INT J COMPUT COMMUN, ISSN 1841-9836
8(2):235-246, April, 2013.

Quality System for Production Software as Tool for Monitoring and Improving Organization KPIs

V.C. Kifor, N. Tudor

Vasile Claudiu Kifor
Lucian Blaga University of Sibiu,
10 Victoriei Blv., 550024, Sibiu, Romania
E-mail: claudiu.kifor@ulbsibiu.ro

Nicolae Tudor
Conti Temic Microelectronic GmbH
Sieboldstrasse 19, D-90411 Nürnberg, Germany
E-mail: nicolae.tudor@continental-corporation.com

Abstract: In this paper we propose a solution as support for quality systems for production software. The motivation behind this study was to reduce that cost in the production area caused by gaps in the quality of the production software. Our proposal: QSPS (Quality System for Production Software) is offering support in the "vulnerable points" of these quality systems which usually generate nonconformities and have proved to be difficult or impossible to control. QSPS is a method in seven steps or modules that integrates also software tools, templates, checklists, evaluating tools elaborated complying to products, process and system quality standards.
If other analyzed methods like: Scrum, XP, Fuzzy, Prompt, PTA, PRINCE2 or norms like: ISO 9001, ISO 9000-3, TickIT, CMM and CMMI, AQAP-110/AQAP-150, IEEE 730/983 are working in a reactive way, after the developing phase was finished, QSPS is an active system, helping the software developer from the beginning of the implementation phase to improve the developing methodology and to fulfill the quality requirements.
QSPS model was applied in one of the largest European automotive company, the result being finalized in a practical approach of the QSPS, named QSMA − Quality System for Manufacturing Application.
Using the QSMA (Quality System for the manufacturing application) for industrial projects and not only therefore, has led to accurate running of the production line from beginning of the SOP (Start of Production).
Once this system was implemented and the production software applications were realized under the principles and rules of the QSMA, we defined strategically measurable KPIs (key performance indicators) out of the seven modules of the QSMA. This KPIs have the role to signal every time a production application has not the desired quality level and presents a high level of risk that could cause additional costs in the production.
Based on the KPIs evolution, the weaknesses in the software applications can be identified in real time, so that the developer can react immediately, before occurrence of substantial damage.
Keywords: quality improvement, control, monitoring, efficiency, capability, performance, organization KPI (key performance indicator).

1 Introduction

This paper describes the deployment of software applications that is used in the production by using different product quality models, it identifies the flaws of these models and proposes a new quality model (QSPS − Quality System for Production Software) and its important KPIs (Key performance indicators), that is designed for the development of software applications used in the automotive industry, a model that includes process and product norms and recommendations of use. The study is based on case studies of application of the most well-known quality standards and models for the development of a software application for production; it identifies the deficiencies of already used methods and proposes a model that is based on the quality requirements in the automotive industry, such as ISO/TS 16949, with an optimal frame for production. We analyzed management methods for software applications, such as Scrum, XP (extreme programming), Fuzzy, Prompt, PTA, PRINCE2 and quality norms such as ISO 9001, ISO 9000-3, TickIT, CMM and CMMI, AQAP-110/AQAP-150, IEEE 730/983.

In order to identify the requirements of the automotive industry, we carried on the study and we have analyzed and applied the ISO/TS 16949 to the earlier mentioned software project, nevertheless, we have analyzed also quality methods, such as Automotive SPICE, ISO/IEC 25000/9126/14598. The choice of quality models for the case study was based on the following:

- the most representatives and/or
- the most frequently used in practice.

Due to special characteristics of software applications that accompany production (difficulty of testing, their impact upon the quality of the finite product), as well as their importance in assuring the continuity of the business environment in an organization (production control), these types of software application have to be treated in a particular way, different than other types of software, whose norms can be found in specialty literature. These differences of production software applications are shown in **Figure 1**:

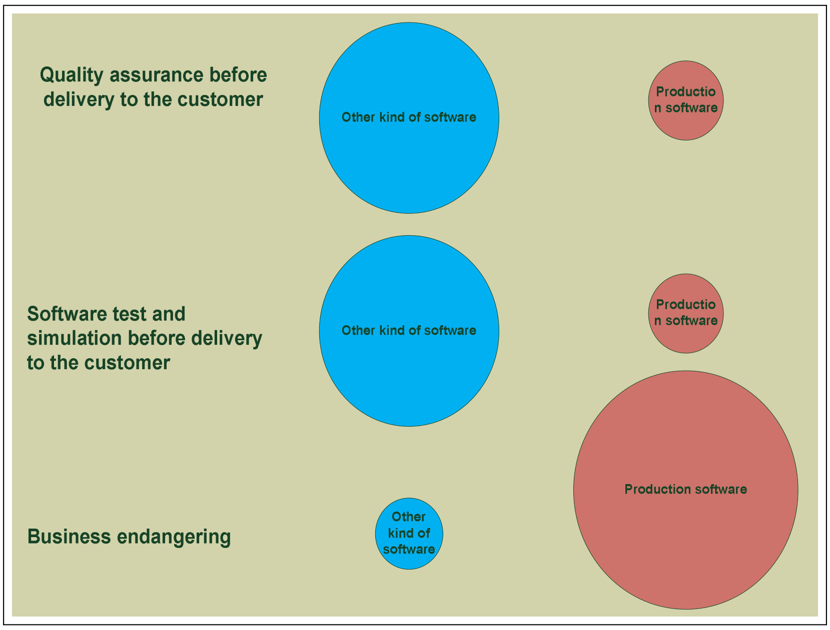

Figure 1: Major differences among production software application

The quality of production software should address two related but distinct notions [16]: functional quality (or fitness for purpose) – reflects how the software complies with specific requirements of the customer or conforms to specifications; structural quality – reflects how the production software meets non-functional requirement that support the delivery of functional requirement, such as robustness or maintainability, the degree to which the software was produced correctly.

Production software quality measurement is about quantifying to what extent a software or system rates along each of five dimensions: reliability, efficiency, security, maintainability and (adequate) size [14].

A quality system for production software should be part of organization quality management system and can be expressed as objectives and means (processes, procedures, tools and responsibilities) designed to fulfill these objectives [8]. The objectives from the above definition are usually addressing customer (external and internal) focus, compliance with standards (effectiveness) waste reduction and a better use of resources (efficiency) and continual improvement. Quality systems for production software are designed according to the requirements of the quality management standards (ISO 9001), software engineering standards (ISO 9000-3, ISO/IEC 9126 etc.), information security standards (ISO 27000), and specific sectors standards (ISO TS 16949 for automotive).

In our view [17], a method for development, implementation and maintaining a production software should be based on traditional principles and methods of software engineering and project management, but has to incorporate specific features in order to cover the particularity of such products:

- support and cover specific gaps of the quality systems for production software;
- degree of difficulty and time required to implement;
- system efficiency in saving resources, costs (or avoid potential additional costs), to control and ensure the project's completion;
- high transparency of project status and difficulties faced by using the method;
- software quality assurance is not to endanger the "normal" flow of production;
- adaptability of the model for extreme situations and different types of such softwares;
- system feasibility;
- easy for managers to understand the software, even with a low level of technical knowledge;
- reduce the communication gap with the involved departments.

2 Objectives

The purpose of the QSPS is to improve the effectiveness and efficiency of the quality systems for production software; applying QSPS correctly would result in a proper running of the production software, a high quality product according to customer and standard requirements and lower production costs.

Effectiveness and efficiency represents today some of the most important Key Performance Indicators and a permanent concern for every organization. While effectiveness is focusing on complying with specific quality outputs and standards requirements, efficiency aims higher results with lower resources [8].

As effectiveness and efficiency are considered to be very general, and difficult to define and measure, specific indicators and objectives are derived from the general objective and proposed for QSPS project, in the **Table 1**.

The objectives are addressing customer (external and internal) focus, compliance with standards (effectiveness) waste reduction and a better use of resources (efficiency) and continuous improvement. QSPS will support the quality system for production software with a method in seven steps or modules that integrates also software tools, templates, checklists, evaluating tools and an experienced database with case based reasoning. QSPS will bring higher effectiveness for the quality system for productions software and give support for reaching more efficient operations.

Objectives/Needs	**Indicators**	**Targets**
Improve the quality level of the production software from product perspective	FPY (first pass yield)	98%
	DPMO (defects per million opportunities)	<10
Reduce the risk of the impact of production software on product quality	CPK	CPK 1,5 (means 6 Dpmo)
Reduce the risk of delay for rump-up	Rate and cycle time of the line. Downtime of the production line due to software deficiency	0
Reduce additional costs caused by production software	Software returns	0
	Software downtime	0
Improve the capability of the production infrastructure, to eliminate the entire problems caused by the production software, so that the production achieves the maximum of yield	At least 2,5% from entire lot of day production and 50% of analyse work per day by specialized engineer and technician, meaning a 10% better yield of the production line.	
Transparency in classification of production problems (Process vs. software)	Identification time	Less than 5 min.
Increase customer satisfaction	Supplier evaluation by customer	90% satisfaction
Quality system audit results - ISO 9001, ISO TS 16949	System compliance levels	95%

Table 1: QSPS Objectives

3 QSPS integration

Based on the objectives proposed, we applied the QSPS in one of the biggest automotive company in the world. Because of the special characteristics of the company, and the existing software lifecycle, the customized QSPS became the name QSMA (Quality system for manufacturing application).

QSMA (Method and tools for improving the efficiency and effectiveness of the quality systems for manufacturing Software) aims to support the quality systems for production software, such systems been developed according to the specific standards specifications. This architecture will be integrated in the Production Information System Lifecycle that controls the product development and industrialization processes. QSMA will give support in the "vulnerable points" of these quality systems which usually generate nonconformities and have proved to be difficult or impossible to control (**Figure 2**).

QSMA is a method in seven steps or modules that integrates also software tools, templates, checklists, evaluating tools elaborated complying to products, process and system quality standards, described also in the next chapter.

4 QSMA practical results

The technical and the economical departments from the automotive company have analyzed, in the period 2010-2011, the QSPS model proposed and have confirmed its feasibility and

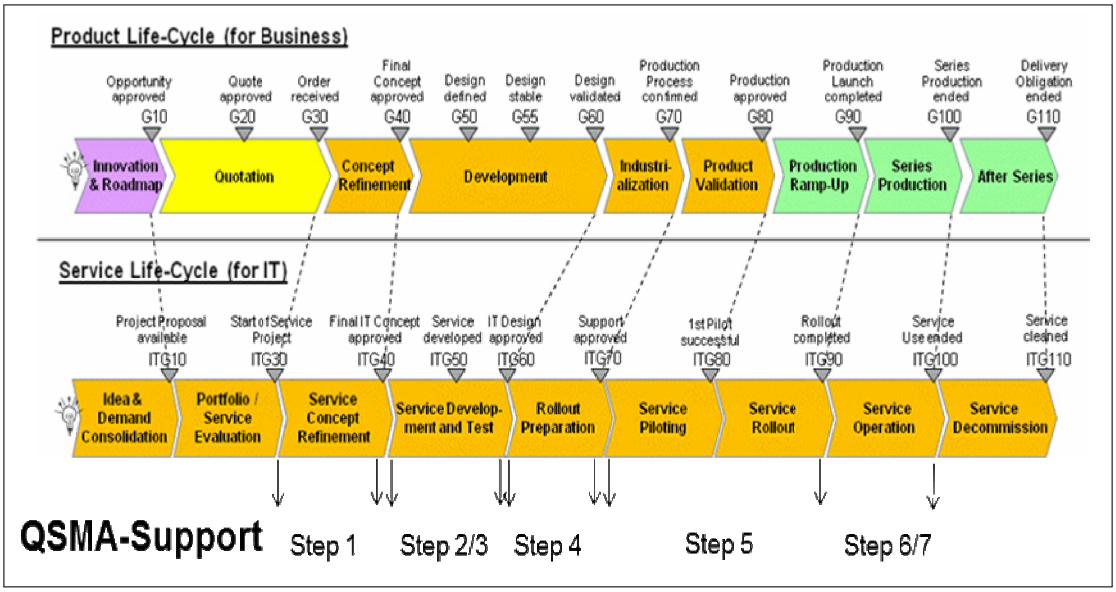

Figure 2: Integration of QSMA

the importance of its implementation in practice.

Therefore, the pilot process of implementation has been approved. This pilot process had considered five representative units of production from this company: Core System, Reporting Solutions, Integration (Client) Solutions, Communication Middleware and Integration Frameworks.

The Quality System for Manufacturing Applications (QSMA), developed on the QSPS structure, will be integrated in MES – **M**anufacturing **E**xecution **S**ystem, in the 92 plants of the company. QSMA comprises 7 stages that will be individually applied for every department from the ones specified above. These stages contain templates, instructions and checklist, as follows:

Step 1:

- Checklist Service development – criteria to start (Functional Spec. template, Training, Service level Agreement, Development closure characteristics etc.);
- Time and cost evaluation (on Component level: reporting, Equipment software etc.);
- Project organization/Software development;
- RASI Chart (who is responsible for what);
- Handover/Q-gates definition (what to do when).

Step 2:

- Process for developer – processing in application (presentation);
- IT requirements – non functional:
  - Coding guideline (Adaptability/ test adaptability/ Stability/ portability/ coexistence/ changeability/ re-use study);
  - GUI guideline;
  - Performance guide line;
  - Test Guideline.
- IT Requirements compare cross check to Business requirements (cover ability check) – Functionality compliance check frequency and timing;
- Best practices (check list) → plugin tcal frames.net;

- Document & source code control & process (version control – "Subversion");
- IT list of risks (template), Technical risk (FMEA);
- IT security requirements (checklist), see IT at conti.;
- MA Dev. Environment checklist-for example Conti configuration model for camline.

Step 3:

- Requirements and recommendations for verification of the software (validation checklist):
  - Design and development review/ verification/validation (checklist);
  - Code Review/Security & Vulnerability Testing;
  - Classification and verification of the software functionality in accordance to the defined IT requirements (performance test) (checklist);
  - Classification and verification of the software functionality in accordance to the defined business requirements. (requirements test).
- Process for verification and validation: template for verification criteria and result.

Step 4:

- Internal criteria of acceptance and approval of the implemented software kit by the developers before delivering it to the client. (check list) (software PPAP);
- Acceptance protocol (with approval signatures) (template) (RASI from Step 2);
- User manual (form and template, content instructions);
- Deviations from specifications (document template/record medium: sharepoint... etc. to document requests/solutions and testing results after implementation);
- Internal functional & process audit for software (score list);
- Accuracy/maturity evaluation and classification (notification criteria for alfa, beta, release... version, definition of software maturity classification rules) (instructions);
- Usability: – Fault tolerance understand ability/ learn ability analysis (instructions);
- Efficiency compliance (statistical reports: iGATE?, capability studies, software CPK, lifetime study) (checklist).

Step 5:

- Verification and approval of the implemented software kit by the customer. (functionality test-Run @ Rate);
- Process of BUGs handling (environment definition test-integration- production) and
- Recording of BUGs situations and solution (lesson learned for developer & Bug Tracking);
- Test scenario based on requirements-specification & implementation (action → expected result → run result);
- Statutory and regulatory conformity check (licenses, local law regulation);
- System integration (rollout procedure: integration system → production system);
- Acceptance and release criteria & Release process and protocol.

Step 6:

- Recommendations and requirements for functional monitoring and measurement of the software product during the production (instructions);
- Monitor results and long term improvements (template);
- Process of software validation and verification extension (template proposal);
- Calculation of ipm (number of software incidents per 1 million executing);
- Process of corrective and preventive action (chart);

- Process of predictive actions (chart);
- Software quality assessment and capability (check/score list);
- Control and record of quality data collection and results (proposal for data structure: sharepoint, iGate Reports);
- Software control plan and handle of bugs in alignment with IT Incident Mng.;
- Lesson learned, collection medium (sharepoint etc.);
- QSMA dynamic adjustment and improvement. Potential Improvement measurement. (Mathematical simulation of known MES software: use software like Statistica).

Step 7:

- Requirements and recommendations for handling customer complaints (instructions);
- Technical support agreement (contract template);
- Problem reporting/monitoring/recording and closing procedure (by customer acceptance of the solution) (system proposal: remedy, HP Service Management);
- Process of change management and problem/incidents resolution (templates for Change Request, templates for incidents: 8D... etc.);
- Training (process, methods and rules) → template with content recommendation;
- Customer communication process (methods, communication medium: e-mail, sharepoint, Portal);
- Periodical customer feedback evaluation/ customer satisfaction reports(template);
- Statistics: over problems and solutions, reaction times, and solution efficiency: statistic proposal like PERT diagram).

5 Key performance indicators

The preliminary data analyzed after implementation confirm a reduction of at least 60% of potential losses caused by software. It was preliminary confirmed the improvement of certain KPIs such as **Figure 3**.

- Functionality compliance:
  - business requirements cover ability: 95%
  - deviations from requirements: 1%
  - missing functionality during verification: 0%
- DPMO (defects per million opportunities): < 10
- downtime of the production line due to software deficiency: 0
- software returns: 0
- supplier evaluation by customer: 90% satisfaction
- non functional compliance: 95%
- software availability: 99,99%

This KPIs have the role to signal every time a production application has not the desired quality level and presents a high level of risk that could cause additional costs in the production.

Based on the KPIs evolution, the weaknesses in the software applications can be identified in real time, so that the developer can react immediately, before occurrence of substantial damage.

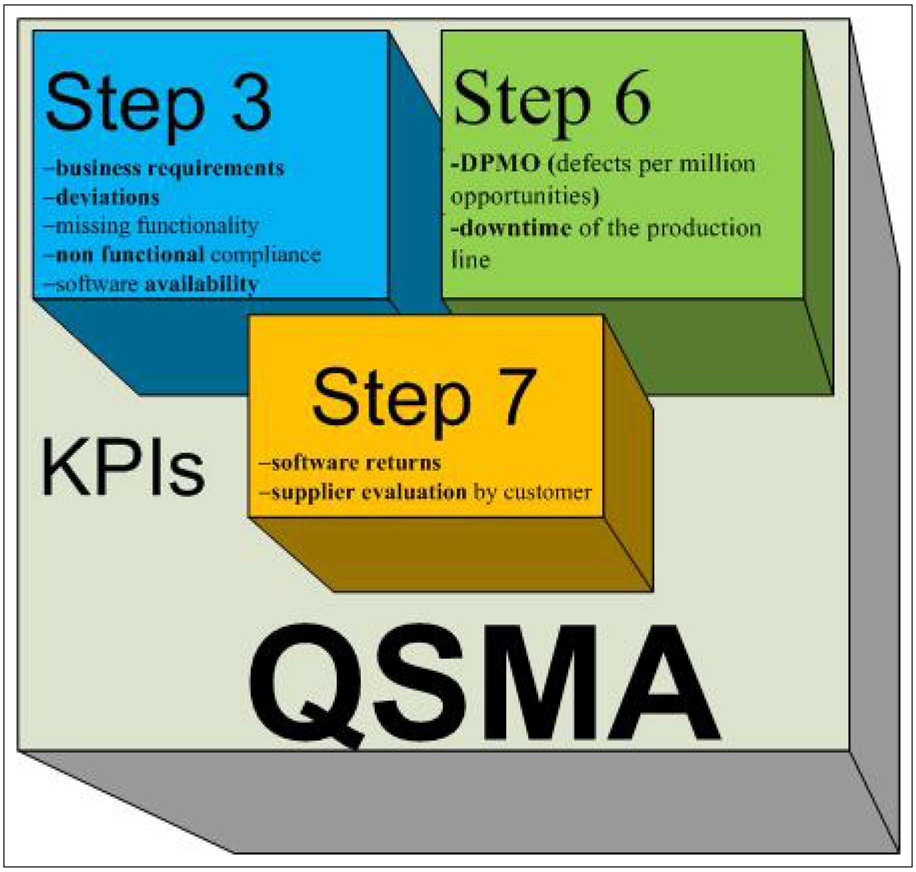

Figure 3: QSMA - KPI overview

6 QSMA continuous improvements

The scoring of the QSMA system means a solid base of data entry in mathematical modeling; thereby the system can be monitored and subjected to continuous improvements (**Figure 4**).

The QSMA system can be taken as a sum of combination quality system elements in the automotive industry ISO/TS 16949 and that of software requirements of this industry Automotive SPICE together with all other elements of the individually studied methods, with the help of formula 2:

$$QSMA = \{E1, E2, E3, E4, E5, E6, E7\} \tag{1}$$

Where the E1÷E7 are the best combination of the elements of the known quality methods.

$$E_1 = \sum_{i=1}^{8} E^o_{1ISOTS16949} E^o_{1AutomotiveSPICE} E_{1M_i} = E^o_{1ISOTS16949} E^o_{1AutomotiveSPICE} \sum_{i=1}^{8} E_{1M_i} \tag{2}$$

$$E_2 = E^o_{2ISOTS16949} E^o_{2AutomotiveSPICE} \sum_{i=1}^{8} E_{2M_i} \tag{3}$$

$$E_3 = E^o_{3ISOTS16949} E^o_{3AutomotiveSPICE} \sum_{i=1}^{8} E_{3M_i} \tag{4}$$

$$E_4 = E^o_{4ISOTS16949} E^o_{4AutomotiveSPICE} \sum_{i=1}^{8} E_{4M_i} \tag{5}$$

$$E_5 = E^o_{5ISOTS16949} E^o_{5AutomotiveSPICE} \sum_{i=1}^{8} E_{5M_i} \tag{6}$$

$$E_6 = E^o_{6ISOTS16949} E^o_{6AutomotiveSPICE} \sum_{i=1}^{8} E_{6M_i} \tag{7}$$

$$E_7 = E^o_{7ISOTS16949} E^o_{7AutomotiveSPICE} \sum_{i=1}^{8} E_{7M_i} \tag{8}$$

So the mathematical representation of the QSPS can be as followed:

$$QSMA = \{E1, E2, E3, E4, E5, E6, E7\} = \sum_{i=1}^{7} E^o_{iISOTS16949} E^o_{iAutomotiveSPICE} \sum_{j=1}^{8} E_{jM_i} \tag{9}$$

The obtained result of the software evaluation with QSPS is a directly proportional function with the arithmetical average of these elements as followed:

$$Evaluation = f(QSMA) = \frac{\sum_{i=1}^{7} E_i}{7} \tag{10}$$

In order to prove how close to these values are to reality, we created the simulation in **Figure 4**, in which we calculated the average value that can be obtained from each element, regardless of the applied method, followed by a simulation of elements, to show their impact on the end result and the satisfaction of customers.

As a result of these simulations, we can sum up the fact, that the already existing methods for software products deeply focus on Element 2 − Software Specific Requirements (management, risk, quality and security) and 3 − Validation and Verification of the Implemented Software Application, so that we have paid an increased attention to the other elements in the implementation of the QSMA system.

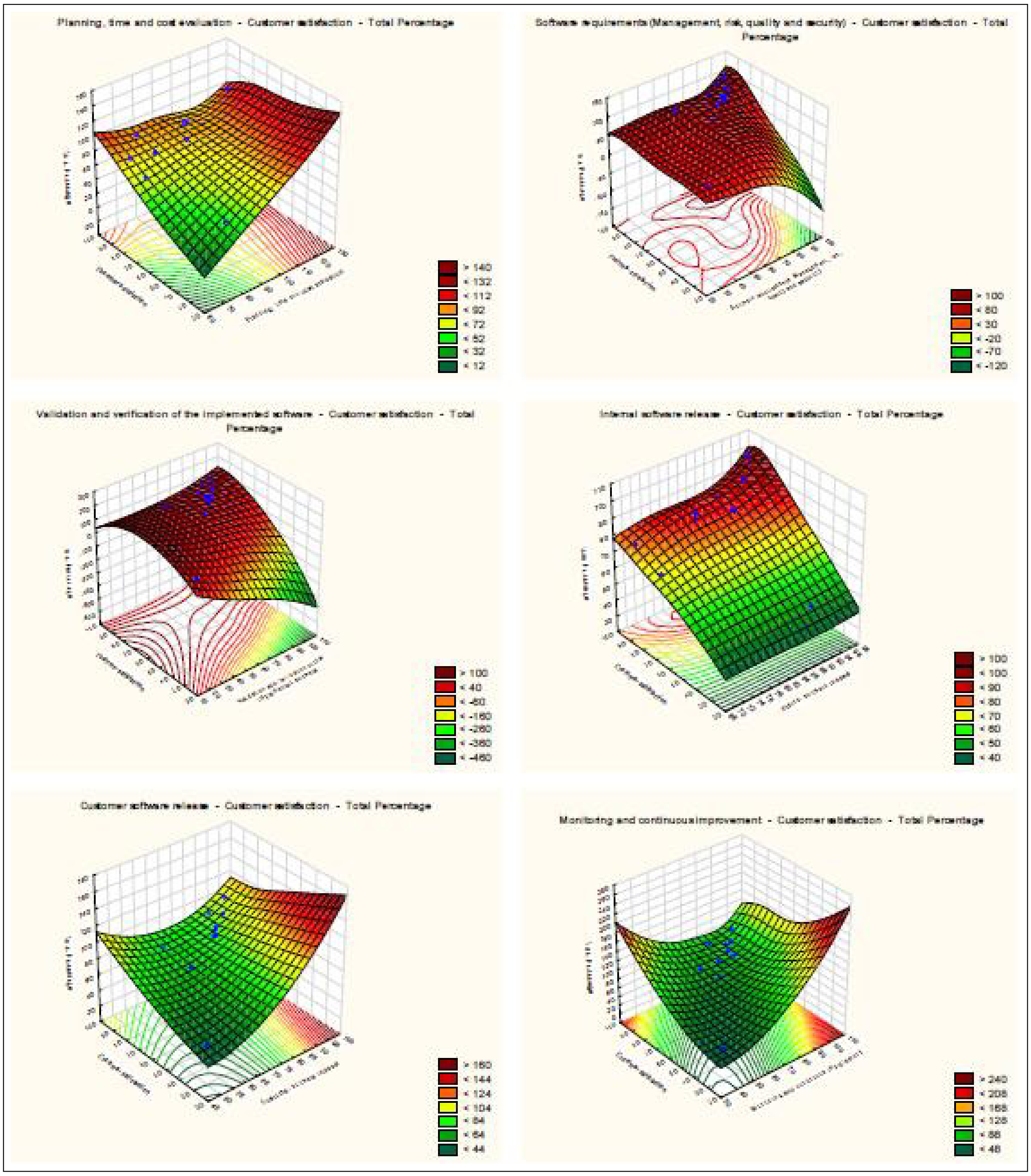

Figure 4: Mathematical modeling of the Method QSMA

7 Conclusions

The QSMA system leads to assurance of quality of production software applications and in the same time to avoidance of extra costs, that can occur in production due to software applications. Other direct advantages, proved to be efficient in practice are:

- assurance of quality of MES software applications in the production of mechanical components for motor vehicles;
- helps in the development of sturdy MES software applications;
- diminishes potential costs that can occur due to software errors and therewith the breakdown of production;

- offers high transparency in the classification and identification of production problems (process problems versus MES software problems);
- acceptance of MES solutions by the customer and verification of fulfillment or requirements;
- growth of customer satisfaction.

The experimental system can be further improved and adapted to the actual needs that occur during practical situations.
The projection of this system was created after a long study, that was the result of the growing need of a method that supports the deployment of software applications in the automotive industry, so that these fulfil the requirements of quality standards in this area assure a normal flow of production in order to avoid economic losses on organizational level.

The Software Quality Management gains more and more important in all areas of development and the economy, due to the following criteria [10]: the software quality is an exponential factor in competition, led by the growing awareness of the qualitative section of the beneficiaries; the error correction of the software has proved to be very expensive, but this can be prevented by early introduction and used of the quality management systems.

It is noteworthy the study conducted within 178 industrial companies where the behaviour and the performance were compared before and after strengthening project management discipline. Thus it was observed that the success rate of projects in terms of quality, program duration and predictability of project cost has increased considerably with the institutionalization of the management element [9].
The management of software projects requires much higher level of discipline for the team involved in project unlike the methods for the management of the "traditional" projects.

Bibliography

[1] Kai-Yuan Cai, Bey-Bey Yin, Software execution processes as an evolving complex network, *Information Science*, 179(12): 1903-1928, April 2008.

[2] Georg Kühner et al., Employing industrial standards in software engineering for W7X, *Fusion Engineering and Design*, 84(7-11): 1130-1135, 2009.

[3] J.-W. Li, Modeling a quality assurance information system for product development projects with the UML approach, *International Journal of Computer Integrated Manufacturing*, 20(4): 370-386, June 2007.

[4] Tudor, N., Kifor C.V. and Oprean, C., Quality system for production software QSPS, *Academic Journal of Manufacturing Engineering*, ISSN 1583-7904, 135 - 140, 2009.

[5] Oprean, C., Kifor C. V., Suciu, O., *Managementul integrat al calitatii*, Ed. Univ. Lucian Blaga din Sibiu, 2005, ISBN 973-739-034-2.

[6] N. Tudor, D. Dumitrascu, *The Benefits of Project Structuring in Sub-projects and Work Packages*,
http://imtuoradea.ro/auo.fmte/files-2008/MIE_files/TUDOR%20NICOLAE%202.pdf.

[7] N. Tudor, D. Dumitrascu, *Advance Estimate Expenses for Project Execution Time*,
http://imtuoradea.ro/auo.fmte/files-2008/MIE_files/TUDOR%20NICOLAE%201.pdf.

[8] Kifor, C.V., Oprean, C., *Ingineria calitatii. Imbunatatirea sase sigma (Quality Engineering. Six sigma improvement)*, Sibiu, Ed. Univ. Lucian Blaga, 2006.

[9] Ebert, C., 2007, The impacts of software product management, *The Journal of Systems and Software*, 80: 850-861, 2007.

[10] Kneuper, R., Sollmann, F., 1995: *Normen zum Qualitätsmanagement bei der Softwareentwicklung Informatik Spektrum*, Band 18 (1995), S. 314-323.

[11] Kifor, C.V., Tudor, N., Oprean, C., A practical approach to a quality system for production software for managing technological changes, *Proceedings of the 6th International Seminar on Quality Management in Higher Education*, ISBN 978-960-99486-2-3, Greece, 2011.

[12] Kifor, C., V.,Oprean, C., Banciu, D.M.,Intelligent system for assisting decisions in advanced product and process planning and design, *Studies in informatics and control*, ISSN 1220 - 1766, 18(3): 247 - 254, 2009.

[13] Oprean, C., Kifor, C. V., Quality Management, Callidus Publishing House, Germany, ISBN 978 - 3 - 940677-50-1, 2008.

[14] Petrash, R., *The definition of software quality; a practical approach*, ISSRE, 1999.

[15] Pires, J.N., Semi-autonomous manufacturing systems: The role of the human-machine interface software and of the manufacturing tracking software, *Mechatronics*, 1191-1205, 2005.

[16] Pressman, Scott, Software Engineering: A Practitioner's Approach, Sixth Int. ed., McGraw-Hill Education, 388, 2005.

[17] Kifor, C., V.,Oprean, C., Banciu, D.M., Intelligent system for assisting decisions in advanced product and process planning and design, *Studies in informatics and control*, ISSN 1220 - 1766, 18(3): 247 - 254, 2009.

[18] Tudor, N., Kifor C. V., Oprean, C., Quality system for Production Software, *Academic Journal of Manufacturing Engineering*, ISSN 1583-7904, 135-140, 2009.

[19] Tudor, N., Kifor C.V., Oprean, C., Using QSPS in developing and realization of a production line in automotive industry, *Int J Comput Commun*, ISSN 1841-9836, 5(5): 945-952, 2010.

[20] McBride, T., The mechanisms of project management of software development, *The Journal of Systems and Software*, ISSN: 0164-1212, 81: 2386-2395, 2008.

[21] Mordechai Ben-Menachem, Towards management of software as assets: A literature review with additional sources, *Information and Software Technology*, 50: 241-258, 2008.

[22] Procaccino, J.D., Verner, J.M., Software project managers and project success: An exploratory study, *The Journal of Systems and Software*, 79 : 1541-1551, 2006.

[23] INTERNATIONAL STANDARD ISO/TS 16949, (2009) Quality management systems - Particular requirements for the application of ISO 9001:2008 for automotive production and relevant service part organizations, Third edition 2009-06-15.

[24] ISO 9001:2008, Quality management systems - Requirements.