# Sigma-TRIZ: Algorithm for Systematic Integration of Innovation within Six Sigma Process Improvement Methodologies

Stelian Brad
*Technical University of Cluj-Napoca*
*Romania*

## 1. Introduction

Continuous process improvement is a constant preoccupation of companies operating in strong competitive markets (Thawani, 2002; Cronemyr, 2007). The goal of process improvement projects is about increasing both efficiency and effectiveness of the business system (Brad, 2008). A widely used methodology for process improvement is Six Sigma DMAIC (Cascini et al., 2008; Hamza, 2008). Some researches reveal a strong relationship between the Six Sigma DMAIC's effectiveness and the qualification of the team involved in its application (Jean-Ming & Jia-Chi, 2004; Treichler et al., 2002). Therefore, top experts are usually hired by potent companies to supervise Six Sigma DMAIC implementation and to generate solutions for process improvement.

Despite the strengths of the Six Sigma DMAIC methodology, the solution generation process is a challenging issue (Smith & Pahdke, 2005). Hence, for formulating reliable results, adequate tools are required to support this activity. Keeping the same register, when significant "noise" factors act upon business processes, creative problem solving and innovation become key approaches for achieving high levels of process maturity and capability (Khavarpour et al., 2008). A powerful tool for inventive problem solving that might be considered in this respect is TRIZ method (Altshuller, 2000).

Integration of TRIZ method within Six Sigma DMAIC methodology has been analyzed by several researchers, recent results being reported in this respect. However, there are no proposals in the current published works on how effectively to integrate TRIZ within Six Sigma DMAIC. For example, Qi et al. (2008) only highlights the positive effect of using TRIZ in connection with Six Sigma DMAIC for stimulating creativity and reducing the effort towards the formulation of mature solutions to the problem under consideration. In the same spirit, the paperwork (Zhao, 2005a) stresses the necessity to use TRIZ together with Six Sigma DMAIC for accelerating the innovation process but it lacks in proposing a detailed solution of integration. In (Zhao et al., 2005b), the use of quality planning tools like QFD in connection with TRIZ for key process identification and innovation within Six Sigma DMAIC framework is put into evidence. However, this research work does not reveal a way to inter-correlate TRIZ and Six Sigma DMAIC. The systematic integration of TRIZ method within the Six Sigma DMAIC methodology was first time proposed by the author of this chapter (Brad, 2008).

The algorithm is called Sigma-TRIZ. It approaches the process improvement problem from a comprehensive perspective, by creating a systematic framework of identification and prioritization of the conflicting zones within the analyzed process. Sigma-TRIZ algorithm starts from the premise that any improvement should increase both efficiency and effectiveness of the analyzed process, without affecting the balance within processes that are correlated with the one analyzed. From this position, Sigma-TRIZ algorithm allows formulation of balanced and robust improvement solutions with respect to the "noise" factors (also called "attractors") acting upon the process. In principle, Sigma-TRIZ connects the innovation vectors generated by the TRIZ framework with the improvement objectives. It does this by considering a complex set of barriers and challenges from the "universe" describing the analyzed process. It starts by prioritizing the intervention areas considering the criticality of the conflicts within the process (Brad, 2008).

In this chapter, an enhanced version of Sigma-TRIZ algorithm is introduced. Enhancements are related to the prioritization of solutions and identification of the correlations between them, as well as to the formulation of the algorithm for being easy-to-implement in a software tool. A case study showing the step-by-step application of the algorithm within Six Sigma DMAIC procedure is also illustrated. The "Conslusions" part of this chapter highlights the practical implications of Sigma-TRIZ for increasing the competitiveness of companies operating in a knowledge-based economic environment.

## 2. The Sigma-TRIZ algorithm

### 2.1 Background philosophy

From practical experience it is known that most of the business-related problems are not simple; and for solving them, consideration of several interrelated and convergent process improvement projects in relation to a given intended improvement objective is required. Under such conditions, integration of innovative problem solving tools like TRIZ should increase the effectiveness of results within the "Improve" phase of the Six Sigma DMAIC methodology.

Denoting with $P = \{p_1, p_2, p_3, ..., p_n\}$ the set of interrelated and convergent process improvement projects linked to the intended improvement objective O, where n is the number of improvement projects in the set P, the objective O is achieved if and only if P leads to a required level of process effectiveness E and efficiency e in a time horizon T; time horizon imposed by the dynamics of the competitive business environment (see Fig. 1). In order to achieve this goal, trade-offs and trial-and-errors approaches (e.g. brainstorming) are not efficient means (Silverstein et al., 2005).

Moreover, to keep a sustainable evolution of performance within the considered process, E and e should be balanced along time. Denoting with t the time variable, with $E_0$ the level of process effectiveness at the initial moment $t_0$, with $E_1$ the expected level of process effectiveness at the moment $t_1$, with $e_0$ the level of process efficiency at the initial moment $t_0$, with $e_1$ the expected level of process efficiency at the moment $t_1$, and with T the difference $t_1 - t_0$, the generic correlation between E and e is described by relationship (1), where the function f depends on the adopted innovation strategy (e.g. upsizing, downsizing).

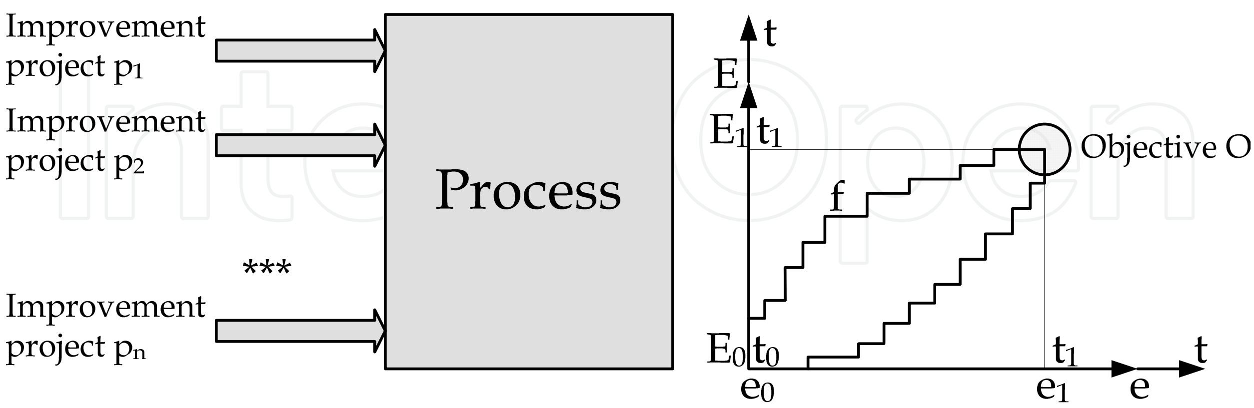



Fig. 1. Competitive approach in process improvement

$$E(t)\Big|_{E_0}^{E_1} = f\left( t\Big|_{t_0}^{t_1}, e(t)\Big|_{e_0}^{e_1} \right). \quad (1)$$

In order to follow a competitive process improvement path, the focus within all improvement projects $p_1$, $p_2$, …, $p_n$ should constantly be on two key paradigms: (a) the ideality paradigm (Altshuller, 2000); (b) the convergence paradigm (Silverstein et al., 2005). The ideal final result (IFR) is the ratio between the sum of all useful functions and effects and the sum of all harmful functions and effects (including the related costs) (Altshuller, 2000). The convergence paradigm focuses on reducing the difficulty of problem resolution (Silverstein et al., 2005). In this respect, the convergence paradigm operates with the ratio between the total number of possible variants and the total number of possible steps that lead to mature solutions (which solve the problem without compromises).

Denoting with I the ideality, with $\Sigma F_U$ the sum of useful functions and effects, with $\Sigma F_H$ the sum of harmful functions and effects, and with $\Sigma C$ the sum of costs because of poor-performances (losses), the mathematical formulation of the law of ideality is (Altshuller, 2000):

$$I = \frac{\sum F_U}{\sum (F_H + C)}. \quad (2)$$

According to relationship (2), the goal is having as low as possible harmful functions, effects and costs, and as much as possible useful functions and effects. Thus, in theory, when ideality is achieved, the result is: $I \rightarrow \infty$. In real systems this cannot happen, but the practical target is to move as close as possible towards the ideal result – this target is known in the literature as "local ideality" (Altshuller, 2000).
Symbolizing with D the difficulty in problem resolution, with TE the number of trial and

error iterations of variants, and with ST the number of steps leading to acceptable solutions, the mathematical formulation of the law of convergence is visualized in relationship (3). Obviously, the goal is having $D \rightarrow 1$.

$$D = \frac{TE}{ST} . \tag{3}$$

TRIZ is a powerful tool towards deployment into practice of the laws described in (2) and (3). Therefore, by systematic integration of TRIZ within Six Sigma DMAIC it is expected to formulate highly mature process improvement projects during the “Improve” phase of DMAIC. An effective way for systematically integrating TRIZ within Six Sigma DMAIC is proposed by Sigma-TRIZ algorithm, which is further described into detail.

### 3.2 Step-by-step Sigma-TRIZ algorithm

Sigma-TRIZ algorithm consists of twelve steps, schematically presented in Fig. 2. The detailed description of these steps covers the next paragraphs of this section.

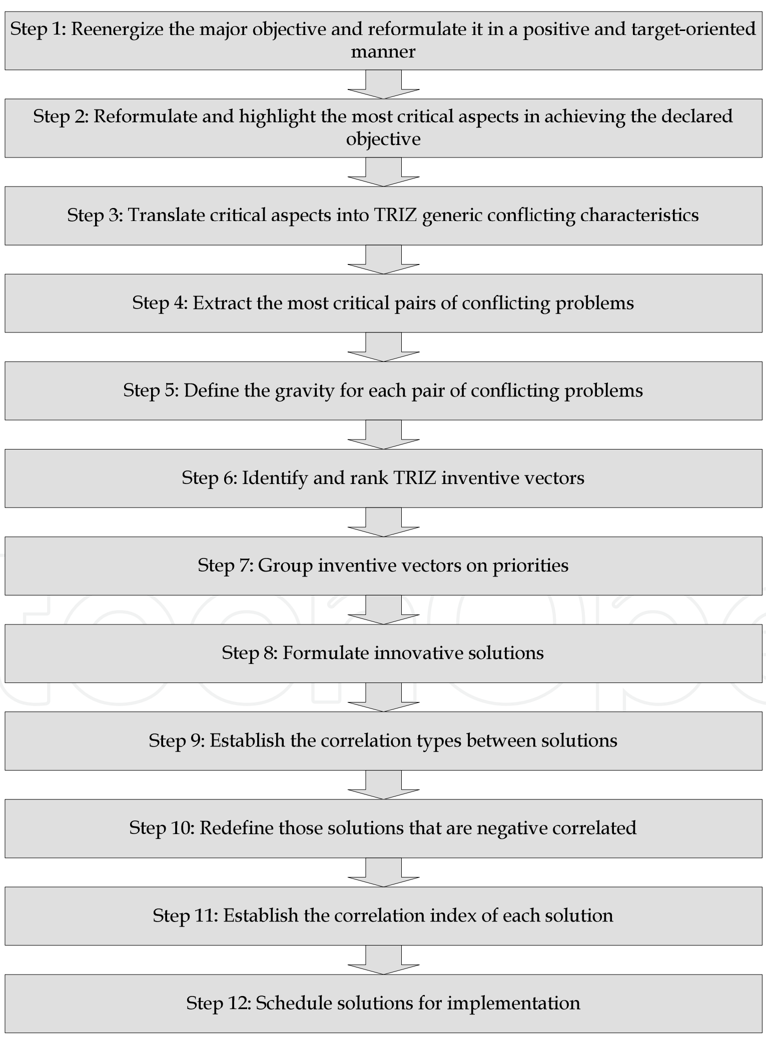



Fig. 2. The main steps of Sigma-TRIZ

*Step 1*: Reenergize the major objective and reformulate it in a positive and target-oriented manner: The improvement objective O is very often expressed by the target group in a negative and/or vague and/or too large manner. Thus, a clear statement of the improvement objective is firstly required. The result is a re-phrased objective $O_p$. For example, considering a software development company, a possible improvement objective O would be: reduction of the number of "bugs" for the work delivered to the customer. The reformulated objective $O_p$ would be: no "bug" in the software application when it is delivered to the customer. This reformulation includes the intended target: "zero bugs".

*Step 2*: Reformulate and highlight the most critical aspects in achieving the declared objective: The set B of significant barriers in achieving the objective $O_p$ is identified. The set B is represented as:

$$B = \{b_1, b_2, \ldots, b_k\}, \tag{4}$$

where: $b_j$, $j = 1, \ldots, k$, are the process-related barriers (k is the number of barriers).

*Step 3*: Problem translation into TRIZ generic conflicting characteristics: For each barrier $b_j$, $j = 1, \ldots, k$, a set of TRIZ-generic parameters that require improvements (maximized or minimized) should be determined. For details about TRIZ-generic parameters reader is advised to consult the reference (Altshuller, 2000). Thus, each barrier $b_j$, $j = 1, \ldots, k$, has a corresponding set of generic improvement requests $GR(b_j)_i$, $i = 1, \ldots, h(b_j)$, $j = 1, \ldots, k$, where $h(b_j)$ is the number of generic improvement requests associated to the barrier $b_j$, $j = 1, \ldots, k$. For each generic parameter $GR(b_j)_i$, $i = 1, \ldots, h(b_j)$, $j = 1, \ldots, k$, a set of generic conflicting parameters should be further determined. They are extracted from the same table of TRIZ parameters (see (Altshuller, 2000)). At the end, a number of k sets of generic conflicting parameters are determined. These sets are denoted with: $GC(GR(b_j)_i)_f$, $f = 1, \ldots, g(GR(b_j)_i)$, $i = 1, \ldots, h(b_j)$, $j = 1, \ldots, k$, where $g(GR(b_j)_i)$ is the number of generic conflicting parameters associated to the generic improvement request $GR(b_j)_i$, $i = 1, \ldots, h(b_j)$, $j = 1, \ldots, k$.

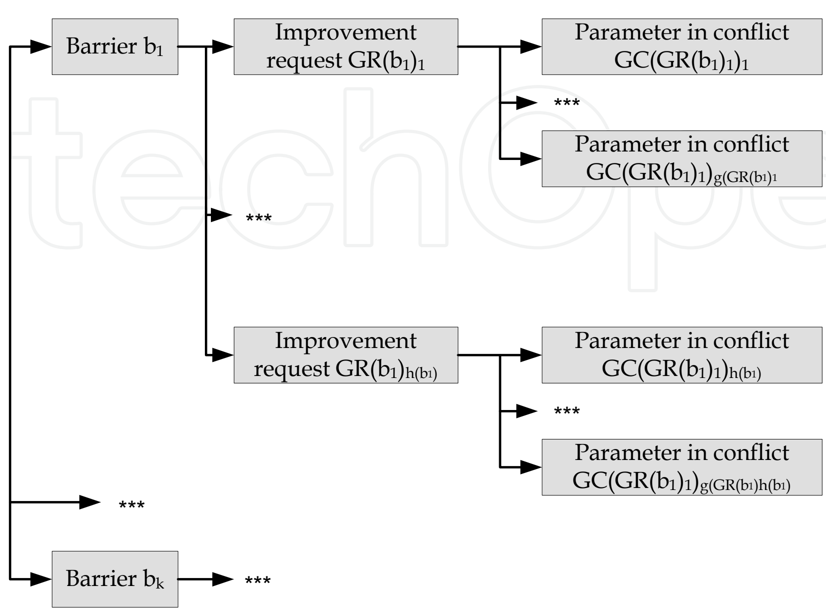



Fig. 3. Step 3 of Sigma-TRIZ

For example, assuming that in a given process one of the barriers identified is: $b_x$ = "Lack of proper infrastructure (there is no stated system for assuring a good coordination between departments)", the TRIZ-generic parameters (see Appendix) requiring improvement in relation with this barrier are: $GR_{x,1}$ = "Energy spent by moving object" (meaning: reduction of human effort in transferring information; optimization trend: minimization); $GR_{x,2}$ = "Loss of information" (meaning: avoid loss of information because of lack of communication; optimization trend: minimization); $GR_{x,3}$ = "Convenience of use" (meaning: reduction of time required to transfer information; optimization trend: minimization).

In the attempt to improve the system with respect to the requests $GR_{x,1}$, $GR_{x,2}$ and $GR_{x,3}$, some other performances of the system might be affected. Thus, in relation with $GR_{x,1}$ the following TRIZ-conflicting parameters are determined: $GC(GR_{x,1})_1$ = "Amount of substance" (meaning: money and effort spent to set up a software information system). In relation with $GR_{x,2}$, the following TRIZ-conflicting parameters are determined: $GC(GR_{x,2})_1$ = "Tension/Pressure"(meaning: increased pressure on people for respecting rules and procedures); $GC(GR_{x,2})_2$ = "Harmful side effects" (meaning: possibility of system breakdown); $GC(GR_{x,2})_3$ = "Amount of substance" (meaning: extra-resources to maintain and enhance the information system". In relation with $GR_{x,3}$, the following TRIZ-conflicting parameters are determined: $GC(GR_{x,3})_1$ = "Energy spent by moving objects" (meaning: effort required by all people in the company learning how to use the information system).

*Step 4*: Extract the most critical pairs of conflicting problems: From the pairs of conflicting problems formulated at step 3, the most critical ones are extracted for further transformations. In some cases it might be possible to keep all pairs of conflicting problems. Thus, in the most general case, the result is a set of pairs of conflicting problems like this:

- $PR_{1,1} = \{GR(b_1)_1$ versus $GC(GR(b_1)_1)_1\}$;
- $PR_{1,2} = \{GR(b_1)_1$ versus $GC(GR(b_1)_1)_2\}$;
- …;
- $PR_{1,g(GR(bj)i)} = \{GR(b_1)_1$ versus $GC(GR(b_1)_1)_{g(GR(bj)i)}\}$;
- …;
- $PR_{h(bk),g(GR(bj)i)} = \{GR(b_k)_{h(bk)}$ versus $GC(GR(b_k)_{h(bk)})_{g(GR(bk)h(bk))}\}$.

In order to simplify the mathematical representation of the pairs of conflicting problems, from this point ahead the set is denoted $PR = \{PR_1, PR_2, \ldots, PR_m\}$, where m is the number of pairs of conflicting problems.

*Step 5*: Define the gravity for each pair of conflicting problems: Using a scale from 1 (enough critical) to 5 (extremely critical), a factor of gravity $fg_t$, $t = 1, \ldots, m$ is associated to each pair $PR_t$, $t = 1, \ldots, m$.

*Step 6*: Identify and rank TRIZ inventive vectors: TRIZ method operates with a set of 40 inventive generic vectors (see (Altshuller, 2000)). For each pair of conflicting problems (which at this point are only generically formulated) a well-defined subset of inventive vectors from the complete set of 40 vectors (counted from 1 to 40) exists; this subset comprises between 0 and 4 inventive vectors (also called inventive principles). If a certain subset comprises 0 vectors, the analyzed case is critical and only radical changes on the system would improve the situation. Thus, for each pair $PR_t$, t = 1, …, m, a set of inventive principles $V_t = \{V_{1,t}, V_{2,t}, V_{3,t}, V_{4,t}\}$, t = 1, …, m, is associated. Each set $V_t$, t = 1, …, m is revealed by the so-called "TRIZ matrix of contradictions" (see (Altshuller, 2000)). According to the TRIZ matrix of contradictions some sets $V_t$, t = 1, …, m, might be null or might have less then 4 members (i.e. only 1, 2 or 3 members).

Once the sets $V_t$, t = 1, …, m, are defined, a rank is given to each inventive vector. The rank is actually the sum of the gravity factors belonging to the pairs for which a certain inventive vector occurs in the sets $V_t$, t = 1, …, m. Thus, if for example, a certain inventive vector $V_e$ is present for the pairs $PR_x$, $PR_y$ and $PR_z$, and if the factors of gravity for these pairs are $fg_x$, $fg_y$ and $fg_z$, the rank of the vector $V_e$ is $r_e = fg_x + fg_y + fg_z$.

It is important to note that the TRIZ matrix of contradictions, as it is defined by its author (G. Altshuller), proposes a certain inventive vector not only once, but several times, depending on the combination of generic conflicting problems. At the end of this process, a set of z unique, ranked inventive vectors is generated. This set is denoted with $U = \{U_1(r_1), U_2(r_2), .., U_z(r_z)\}$, $z < 40$, where each inventive vector $U_l$, l = 1, …, z, has a rank $r_l$, l = 1, …, z.

For a better visualization, a certain inventive vector from the set U could be denoted as: X(Y/Z), where X is the position of the inventive vector in the table of TRIZ inventive vectors, Y is the number of times the inventive vector is called in the set $V_t$, t = 1, …, m, and Z is the rank of the respective inventive vector (the sum of the factors of gravity of the pairs of conflicting problems that have associated the respective inventive vector).

*Step 7*: Group inventive vectors on priorities: A qualitative analysis is done for each inventive vector X(Y/Z). According to the value of Z and then of Y, the inventive vectors of the set U are structured on priority groups. This structuring is not a "mechanical" process. The expert must analyze the potential impact of the vectors based on the values Z and Y. Thus, vectors having a close value of their gravities (Z) and with close values of their occurrences (Y) could be grouped together. Each group has a certain priority. The group having the vectors with the highest gravities (Z) and number of occurrences (Y) is of first priority, and so on. Actually, each inventive vector comprises some generic directions of intervention where innovative solutions should be searched and defined. It is important to mention that, in the table of 40 TRIZ inventive vectors, each inventive vector has associated several sub-vectors (see (Altshuller, 2000)). For example, if the inventive vector is "segmentation", the related directions of intervention are: "divide the system into independent parts", "increase the level of segmentation", "make the system sectional".

Thus, at the end of this process, for each priority group a number of generic directions of interventions will be revealed. The number of priority groups is not a fixed one; it comes up after the qualitative analysis done by the experts. For a better visualization of the results, the affinity groups are denoted with a(s), s = 1, ..., w, where s is the priority associated to the respective group and w is the number of groups generated at the end of the process. A direction of intervention of a certain group is symbolized $DI_{a(s),q}$, where q = 1, …, y(a(s)), with y(a(s)) the number of directions of intervention in the group a(s), s = 1, …, w.

*Step 8*: Formulate innovative solutions: For each direction of intervention $DI_{a(s),q}$, q = 1, …, y(a(s)), with y(a(s)) the number of directions of intervention in the group a(s), s = 1, …, w, and in the spirit of the direction of intervention, one or several innovative solutions might be proposed. A solution is innovative when it solves the conflict without compromises. The process of solution generation is a creative one; this task requires "openness" in "translating" the generic direction of intervention into effective, practical solutions. The process should start with the directions of intervention from the first priority group and continue up to the last priority group. At the end of this step a set of solutions is generated. This set is denoted with $S = \{S_1(z_1), S_2(z_2), \ldots, S_d(z_d)\}$, where d is the number of solutions, $z_i$, i = 1, …, d, is the factor of gravity associated to the inventive vector to which the direction of intervention $DI_i$, i = 1, …, d, belongs; $DI_i$, i = 1, …, d, being the direction of intervention to which the solution $S_i$, i = 1, …, d, is associated.

*Step 9*: Establish the correlation types between solutions: It is important having all solutions positive correlated for complying with the laws of ideality and convergence (see relationships (2) and (3)). Hence, each solution is analyzed with respect to all the other solutions in order to establish the type of correlations between them. To perform this task, a correlation matrix is worked out. It consists of a number of columns and rows equal with the number of solutions. The main diagonal of the matrix is not taken into account. Each correlation is analyzed following each column in turn, from top to bottom.

*Step 10*: Redefine solutions that are negative correlated: If there are two negative correlated solutions, the one having a lower value of the factor of gravity z will be primarily eliminated and a new solution will be proposed in place, such as the positive correlation to be established. It might be possible that some solutions to have no correlation with the other solutions. This is not at all a drawback in solution definition.

*Step 11*: Establish the correlation index of each solution: Using the same matrix of correlation from steps 9 and 10, the correlation level related to each pair of solutions is determined. In this respect the following scale is used: 0 – no correlation; 1 – weak/possible correlation; 3 – medium correlation; 9 – strong correlation; 27 – extremely strong correlation (almost indispensable each other). Denoting with $a_{ij}$, i, j = 1, …, d, $i \neq j$, the correlation level between solution $S_i$ and solution $S_j$, the correlation index $C_i$, i = 1, …, d, of the solution $S_i$, i = 1, …, d, is calculated with the following formula:

$$C_i = z_i \cdot \sum_{j=1;\, j \neq i}^{d} a_{ji}; \; i = \overline{1,d}\,. \tag{5}$$

*Step 12*: Schedule solutions for implementation: Considering the correlations between solutions as qualitative indicators of prioritization and considering the correlation indexes as quantitative indicators of prioritization, experts should schedule the implementation of solutions. Actually, each solution is a kind of mini-project that requires planning and implementation. Results from a mini-project could influence the results in other mini-projects or require running other mini-projects, according to the correlations between mini-projects. For each mini-project, several issues have to be clearly defined, like: time, costs, responsibilities, tools, etc.

## 3. Case study

The integration of Sigma-TRIZ algorithm within the Six Sigma DMAIC methodology is further illustrated via a case study from the IT sector. It is actually about an IT company which provides outsourcing services in software development. In order to have the overall picture of the improvement project, the case study introduces all phases of the Six Sigma DMAIC process, with highlights on the "Improve" phase, where Sigma-TRIZ algorithm is effectively implemented.

### 3.1 "Define" (D) phase within Six Sigma DMAIC methodology

*The project*: Reorganization of the whole process of outsourcing service provision in software development.

*Intended objectives*: (O1) Reduction of "bugs" for deliverables sent to the client (bug = nonconformity in software jargon): this objective automatically leads to increase in customer satisfaction, reduction of failure costs and on-time delivery of results; (O2) Reduction of the non-productive time in software development: this objective automatically leads to increase of labor productivity.

*Performance metrics*: (M1) Deviation from the scheduled deadline; (M2) Level of poor-quality costs with respect to the income of the project; (M3) Return on net assets; (M4) Customer satisfaction (on a scale from 0 to 5).

*Actors in the process*: (A1) The project manager; (A2) The project manager of the client; (A3) The software programmers (members of the development team); (A4) The testers (members of the quality assurance team); (A5) The contact person with the customer; (A6) The technical director; (A7) The quality management director; (A8) The executive director; and (A9) The executive director of the customer.

*Key requirements and expectations*:

- The project manager: (R1) Short feedback time from the customer (getting answer to the questions asked); (R2) Enough time to perform a detailed project planning;
- The programmer: (R3) Requirements document to be clear and detailed; (R4) Enough time for development; (R5) Less cases of change requests in the project;
- The tester: (R6) Enough time to perform testing; (R7) Adequate and detailed documentation about test scenarios and test plans;
- The project manager of the client: (R8) Good communication with the development team of the supplier; (R9) Detailed documentation of the delivered solution; (R10) High quality of services – no critical "bugs";
- The technical director: (R11) On-time delivery of the project; (R12) High quality (from all perspectives: functionality, reliability, etc.); (R13) Low fluctuations of the team members during project development;
- The executive director: (R14) Project delivered on-time; (R15) Adequate quality to avoid penalties stipulated in the contract; (R16) Low costs with "bug" fixing;
- The quality management director: (R17) Adequate project documentation (records, etc.); (R18) High level of customer satisfaction;

- The executive director of the customer: (R19) On-time project delivery; (R20) Project delivered within the scheduled budget; (R21) Quality of solution at the level specified in the requirements document.

### 3.2 "Measure" (M) phase within Six Sigma DMAIC methodology

*"Suppliers" in the process:* Project manager + development team + testing team + company (infrastructure, procedures, documentation, training, etc.) + customer (documentation, access to certain resources, etc.) + contact person with the client.

*Constrains*: (C1) The project duration is, in most of the cases, imposed by market (e.g. the final customer); (C2) The project budget is relatively rigid; (C3) The previous experience of the team in the project field is sometime insufficient.

*The process (main phases)*:
- Main flow: MF1. Requirements analysis → MF2. Capability definition → MF3. Effort estimation → MF4. Contract negotiation → MF5. Technical analysis → MF6. Design → MF7. Code implementation → MF8. Module testing → MF9. Integration testing and validation → MF10. Deployment → MF11. Delivery.
- Supporting flow: SF1. Project planning → SF2. Monitoring and control.

*Details about process phases*: This information is extracted from quality management system's documentation (records, procedures, instructions, etc.) and/or from one-to-one interviews with actors involved in each phase of the process.

- Requirements analysis: MF1.1. Product vision analysis → MF1.2. Actor definition → MF1.3. Requirements formulation → MF1.4. Business use-case formulation → MF1.5. Requirements prioritization → MF1.6. Formulation of possible solutions;
- Capability analysis: MF2.1. Technical characteristics analysis → MF2.2. Preliminary effort estimation → MF2.3. Technical risk identification → MF2.4. Technical risk estimation → MF2.5. Technical risk assessment → MF2.6. Preliminary data formulation for project planning and contract negotiation → MF2.7. Establish quality objectives → MF2.8. Define and communicate iterations;
- Effort estimation: MF3.1. Detailed data analysis → MF3.2. Detailed effort estimation;
- Contract negotiation: MF4.1. Define key issues in the framework agreement → MF4.2. Analysis of contractual requirements → MF4.3. Negotiation → MF4.4. Contract signing → MF4.5. Order management;
- Technical analysis: MF5.1. Set up the specification document → MF5.2. Use-case formulation;
- Design: MF6.1. Top level architecture design → MF6.2. Interface design → MF6.3. Unit design → MF6.4. Data structure design → MF6.5. Description of system workflow → MF6.6. Supporting unit design → MF6.7. Test scenarios design;
- Code implementation: MF7.1. Coding and code documentation → MF7.2. Unit testing;
- Module testing: MF8.1. Testing of modules;
- Integration testing and validation: MF9.1. Module integration → MF9.2. Testing → MF9.3. Validation;

- Deployment: MF10.1. Functionality documentation → MF10.2. Elaboration of the deployed structure → MF10.3. Structure documentation;
- Delivery: MF11.1. Internal check of solution → MF11.2. Delivery to the customer → MF11.3. Assessment of results;
- Project planning: SF1.1. Refining and description of iterations → SF1.2. Plan and schedule iterations → SF1.3. Define deadlines → SF1.4. Resource allocation → SF1.5. Detailed planning for various project phases;
- Monitoring and control: SF2.1. Cost monitoring and control → SF2.2. Time monitoring and control → SF2.3. Quality monitoring and control.

*Expected results*: (U1) Projects to be delivered on-time, each time; (U2) No major "bug" at delivery; (U3) No major "bug" at internal testing (before delivery); (U4) Actual effort not exceeding the scheduled effort.

*Beneficiaries of results*: (B1) The software outsourcing company; (B2) The customer; (B3) The development team (e.g. less stress, professional satisfaction; success bonus, etc.).

*Necessary inputs to ensure an adequate progress of the process*: The most important inputs are the followings: (I1) The estimated effort not below the real effort; (I2) Clear and detailed requirements and specifications; (I3) Adequate communication infrastructure for distance cooperative work; (I4) Adequate data and document management infrastructure; (I5) Very high skills and expertise of the project manager in the programming technologies; (I6) High skills of the team members in the programming technologies; (I7) Existence of predefined templates for recording deliverables in all phases of the project; (I8) Existence of clear methodologies for project monitoring and control, etc.

### 3.3 "Analyze" (A) phase within Six Sigma DMAIC methodology

*Current nonconformities*: (N1) Errors in the code delivered to the customer; (N2) Difficulties in monitoring the "true" labor productivity; (N3) Difficulties to have a direct and controlled intervention on labor productivity (because of job specificity).

*Root causes for existence of current nonconformities*:

- A software product is characterized by flexibility, "invisibility" and very high complexity per unit of effort, being from these perspectives extremely difficult (or even impossible) to estimate the real effort from the planning phase;
- The object-oriented programming technologies ensure a high flexibility in design and coding, with side effects on covering all possible user scenarios in the testing phase;
- Not enough time allocated for testing because of delays in coding, rigidity in deadlines, low budget allocated by the customer for an extensive testing, etc.;
- Testing scenarios are not complete (and sometimes not applied 100%);
- Lack of effectiveness in the case of testing scenarios;
- Insufficient programming skills;
- Programmers do not work all the time with maximum responsibility;
- It is almost impossible to impose a constant labor productivity (human brain does not work linear);

- Incomplete and ambiguous specification document;
- Too many change requests come up from the customer in the late phases of the development, etc.

Once the root causes are identified, specific actions must be taken to overpass problems or at least to minimize their effect; and this thing is a very challenging one.

### 3.4 "Improve" (I) phase within Six Sigma DMAIC methodology

At this point, integration of Sigma-TRIZ algorithm within the Six Sigma DMAIC methodology has to be done. The goal is to minimize the effort in searching for and setting up mature solutions (if it is possible, solutions that are „free of conflict"), as well as to minimize the capability of identifying these solutions.

*Reenergizing the major objective and reformulate it in a positive way*: No bug at delivery and capacity to deliver on-time.

*Highlighting the most critical issues in achieving the intended objective*: The following barriers have high significance: (a) Insufficient planning of software product/application quality in the initial phases of the development process; (b) customer's pressure to reduce both the budget and the delivery date.

*Problem translation into generic characteristics that need improvement*: For the above mentioned barriers, equivalences within the TRIZ parameters are searched. For this case study, the following generic TRIZ characteristics requiring improvements have been identified: ($a_1$) engagement of employees (maximized); ($b_1$) pressure upon employees (minimized); ($c_1$) solidity of the software development process to various external disturbances (maximized); ($d_1$) effort required to the top management for involving dynamic systems (employees, customers) (minimized); ($e_1$) effort spent per unit of time by employees without affecting productivity (minimized); ($f_1$) waste of energy/resources (minimized); ($g_1$) software system reliability when it becomes operational (maximized); ($h_1$) labor productivity (maximized); ($i_1$) clarity of the process flow (maximized).

*Identification of conflicting generic characteristics*: In the attempt of improving the generic characteristics presented in the previous paragraph, some other generic characteristics might be affected. They are: ($a_2$) quantity of money spent (minimized); ($b_2$) project duration (minimized); ($c_2$) effort spent by dynamic elements (effort required to the customer for providing clear and complete information on-time) (minimized).

*Formulation of the most critical pairs of conflicting characteristics*: From the analysis of the generic characteristics that need to be improved and the generic characteristics that might be affected because of the expected improvement in relation with the intended objective, the following pairs of conflicting issues are identified: A) $a_1 - a_2$; B) $b_1 - a_2$; C) $b_1 - b_2$; D) $b_1 - c_2$; E) $c_1 - a_2$; F) $c_1 - b_2$; G) $c_1 - c_2$; H) $d_1 - a_2$; I) $e_1 - a_2$; J) $e_1 - b_2$; K) $e_1 - c_2$; L) $f_1 - b_2$; M) $g_1 - b_2$; N) $g_1 - c_2$; O) $h_1 - a_2$; P) $h_1 - c_2$; Q) $i_1 - b_2$; R) $i_1 - c_2$. Thus, 18 conflicting problems have been formulated within the analyzed process. All these conflicts must be proper solved; otherwise, significant improvements in process performances cannot be expected.

*Establishing a gravity factor to each pair of problems*: The gravity factor is denoted with $f_g$. A scale from 1 (enough critical) to 5 (extremely critical) is associated to $f_g$. For the pairs of problems in this case study, the following results are obtained: A) $a_1 - a_2$ ($f_g = 5$); B) $b_1 - a_2$ ($f_g = 5$); C) $b_1 - b_2$ ($f_g = 4$); D) $b_1 - c_2$ ($f_g = 3$); E) $c_1 - a_2$ ($f_g = 2$); F) $c_1 - b_2$ ($f_g = 5$); G) $c_1 - c_2$ ($f_g = 3$); H) $d_1 - a_2$ ($f_g = 1$); I) $e_1 - a_2$ ($f_g = 5$); J) $e_1 - b_2$ ($f_g = 5$); K) $e_1 - c_2$ ($f_g = 3$); L) $f_1 - b_2$ ($f_g = 3$); M) $g_1 - b_2$ ($f_g = 5$); N) $g_1 - c_2$ ($f_g = 3$); O) $h_1 - a_2$ ($f_g = 5$); P) $h_1 - c_2$ ($f_g = 3$); Q) $i_1 - b_2$ ($f_g = 2$); R) $i_1 - c_2$ ($f_g = 1$).

*Identification and ranking of TRIZ inventive vectors*: The TRIZ inventive vectors for the 18 pairs of problems in this case study are further presented. Numbers associated to each pair represent positions of the inventive vectors in the TRIZ table of 40 inventive vectors (see Altshuller (2000)). A) $a_1 - a_2$: 14, 29, 18, 36: ; B) $b_1 - a_2$: 10, 14, 36; C) $b_1 - b_2$: 6, 35, 36; D) $b_1 - c_2$: 14, 24, 10, 37; E) $c_1 - a_2$: 29, 10, 27; F) $c_1 - b_2$: 8, 13, 26, 14; G) $c_1 - c_2$: 19, 35, 10; H) $d_1 - a_2$: 34, 23, 16, 18; I) $e_1 - a_2$: 4, 34, 19; J) $e_1 - b_2$: 15, 35, 2; K) $e_1 - c_2$: 16, 6, 19, 37; L) $f_1 - b_2$: 10, 13, 28, 38; M) $g_1 - b_2$: 21, 35, 11, 28; N) $g_1 - c_2$: 21, 11, 27, 19; O) $h_1 - a_2$: 35, 38; P) $h_1 - c_2$: 1; Q) $i_1 - b_2$: 35, 37, 10, 2; R) $i_1 - c_2$: 32, 1, 19. Just for exemplification, if we take the pair $a_1 - a_2$, the numbers 14, 29, 18 and 36 are the positions in the TRIZ table of 40 inventive vectors of the following vectors: 14 – replace a linear "approach" with a nonlinear "approach"; 29 – replace "rigid" components of the system with "fluid" components; 18 – use the "resonance frequency"; and 36 – use effects generated during a transition phase.

According to the Sigma-TRIZ algorithm, denoting with X the position of the vector in the TRIZ table of 40 inventive vectors, with Y the number of occurrences of the vector in the set of 18 pairs of problems and with Z the sum of the gravity factors of the pairs of problems having associated the respective vector, the vector is symbolized as: X (Y/Z). Thus, for this case study, the following situation occurs:

- Vectors of rank 1: {35 (6/24), 10 (6/18), 14 (4/18)};
- Vectors of rank 2: {19 (5/15), 36 (3/14)};
- Vectors of rank 3: {37 (3/8), 38 (2/8), 28 (2/8), 21 (2/8), 13 (2/8), 11 (2/8), 29 (2/7), 6 (2/7), 2 (2/7)};
- Vectors of rank 4: {34 (2/6), 18 (2/6), 27 (2/5), 26 (1/5), 15 (1/5), 4 (1/5), 8 (1/5)};
- Vectors of rank 5: {1 (2/4), 16 (2/4), 24 (1/3)}.

The inventive vectors of rank 1 should be considered as priority 1, the inventive vectors of rank 2 should be considered as priority 2 and so on. As the above data reveal, vectors with close values of the sum of their gravity factors and with close number of occurrences are included in the same rank.

*Define generic directions of intervention*: The generic directions of intervention (DI) describe possible generic facets of the inventive vectors. The reader can find them in the TRIZ table of 40 inventive vectors (Altshuller, 2000). For this case study, results are presented below.

Priority 1:
- DI1. Change the degree of flexibility (vector 35);
- DI2. Change the state "concentration" (vector 35);

- DI3. Perform in advance, completely or partial, the required actions upon the system (vector 10);
- DI4. Replace linear approaches with nonlinear ones (vector 14);

Priority 2:
- DI5. Replace a continuous action with a periodical one or with an impulse; and if the action is periodical, change its frequency (vector 19);
- DI6. Use various effects of "phase transition" (vector 36);

Priority 3:
- DI7. "Thermal expansion" (see motivation) (vector 37);
- DI8. "Strong" interactions (see transition towards optimal approaches for effort and time reduction) (38);
- DI9. Replace rigid parts with soft parts (vector 28);
- DI10. Perform harmful operations at high speed (vector 21);
- DI11. Instead of doing something according to specification, implement a completely opposite action (vector 13);
- DI12. Compensate a low reliability with some actions done in advance (vector 11);
- DI13. "Fluid" construction of the system (vector 29);
- DI14. Make the system able to perform multiple functions (vector 6);
- DI15. Remove from the system the part or property which disturbs (vector 2);

Priority 4:
- DI16. "Modify" some elements of the system during process operation once those elements have completed their tasks (vector 34);
- DI17. Use the "resonance frequency" to activate the system (vector 18);
- DI18. Replace an expensive system with several cheap systems (vector 27);
- DI19. Use simple copies instead of a single complex system (vector 26);
- DI20. Some parts of the system or of its environment must be automatically adjusted for an optimal performance (vector 15);
- DI21. Replace a symmetrical "unit" or "element" with one or several asymmetrical "units" or "elements" (vector 4);
- DI22. For a better balance, compensate the system's "weight" with another system (vector 8);

Priority 5:
- DI23. Increase the level of segmentation (vector 1);
- DI24. If it is difficult to get 100% of the intended effect, try however to achieve as much as possible from it (vector 16);
- DI25. Temporarily use an intermediary and easy replaceable system for performing some actions (vector 24).

Numbers in the brackets represent the position of the inventive vectors in the TRIZ table of 40 inventive vectors. According to the above data, 25 generic directions of intervention are proposed in this case study. With respect to these directions of intervention, adequate solutions have to be formulated.

*Formulation of innovative solutions*: This process, even if it is guided by the generic directions of intervention, is a highly creative one. Innovative solutions are actually the improvement projects which the company should implement in order to achieve the intended objective. Thus, in the next paragraphs, the term "innovative solution" is replaced with the term "improvement project". In this case study, under the guidance of the directions of intervention, the following improvement projects have been proposed:

- Project P1: Define your development process to be easy customizable to the specificity of the customer processes {DI1};
- Project P2: Flexible organized teams, according to project requirements {DI2}, {DI13};
- Project P3: Apply simultaneous engineering (some processes are run quasi-parallel: analysis-design; coding-testing) {DI3};
- Project P4: Prepare and use optimized templates for each process – new people can be very fast integrated in the company {DI3};
- Project P5: Rotate the team leadership {DI4};
- Project P6: Apply quality circles for knowledge shearing within teams and between teams {DI4};
- Project P7: Apply planning and innovation tools to help the customer in formulating its needs (provide solutions, not just execute orders) {DI4};
- Project P8: Consider rapid application development approaches to early verification of some concepts {DI4};
- Project P9: Use feature teams (your best people, for very short time) for rapid reaction in the starting phase of critical projects {DI4};
- Project P10: Cross testing {DI5};
- Project P11: Periodical review (e.g. weekly) of code samples by the best people in the company {DI5};
- Project P12: Internal audits at irregular time intervals {DI5};
- Project P13: Define performance requirements for each phase of the software development process {DI6};
- Project P14: Apply the concept of "internal client-internal supplier" {DI6};
- Project P15: Individual recognition (bonuses) {DI7};
- Project P16: Collective recognition (success bonus) {DI7};
- Project P17: Use the best people in the company in various phases of a project for advising the team {DI8};
- Project P18: Periodically, "inject" professional challenges to the team members {DI8};
- Project P19: Electronic management of all documents related to a certain project {DI9};
- Project P20: Use creativity techniques (e.g. mind-map) and innovative problem solving techniques (e.g. TRIZ, ASIT) in the design phase of a project {DI9};
- Project P21: Use spiral development model to approach highly innovative projects {DI4}, {DI10};
- Project P22: Urgent change of a member if he/she does not handle the project {DI10};
- Project P23: When problems occur, find solutions to improve the process not to blame the team {DI11};
- Project P24: Think to contingency plans from the early phases of a project {DI12};
- Project P25: Negotiate projects considering pessimistic scenarios {DI12};
- Project P26: Consider periodical „back-up" actions {DI12};

- Project P27: Start development with „C” in the PDCA cycle {DI12};
- Project P28: Multi-qualified staff and multiple roles in the project (e.g. the project manager has also some tasks of coding and testing) {DI13};
- Project P29: Fast and facile communication between the top management and the team members (breakdown the communication barriers) {DI14};
- Project P30: Flexible teams, of variable size, in various phases of the project {DI16};
- Project P31: Use external consultants for certain phases of a critical project {DI16}, {DI25};
- Project P32: Daily communication, in multiple modes, with the customer (e-mail, messenger, phone, etc.) {DI17};
- Project P33: Apply 360° review {DI17}, {DI21};
- Project P34: Before delivery, perform multiple and various module tests {DI18}, {DI19};
- Project P35: Flexible organization (dynamic-oriented teams) {DI20};
- Project P36: More time allocated to P and C within PDCA cycle {DI21};
- Project P37: Focus on solutions not on service (thus, the process of analysis, design and technical innovation should be highly mature) {DI4}, {DI22};
- Project P38: Monitor team profitability (autonomous profit units) {DI23};
- Project P39: From time-to-time, for very short periods, apply tele-work {DI23};
- Project P40: Rotate team members in projects to avoid monotony {DI24};
- Project P41: Better adaptation to project diversity by increasing flexibility in customer selection {DI24};
- Project P42: Subcontract auxiliary functions {DI25}.

To each project, one or more directions of intervention are associated (see the symbols in the brackets { }). None of the 42 projects are in conflict each other, thus none of them should be removed. According to step 11 of the Sigma-TRIZ algorithm, a correlation index can be associated to each project (see the relationship (5)). This information is very useful to establish priorities in starting the implementation of the projects in the list. However, this process should not be applied in a "mechanical" way. First of all, projects should be analyzed; and selected from the list those which can start immediately (e.g. because they are simple rules; because do not involve so much resources and time, etc.). There should be kept for prioritization only those projects which effectively involve more resources and time.

Actually, in this case study, from the set of 42 projects, a subset of 27 projects can be distributed to various units of the company to be implemented immediately, as long as they primarily describe good-practice rules and simple working routines. In this category are included the following projects: P2, P3, P5, P6, P9, P10, P11, P12, P14, P15, P16, P17, P18, P22, P23, P24, P25, P26, P27, P28, P29, P30, P31, P32, P34, P36, P42. The remaining subset, constituted from 15 projects: P1, P4, P7, P8, P13, P19, P20, P21, P33, P35, P37, P38, P39, P40, P41, should be further analyzed for prioritization.

*Project prioritization*: For the subset of 15 projects above mentioned, a matrix of correlation is set up. The results are shown in Table 1. In the matrix, the correlations between projects and their correlation indexes are put into evidence. According to the results in Table 1, project P4 is of first priority, followed by projects P1, P7, P8 and so on. This priority is not necessarily the order in which implementation will effectively happen in practice.

| | P1 | P4 | P7 | P8 | P13 | P19 | P20 | P21 | P33 | P35 | P37 | P38 | P39 | P40 | P41 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | 27 | 27 | 27 | 9 | 27 | 27 | 27 | 9 | 27 | 27 | 27 | 9 | | 9 |
| P4 | 9 | | | | 1 | 27 | | | | | | 3 | | 9 | 27 |
| P7 | 9 | 9 | | 9 | | | 27 | 9 | | 3 | | | | | |
| P8 | 9 | | 9 | | | | 1 | 27 | | | | | | | |
| P13 | 9 | 9 | | | | | | | | | | 3 | | | |
| P19 | 9 | 27 | | | | | | | | | | | | | |
| P20 | 9 | | 27 | 9 | | | | 1 | | | | | | | |
| P21 | 9 | 9 | 27 | 27 | | | 27 | | | 1 | | | | 1 | |
| P33 | | 9 | | | | | | | | 27 | | 3 | | 3 | |
| P35 | 9 | 27 | | 9 | | 9 | 1 | | 1 | | 9 | 9 | | 9 | 27 |
| P37 | 9 | 27 | | | | 9 | | | | 9 | | 3 | | | |
| P38 | 9 | 9 | | | 3 | 9 | | | 3 | | | | | 9 | |
| P39 | 3 | 27 | | | | 27 | | | | | | | | | |
| P40 | | 27 | | | | 9 | 1 | | | 27 | | 9 | | | |
| P41 | 27 | | | | | 27 | 27 | | | 27 | 3 | | | | |
| $z_i$ | 24 | 18 | 18 | 18 | 14 | 8 | 8 | 18 | 6 | 5 | 18 | 4 | 4 | 4 | 4 |
| $\Sigma a_i$ | 120 | 207 | 90 | 81 | 13 | 144 | 111 | 64 | 13 | 121 | 39 | 57 | 9 | 31 | 63 |
| $C_i$ | 2880 | 3726 | 1620 | 1458 | 182 | 1152 | 888 | 1152 | 78 | 605 | 702 | 228 | 36 | 124 | 252 |
| Priority | 2 | 1 | 3 | 4 | 12 | 5 | 7 | 6 | 14 | 9 | 8 | 11 | 15 | 13 | 10 |

Table 1. Prioritization of the subset of 15 projects

Results in Table 1 highlight the importance which a well-documented, highly mature quality management system plays in the equation of competitiveness of a software outsourcing company (P4). They also show how important is to define a customizable software development process (P1), as well as to provide solutions, not just being a simple executive (P7) and to consider evolutionary approaches in software development (P8, P21). An important role is also played by running the whole process in a virtual environment (P19), etc.

Please remember that Table 1 does not include the rest of 27 projects in the analysis. The fact these projects are simpler in terms of time and resources required for their implementation does not necessarily mean they are less important than the 15 projects analyzed in Table 1. Only an exhaustive analysis could give an answer to this issue. However, for the scope of improvement, such discussions are not at all relevant as long as all projects have to be implemented in order to set up a mature business process.

*Implementation plan*: Implementation of projects requires several actions, like: definition of implementation means, resources, scheduling, etc. This task strongly depends on the context in which the company operates; therefore, this part of the exercise is not included here for presentation.

### 3.5 "Control" (C) phase within Six Sigma DMAIC methodology

This phase includes two important steps: a) development of the monitoring plan; and b) institutionalization of improvements (e.g. by means of procedures, rules, instructions, records, etc.). These kinds of issues overpass the scope of this paper, thus they are not further treated here.

## 4. Conclusion

Conflicts occurring between various project actions represent the true barriers in setting up mature business processes. However, identification of the "real" conflicts is not a simple task. To this, formulation of effective solutions to various problems requires innovations in many cases. Starting from these premises, this chapter explores the integration of innovative problem solving tools within Six Sigma DMAIC methodology to increase the effectiveness of process improvement projects. The result is a novel algorithm for systematic approach of the "Improve" phase within Six Sigma DMAIC methodology. It is called Sigma-TRIZ algorithm. The success in formulating effective solutions of process improvement with Sigma-TRIZ is direct proportional with the capacity of solving without compromises conflicts within the analyzed process. This effort is guided by the paradigm of ideality and the paradigm of convergence.

Some other concluding remarks derive from the case study presented in this chapter. First, improvements within business processes cannot be effectively done without systematic approaches, where innovation plays a key role. Second, generation of visible results in the effort of process improvement necessitates identification and implementation of convergent, positive correlated improvement projects. Third, understanding the impact of each project in the equation of process improvement increases the level of effectiveness, especially for those situations where resources are limited. Fourth, the number of improvement projects is proportional with the number of conflicts occurring in the process. Fifth, initiatives of process improvement are not so simple, requiring simultaneous implementation of more improvement projects. Without a clear commitment and support from the top management, such initiatives could fail. Sixth, to have mature business processes, a clear focus should be on extended business models, where customers are key parts of the business process.

The research in this paper is also subjected to some limitations which open new opportunities for further works. In this respect, consideration of a single tool for inventive problem solving (specifically, the TRIZ method) would be seen a restrictive condition. In fact, a space for exploring similar methods within the Six Sigma DMAIC methodology is open. A kind of drawback in the Sigma-TRIZ algorithm comes up from the fact that TRIZ method, by itself, raises up some difficulties in being properly use by usual people; but "without pain, no gain".

In addition, results presented in the case study could be of real support for people operating in the field of software outsourcing industry. In the same register, the case study details a significant part of a Six Sigma DMAIC methodology, being from this perspective a useful guide for practitioners in the field of quality management, as well as an instructive material for students.

## 5. Abbreviations

DMAIC = Define – Measure – Analyse – Improve - Control.
TRIZ = Theory of Inventive Problem Solving.
QFD = Quality Function Deployment.

## 6. References

Altshuller, G. (2000). *The Innovation Algorithm: TRIZ*, Technical Innovation Center, ISBN 0-9640740-4-4, Worcester

Brad, S. (2008). Integration of innovation tools within Six Sigma process improvement methodologies, *Proceedings of the International Conference Quality-Innovation-European Integration-Special Issue Calitatea AS*, Vol. 93, No. 1, pp. 66-72, ISSN 1582-2559, Romania, September 2008, Publisher Calitatea AS, Sibiu

Cascini, G.; Rissone, P. & Rotini, F. (2008). Business re-engineering through integration of methods and tools for process innovation, *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, Vol. 222, No. 12, pp. 1715-1728, ISSN 0954-4054, Italy, December 2008, Publisher Professional Engineering Publishing Ltd, Florence

Cronemyr, P. (2007). DMAIC versus DMADV. Differences, similarities and synergies. *International Journal of Six Sigma and Competitive Advantage*, Vol. 3, No. 3, pp. 193-209, ISSN 1479-2494

Hamza, A. (2008). Design process improvement through the DMAIC six sigma approach. *International Journal of Six Sigma and Competitive Advantage*, Vol. 4, No. 1, pp. 35-47, ISSN 1479-2494

Jean-Ming, C. & Jia-Chi, T. (2004). An optimal design for process quality improvement: modeling and application. *Production Planning and Control*, Vol. 14, No. 7, pp. 603-612, ISSN 0953-7287

Khavarpour, M.; Asadian, N.; Hoseinpour, S. & Malihi, H. (2008). A new approach to cost reduction with TRIZ method, *Proceedings of the 38th International Conference on Computers and Industrial Engineering*, Vol. 1-3, pp. 1551-1557, ISBN 978-7-121-07437-0, China, November 2008, Publisher Publishing House Electronics Industry, Beijing

Qi, M.; Dai, Q.Q.; Lin, Q.Y. & Shang, W.F. (2008). The design of enterprise innovation model that integrated TRIZ into six sigma, *Proceedings of the International Workshop on Computer Science and Engineering*, pp. 418-421, ISBN 978-1-84626-172-5, Russia, May 2008, Publisher World Acad Press, Moscow

Silverstein, D.; DeCarlo, N. & Slocum, M. (2005). *In-sourcing Innovation: How to Transform Business as Usual into Business as Exceptional*, Breakthrough Performances Press, ISBN 0-9769010-0-5, Longmont

Smith, L. & Pahdke, M. (2005). Some thoughts about problem solving in a DMAIC framework. *International Journal of Six Sigma and Competitive Advantage*, Vol. 1/2, pp. 151-166, ISSN 1479-2494

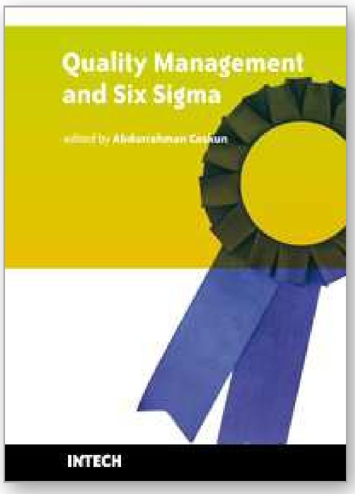
Thawani, S. (2002). Six sigma: strategy for organizational excellence. *Total Quality Management & Business Excellence*, Vol. 15, No. 5-6, pp. 655-664, ISSN 1478-3371

Treichler, D.; Caramichael, R.; Kusamanoff, A.; Lewis, J. & Berthiez, G. (2002). Design for six sigma: 15 lessons learned. *Quality Progress*, Vol. 35, No. 1, pp. 33-42, ISSN 0033-524X

Zhao, X.J. (2005a). Integrated TRIZ and six sigma theories for service/process innovation, *Proceedings of the International Conference on Services Systems and Services Management*, Vol. 1-2, pp. 529-532, ISBN 0-7803-8971-9, China, June 2005, Publisher IEEE, Chongqing
Zhao, X.J.; Feng, C.X. & Zhou, X. (2005b). Integrating TRIZ and QFD into six sigma processes for satisfying customers, *Proceedings of the 2005 International Conference on Management Science & Engineering (12th)*, Vol. 1-3, pp. 732-736, ISBN 7-5603-2189-5, Korea, July 2005, Publisher Harbin Institute Technology Publishers, Incheon

## Appendix: TRIZ-generic parameters

| No. | Characteristic | No. | Characteristic | No. | Characteristic |
|---|---|---|---|---|---|
| 1 | Weight of moving object | 14 | Strength | 27 | Reliability |
| 2 | Weight of nonmoving object | 15 | Durability of moving object | 28 | Accuracy of measurement |
| 3 | Length of moving object | 16 | Durability of nonmoving object | 29 | Accuracy of manufacturing |
| 4 | Length of nonmoving object | 17 | Temperature | 30 | Harmful factors acting on object |
| 5 | Area of moving object | 18 | Brightness | 31 | Harmful side effects |
| 6 | Area of nonmoving object | 19 | Energy spent by moving object | 32 | Manufacturability |
| 7 | Volume of moving object | 20 | Energy spent by nonmoving object | 33 | Convenience of use |
| 8 | Volume of nonmoving object | 21 | Power | 34 | Repairability |
| 9 | Speed | 22 | Waste of energy | 35 | Adaptability |
| 10 | Force | 23 | Waste of substance | 36 | Complexity of device |
| 11 | Tension/Pressure | 24 | Loss of information | 37 | Complexity of control |
| 12 | Shape | 25 | Waste of time | 38 | Level of system automation |
| 13 | Stability of object | 26 | Amount of substance | 39 | Capacity or productivity |




**Quality Management and Six Sigma**

Edited by Abdurrahman Coskun

ISBN 978-953-307-130-5

Hard cover, 276 pages

**Publisher** Sciyo

**Published online** 16, August, 2010

**Published in print edition** August, 2010

If you do not measure, you do not know, and if you do not know, you cannot manage. Modern Quality Management and Six Sigma shows us how to measure and, consequently, how to manage the companies in business and industries. Six Sigma provides principles and tools that can be applied to any process as a means used to measure defects and/or error rates. In the new millennium thousands of people work in various companies that use Modern Quality Management and Six Sigma to reduce the cost of products and eliminate the defects. This book provides the necessary guidance for selecting, performing and evaluating various procedures of Quality Management and particularly Six Sigma. In the book you will see how to use data, i.e. plot, interpret and validate it for Six Sigma projects in business, industry and even in medical laboratories.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Stelian Brad (2010). Sigma-TRIZ: Algorithm for Systematic Integration of Innovation within Six Sigma Process Improvement Methodologies, Quality Management and Six Sigma, Abdurrahman Coskun (Ed.), ISBN: 978-953-307-130-5, InTech, Available from: http://www.intechopen.com/books/quality-management-and-six-sigma/sigma-triz-algorithm-for-systematic-integration-of-innovation-within-six-sigma-process-improvement-m



**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821