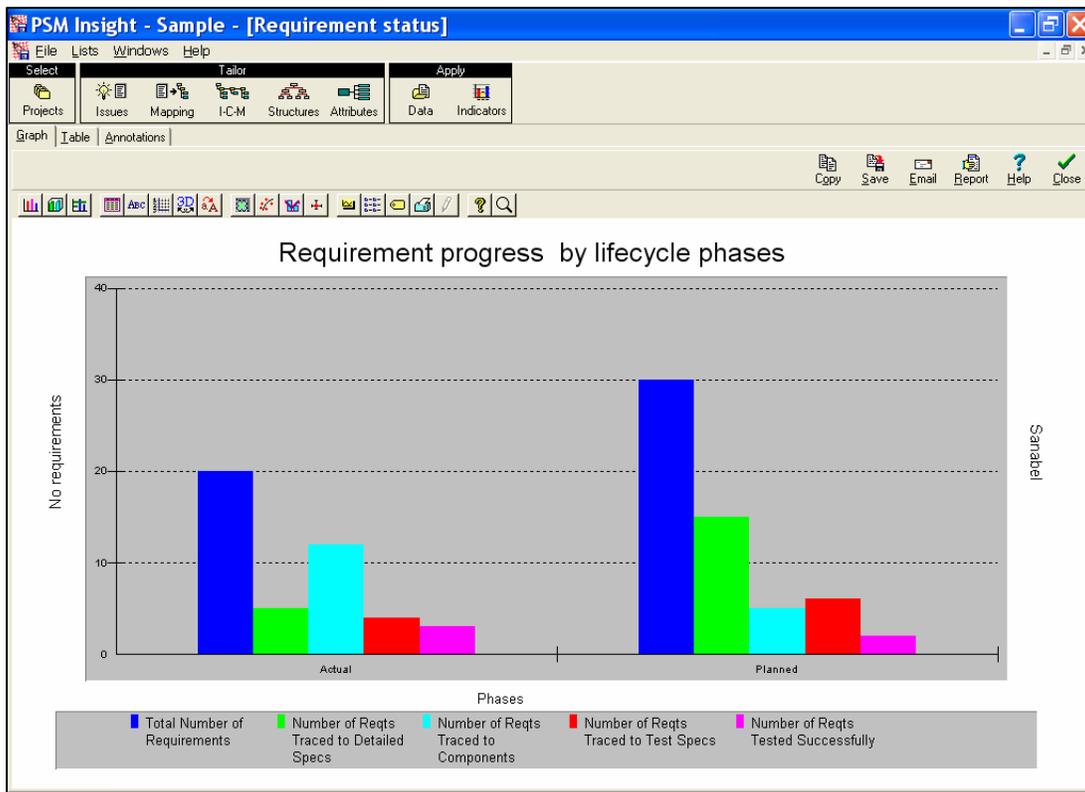


Figure 3: Sample requirement indicator

## Biography:

**Manal Ahmed Helmy**, is a Software process improvement consultant at SECC. She got her B.Sc. in Computer Engineering from the Faculty of Engineering, Ain Shams University, Cairo (Egypt). She has over 15 years of experience in information technology. She has the "NCC" certificate in structured design and analysis and experience in quality management systems. She is a certified quality analyst "CSQA". Provide Consultation for Process improvement initiative based on SE/SW-CMMI. Participating in the projects for process improvement in Egypt helping the companies in getting CMMI certification through gap analysis, consultation visits and formal SCAMPI appraisal class "A". She is an SEI authorized instructor for "Introduction to CMMI", Six sigma

certified green belt and preparing for the black belt.

## Feedback contacts

Feedback, comments and questions are appreciated by the author.

Email:

[mhelmy@mcit.gov.eg](mailto:mhelmy@mcit.gov.eg)

# Software Process Improvement

*By: Ahmed M. Hammad*

In this article, the author highlights a software process improvement approach that works effectively in his opinion. Next, he will highlight the importance of having central knowledge base to store organization assets. Finally the IDEAL methodology is briefed.

## **SPI in a nutshell**

For Software Process Improvement (SPI) new comers, process improvement seems to be confusing. In the following paragraphs I will try to explain process improvement intent and rational behind it.

Any software Company has 3 basic elements that work together to develop software products. They are People, tools and processes. People use tools in a certain way to develop products. People need leadership, ethical management, and rewarding. Tools require proper selection and training to be used effectively. The remaining part is the way people collaborate together to plan and execute development projects. The way people collaborate and use tools is called processes. Software process improvement focuses on improving the management procedures used by team members to manage work tasks and the way and order tools are used.

Still confused! Process improvement can be called management practices improvements. As many people believe they lack proper management to their human and non-human resources. SPI helps you learn how to manage your team members, projects and tools in an organized and scientific way.

## **First Steps in SPI:**

Every company has processes, even if large parts of it are not written and not consistently followed. Regardless of your organization size and characteristics, if you have developed products, then you have used some processes to develop it, even if known verbally.

The first days of process improvement should focus on discovering the current way you manage your projects, whether they are good or bad. Start by documenting your organization structure, job description, responsibilities, the processes/procedures used in the company, and any assumptions in action. Now you have something to start improvement efforts.

Second step is to focus on your business needs and critic your system. Document your current problems, challenges, advantages, disadvantages, and areas you believe you should improve.

Now you have basis to start with and rational behind your SPI activities. Trying to ignore your company culture, strengths, weaknesses and experience in doing software development in favor of a suggested magic solution that will overnight convert you to a mature company is unrealistic.

The idea is to discover your system, document it, and retain your strengths. Study your current problems and your business needs. Learn from other's method of doing development, and then find ways to improve your weaknesses.

What if you reuse ready made process descriptions to move faster and use a reliable quality system that is proven to be perfect? Sure you will start fast but in the wrong road. It is not yours, so the gaps between what is been already documented and the actual practices will be widened. Even if you enforce team members to follow it, they will find ways to avoid it.

## **And what is next in SPI?**

Most of the companies, probably your company have the foundation of project management. So treat software process improvement as high priority project. Define requirements (i.e. what to be achieved), tasks, assign resources, and define budgets. Then track tasks completion and cost. Make the project as a 6 months project, and at the end of the project evaluate your position and plan for the next 6 month project.

Software process improvement is a continuous effort, so the 6 months projects will never end. You should move from initial software process improvement to CMMI level 2 then to higher levels. The only level that you can stay at is level 5. Level 5 is the "Optimizing level", which means continuous process improvements.

Start by documenting your best practices that work great with you, the current processes and procedures that you use, then document current organization structure, job descriptions, and document all templates used in your company. It is easy to create Plan templates from the best plan you have.

Define clearly what is good in your organizations and what works bad, what are the future challenges you have to meet. Define your business goals and how they will affect your current processes.

Make 3 or 4 iterations per project, which is 6 months time frame. In all iterations, follow the IDEAL model. Although there are several improvement models that you can use such as PDCA (Plan, Do, Check, Act), PDSA (Plan, Do, Study, Act) or Six Sigma DMAIC (Define, Measure, Analyze, Improve, Control), I suggest IDEAL for its comprehensive guidance.

## **Organization Repository**

If you don't have a content management system hosted in your current Intranet, you should have one very fast.

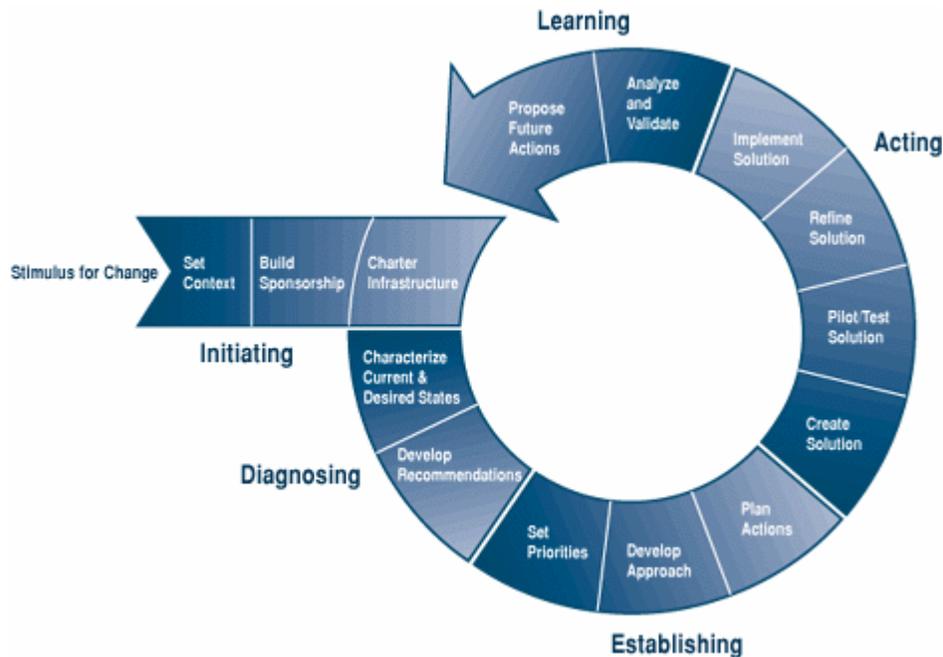
Many organization experience, culture, knowledge are stored in the memories of team members and on many emails stored on personal computers. Some have file server, but have never being up to date and are usually incomplete. File server in my experience is not efficient in organizing organization assets. It is inefficient to browse for a certain document and you can't effectively search.

Organization should move from vocal knowledge to written knowledge. When human invented writing, it was the first time people can share knowledge between them even if they have never met. Your organization should have central repository to host all of the company documents, all project artifacts, lessons learned, guidance to current and new employees, and all digital asset. By doing so, you will build a real organization asset that foster team learning, sharing, and reusing across your organization.

Happily there exist many open source and commercial systems, selecting which system to select is a critical task and beyond the scope of this article.

## IDEAL brief description

Here is a brief about IDEAL, for full description look at:  
<http://www.sei.cmu.edu/ideal/ideal.bridge.html>



IDEAL is a model for continuous improvement. It helps by describing the main steps necessary to establish a successful improvement program.

IDEAL is acronym for Initiating, Diagnosing, Establishing, Acting, and Learning.

**I – Initiating:** Laying the groundwork for a successful improvement effort.

- **Stimulus for change:** Change for the sake of change rarely results in significant improvement, so state business reasons for process improvement. When the business reasons for change are more evident, there is greater buy-in throughout the organization and there are greater chances for success.

- What benefits (such as return on investment or improved capabilities and morale) will result?

- **Sponsorship:** It is necessary to maintain sponsorship levels throughout an improvement effort. Sponsors can be most effective if they give personal attention to the effort and stick with it through difficult times.

**D – Diagnosing:** Determining where you are relative to where you want to be.

- **Characterizing** the current and desired states is similar to identifying the origin and destination of a journey. A good starting point is "stimulus for change" activity in Initiating phase.

- **Develop Recommendations**

**E – Establishing:** Planning the specifics of how you will reach your destination.

- Set Priorities
- Develop Approach
- Plan Actions

**A – Acting:** Doing the work according to the plan.

- **Create Solution:** Write the new process descriptions (or the modifications proposed), templates and procedures that are best guess solution.
- **Pilot/Test Solution:** Try executing the process on a pilot project to make sure of applicability.
- **Refine Solution:** Once the pilot done, refine per the lessons learned in the Pilot project.
- **Implement Solution:** You are ready now to apply the new process changes organization wide.

**L – Learning:** Learning from the experience and improving your ability to adopt new technologies in the future.

- **Analyze and Validate:** In what ways did the effort accomplish its intended purpose? What worked well? What could be done more effectively or efficiently? Lessons are collected, analyzed, summarized, and documented. The business needs identified during the initiating phase are reexamined to see if they have been met.

- **Propose Future Actions:** During this activity, recommendations based on analysis and validation are developed and documented. Proposals for improving future change implementations are provided to appropriate levels of management for consideration.

Note: many statements are just copied from "The IDEAL<sup>(SM)</sup> Model: A Practical Guide for Improvement" at <http://www.sei.cmu.edu/ideal/ideal.bridge.html>

## References

- <http://www.sei.cmu.edu/ideal/>
- <http://www.sei.cmu.edu/ideal/ideal.bridge.html>
- IDEAL: A User's Guide for Software Process Improvement at <http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.001.html>
- <http://www.jspwiki.org>

## Biography

**Ahmed M. Hammad**, Senior Software Process Improvement Engineer in SECC. Before joining SECC, he was working as a software manager in the Research & Development Dept. in the Egyptian Telephone Company "QuickTel". He has experience in software development and management for more than 13 years, in desktop, web based and embedded development.

## Feedback Contacts

Feedback, comments and questions are appreciated by the author.

Email:

[ahammad@mcit.gov.eg](mailto:ahammad@mcit.gov.eg)