Egypt–SPIN Newsletter

Issue 6, Apr. – Jun., 2004

Sponsored by SECC

From the Editor (*Ahmed El Shikh*)

Welcome to our 6th 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.

> First of all, we would like to congratulate **ITSoft Company** and its team about their big achievement of reaching **CMM level-3** after a record time of becoming CMM level-2. We hope that this milestone becoming a sign - and motivation - to other companies that achieving the CMM dream can be in hand by just serious and loyal efforts.

In this issue we tried to record some of the most important events that happened inside our community (1st and 2nd articles), share real life experience (1st and 3rd article) and discuss some hot topics in software engineering, software project planning and software quality (4th, 5th and 6th articles respectively).

Eng. Sameh Zeid shares with the community **ITSoft's journey from CMM Level-2 to CMM Level-3.** The article explains how ITSoft managed the transition from level-2 to level-3 in a record time.

Eng. Omar Kamal is writing about the **Software Sizing and Estimation** workshop developed by David Consulting Group Inc from 31 May to 3 Jun. and sponsored by SECC in collaboration with USAID/ICT. He summarized the explained methodology and its benefits.

Eng. Mona Arishi talking about her experiences in the **Software Process Improvement Implementation** inside IBM's Cairo branch. She discusses the planning, implementation and pitfalls. Also, enjoy a life presentation that you can download form SECC/SPIN website for more details.

Eng. Madiha suggests a critical topic for discussion about the suitability of **Automated Testing Tools** for the maturity level of the software process. She talks about what should be automated and what shouldn't, the prerequisite infrastructure, and return of investment in automated testing.

Eng. Mohamed A. AbuSen is writing about the **Estimation** in the software project planning. What is its definitions, what to estimate and who can do it? Also, some problems associated with the estimation process had been mentioned and how to prepare contingencies.

Eng. Ahmed El Shikh discusses the mangers situation inside SMEs that has no well defined measuring program. How much the **Software Measurement** is important, what to measure, practical roadmap of success, traps to be avoided, tips and long term benefits.

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 el_shikh@sas-sys.com

Table of Contents

Moving from CMM level-2 to CMM level-3

By Sameh S. Zeid

1. Introduction

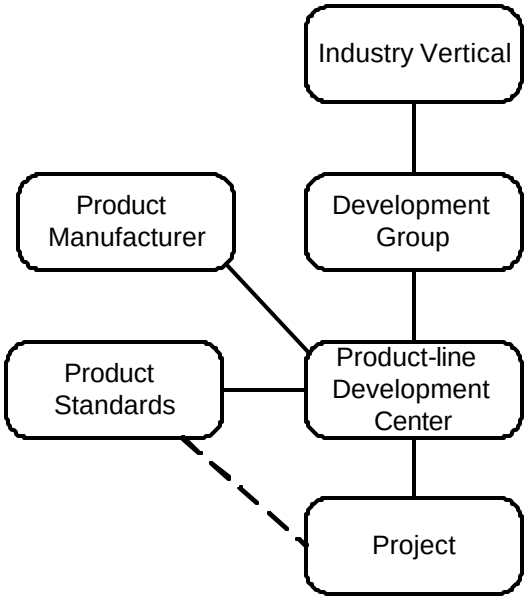
ITSoft has recently achieved CMM level-3 making the transition from CMM level-2. When assessed at level-2, ITSoft had incorporated many of the requirements of level-3 KPAs of CMM. This has paved the way to move to level-3 as a manageable project in record schedule.
This article discusses the transition from level-2 to level-3 from the perspective of what needed to be done, main challenges and project management consideration.

2. Background

Upon achieving level-2 early May/2004, ITSoft has already satisfying many of the requirements of level-3. Taking advantage of this situation, it was decided to make another round of formal assessment for CMM level-3 on mid June, 2004. The work on level-3 was done on parallel while level-2 project was at its final stages. This was possible based on the fact that many of KPAs at level-3 address organization concerns rather than being project related.

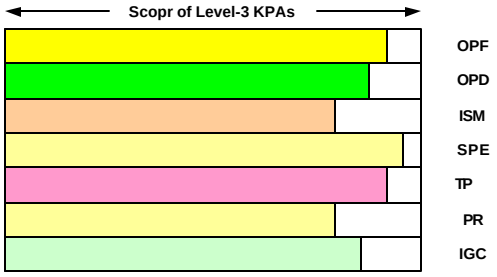
3. ITSoft Business Structure

ITSoft is formed of multiple development groups representing different vertical industries (e.g.: Core Banking, Delivery Channel, Trade Finance, Higher Education Management, and others). Every group provides software development and customization services around a core product –line supplied by a market leader supplier of the product.



4. Base-line of level-2

The extent of satisfying CMM level-3 requirements upon completing CMM level-2 in early May/2004 can be represented as follows:



The baseline for level-2 at ITSoft had the following components from level-3:

4.1. Organization Process Focus (OPF)

When CMM initiative was chartered early September/ 2003, the charter included the establishment of SEPG.

SEPG was headed by full-time member and many other part-time members, in-addition to all organization worked as implementers of processes generated at SEPG.

4.2. Organization Process Definition (OPD)

The CMM level-2 project has itself produced a documented OSSP that is shared across the whole organization through implementation of Software Process Database (SPDB).

4.3. Inter-Group Coordination (IGC)

Though we did not have a document or process manual stating the details for IGC, the IGC requirements were stated and implemented in an inter-leaved way across various processes. The main implementation of IGC is done during Requirements Management to have consensus across all parties and related groups on any commitment made on behalf of the organization. Also, during Project Planning, a kick-off meeting must be held for any project, where all related groups are invited, they review the Software Development Plan (SDP) and commit on their responsibilities towards the project. During Project Tracking, any change in the requirement involved the participation of all relevant groups and this is tracked through the whole project.

4.4. Integrated Software Management (ISM)

From our business structure, every development group had product-specific standards for software development provided by the supplier of the product. These standards represented Project Software Development Process (PSDP) to be used for any project related to this product-line. This PSDP have been used as the basis for pre-project work, project planning, project tracking and other management based processes.

4.5. Training Program (TP)

Being an ISO 9001:2000 organization, ITSoft had procedures for conducting training programs to leverage the skills of its staffs. These included procedures for course registration, course attendance, course assessment, and planning for training.

4.6. Peer Review (PR)

Peer Review was implemented at ITSoft as a vital means for discovering defects before they leak from one phase to another. Reviews are complementary part of Software Life Cycle of any product-line and were required before baselining any work product.

4.7. Software Product Engineering (SPE)

All technical activities for doing the technical work of the project had expertise and trained staffs to implement. We extensively used on-job training to make our staffs able to perform the technical work. Every product-line had its technical development manual and the specific details of doing micro-level work.

5. CMM Level-3 Project

During the planning for the project and preparation for action plans, certain gap areas have been identified as needing immediate attention. The main challenge was to maintain the same hype and to prevent the natural tendency of involved staffs to relax after we had acquired level 2 in early May/2004. Many stakeholders were under belief that the schedule and objectives are not realistic. In the

following paragraphs we will discuss the key areas of concentration to make quick improvement.

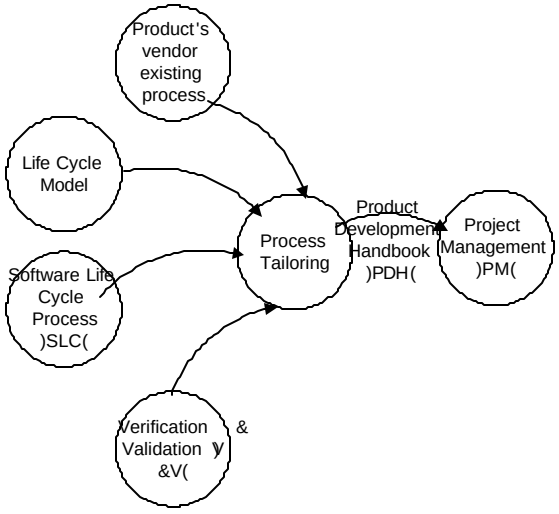
5.1. Integrated Software Management (ISM)

This KPA integrates the management and engineering activities of project. It requires that project planning and tracking activities to be based on a Project Defined Software Process (PDSP) that is based on a certain Software Life Cycle (SLC) model.

We have defined a Product-line Development Handbook (PDH) that was derived based on the following:

- Software Life Cycle Model,
- Organization-wide software development life cycle,
- Standards and specific technical procedures adopted by the product-line, and
- Applying the organization-wide Tailoring guidelines on any relevant OSSP process.

Please refer to the next diagram showing the cycle for developing the PDH.



The PDH has represented the outcome of applying Tailoring Guidelines on the OSSP taking into account the specifics of every product. PDH was developed by practitioners from product-line who have adequate relevant experience. The PDH was checked for compliance with OSSP and was under ownership of SEPG.

The PDH has served the following:

- became a basis for any project planning for any project under this particular product-line,
- accelerated the implementation by having the PSDP ready defined for product-line practitioners,
- have a common language for all stakeholders involved in the product-line development, and
- satisfied the expectation of our partners for using their product specific language and same time met the requirements of OSSP which was already CMM level-3.

5.2. Software Product Engineering (SPE)

The PDH has provided reference on the how-to of the micro level tasks that are involved in the software development projects under the product-line. PDH has served as bible for all tasks related to the product-line and satisfied OSSP requirements for SPE. We have capitalized on the available standards already in place per product-line to reduce learning curve and speed the implementation.

5.3. Peer Review (PR)

The Verification and Validation (V&V) process has defined requirements for reviews. Types of reviews were provided in separated guidelines and

training sessions. PR was considered from time proposal was prepared, before making any client commitment, passing through project planning, project tracking and maintenance. PR has been proven to be costly commitment on behalf of the organization and increases the costs at early stages of project with pay-off at later stage. This pay-off is manifested by reduction of defects during testing. However, in the first few projects the net cost will be higher till staffs involved gained the skills required for doing effective reviews.

5.4. Metrics & Measurements

Metrics were implemented to have basis for decision making and to carry-out improvements in the various facets of a project. These included the following:

Project Status Report

This periodic report included metrics on:

- Schedule Variance
- Effort Variance
- Defects Density
- Efficiency of Reviews in finding defects
- Rework due to defects
- Status of Non-conformities
- Status of Change Requests
- Risks Status

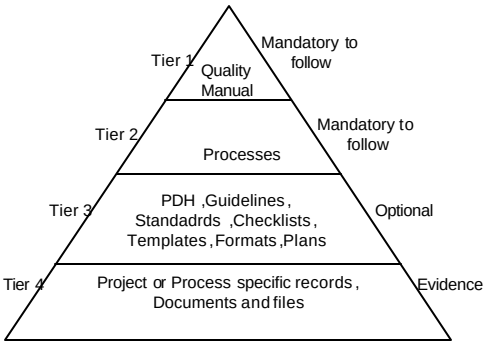
The above metrics and summaries served quantitative tracking of project status. The Project Software Manager had to change their way of work to ensure the possibility of getting these figures. This has served that PSM began actually to implement many of ISM, PP, PTO and RM KPA requirements.

5.5. Organization Process Focus (OPF)

Though OPF is at level 3, it was implemented as the first step in the level-2 project itself. This gave strong ownership for whole SPI initiative from the beginning, and paved the way for enabling many changes to make progress possible. The SEPG guidelines and quarterly plans served the requirements for having planned SPI efforts that have management support.

5.6. Organization Process Definition (OPD)

The structure of OSSP is represented by the following diagram:



Tier 1:
The *Quality Manual* established the following:

- Quality Organization within ITSoft,
- Quality Policy,
- Polices for CMM KPAs of levels 2 and 3, and
- Structure of OSSP.

Tier 2:
This tier included the OSSP processes to be used across the whole organization for various projects. Tier 2 is derived by the higher level policies established at Tier 1.

Tier 3:
This Tier contained the outcomes of applying tailoring guidelines. Also, this Tier contained templates, checklist and other formats that can readily be used by various projects and organization units.

Tier 4:
This Tier has the evidence results from applying various processes and guidelines to projects and organization units within ITSoft.

All data related to the previous Tiers were maintained in the SPDB and accessible to all staffs at ITSoft.

Metrics were introduced for OPD that included the average time to close an OSSP Change Request.

5.7. Inter-Group Coordination (IGC)

We introduced the notion of teams in various critical tasks that require participation from more than one group. Examples of these teams are the following:

- Proposal Preparation Team
- Contract Preparation Team
- SDP Preparation Team

In addition, we enforced the conduct of certain meetings to act as control points for implementing IGC. This may be summarized as follows:

- Project Kick-off Meeting
- Project Status Meeting
- Contract Follow-up Meeting

5.8. Training Program (TP)

Training is a crucial ingredient of every project and it is rare to have a project that does not include Training requirement. The following enhancements were made on existing training procedures:

Training Needs Definition
The needs are derived based on:

- project requirements (which are specified during proposal stage),
- staffs appraisal, and
- Business Plan of the organization.

These needs are source for updating Skills database for the required skills.
Also, these needs were the basis for Training Planning, which is conducted quarterly at the organization level.

Maintain Skills Database
Every staff in the organization has a record of her existing skills, required skills and skills acquired by training.

Course Post Assessment
This served as means to calculate training scores and metrics for evaluation of training. It is used for updating skills database.

5.9. Organization Learning

We introduced quality gates at our OSSP that their implementation has facilitated establishing basis for knowledge management across the organization.

Project Closure Meeting
Project team meet in a free learning environment to agree on lesson learned and update SPDB.

Productivity
Productivity measures are updated through the project. These measures are used for planning new projects.

SPDB Guidelines
SPDB is repository for holding all assets produced by OPD and derivatives from execution of various projects. SPDB is strictly maintained according to guidelines and procedures implemented across the organization.

5.10. Time Sheet System

We have given a code for every activity mentioned at the process, sub-process or task level of OSSP. This has helped to measure the efforts spent in various activities regardless whether they are Engineering, Project Management, or Organization related.

6. Conclusion

CMM level-3 addresses KPAs that are:

- Organizationally related (OPF, OPD, TP),
- Engineering related (SPE, PR), and
- Project Management related (ISM, IGC and level-2).

Organizationally related KPAs can facilitate the implementation of Project Management KPAs. For example, OPF is a key enabler to have clear ownership on SPI initiatives and KPAs implementation. OPD if established at level-2 can serve to have basis for OSSP that supports whole Organization Process Asset Libraries.

ITSoft has been formally assessed at level-3 within 2 months of acquiring level-2. This is primarily attributed to considering level-3 KPAs while implementing level-2.

7. Glossary of Terms

OSSP	Organization Standard Software Process
KPA	Key Process Area
CMM	Capability Maturity Model for Software version 1.1
SEI	Software Engineering Institute
SPI	Software Process Improvement
SDP	Software Development Plan
PSM	Project Software Manager
RCB	Requirements Control Board
PDH	Product-line Development Handbook
PDSP	Project Defined Software Process
OPAL	Organization Process Asset Library
SPDB	Software Process Database

Biography

Sameh S. Zeid is the Program Manager of Process Improvement in ITSoft. He has been involved with software development and management since 1985. Sameh played various roles on the technical and managerial levels for establishing core business solutions. In last one year Sameh has managed Process Improvement Program that accredited the organization CMM level-3. Sameh has a B.Sc. degree in Computer Science from Faculty of Engineering and Post-Graduate degree in Operations Research.

Feedback contacts

Feedback, comments and questions are appreciated by the author.

Author contacts:

E-Mail: szeid@itsoft.com.eg

Customized Software Sizing and Estimation Workshop.

By: Omar Kamal.

June2004, the "Customized Software Sizing and Estimation" workshop developed by David Consulting Group Inc. was carried out on Cairo by Steven Neuendorf. Mr. Neuendorf's experience includes software sizing, estimation, training, mentoring and auditing. He is a member of the International Function Point Users Group (IFPUG) and is a Certified Function Point Specialist.

The course was organized and sponsored by the Software Engineering Competence Center (SECC) in collaboration with USAID/ICT. The attendees came from IBM, Harf, Delta, SAS, MCIT, ICT, DMS, Nozha IT, ITWorx, United Ofoq, Mentor Graphics, Link.Net, Giza Systems, Primasoft, Raya software, EDS, CITC, and Quicktel.

Steven started by explaining what is functional point, stating that the objectives for workshop were transferring function point knowledge and skills to the participants, demonstrating Function Point Analysis through class exercises and case studies and applying Function Point Analysis to improve estimation practices.

It was clear that functional point analysis is useful and beneficial as a:

- A tool to determine the benefit of an application to an organization by counting function that specifically matches requirements.
- A tool to measure the units of software product to support quality and productivity analysis.
- A vehicle to estimate cost and resources required for software development and maintenance.
- A tool to size purchased application package.
- A normalization factor for software comparison.

Steven explained the Function Point History in a detailed matter, which I feel from my point of view wasn't too useful and it could be more squeezed in a couple of slides. Then the pace of the course was steadier and covered the following topics:

The function point types are:
- Development Project.
- Enhancement Project.
- Application.

Function Point Counting Rules.
Types of Function Point Counts.
- Development Project.
- Enhancement Project.
- Application.

Function Points Components (Figure1).
- Data Functions.
- Transaction Functions.
- General System Characteristics.

Function Point Process.
Case Study.

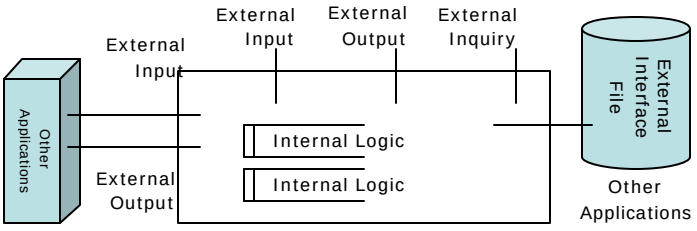


Figure 1: Data Functions and Transaction Functions

At this point counting functional points was clear, but there were more issues that rose about how to include some factors that aren't directly tied to the size, but can affect the project size indirectly. Steven showed that those factors can be mapped to what is called the "General System Characteristics" which were categorized as follows:

- Data communications
- Distributed data processing
- Performance
- Heavily used configuration
- Transaction rate
- On-Line data entry
- End-user efficiency
- On-Line update
- Complex processing
- Reusability
- Installation ease
- Operational ease
- Multiple sites
- Facilitate change

Steven asked the attendees about their preference in choosing one or two of different application types to study further. Web applications, Use cases, and applications with extensive Graphical User Interface (GUI) interaction were selected to discuss further advanced functional point analysis.

I got the feeling that I knew how to calculate the functions points but I still didn't recognize how to use those counts to ultimately estimate cost, time, and effort for software projects. At that time Steven started talking about building estimated based on functional point sizing. The model for estimation is abstractly conceptualized in figure 2.

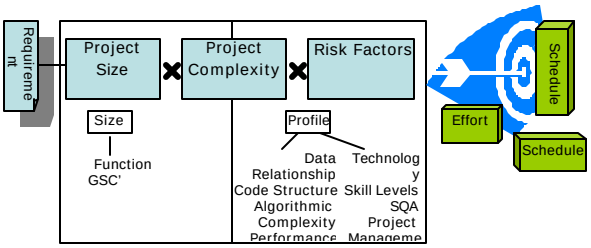


Figure 2: Project Estimation Process.

The following steps summarize the project estimation process:

- After computing the project size using function points, which represent the project definition.
- The job now is to determine the project complexity, the delivery rate, and possible risk factors.
- Complexity factors include the following:
  - Logical and mathematical algorithms.
  - Code structure.
  - Data relationships.
  - Reuse.
  - Memory.
  - Security.
  - Warranty.
  - Data field size (number).
- Risk factors are calculated and included in the big sum. Risk factors include the following:
  - Technology applied such as tools, languages, reuse, and platforms.
  - Process/Methodology including task performed, reviews, testing, object oriented.
  - Customer/User and Developer skills, knowledge, experience.
  - Environment including locations, office space.
  - System type such as information systems, control systems, telecom, real-time, client server,

scientific knowledge-based, web.
- Industry such as automotive, banking, financial, insurance, retail, telecommunications.

Functions point workshop was very useful in determining how to size software project, although another course that cover estimation based on function points sizing will complete the picture.

Biographies:

Omar Kamal has 7 years of experience in wireless telecommunications, software development, training, and software quality management. Omar is working as a senior software engineer in QuickTel Research and Development. He used to work with Lucent Technologies, Hewlett Packard, and Eitisalat. He holds a bachelor's degree in telecommunications engineering from Cairo University, and master's degree in business administration from City University. He is also Certified Quality Manager by the American Society for Quality.

Feedback contacts

Feedback, comments and questions are appreciated by the author.

Author contacts:

E-Mail: omkamal@yahoo.com

Implementing Software Process Improvement at IBM "Cairo Technology Development Center"

By: Mona Arishi

Table of Contents

In this Report we will share with you Cairo Technology Development Center's (C-TDC) experience and lessons learnt in its journey of software process improvement (SPI). We will start with an overview of introducing SPI to C-TDC, hints and tips, and pitfalls encountered while implementing SPI.

Section 1: Overview

The IBM Cairo Technology Development Center TDC is currently one of the series of IBM labs scattered all over the world. To join that elite society of highly technical highly qualified resources, the lab had to undergo a number of major changes. TDC Senior Manager Dr. Ahmed Tantawy had decided to launch the Software Process Improvement task force by end of the 1st quarter 2002.

In those two years, the TDC went through the experience of two process improvement cycles. The TDC process Improvement Team would like to share with you the lessons learnt during this enlightening and highly educative journey.

The Process Improvement Team consists of dedicated and cross departmental team members implementing the quality and the software engineering aspects under the leadership of the Management Steering Group consisting of The Organization Senior manager, the Quality Manager, Management team, in addition to a dedicated technical team to support the process tools.

Section 2: SPI Path

Section 2.1: Planning Phase

The most important step in any Process Improvement approach is senior management's direct involvement. Senior manager should set quality goals, provide resources, review progress, give recognition, revise the reward system, contain employee fears and encourage them to excel.

This was demonstrated in our process improvement approach through:

- Attending SEPG early meetings.
- Setting the quality goals for the organization.
- Assigning SEPG and SPI teams.
- Reviewing the progress through periodic meetings with the SEPG manager.
- Giving recognition to noticeable achievement.
- Setting a rewards system that supports process improvement efforts.

- Emphasizing the importance of quality and improvements.

■ *Budget assigned as a long term investment*

Most organizations that get involved with Process Improvement are in the business to make a profit. Process Improvement is not free; it takes money computer resources, human resources, tools and techniques, training and consulting support. It is important to determine how much process improvement an organization can afford.

■ *Best people are put on SPI & SEPG teams*

Selecting the SPI team and the SEPG team should be based on defined criterion. They will be the leader of changes needed. They must understand the value of the improvement effort. Choosing the most effective and experienced project managers in the SEPG team as part of their time was the criteria our organization took in addition to insuring that different business units are involved. As for the SPI team, it is a team dedicated to plan for improvement and follow up on the activities done, present to the SEPG any improvement opportunity that needs discussion and decision.

■ *SPI activities are an organizational project*

SPI activities should be planned, tracked and controlled, measured for their success and failures then get lessons learnt. Assign responsibilities and identify tasks for people within the organization to achieve improvement objectives and success factors.

■ *Dedicated & motivated people on the job*

People, Process and Technology forms the basis for new organizations. Selecting the right people to do the job is a major key contributing in our success. Training people to achieve the defined goals is another key for success. These are some hints how that can be achieved:

- Roles & Responsibilities should be defined clearly without any redundancy.
- People are selected according to defined standards.
- Training is the vehicle for proficiency.
- Tools are provided to automate processes.

■ *Process Change Method (PCM) is used as an SPI model*

Finding your way can be difficult and it's easy to get lost if it's your first take. Implementing CMM-based process improvement is like finding a path through the forest. The SEI has worked with community Partners to find effective practices in CMM-based Process Improvement. The PCM is a collection of these practices.

■ *SPI activities are advertised*

Because we believe that Continuous process improvement is the only way to lead in today's world, several activities are carried to ensure that everyone are contributing, as follows:

- A Periodical news letter is being sent to all team members encouraging them to join our improvement activities.
- Periodical process improvement meetings are held welcoming any team member to participate in the improvement
- A tool is used to facilitate and manage the process improvement suggestions.

Section 2.2: Implementation Phase

- ***Respect the organizational culture***

Like any change initiative, processes cannot be improved in isolation of the environment in which they are implemented. Thus, understanding your organization's culture is key to success. Understanding the culture's elements and dynamics will help you to effectively manage change and work along with - rather than against - your organization's culture.

- ***Use current Processes as a starting point***

Putting together ideal processes may seem to be the right starting point: logical activities, put in logical order, which people will not have any problem following. Yet, experience proves the opposite. Your best starting point is your current processes: document them, analyze them, identify problem areas, then work to gradually close the gap between the process "as is" and the process as you would like it to be.

- ***Pilot fast and in parallel***

Your next step is to pilot. Piloting is the way to judge whether the theoretical process is suitable for practical implementation. Naturally, you will find some aspects that need to be improved or adjusted. Monitor piloting results, as well as people's reactions to and problems with the process; use these as indicators for areas that are still "not right". If you have enough projects, pilot many processes at the same time.

- ***Involve the implementers in the process***

Involve practitioners in process improvement teams: they are the best information source for the current process. After all, they work with it every day. On a side note, involving them in the improvement effort from the start ensures their buy-in and decreases their resistance.

- ***Use automation for control & acceleration***

If there is a budget, develop tools. Tools facilitate work, standardize implementation details and eliminate time consuming tasks. They will also control the flow of the process and ensure that the checkpoints are not missed.

- ***Award star implementers***

Use rewards to get and keep people's commitment and enthusiasm. Recognize star and early implementers; they will serve as models for the rest of the organization.

- ***Measure as early as you can***

Collecting measures is a new practice for most organizations, so start with measurements early on. Educate people on the benefits of using measurements for monitoring and control, and encourage them to collect the measurements appropriate to their work. By the time your processes are all in place, your practitioners would be used to managing their work quantitatively, and your organization would have a good base of data to guide analysis and decision-making.

Section 3: Pitfalls

- ***Forcing change.***

Changing the organization culture means changing people's mindset, attitude, behavior, and mainly changing the way work is done. These are areas that need patience when dealing with, forcing change suddenly can lead to discrepancies in the organization behavior. Change

management is the main challenge that is faced when deploying the quality system, using the suitable approach accelerates goal achievement.

■ *Working against middle management*

"Buy-In" is considered the magic spell if acquired from middle managers the change will flow like a breeze, otherwise:

- o Managers will reject change in a direct or indirect ways.
- o Low morale will start spreading across teams.
- o Quality of work products will decrease.

■ *Going big on automation from the start*

Automation is known as the activity of speeding up the processing of steps to achieve results using computerized systems. By default, a defective process leads to a defective output. So imagine what can happen in the case of a defective automated process. MESS I guess.

Automation is expensive. That is why managers should study this decision carefully and should know when and how this will start to payback. From our experience we are highlighting the following:

- o Start using an evolutionary concept in mind concerning the application.
- o Clear and simple User manual is mandatory.
- o Pick a platform which is easily scalable to avoid having bottlenecks when implementing improvement in the processes.
- o Know that the automated system will be the heart of the project management systems in your organization.

■ *Working in an island*

Any quality system is based on collaboration between all organization functions. The higher the level of collaboration the organization reaches, the higher the level of its success. Tying the organization functions should be set as a high priority action for achieving our goal as follows:

- o Consistency of management decisions across organizational functions concerning the implementation of the quality system is considered a catalyst.
- o Saving effort and money that might be wasted if different functions are working on similar activities.
- o Enhancing the morals between organization members that will leverage their output.

■ *Using outside processes*

Using outside processes is a major problem that the organization faces when starting to deploy the quality system. At the start this usually happens because of the lack of experience in this area. But after implementing that; problems arise mainly due to:

- o External processes are not fit for the organization activities. They are like wearing somebody else's clothes. They do not fit.
- o Organization members find it hard to commit to external processes because of the lack of ownership.

- Imposing processes that teams do not feel comfortable with negatively affects the organization's morale.

- ***Interrupting the process for other priorities***

It is not only about willingness, it is also about commitment. Failing to put the process on top of the priority list for those developing the process and those supporting it shall cause implementers to deal with the process with a sense of triviality and indifference. This in turn will be an obstacle in the organization's path to success.

- ***Creating hostility with implementers***

It is all about communication, giving the impression –intentionally or un-intentionally- that the process is here to uncover mistakes or to single out individuals with erroneous practices. This will result in creating a state of hostility not only between implementers and the process, but also between implementers and those developing the process. A state of hostility may also arise from a wrongful impression that those developing the process are superior to those implementing it, a case that gives hostility a personal edge, almost not curable over time.

Glossary

1- **SPI:** Software Process Improvement.

2- **C-TDC:** Cairo Technology Development Center.

3- **SEPG:** Software Engineering Process Group.

4- **PCM:** Process Change Method.

References

IBM C-TDC Organization policy, standards, and procedures.

Biography

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

Feedback contacts

Feedback, comments and questions are appreciated by the author.

Author contacts:

E-Mail: mona@eg.ibm.com

Presentations

You can download a life presentation from the SECC/SPIN website for more details.

Are we ready for Automated Testing Tools?

By: Madiha A. Hassan

By definition testing is a repetitive process through out the development life cycle. Automation of some activities in the software testing process allows the tests to be completed many times faster. Furthermore some types of testing such as load / stress testing are very hard to be performed manually. Automation allows performing consistent and repeatable tests through enforcing consistent procedures. The productivity gains delivered by automated testing allow and encourage to test more often and more completely. Full-featured automated testing systems produce convenient test reporting and analysis. These reports provide a standardized measure of test status and results, thus allowing more accurate interpretation of testing outcomes.

The introduction of automated testing into the business environment involves more than buying and installing automated testing tool. In fact effective automation is predicated on the idea that a manual testing process already exists. Before attempting to automate a test, a solid grasp of basic testing processes is needed.

In fact not all tests should be automated. When considering which tests to automate, focus should be placed on those manual test activities which take the longest time to set up, those manual tests that require the higher number of repetitive tasks and which are to be run most frequently. Automated testing should be used to augment current testing process, so planning for using automated testing tools should be an extension of a solid testing process.

Most but not all types of tests can be automated. Certain types of tests like user comprehension tests, and tests that require constant human intervention are usually not worth the investment to automate. Manual testing never goes away and both should go supporting each other.

Automated tools should not be viewed as a cure-all that will instantly reduce all problems (cost, effort …). Without up-front planning and infrastructure design, automated tests can likely become difficult to create and maintain. By that we can achieve an expensive shelf-ware tool.

So don't think that an automated test tool can generate test cases in the click of mouse. It will not solve problems right out of t he box, nor will be successful without a well designed plan and a lot of hard work.

Purchasing and using an automated testing tool is a major investment so we need to look for getting return on this investment. The ultimate benefit of using automated test tools is the ability of running these tools on subsequent of builds / releases.

Beside the basic manual testing process which proved to be successful for the organization we should look for other infrastructures to be incorporated as basics like developing the product with testability feature, writing scripts in standard way, developing the supporting libraries,...

One of the most important things to remember is that there is no one size fits all for applying using automated testing tools. An automation effort depends on the criticality of the

software under test, the level of investment we are willing to make, the maturity of the software engineering processes we are using and the time frame in which we expect to see a return on investment.

The decision to use automated testing tools can start small or start large. Starting small means to deviate resources for incremental efforts, showing success then grow using automated testing tools in other areas where it makes sense to do so. In that aspect people begin with parts easy to automate to provide quick results. By starting small we will be able to learn how to use the tool effectively.

It is very important to notice that all success story noticed in this area were built on focused areas of application where it made sense to start augmented by skilled people either technically or managerially (the right people) to support the mission.

I hope by sharing these ideas together (I feel important to open now) we start putting together our roadmap to start right.

Feedback contacts

Feedback, comments and questions are appreciated by the author.

Author contacts:

E-Mail: mad_abdalla@mcit.gov.eg

What is an Estimate?

By: Mohamed A. AbuSen

An Estimate is a decision about how much time and resource are required to carry out a piece of work to acceptable standards of performance. This therefore requires you to determine:

- The "**Size**" of the task or group of tasks – as determined from measurements if possible;
- The amount of "**Effort**" required to complete the work – how can the work is broken down. Can it be divided between two or more people? Effort is measured in project time units – hours / days / weeks.

Once the efforts is known then optimize the resource needs, taking individual capacities or available time into account to determine the levels of effort required from each.

Effort is a direct measure of person's time to do a piece of work in normal workdays. Unfortunately, that person will often have other non-project activities to complete which reduce their capacity to do work.

At a capacity of 50 %, the work will take at least double the number of days. In practice, it takes longer because of the "back-track" effect due to the breaks in the flow of the work. Effort is measurable as continuous work with no interruptions.

Duration is a conversion of effort taking into account the number of people involved, their capacities and an allowance for non-productive time. Since duration is measured in real working days this is never the same as the schedule, which has to take into account:

- Non available days for project work.
- Non working days – weekends.
- Public and organization holidays.
- Staff holidays.

The first step for you is to derive some realistic duration and then apply these to a calendar to derive a schedule.

Estimating the durations

As the duration of each key stage is the real time it will take to complete the work this is usually the most difficult part of the planning process. The process appears to be one of part art and part science that is hardly surprising since you are really trying to predict the future! So far, no one has produced a reliable crystal ball!

The sources for accurate estimates are limited:

- Experience of others;
- The expert view;
- Historical data from other projects.

There is no substitute for experience. If similar work has been done before then you can ask ***others*** for their own previous experience and adjust the data for your project. It is a reasonable way to start but always take a cautious approach.

The data you collect this way will often hide important relevant information. No one will easily admit taking longer than the plan predicted for a piece of work – particularly if slippages caused problems. In addition, people's memories have a habit of only remembering the good news with the passage of time.

If good plans and records exist review, these to determine what actually happened compared to what was planned to happen. Take more than one opinion if you can and remember no two people ever do the same piece of work at the same pace. The equation relating effort and performance is different for us all.

Who are the experts? There may be a few – or so they believe! Always ask questions about how reality compared with original estimates for some works. Check that the nature or content of the work did not change. You soon discover who is above average at estimating accurately – the expert (s) you desperately seek.

Since, it is relatively rare for work to be identical between different projects then apply an adjusting factor to arrive at duration for your activities.

Keep a record of how you derived the estimates in case you are wrong, then you can improve your estimate skills.

People Problems
Ask anyone how long a piece of work will take and you are likely to be given a shrug, smile, and wildly inaccurate answer.

This is because they do not ask themselves some simple questions:

- Do I really understand what is involved?
- Do I have all the necessary skills and tools for the work?
- What else must I do at the same time?
- What is the priority of the project work over other work?
- When is it really needed by?
- Can I break the job down into chunks to do different times?
- Will I be taking any holiday during the time concerned?
- Do I have any other obligatory commitments during the time concerned?
- What does my manager know about my future commitments that I do not know yet?

The reality is that the majority of people are not productive 100 % of the time! As much as 20% of the working week is taken up by (Meetings, General interrupts, unforeseen events, seeking advise from others, communication failures, personal organization, engaging in conflict**, *inability to say "no" to others' requests***).

Contingencies

The purpose of contingences is to attempt to quantify the extent of uncertainty in the estimating process that make up the project plans.
To determine the contingence values ask:

- What factor can you use for adjusting people's estimates?
- Is the factor global for all estimates or different for different types of work and for different people?
- Should you expose your adjustment factor (s)?

- What limits must you use in applying contingency?
- Should you multiply some estimates by an additional weighting for:
  - Team size?
  - Team experience (in individuals)?
  - Team working history of this team?
  - Project complexity?
  - Project use of new technology or techniques?

If so, what should it be? You take the final decision what figures for "durations" you intend to insert into your plan.

Obviously there is a balance between the desired project completion date and the projected or forecast completion date based only on estimates.

References

Trevor L Young, ***the handbook of project management****, Kogan page limited 2001.*

Biography

Mohamed A AbuSen holds a bachelor degree in Computer Science from Alex university in1991, Project Management diploma - AUC 2001 and Master of International Business Administration (MIBA) ESLSCA 2004. Mohamed has 6 year of experience in software development and 5 years in project management, he is a member in Fujitsu Service Egypt SEPG. Mohamed is working as Projects Control Manager for Fujitsu Services Egypt.

Feedback contacts

Feedback, comments and questions are appreciated by the author.

Author contacts:

Email **:** mohamed.abusen@Services.Fujitsu.com

The Blind Manager, Can he drive through the competition? "Software Measurements"

By: Ahmed S. El-Shikh

In SMEs, according to the limited resources and hard competition; the project manager (some times under the influence from top management) usually focuses to make the delivery date as soon as possible (i.e. to reduce the cycle time of the software production process). They order their teams to do what appears to be "*essential*" only. From the managerial aspects, that is true. But the problem here is in what can be considered to be under the umbrella of the word "essential". Recording, documenting, measuring, auditing and other software production related activities are usually being the victim. Recording, documenting and auditing help the organization to build its history (that will be helpful in the near future) but omitting measurement makes manager loosing vital information that are needed even in day-by-day management activities.

It is clear that you can not control what you can not measure and as we know controlling is one of the key managerial skills besides planning, organizing and leadership. The manager that has no measured data - to be analyzed and used as a sign - is like a **blind car driver** that doesn't see the road he drives on.

The story is not over yet; he also has no ability to predict the problem and take preventive actions. He feels the disaster after it happens with lacking the ability to even correct it.

As a manager you need to objectively identifying some entities inside your production environment and there properties.

Tracy O'Rourke said "**Without the right information, you are just another person with an opinion**", and as I think he had summarized the real situation in all companies with no defined measuring program.

Formally, the measurement activities can focus on two different **metrics categories**, "*Process Metrics*" and "*Product Metrics*". Effort, time, productivity and error rate are clear examples of the process metrics, while reliability, testability and configurability fall into the product metrics.

Although there is a wide spectrum of metrics, it is strongly recommended to **start with** metrics that are easy to be implemented and understood to company team, such as effort (in work hours).

The customers of the measurement process usually fall into two categories:

- *Inside* the software production process, such as project managers, project team members and quality representatives.
- *Outside* the production process, such as top management and external customers.

Generally, the measurement process has an **effect on** the three elements of the quality trilogy:

- *Quality Control:*
  It enables the day by day managerial corrective activities.
- *Quality Improvement*:
  It is considered to be the base line for all improvement activities done by quality teams.
- *Quality Planning:*
  A lot of data types and categories are needed to guide the planning roadmap.

Measuring all available indicators leads to an overhead that can exceed the production process itself. So, it is important to have a **selection methodology** to enable you to decide what need to be measured and what

don't. The "**Goal/Question/Metrics, GQM**" approach is the common one; it has a lot of advantages, such as:

- Let you focus on what is important for you.
- Tightly couples the measurement activities and the company strategic goals.
- Clearly shows the company priorities to each metric.

GQ(I)M –considered as the natural promotion of the GQM- can be used for the same purpose. GQ(I)M is an acronym for **Goal-Question-(Indicator)- Measure**. The "I" in parentheses distinguishes this from the closely related GQM methodology. For an introduction to GQM methodology, see [Tadros, 2003] and for complete details about GQM and GQ(I)M see [Park, 1996].

The most important part in the GQ(I)M is **where to apply the indicators** through the production process, there are three categories:

- *Leading Indicators*.
  That appear early in the process, this type gives alarms before things go worse.
- *Concurrent Indicators*.
  Those are in the heart of the process
- *Lagger Indicators*,
  Which evaluate the results coming out from the process.

As I think, it is clear that the *leading* ones must take the focus of the measuring team, because they enable managers to take early corrective actions. The *lagger* ones are used to evaluate the overall process performance.

Collecting the data is a process that needs to be designed correctly, be careful, do not make a separate data collection team. I.e. do not give them the image of the (*Spy*) person or the hidden enemy. Formally, functional organization structure (with separate permanent measurement team) is not the good choice. Try to form a *cross functional team* that supports a full coverage of the process activities distributed over all company departments and rules.

Analyzing the collected data is the next step that shows the true added value of the measurement process. "*Statistical Quality Control, SQC*" is the technique that must be used. It enables you to take a snapshot image, long historical view and accurate evaluation of your process capability. Also, it gives you a good methodology to test your hypnosis about the improvement results. For more information about the SQC see [Ghannam, 2003] for an introduction. For an example of using the SQC in improving the quality of the software inspection process see [GEORGE,2003]

Until now (after you take care off all of the above) you have a good plan. Wait; do not go a step further before building the **measurement culture** inside your company. The Fear is the first reaction towards applying measurement program. Every one imagines that this program will be the enemy that had been built specially to discover his errors. Another myth is that it will take too much time to collect and analyze the data and that the team will fixate on getting the numbers right rather than building good software. Try to remove these bad reactions and emotions. If you do not, you will face a big resistance that will be added to the traditional resistance towards any new changes.

Some **tips for success** can be helpful. First of all, start slow, small, and train people. But be careful, the fear of measuring (from working force, middle mangers and top management) will try to make you very slow.

You need to avoid a lot of **traps,** such as,

- Lack of Management Commitment
- Measuring Too Much, Too Soon
- Measuring Too Little, Too Late
- Measuring the Wrong Things
- Imprecise Metrics Definitions
- Using Metrics Data to Evaluate Individuals

- o Using Metrics to Motivate, Rather than to Understand
- o Collecting Data That Is Not Used
- o Lack of Communication and Training
- o Misinterpreting Metrics Data

For more information about possible traps, see [Wiegers, 1997]. Just wait, I forget to tell you, the most important trap to be avoided is the *"Hidden Agendas"* of some mangers and as I think you had understood me, be careful.

A **practical step-by-step guide** for implementing a measurement program with a case study will be helpful now. You can use it to transfer what we mentioned above into a real world entity. Additional appendixes for calculation formulas are available too, see [Park,1996].

Now, we must ask a question, what are the **real benefits** of a measurement program that had been implemented with all these features? The field surveys show that:

- o Design and testing efforts have steadily increased over the years.
- o The fraction of time devoted for implementation has trended downward.
- o Specification & implementation reveal complementary trends, with one being high when the other is low.
- o An initial downward trend in preliminaries and project planning was reversed if you begin to concentrate more on formal project management.

At the end, you will face a resistance to change that will be affected by the amount, velocity and type of the changes that you are going to do. Also, do not wake up any lion inside your company (that resist your program)….you knows what can be the next change…. be careful.

References

[Baumert,1992] John H. Baumert and Mark S. McWhinney, "***Software Measures and the Capability Maturity Model***", Technical Report CMU/SEI-92-TR-25, September1992.

[GEORGE,2003] ELLEN GEORGE AND STEPHEN JANISZEWSKI "***Optimizing Software Inspections with Statistical Quality Techniques***", SQP VOL. 6, ASQ, 2003,

[Ghannam,2003] Adel Ghannam "***Software Product Quality Engineering for SME***",SPIN Newsletter, Issue 1, 2003.

[Goethert,2001] Wolfhart Goethert and Will Hayes, "***Experiences in Implementing Measurement Programs***", Technical Note,CMU/SEI-2001-TN-026,November 2001.

[Tadros, 2003] Marian Tadros, **"Software Measurement (Why, What and How)",** SPIN Newsletter, Issue 2, 2003.

[Park,1996] Robert E. Park, Wolfhart B. Goethert and A. Florac, "***Goal-Driven Software Measurement- A Guidebook***",CMU/SEI-96-HB-002, August 1996

[Wiegers,1994] Karl E. Wiegers, "***Lessons from Software Work Effort Metrics***", published in the Software Development magazine, October, 1994.

[Wiegers,1997] Karl E. Wiegers, "***Software Metrics: Ten Traps to Avoid***", published in the Software Development magazine, October 1997.

[Wiegers,1999] Karl E. Wiegers, "***A Software Metrics Primer***", published in the Software Development magazine, July 1999.

Biography

Eng/ Ahmed S. El-Shikh is the Quality Assurance Manager in the “Systems Advisory & Solutions, (SAS)” company. His interests include software engineering aspects, software quality management, statistical quality control and process Improvement approaches specially the “Six Sigma, DMAIC”.

Feedback contacts

Feedback, comments and questions are appreciated by the author.

Author contacts:

E-Mail: el_shikh@sas-sys.com