

Ing. Nicolae TUDOR

TEZĂ DE DOCTORAT
-Rezumat-

Conducător științific:

Prof. univ. dr. ing. Claudiu Vasile KIFOR

Sibiu, 2011

Ing. Nicolae TUDOR

TEZĂ DE DOCTORAT

-Rezumat-

CONTRIBUȚII PRIVIND REALIZAREA UNUI SISTEM DE MANAGEMENT AL CALITĂȚII PENTRU APLICAȚIILE SOFTWARE DIN INDUSTRIA COMPONENTELOR PENTRU AUTOVEHICULE

Comisia de evaluare a tezei de doctorat:

Președinte: - Prof.univ.dr.ing. Bandrea Ioan, Universitatea Lucian Blaga din Sibiu

Membri:

Prof. univ. dr. ing. Kifor Vasile Claudiu

Conducător științific, Universitatea Lucian Blaga din Sibiu

Prof. univ. dr. ing. Dumitrașcu Dănuț

Universitatea Lucian Blaga din Sibiu

Prof.univ.dr.ing. Banciu Doina

Institutul Național de Cercetare - Dezvoltare

în Informatică – I.C.I. BUCUREȘTI

Prof.univ.dr.ing. Amza Gheorghe

Universitatea Politehnica din București

Prof.univ.dr.ing. Mărăscu-Klein Vladimir

Universitatea Transilvania Brașov

CUPRINS	TEZĂ	Rezumat
Prefață	3	
Lista de prescurtări și simboluri	10	8
Capitolul-1 Introducere; obiectivele tezei de doctorat și metodologia de cercetare	14	12
1.1. Introducere	14	12
1.1.1 Factori decisivi în inițierea studiului. Motivul alegerii subiectului.	14	
1.1.2 Stadiul actual	15	12
1.1.3 De ce s-a ales managementul calității proiectelor software de producție ?	19	
1.1.4 De ce s-au studiat metode ale managementului proiectelor software ?	19	
1.1.5 De ce s-au studiat modele ale calității aplicațiilor software ?	20	
1.1.6 De ce s-a ales industria constructoare de componente de autovehicule ?	21	
1.1.7 Elemente inovative	21	
1.2. Obiective	22	14
1.2.1 Obiective de analizat și rezolvat	22	14
1.2.2 Metodologia de cercetare	23	15
1.2.3 Structura tezei de doctorat	24	
1.2.4 Dificultăți întâlnite și rezolvarea acestora	27	
1.2.5 Cine sunt beneficiarii potențiali ai cercetării propuse?	27	
Capitolul-2 Managementul proiectelor software	28	16
2.1. Introducere	28	16
2.2. Știința managementului	28	
2.2.1 Principalele caracteristici ale științei managementului	28	
2.2.2 Principalele caracteristici ale managementului științific	29	
2.2.3 Conceptul de management	29	
2.2.4 Funcțiile managementului	31	
2.2.5. Produsul și procesul	32	
2.3. Conceptul de proiect	33	
2.3.1 Scurt istoric	33	
2.3.2 Definiții	33	
2.3.3 Clasificarea proiectelor	34	
2.3.4 Particularități ale unor proiecte economice	35	
2.3.5 Obiectivele proiectelor	35	
2.3.6 Etapele proiectului	36	
2.3.6.1 Avantajele împărțirii proiectelor în etape	37	
2.3.6.2 Structurarea proiectului	38	
2.3.6.3 Ciclul de viață al proiectelor	40	
2.4. Managementul de proiect	42	
2.4.1 Noțiuni introductive, scurt istoric	42	
2.4.2 Definiții ale managementului de proiect	43	
2.4.2.1 Definiții clasice	43	

2.4.2.2 Definiții din stadiul actual	44	
2.4.3 Metodologii și abordări moderne ale managementului de proiect	45	
2.4.3.1 Metodologii clasice	45	
2.4.3.2 Metodologiile actuale utilizate în managementul proiectelor	54	
2.4.4 Etapele managementului de proiect	55	
2.4.5 Modele și instrumente în managementul de proiect	57	
2.4.5.1 Modelul SWOT	57	
2.4.5.2 Graficul RASI	58	
2.5. Managementul proiectelor software	59	16
2.5.1 Metode și abordări în stadiul actual	59	16
2.6. Elemente de sinteză ale managementului proiectelor software	66	
2.6.1 Elemente de interes ale managementului de proiect pentru proiectele software	66	
2.6.2 Cercetare științifică documentară conceptuală și selectivă privind impactul managementului proiectelor asupra proiectelor software	67	
2.6.3 Analiză sintetică a diverselor modele de management pentru proiectele software	69	20
2.6.3.1 Comparații ale caracteristicilor de modelare	69	
2.6.3.2 Comparații de utilitate a modelelor proiectelor software	71	
2.7 Concluzii, opinii și soluții	73	
Capitolul-3 Sisteme de management al calității și securității aplicațiilor software	74	23
3.1. Introducere	74	
3.2. Calitatea produselor	75	
3.2.1 Principalele caracteristici ale calității produselor	75	
3.2.2 Principalele caracteristici ale securității informațiilor	79	
3.3. Sisteme ale calității și securității aplicațiilor software	82	23
3.3.1 Calitatea aplicațiilor software	82	
3.3.1.1 Calitatea în procesul de obținere a aplicațiilor software (ISO/IEC 9001/9000-3)	86	
3.3.1.2 Studiu comparativ asupra normelor de calitate de proces	100	
3.3.1.3 Exemple practice de aplicare a normelor	103	29
3.3.1.4 Calitatea axată pe produs în proiectele software	119	
3.3.1.5 Modelul calității bazat pe conceptul intervalului și sistemului fuzzy cu algoritmi genetici(GA)	126	
3.3.2 Securitatea aplicațiilor software	128	34
3.3.2.1 Obiectivele securității aplicațiilor software	129	
3.3.2.2 Aplicațiile software securizate	129	
3.3.2.3 Integrarea procedurilor de securizare în ciclul de viață al aplicațiilor software	130	
3.3.2.4 Modelul de securitate SAMM din cadrul OWASP	131	34
3.3.2.5 Modelul de securitate Microsoft SDL	133	35

3.3.2.6 Modelul de securitate BSIMM	135	36
3.4. Concluzii	137	
Capitolul-4 Model de sistem de calitate pentru aplicațiile software din industria construcției componentelor de autovehicule	138	37
4.1. Introducere	138	37
4.2. Modele de calitate de produs pentru software și aplicabilitatea lor în practică	139	38
4.2.1 Descrierea aplicației software folosit în exemplul practic	139	
4.2.2 Evaluarea și verificarea aplicației software cu ISO/TS 16949	140	
4.2.3 Evaluarea și verificarea aplicațiilor software cu Automotive SPICE	146	
4.2.4 Aplicarea normei ISO/IEC 25000	158	
4.3. Cercetări privind realizarea unui model de calitate pentru aplicațiile software de producție: reguli, cerințe și repere îndrumătoare	159	38
4.3.1 Modelul conceptual (cadru) QSPS	160	39
4.3.2 Procesul și mediile de aplicare ale metodei QSPS	161	40
4.3.3 Arhitectura modelului de calitate QSPS	163	42
4.3.4 Etapele și elementele modelului de calitate propus QSPS	164	43
4.4. Modelarea matematică a metodei QSPS	175	45
4.5. Concluzii	183	
Capitolul-5 Cercetări privind realizarea unui model experimental de sistem de calitate pentru aplicațiile software din industria componentelor de autovehicule	184	49
5.1. Introducere	184	
5.2. Aplicațiile software MES – Aplicații software de suport ale producției	184	
5.2.1 Arhitectura sistemului MES	185	
5.2.1.1 Sistemul MES CamLine	187	
5.2.1.2 Sistemul MES la una dintre cele mai mari companii europene constructoare de componente de autovehicule	188	
5.2.1.3 Componentele interactive ale aplicațiilor MES la una dintre cele mai mari companii europene constructoare de componente de autovehicule	189	
5.3. Ciclul de viață a serviciilor software la una dintre cele mai mari companii europene constructoare de componente de autovehicule și integrarea sistemului QSMA în cadrul acestuia	192	
5.3.1 ITG10- Propunerea de proiect (Project Proposal available):	192	
5.3.2 ITG30 – Demararea Proiectului de servicii (Start of Service Project):	193	
5.3.3 ITG40-Aprobarea conceptului final (Final IT Concept Approved):	193	
5.3.4 ITG50 – Implementarea serviciului și testare (Service development and test):	193	

5.3.5 ITG60 – Aprobări de Design (IT design approved):	193	
5.3.6 ITG70 – Aprobarea serviciului de instalare și suport (Support approved – Rollout preparation):	194	
5.3.7 ITG80 Prima instalare Pilot cu succes (1st pilot successful – service piloting):	194	
5.3.8 ITG90 – Finalizarea instalării serviciului (Rollout completed – service rollout):	194	
5.3.9 ITG100 – Terminarea serviciului (Service use ended – service operation):	194	
5.3.10 ITG110 –Ștergera serviciului din portofoliul de servicii (Service cleaned - Service decommission):	194	
5.4. Sistemul QSMA – Model experimental și practic	195	49
5.4.1 Structura organizațională a sistemului MES	197	
5.4.1.1 Core System- Sistemul de bază:	198	
5.4.1.2 Communication Middleware- Mediul de comunicare	198	
5.4.1.3 Reporting Solution- soluții de raportare	199	
5.4.1.4 Integration Frameworks- integrări client Frameworks	199	
5.4.1.5 Integration Solutions / Clients - soluții pentru integrarea programelor de tip client	199	
5.4.2 MODUL1-Evaluări de timp și costuri (PM) – CA IT Gate ITG:30->40	200	50
5.4.3 MODUL2-Cerințe Software (SW) – CAS IT Gate ITG 40->60	200	51
5.4.4 MODUL3-Validare și Verificare (VaVe) – CA IT Gate 40->60	200	51
5.4.5 MODUL4-Aprobări Interne (IR) – CAS IT Gate 60->70	200	51
5.4.6 MODUL5-Aprobările Clientului(CR) – CAS IT Gate 70->90	201	52
5.4.7 MODUL6-Monitorizări Funcționale (FM) – CAS IT Gate 90->100	201	52
5.4.8 MODUL7-Satisfacția Clientului (CS) – CAS IT Gate 90->100	202	52
5.5. Rezultate practice obținute în urma aplicării metodei QSMA	204	
5.5.1 Elementul 1. Structurarea proiectelor în subproiecte și pachete de activitate	204	
5.5.1.1. Planul structurat al unui proiect	205	
5.5.1.2. Calculul de costuri în situații reale	205	
5.5.2 Modulul 1. Evaluarea timpului de implementare al aplicațiilor software de producție	209	
5.5.3 Cercetări a beneficiilor aplicării metodei QSMA în cazuri practice	212	
5.6. Concluzii	213	55
Capitolul-6 Contribuții proprii și arii potențiale de cercetare viitoare ale studiului	214	56
Referințe bibliografice:	217	59
Lista de tabele	237	
Lista de figuri	238	
Anexe:	240	76

Anexa 1 – Checklist pentru condiția de demarare	240	
Anexa 2 – Calcul estimativ de Timp și Costuri (resurse)	242	76
Anexa 3 – Graficul RASI	244	
Anexa 4 – Definirea etapelor de calitate (Q-Gate) pentru proiect	245	
Anexa 5 – Checklist de maturitate - tehnologie software	248	78
Anexa 6 – Chestionar responsabilități pe timpul ciclului de viață	249	79
Anexa 7 – Chestionar de evaluare	250	80
Anexa 8 – Confirmare aplicare practică	257	87
Anexa 9 – Curriculum Vitae	261	91
Anexa 10 – Lista Publicațiilor științifice	264	94

Lista de prescurtări și simboluri

AACE:	<i>Association for the Advancement of Cost Engineering</i>	<i>Asociația de promovare a ingineriei de cost</i>
ACQ:	<i>Acquisition</i>	<i>Procesul de achiziție</i>
AMD:		<i>Analiza modurilor de defectare</i>
ASRO:		<i>Asociația de Standardizare din România</i>
AQAP:	<i>Allied Quality Assurance Publication</i>	<i>Publicația asociației pentru asigurarea calității</i>
BS6079:	<i>British Standard 6079</i>	<i>Standardul britanic 6079</i>
BSIMM:	<i>Building security in maturity model</i>	<i>Realizarea de securitate în modelul de maturitate</i>
CAD:	<i>Computer-aided design</i>	<i>Proiectarea asistată de calculator</i>
CAPS:	<i>CamLine Application Server</i>	<i>Serverul de aplicații CamLine</i>
CarMa:	<i>Carrier Management</i>	<i>Managementul purtătorilor de producție</i>
CCPM:	<i>Critical Chain Project Management</i>	<i>Managementul proiectelor prin lanț critic</i>
CDR:	<i>Critical Design Review</i>	<i>Revizii critice de design</i>
CMMI:	<i>Capability Maturity Model Integration</i>	<i>Capabilitatea de integrare a modelului de maturitate</i>
COTS:	<i>Comercial off-the-shelf</i>	<i>Produce finite de serie</i>
CPK:	<i>Capability Key</i>	<i>Indicatori de capabilitate</i>
CPM:	<i>Critical Path Method</i>	<i>Metoda drumului critic</i>
CR:	<i>Customer Software Release</i>	<i>Aprobări client pentru software</i>
CS:	<i>Customer Satisfaction</i>	<i>Satisfacția clientului</i>
CSA Research Pack:	<i>Cambridge Scientific Abstracts</i>	<i>Pachetul de lucrări științifice abstracte de la Cambridge</i>
CVS:	<i>Concurrent Version system</i>	<i>Sistemul versiunilor concurente</i>
CWQC:	<i>Company Wide Quality Control</i>	<i>Controlul vast al calității companiei</i>
DB:	<i>Database</i>	<i>Bază de date</i>
DGQ:	<i>Deutsche Gesellschaft für Qualität</i>	<i>Asociația germană pentru calitate</i>
DIN:	<i>Deutsches Institut für Normung</i>	<i>Institutul german de norme</i>
DPMO:	<i>Defects per million opportunities</i>	<i>Defecte pe un milion de oportunități</i>
DSDM:	<i>Dynamic System Development Method</i>	<i>Metoda de dezvoltare a sistemelor dinamice</i>
DWH:	<i>Data Ware House</i>	<i>Depozit de date</i>
ECoFrame:	<i>Equipment Connection Framework</i>	<i>Arhitectura de conectare a echipamentului</i>
ECSS-M-30A:	<i>European Cooperation for Space Standardization</i>	<i>Cooperarea europeană de standardizare a spațiului</i>
EFQM:	<i>European Foundation for Quality management</i>	<i>Fundația Europeană pentru managementul calității</i>
EN:	<i>European Norm</i>	<i>Normă Europeană</i>
ENG:	<i>Engineering</i>	<i>Proiectare / dezvoltare</i>
ERP:	<i>Enterprise Resource Planning</i>	<i>Activitatea de planificare a resurselor</i>
EVAPROD:	<i>Evaluation of Production</i>	<i>Evaluarea producției</i>
FM:	<i>Functional monitoring</i>	<i>Monitorizări funcționale</i>
FMEA:	<i>Failure Mode and effect analysis</i>	<i>Modul de defectare și analiza afecțiunilor</i>

FPY:	<i>First Pass Yield</i>	<i>Procesări cu succes</i>
FraMES:	<i>Framework Manufacturing execution system</i>	<i>Arhitectura sistemului de execuție a producției</i>
GA:	<i>Genetic Algorithm</i>	<i>Algoritm genetic</i>
GRNN:	<i>General Regression neural network</i>	<i>Rețea neuronală de regresie generală</i>
GUI:	<i>Graphical user Interface</i>	<i>Interfață grafică pentru utilizatori</i>
HP:	<i>Hewlett Packard</i>	
HPGL:	<i>Hewlett Packard Graphic Language</i>	<i>Limbaaj Grafic Hewlett Packard</i>
ID:	<i>International development</i>	<i>Proiectele de dezvoltare internaționale</i>
IEC:	<i>International Electrotechnical Comission</i>	<i>Comisia electrotehnică internațională</i>
IEEE:	<i>Institute of Electrical and Electronics Engineers</i>	<i>Institutul ingineresc de electronică și electrotehnică</i>
iGATE:	<i>Information Gateway</i>	<i>Poartă informațională</i>
IPM:		<i>Incidente pe un milion</i>
IPMA	<i>The International Project Management Association</i>	<i>Asociația internațională a managementului de proiect</i>
IR:	<i>Internal Software Release</i>	<i>Aprobări interne de software</i>
ISBN:	<i>International Standard Book Number</i>	<i>Număr internațional standard de carte</i>
ISO/TS:	<i>International organization for standardization/Technical specification</i>	<i>Organizația internațională de standardizare / Specificația tehnică</i>
IT:	<i>Information technologie</i>	<i>Tehnologia informațională</i>
ITG:	<i>IT Gate</i>	<i>Poartă tehnologică informațională</i>
JCM:	<i>Job characteristics model</i>	<i>Model de caracteristici a activității</i>
KPA:	<i>Key Process Area</i>	<i>Domeniul indicatorilor de proces</i>
KPI:	<i>Key performance indicator</i>	<i>Indicatori deperformanță proces</i>
LASP:	<i>Lightweight Application Security Process</i>	<i>Gradul de aplicare al procesului de securitate</i>
MaMa:	<i>Material Management</i>	<i>Managementul materialelor</i>
MAN:	<i>Management</i>	<i>Procesul de management</i>
MBA:	<i>Master of Business Administration</i>	<i>Master în administrația afacerilor</i>
MES:	<i>Manufacturing execution system</i>	<i>Sistemul de execuție al producției</i>
MIL:	<i>Military specification</i>	<i>Specificație militară</i>
MPM:	<i>Master Proces Monitor</i>	<i>Monitorizarea procesului superior</i>
MPX:		<i>Managementul de proiect extrem</i>
NA:	<i>Not applicable</i>	<i>Neaplicabil</i>
NAFTA:	<i>North American Free Trade Agreement</i>	<i>Acordul nord american de taxe vamale libere</i>
NATO:	<i>North Atlantic Treaty Organization</i>	<i>Organizația tratatului nord atlantic</i>
NIST:	<i>National Institute of Standards and Technology</i>	<i>Institutul național de standarde și tehnologii</i>
OGC:	<i>Office of Government Commerce</i>	<i>Oficiul de comerț guvernamental</i>
OOPSLA:	<i>Object-Oriented Programming, Systems, Languages & Applications</i>	<i>Programarea orientată pe obiect, Sisteme, limbaje de programare & Aplicații</i>
OS:	<i>Operating system</i>	<i>Sistem de operare</i>
OWASP:	<i>Open Web Application Security Project</i>	<i>Proiectul de securitate pentru aplicații de rețea deschisă</i>

PAA:	<i>Part average analysis</i>	<i>Analiza mediei</i>
PDCA:	<i>Plan Do Check Act</i>	<i>Planifică Execută Verifică Acționează</i>
PDR:	<i>Preliminary Design Review</i>	<i>Revizuire preliminară de design</i>
PERT:	<i>Program Evaluation and Review Technique</i>	<i>Evaluări de program și tehnici de revizuire</i>
PIM:	<i>Project improvement</i>	<i>Îmbunătățirea continuă</i>
PL/SQL:	<i>Procedural Language /Select Query language</i>	<i>Limbaje procedurale / limbaje de interogare</i>
PM:	<i>Project management</i>	<i>Managementul de proiect</i>
PM@IT:	<i>Project Management at IT</i>	<i>Managementul de proiecte IT</i>
PMBOK:	<i>Project Management Body of Knowledge</i>	<i>Managementul de proiect, partea de cunoștințe</i>
PMDA:	<i>Product Main Data Administration</i>	<i>Administrarea datelor primare de produs</i>
PMI :	<i>Project Management Institute</i>	<i>Institutul managementului de proiect</i>
PPAP:	<i>Production part approval process</i>	<i>Procesul de aprobare a produsului finit de producție</i>
PROMPT:	<i>PROject Management of Process Tradeoffs</i>	<i>Managementul deviațiilor de proces</i>
PTA:	<i>Process tradeoff analysis</i>	<i>Procesul de analiză a deviațiilor</i>
QIP:	<i>Quality Improvement Paradigm</i>	<i>Paradigma îmbunătățirii calității</i>
QM:	<i>Quality Management</i>	<i>Managementul calității</i>
QSMAS:	<i>Quality System for Manufacturing Application</i>	<i>Sistem de calitate pentru aplicațiile de producție</i>
QSPS:	<i>Quality System for Production Software</i>	<i>Sistem de calitate pentru aplicații software de producție</i>
RAD:	<i>Rapid Application Development</i>	<i>Dezvoltarea rapidă de aplicații</i>
RASI:	<i>Responsible, approve, support, inform</i>	<i>Responsabilitate, Aprobare, Suport, Informare</i>
REU:	<i>Reuse</i>	<i>Procesul de reutilizare</i>
RM:	<i>Recipe Management</i>	<i>Managementul de recoltare</i>
SAMM:	<i>Software Assurance Maturity Model</i>	<i>Modelul de asigurare a maturității software</i>
SAP:	<i>Special Assistance Plan</i>	<i>Plan de asistență specială</i>
SCR:	<i>System Concept Review</i>	<i>Revizuirea conceptului de sistem</i>
SCSP:		<i>Sistem de Calitate pentru Software de Producție</i>
SDL:	<i>Security Development Lifecycle</i>	<i>Ciclu de viață al procesului de dezvoltare a securității</i>
SEI:	<i>Software Engineering Institute</i>	<i>Institutul de proiectare software</i>
SerGen:	<i>Serial Number Generator</i>	<i>Generator de numere de serie</i>
SIG:	<i>Automotive Special Interest Group</i>	<i>Grupul de interese speciale din industria auto</i>
SLA:	<i>Service Level agreement</i>	<i>Acordul serviciului de suport</i>
SMART:		<i>Specifice Măsurabile Acceptate Realiste Timp precizat</i>
SMC:		<i>Sisteme de management ale calității</i>
SPACE:	<i>Statistical Process Analysis and Control Environment</i>	<i>Analize statistice de proces și control al mediului</i>
SPC:	<i>Special characteristics</i>	<i>Setul de caracteristici speciale</i>

SPICE:	<i>Software Process Improvement and Capability Estimation</i>	<i>Procesul de îmbunătățire software și estimări de capacitate</i>
SPL:	<i>Supplier Process</i>	<i>Procesul de furnizare și al furnizorilor</i>
SQAP:	<i>Software Quality Assurance Plan</i>	<i>Planul de asigurare a calității software</i>
SQL:	<i>Select Query Language</i>	<i>Limbaj de selecție/ interogare</i>
SQM:	<i>Supplier Quality Management</i>	<i>Managementul calității furnizorilor</i>
SQuaRE:	<i>Software product Quality Requirements and Evaluation</i>	<i>Evaluarea cerințelor de calitate a produselor software</i>
SSG:	<i>Software security group</i>	<i>Grupul de securitate software</i>
SSR:	<i>System Specification Review</i>	<i>Revizuirea specificației de sistem</i>
SST:	<i>Software Security Touchpoints</i>	<i>Aspecte de securitate software</i>
SUP:	<i>Support</i>	<i>Asistență și suport</i>
SWOT:	<i>Strength-Weakness Opportunities-Threats</i>	
UA:	<i>User Administration</i>	<i>Administrarea utilizatorilor</i>
UG:	<i>User Group</i>	<i>Grupul utilizatorilor</i>
UNDP:	<i>United Nations Development Program</i>	<i>Programul de dezvoltare a națiunilor unite</i>
USFCA.edu:	<i>University of San Francisco Educase</i>	<i>Universitatea EDUCASE din San Francisco</i>
VaVe:	<i>Validation and Verification</i>	<i>Validare și verificare</i>
VDA:	<i>Verband der Automobilindustrie</i>	<i>Asociația industriei auto</i>
WBS:	<i>Work breakdown structure</i>	<i>Structurarea activităților de lucru</i>
WIP:	<i>Work in Progress</i>	<i>Activitate în desfășurare</i>
WNN:	<i>Ward neural network</i>	<i>SubRețea neuronală</i>
XML:	<i>Extensible Markup Language</i>	<i>Limbaj de marcare extins</i>
XP:	<i>Extreme Programming</i>	<i>Programare extremă</i>

Capitolul-1

Introducere; obiectivele tezei de doctorat și metodologia de cercetare

1.1 INTRODUCERE

Cerințele unei piețe economice din ce în ce mai competitive în care calitatea și prețurile scăzute ale produselor finite sunt determinante, au reprezentat presiuni asupra întreprinderilor care, pentru a prospera sau poate chiar a supraviețui într-o astfel de piață economică, au inițiat o multitudine de studii în ceea ce privește globalizarea, standardizarea și dezvoltarea de strategii și tehnici pentru a deveni competitive.

Studiul de față este realizat din punct de vedere managerial din perspectiva valorii produsului finit și își propune astfel identificarea factorilor cauzatori de pierderi economice pentru o întreprindere, delimitarea acestora și propunerea unei soluții de eliminare totală sau diminuare a unuia sau mai multora dintre acești factori.

Astfel, obiectul studiului se concentrează asupra creșterii valorii produsului finit; această valoare este reprezentată de elemente statice, ce pot fi controlate precum: costuri de material în proporție de 50%, costuri de transport și logistică în proporție de 10%-15%, costuri de fabricație, echipament și personal 20%-30% și asigurarea de profit în raport de 10% și elemente dinamice: numărul de piese produse într-un interval de timp definit și aprobat de comun acord cu clienții (durata ciclului de viață a proiectului) din care se calculează tactul liniei de producție, capacitatea echipamentului industrial și stabilirea procesului industrial, precum și definirea cerințelor pentru aplicația software care însoțește producția și interacționează cu echipamentul industrial. Aceste elemente dinamice nu pot fi controlate pe deplin, ele fiind doar anticipate și proiectate. Toate aceste elemente dinamice: tactul liniei, echipamentul industrial ales, proiectarea procesului de producție și aplicația software de producție sunt elementele dinamice ce pot cauza costuri impredictibile, greu de anticipat.

Studiul de față vizează unul din aceste elemente și anume **asigurarea calității aplicațiilor software care însoțesc producția**, astfel încât costurile cauzate de acest element să fie mult diminuate sau chiar eliminate.

1.1.2 Stadiul actual

Întreprinderile moderne se confruntă la ora actuală cu două imperative ale societății contemporane: calitate ridicată la costuri scăzute, ceea ce necesită în mod inevitabil îmbunătățirea producției. Din această cauză procesul de prelucrare trebuie să fie foarte eficient și bine controlat, atenția îndreptându-se asupra automatizării, computerelor și aplicațiilor software (Pires, 2005). În multe industrii, producția este foarte detaliat controlată în toate etapele ei, pe fiecare segment integrat *in-line* de producție, menit să execute operațiile necesare de transformare a materiei prime în produs final (Pires, 2005).

Metodele pe care le abordează acest studiu vin în sprijinul îmbunătățirii continue a performanței aplicațiilor software (de ex. diminuarea perioadei de realizare, creșterea calității, reducerea cerințelor de

modificări etc.) (Peterson, Wohlin, 2010). Pentru aceste îmbunătățiri au fost propuse nenumărate modele printre care CMMI-Capability Maturity Model Integration (CMMI) (CMMI-Product-Team, 2006) sau QIP- Quality Improvement Paradigm (QIP) (Basili,1985; Basili and Green, 1994).

Odată cu introducerea normelor și regulilor de calitate în producția automatizată, au fost introduse și nenumărate strategii și algoritmi cu scopul de a controla și asigura calitatea produselor. Pentru implementarea aplicațiilor software de producție există de asemenea nenumărate studii cu scopul de a realiza aplicații software robuste și conforme cerințelor producției. Ambele direcții de studiu sunt foarte profunde și au un rezultat remarcabil, însă nu sunt CORELATE între ele. De cele mai multe ori, implementarea aplicațiilor software de producție se realizează pe bază de metode, tehnici și norme dezvoltate pentru software fără a se ține cont de cerințele normelor de calitate ce trebuie să fie îndeplinite de procesul de producție pentru a obține calitatea cerută de clienți. Aceste incompatibilități ale aplicațiilor software cu cerințele de calitate ale produselor finite, cauzează revizuirii complete ale aplicațiilor software de producție după rularea și aplicarea lor, ceea ce duce la costuri adiționale cauzate de întreruperea producției și neîndeplinirea volumului necesar pentru livrări. Pe de altă parte aplicațiile software trebuie realizate în conformitate cu anumite criterii ale echipamentului industrial, fără a se lua în considerare impactul asupra calității produselor.

Demararea dezvoltării procesului industrial trebuie să fie însoțită de procesul de concepere a aplicațiilor software pentru producție, iar acesta din urmă trebuie să fie integrat și omologat în procesul industrial ca o componentă a acestuia. De cele mai multe ori aceste aplicații software sunt realizate după ce echipamentul industrial a fost proiectat și chiar realizat, și doar pe baza unor specificații ale echipamentului, fără cunoștințe adiționale asupra produsului de realizat sau asupra cerințelor clientului în privința calității dorite. În aceste cazuri cerințele de calitate de produs nu mai sunt luate în considerare.

De asemenea una dintre dificultățile adesea întâlnite, este realizarea unei specificații tehnice complete și corecte și de asemenea în timp util. „În general, la ora actuală se consumă mult mai mult timp și efort pentru specificațiile tehnice decât se consuma în trecut, când aceste specificații tehnice nu erau realizate destul de explicit deoarece ele erau considerate a fi de domeniul cunoașterii elementare.” (Kühner, G., Torsten, B., 2009).

„De altfel managementul calității aplicațiilor software câștigă tot mai multă importanță în toate domeniile de implementare și economice datorită următoarelor criterii (Kneuper, R., Sollmann, F., 1995):

- pe de o parte, calitatea a devenit esențială într-o piață competitivă, fapt cauzat de conștientizarea tot mai mare de către beneficiari a factorului calitativ.
- pe de altă parte, corectarea erorilor din software s-a dovedit a fi foarte costisitoare, aspect ce poate fi prevenit prin introducerea rapidă și utilizarea sistemelor de management al calității”.

Collyer și Warren (2009), în urma studiului efectuat, ajung la concluzia că managementul de proiect este benefic chiar și în proiectele cu medii dinamice și schimbări dese, precum proiectele software.

Kouskouras și Georgiou (2007) realizează în baza mecanismelor managementului de proiect o metodă discretă de simulare și coordonare a aplicațiilor software. Această metodă oferă suportul în definirea strategiilor și luarea deciziilor de management.

Un alt element de mare importanță în proiectele software este metoda CCPM (Critical Chain Project Management). Lee și Miller (2004) consideră această metodă ca fiind una crucială în proiectele software în care realizarea în paralel a activităților este inevitabilă. CCPM nu construiește doar rețeaua proiectului dat recunoaște și interdependențele între activități.

Tehnicile managementului de proiect, au un mare impact asupra calității aplicațiilor software. Aplicând reguli și metode de management clasic în proiectele software, trebuie avut în vedere impactul pe care acestea le au asupra rezultatului dorit. Astfel, mecanismele de management trebuie alese cu precauție în funcție de natura și complexitatea proiectului. Dinamicitatea aplicațiilor software, dar și urmările pe care modificările le au în aceste proiecte, necesită o analiză foarte laborioasă încă dinainte de începerea proiectelor și de planificarea lor. Cele mai dezavantajoase cazuri (*worse case*) ce pot apărea nu trebuie neglijate, ci tratate cu seriozitate, mai ales că aceste cazuri au o probabilitate foarte mare de a apărea pe parcursul aplicațiilor software.

Astfel apar mai multe denumiri și clasificări ale versiunilor aplicațiilor software:

- **pre alfa:** în acest stadiu proiectele au o funcționalitate minimă (interfața și una două funcțiuni) fiind folosite pentru prezentări intermediare;
- **alfa:** reprezintă prima versiune care poate fi testată de o a treia persoană, alta decât programatorul;
- **beta:** este versiunea care se poate deja comercializa, stadiul fiind încă un stadiu de test;
- **beta perpetuu:** este o versiune Beta descrisă mai sus, însă în continuă schimbare;
- **pre-release:** acest pre-release semnifică o aprobare a aplicației finale de software. Acum se efectuează teste complete ale tuturor funcționalităților în diferite situații și se efectuează corecturile erorilor noi apărute. În cazul apariției de erori, o nouă versiune de pre-release va trebui stabilită și retestată;
- **release:** versiunea finală de comercializat. Aici are loc numerotarea versiunii după modelul următor:
 - gata de fabricare: gata pentru comercializare;
 - stabil: versiune stabilă care nu se mai modifică;
 - final: pentru versiunea finală;
 - acces general: liberă și accesibilă pentru utilizarea de larg consum;
 - versiunea de aur: produs matur și liber de orice potențiale erori;
- **corectare de erori:** acest stadiu este folosit pentru instalare de pachete corectate de erori apărute și raportate ulterior de beneficiarii acestor pachete. Aceste pachete sunt de regulă libere de costuri pentru beneficiari.

Cu toate că aceste stadii descriu faze finale ale aplicațiilor software și deci în teorie ar trebui să conțină doar mici diferențe software de la un stadiu la altul, în realitate, o aplicație software conține diferențe foarte mari între două stadii consecutive, astfel încât aplicațiile software aferente acestor stadii ajung să fie complet diferite.

1.2 OBIECTIVE

1.2.1 Obiective de analizat și rezolvat

Aceste obiective pot fi enumerate după cum urmează:

- asigurarea calității aplicațiilor software de producție din industria componentelor de autovehicule în vederea reducerii costurilor cauzate de neconformitățile/erorile din aplicațiile software;
- aplicarea normelor din domeniul calității și găsirea unor norme analoage normelor de calitate în realizarea și punerea în funcțiune a aplicațiilor software de producție în industria componentelor de autovehicule în conformitate cu cerințele acestei industrii;
- realizarea aplicațiilor software de producție în timp util;

- îmbunătățirea calității producției;
- înlăturarea riscurilor de întrerupere a desfășurării normale a producției;
- **realizarea unui sistem de calitate pentru aplicațiile software de producție** asemănător cu sistemele de calitate de procese și produse pentru producția din industria componentelor de autovehicule (ISO/TS 16949, VDA 6.3...etc.);
- **integrarea, adaptarea și generalizarea sistemului** într-una dintre cele mai mari companii Europene constructoare de componente pentru autovehicule.

Scopul proiectului este de a dezvolta un sistem al calității pentru aplicațiile software de producție, ca element integrant al dezvoltării procesului industrial, cu scopul de a asigura desfășurarea adecvată a producției și nivelul ridicat al produselor finite în conformitate cu cerințele clienților.

Algoritmii de estimare a duratei proiectului și procedurile de analiză a riscurilor, module integrante ale sistemului QSPS, demonstrează acțiunea directă a acestui sistem asupra asigurării rulării optime a producției și asupra asigurării certitudinii de pornire a producției în perioada planificată.

Astfel obiectivele specifice ale acestui studiu sunt constituite din:

- creșterea nivelului calității aplicațiilor software de producție;
- reducerea impactului negativ al aplicațiilor software asupra calității produselor finite;
- diminuarea riscului de întârziere în pornirea producției;
- reducerea costurilor rezultate din cauza aplicațiilor software din producție.

1.2.2 Metodologia de cercetare

Pornind de la studierea *tehnicilor managementului de proiecte software*, iar apoi a normelor de *calitate și securitate* al acestor produse și a cerințelor de calitate din *industria componentelor de autovehicule* s-au realizat chestionare de evaluare care au fost aplicate în mod simulativ asupra unor proiecte software de producție din industria autovehiculelor. În mod observabil au fost eliminate pe de o parte elementele care nu și-au găsit aplicabilitatea pentru acest tip de producție, iar pe de altă parte, au fost propuse elemente adiționale, care nu a fost regăsite în metodele sau standardele studiate. Rezultatul constă într-un sistem teoretic conglomerat, expresia detaliată a tuturor cerințelor din domeniile anterior amintite, sistem ce a fost supus încă o dată unei filtrări, pentru facilitarea aplicabilității practice și generalizării (transmiterii) într-o companie multinațională.

Astfel etapele cercetării au fost următoarele:

- identificarea necesității aplicării modelelor de calitate pentru aplicațiile software de producție;
- aplicarea experimentală a diverselor modele reprezentative de calitate pentru aplicațiile software din producție;
- analiza comportamentului acestor metode în diferite situații ale producției;
- selecția elementelor reprezentative respectiv adaptarea acestora pentru cerințele producției;
- modelarea unei metode (QSPS) a calității formată din elemente ale unor norme și modele cunoscute precum și din elemente de contribuție proprie;
- testarea sistemului în diferite situații practice, rezultatele obținute fiind baza de calcul al
- modelării matematice ale sistemului în vederea optimizării.

Capitolul-2

Managementul proiectelor software

2.1 INTRODUCERE

Datorită caracterului eclectic al disciplinei de management, în care managerii au de a face în același timp și cu resurse umane dar și cu tehnologii, dezvoltarea managementului nu a cunoscut o serie de trepte cronologice în evoluția sa. Astfel modelul de dezvoltare a fost unul cu o diversitate de abordări, care adesea s-au suprapus în dezvoltare.

2.5 MANAGEMENTUL PROIECTELOR SOFTWARE

2.5.1 Metode și abordări în stadiul actual

Metoda Scrum. Prima abordare a acestei metode a fost descrisă de Takeuchi și Nonaka în „The New Product Development Game” (Harvard Business Review, Jan-Feb 1986). Ei au scris că de-a lungul timpului proiectele care folosesc echipe mici, care au posibilitatea de a transfera una de la alta sarcini, produc cele mai bune rezultate. De asemenea, ei au asemănat aceste echipe performante cu grămada folosită în rugby (în engleză „*scrum*”), referindu-se la această metodă de organizare a proiectelor ca o „grămadă”.

Caracteristicile metodei SCRUM sunt următoarele:

- un set de sarcini nerezolvate care descriu ceea ce trebuie făcut și în ce ordine;
- îndeplinirea unui set fixat de sarcini nerezolvate în serii scurte numite sprinturi;
- întâlnire scurtă în fiecare zi (o ședință scrum) în care este stabilit progresul efectuat, munca ce urmează și eventualele impedimente;
- scurtă sesiune de planificare a sprintului în care vor fi definite sarcinile nerezolvate ce vor fi incluse în sprint;
- scurtă retrospectivă a sprintului în care toți membrii echipei reflectează asupra sprintului încheiat;

Metoda XP (Programare extremă): este o metodă care în ultimul timp a luat tot mai multă amploare și este foarte des utilizată în realizarea proiectelor de orice gen, mai ales în proiectele software. Principiul acestei metode este de a pune în prim plan rezolvarea unei probleme de programare în detrimentul formulării de dezvoltare software, abordarea fiind astfel una formală, mai puțin importantă. La baza acestei metode stau tehnici de genul „*Best practice*” cele mai bune experiențe din practică (metodă preluată din managementul clasic de proiect). Comunicarea între membrii echipei este caracteristica de bază a acestei metode. Această metodă se aplică în proiectele în care cerințele clientului nu sunt bine definite sau chiar cunoscute de la bun început, astfel încât scopul și durata proiectului sunt vag de recunoscut.

Figura 1 ne exemplifică principiul funcționării acestei metode (Beck, K.,2004):

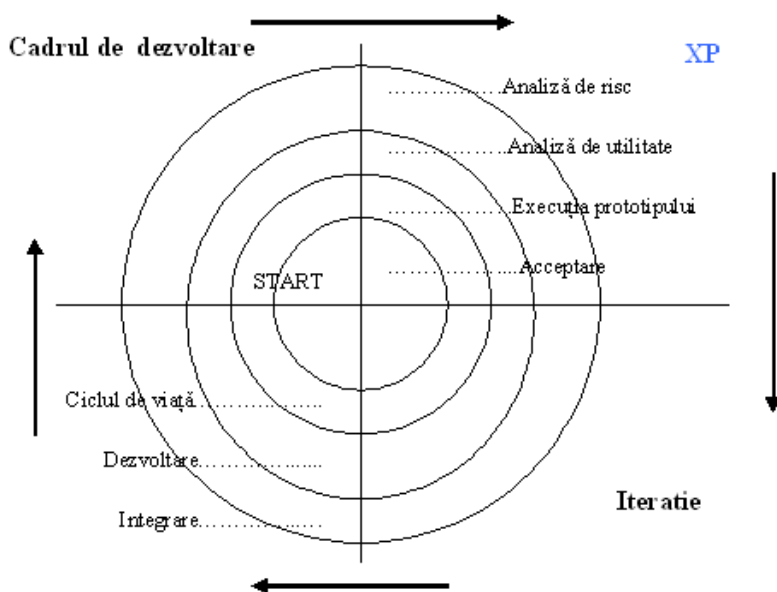


Figura 1 Metoda XP

Astfel clientul este implicat în proiect pe deplin de la bun început, având acces la demersul proiectului și totodată influență în cazul în care cerințele devin mai ambigue. Acest aspect constituie totodată un dezavantaj, în cazurile în care clientul primește o soluție interimară, alta decât soluția finală avută în vedere. Astfel, prin schimbarea continuă a cerințelor, programatorii sunt expuși la o adaptare rapidă la cerințele apărute, dezvoltându-se astfel terminologia de „programator ideal”. Dezavantajul acestui aspect este legat de riscul ridicat de confuzie ce poate apărea în cadrul echipei, în care nimeni nu se mai simte direct răspunzător. Astfel apar roluri noi ce pot fi mai bine vizualizate în tabelul 1 (Beck,K., Andres,C., 2004):

Rolurile în metoda XP		
Rol	Activitate	Exemplu
Proprietarul produsului	Deține responsabilitatea, stabilește priorități, ia decizii privitoare la cel mai bun ROI (Return of investments)	Managerul de produs, un sponsor, un client, un utilizator, un beneficiar, un analist, un manager al utilizatorului
Client	Ia decizii asupra ce urmează a fi executat, raportează / răspunde / aprobă periodic, compune cerințele	Proprietarul produsului, nu neapărat utilizatorul produsului
Programator/dezvoltator	Dezvoltă produsul	Întreaga echipă de dezvoltare: programator, responsabil de testare, proiectant, arhitect, expert baze de date...etc.
Manager de proiect	Conduce echipa	De obicei este proprietarul produsului. Poate fi chiar și

		un programator din cadrul echipei.
Utilizator	Folosește produsul	Utilizatorul produsului

Tabelul 1 Rolurile în metoda XP

Metoda simulării comportamentului aplicațiilor software: această metodă reprezintă un utilitar de estimare a trei variabile de ieșire (calitate, timp și efort depus) ale unei aplicații software, pentru a ajuta managerul de proiect în a lua decizii adecvate (María N. Moreno Garcí'a , Isabel Ramos Román , Francisco J. Garcia Penalvo , Miguel Toro Bonilla , 2008).

De menționat la această metodă ar fi faptul că de cele mai multe ori eșecul aplicațiilor software este datorat mai mult politicii firmei și principiilor acesteia decât deciziilor managerului.

Spre deosebire de metoda XP, un alt studiu demonstrează raportul direct între succesul aplicațiilor software și competența managerială. Acest studiu practic a fost realizat în 178 de companii industriale. Pentru analiză a fost comparat comportamentul și performanțele proiectelor înainte și după întărirea disciplinei de management. Astfel s-a observat că odată cu instituționalizarea elementului de management, rata de succes a proiectelor în ceea ce privește calitatea, programul de predictibilitate a duratei și costului proiectului a crescut considerabil (Ebert,C., 2007).

Un alt studiu dezvoltă metoda bazată pe **agenți inteligenți** pentru management de proiect adaptiv. Sistemul realizat conține o componentă de simulare bazată pe arhitectură de nivel înalt și o componentă agent care realizează modelarea convingere – dorință - intenție. Scopul studiului este de modelare integratoare a problemelor întâlnite pe scară largă cu elemente de nesiguranță în managementul aplicațiilor software și nu numai (Ourdev, I., Xie, H., AbouRizk,S., 2008).

Metoda adaptării la circumstanțe noi abordează mecanismele managementului de proiect, mecanisme care conduc la realizarea și atingerea scopului proiectelor. Odată cu schimbarea cererilor în proiectele software, acestea trebuie să se adapteze foarte rapid noilor cerințe, astfel putând interveni schimbări în ciclurile de implementare, precum revenirea la modelul de bază al aplicației software sau chiar distribuirea aplicațiilor în alte departamente sau companii (*outsourcing*). Ca atare, managementul de proiect va trebui să fie unul adaptiv și reactiv, cu toate că obiectivele proiectului rămân aceleași. Studiul investighează mecanismele managementului de proiect în faza de dezvoltare a produsului, fază în care schimbările necesită o adaptabilitate ridicată și demonstrează cum cunoașterea și alegerea adecvată a acestor mecanisme încă din fazele timpurii ale proiectelor, duc la succesul acestora cu toate schimbările ce pot apărea (McBride,T., 2008). Metoda presupune însă un management de proiect foarte experimentat, precum și o analiză riguroasă înainte de începerea unui proiect, fapt care este foarte rar întâlnit în realitate, din cauză de resurse și timp.

Metoda motivației. De cele mai multe ori s-a demonstrat că în proiectele software și nu numai, motivație reprezintă factorul de succes al acestora. Studiul efectuează o analiză a modelelor de motivație în proiecte, existente în literatura de specialitate și aduce cu sine propunerea unui nou model care înglobează modele de motivație și acoperă totodată carențele modelelor existente de motivație. Astfel modele de motivație precum Job characteristics model –JCM a ingineriilor software, modele axate pe satisfacția la locul de muncă, modele de motivație a programatorilor de sisteme deschise (*open sourcing*), modele ale influenței conducătorilor de echipe asupra motivației inginerilor de software, modele bazate pe teoria așteptărilor, teoria scopurilor și a comportamentului organizațional specifice procesului de dezvoltare software, modele ale arhitecturii activităților asupra motivației inginerilor de software, modele ale

influenței progresului în cariera profesională asupra motivației dar și ale influenței suportului social asupra motivației, au fost studiate și evaluate, reliefând atât părțile negative dar și cele pozitive. Rezultatul acestui studiu este un nou sistem de motivație atotcuprinzător (Sharp,H., Baddoo,N., Beecham,S., Hall,T., Robinson,H., 2009).

Citind și interpretând acest studiu, am putut identifica o mare problemă cu care companiile se confruntă în stadiul actual, un stadiu al globalizării și incertitudinii economice, pe care lucrarea de mai sus nu o tratează. Plecând de la teoria „angajat mulțumit” însemnând angajat motivat și interesat în atingerea scopului proiectului (figura 2) putem observa partea negativă a acestei teorii în care angajații nu sunt mulțumiți, ceea ce cauzează schimbarea locului de muncă deci implică o fluctuație de personal puternică în cadrul companiei. Aceste fluctuații aduc cu sine pierderea de cunoștințe de specialitate în domeniu, prelungirea duratei proiectelor, deci implică costuri mai mari în realizarea proiectelor. Aceste costuri sunt de cele mai multe ori mult mai mari decât dacă s-ar fi investit din timp în motivarea angajaților pentru a evita astfel de fluctuații. Din această cauză, trebuie bine calculate investițiile în motivarea angajaților în funcție de necesitatea firmei de a păstra cunoștințele de specialitate ce pot fi pierdute odată cu plecarea angajaților din firmă, și de asemenea trebuie avută în vedere rata de fluctuație de personal pe care firma o poate accepta sau suporta fără a influența funcționarea normală a acesteia.

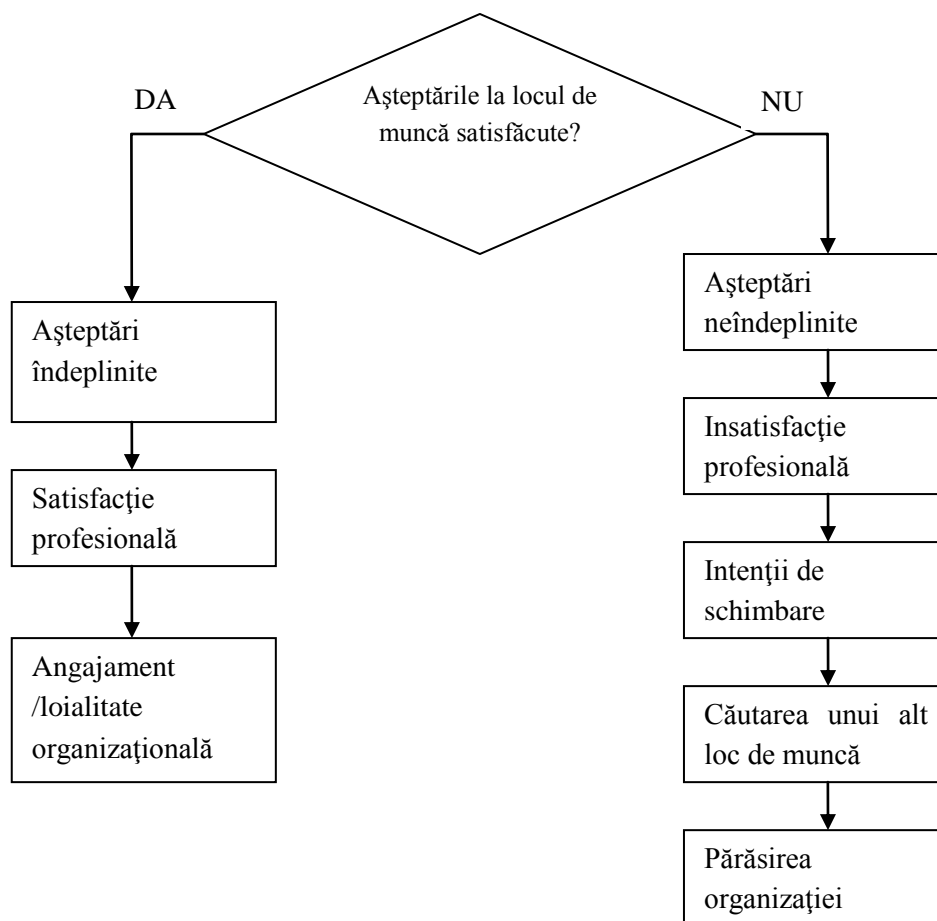


Figura 2 Teoria „angajat mulțumit”

Abordarea aplicațiilor software ca și bunuri de consum în planificarea și managementul lor are la bază faptul că în viziunea specialiștilor software, managementul aplicațiilor software nu este altceva decât o activitate de configurare. Această abordare este însă una falsă, activitățile de planificare fiind astfel

neglijate, ceea ce poate conduce ușor la eșecul proiectelor. Totodată, datorită gradului de complexitate foarte ridicat, unele proiecte devin foarte dificil de controlat și monitorizat de managerul de produs (în cazul nostru, aplicația software), astfel de cele mai multe ori un membru al echipei de implementare cunoaște mult mai multe despre produs decât însuși managerul, acestuia din urmă însă lipsindu-i cunoștințe de management și de valori ale produsului pentru companie. Din lipsa de cunoștințe asupra produsului rezultă o coordonare nesatisfăcătoare a proiectului. Această metodă își propune să ajute managerul de proiect să atingă un mai bun nivel al înțelegerii tehnice a proiectului pentru a putea desfășura și alege mecanismele adecvate ale managementului de proiect (Mordechai Ben-Menachem, 2008).

Metoda PTA (*Process tradeoff analysis*): dezvoltată de Raffo în 1996, este o dezvoltare a unei abordări cantitative asupra evaluării potențialelor schimbări de proces cu scopul de a determina costurile de dezvoltare, calitatea produsului, precum și resursele de timp ale proiectului. Această metodă este folosită cu preponderență în proiectele software, și anume în funcția evaluării resurselor de timp, care constituie totodată una dintre criteriile majore în simularea aplicațiilor software descrise de Kellner și Raffo (1996).

Metoda PROMPT (*PROject Management of Process Tradeoffs*) are la bază metoda PTA și are ca scop realizarea unei abordări predictibile care să aducă suport managerului proiectului pentru funcția de control. Informația utilă în această metodă este constituită din datele actuale de proiect, astfel încât predicția rezultatului devine foarte aproape de realitate (Raffo, D.M., 2005).

2.6.3 Analiză sintetică a diverselor modele de management pentru proiectele software

Acest studiu a fost realizat în mod comparativ, modelele selectate fiind următoarele: SCRUM, XP, FUZZY, PTA, PROMPT, PRINCE2 precum și alte modele care aduc contribuții semnificative la modelele de bază enumerate mai sus.

Tabelul 2 prezintă caracteristicile metodelor enumerate mai sus pentru o vizualizare directă a caracteristicilor acestor metode:

Criterii caracteristice de modelare a aplicațiilor software						
Metoda	Definirea de roluri/ responsabilități	Timpul pentru comunicare	Necesitatea de adaptare membrilor echipei	Rata pierderii de informație utilă	Experiență managerială	Cunoștințe tehnice ale managerului de proiect
Scrum	nu	mare	da	mare	medie	vastă
XP	nu	mare	da	mare	mare	mare
Fuzzy	da	mare	condiționat	medie	vastă	mare
Prompt	da	mare	condiționat	medie	vastă	mediu
PTA	da	mare	condiționat	medie	vastă	mare
PRINCE2	da	mediu	condiționat	mica	vastă	mediu

Tabelul 2 Criterii caracteristice de modelare a aplicațiilor software

Datorită timpului foarte mare necesar în comunicare și al altor factori care pot periclita desfășurarea normală a proiectului atrăgând întârzieri/costuri, Procaccino și Verner (2006) au dezvoltat metode pentru

sprijinirea managerului de proiect în alegerea adecvată a mecanismelor de management. Mordechai Ben-Menachem, (2008) studiază problema comunicării dezvoltând astfel o metodă care ajută managerul de proiect să atingă un nivel mai ridicat al înțelegerii tehnice, diminuând astfel necesitatea comunicării pe ramură tehnică. McBride, T. (2008) studiază adaptabilitatea membrilor echipei, realizând astfel o metodă pentru adaptarea aplicațiilor software la cerințe noi. Totodată metodele trebuie să fie robuste, durabile și cu un grad mare de adaptabilitate.

Tabelul 3 ne arată performanțele oferite de aceste metode în funcție de cerințe. Pentru realizarea acestei comparații s-a utilizat un punctaj pe o scară între 1 și 5:

Caracteristici de modelare					
Metoda	Fiabilitatea	Robustețea	Adaptabilitate	Standardizare prezentă	Generic
Scrum	3	3	3	da	nu
XP	3	3	3	da	nu
Fuzzy	4	3	2	nu	nu
Prompt	5	4	3	nu	nu
PTA	5	4	3	nu	nu
PRINCE2	5	4	3	da	nu

Tabelul 3 *Caracteristici de modelare*

După cum se poate observa modelele Prompt, PRINCE2 și PTA au obținut punctaj maxim în ceea ce privește fiabilitatea acestora. Singurul dezavantaj însă pe care aceste două modele îl reprezintă este necesitatea unui timp foarte mare de analiză și de studiu al proiectului de realizat, acest timp nefiind întotdeauna disponibil.

Cu toate deficiențele pe care unele modele le reprezintă, managerii de proiect utilizează metode care nu sunt propice proiectului de îndeplinit. Aceasta se datorează lipsei de informații asupra metodelor existente, lipsei popularității metodelor, precum și unor politici manageriale din cadrul unor companii. De regulă majoritatea companiilor folosesc metode moștenite de-a lungul timpului, bazându-se pe principiul „best practice” cu toate că natura și profilul proiectelor noi câștigate este complet diferită. Tabelul 4 oferă o privire de ansamblu asupra metodelor clasice și utilitatea acestor:

Utilitatea metodelor			
Metoda	Nivelul de utilizare	Gradul de dificultate în utilizarea metodei	Popularitatea sistemului
Scrum	des	mediu	mare
XP	des	mediu	mare
Fuzzy	rar	mare	mică
Prompt	rar	mare	mică
PTA	rar	mare	mică
PRINCE2	mediu	mare	medie

Tabelul 4 *Utilitatea metodelor*

În urma acestor comparații se pot observa lipsurile acestor metode și riscul de nerealizare pe care proiectele software încă îl au, indiferent de metoda aplicată. Acest risc este unul esențial mai ales dacă aceste proiecte software nu sunt altceva decât subproiecte ale unor proiecte principale din industria construcției componentelor de autovehicule. Aceste nerealizări pot duce la întârzieri în realizarea autovehiculelor, costurile în aceste cazuri fiind uriașe.

În figura 3 se pot observa mai bine punctele slabe ale metodelor existente în managementul aplicațiilor software din stadiul actual. Punctajul folosit este pe o scară de la 1 la 10:

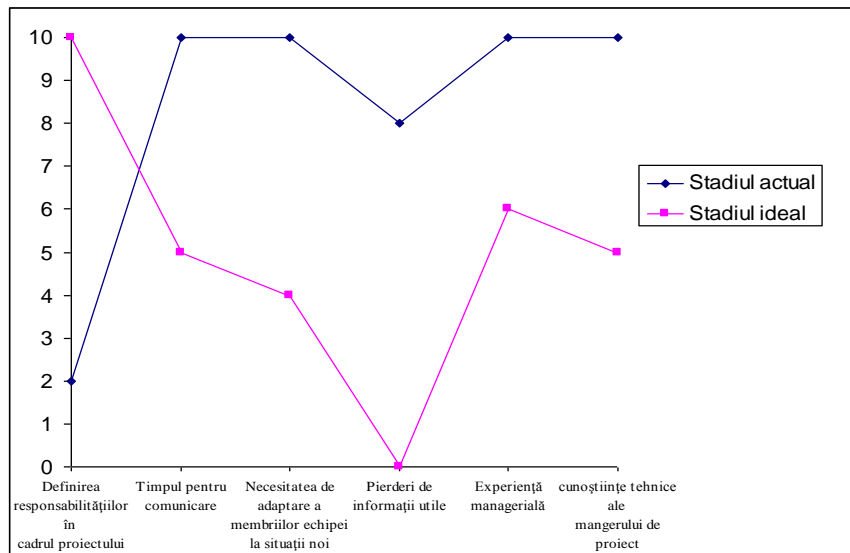


Figura 3 Studiu Comparativ între stadiul actual și cel ideal

Domeniile care trebuie urmărite pentru obținerea stadiului ideal sunt reprezentate în figura 4:

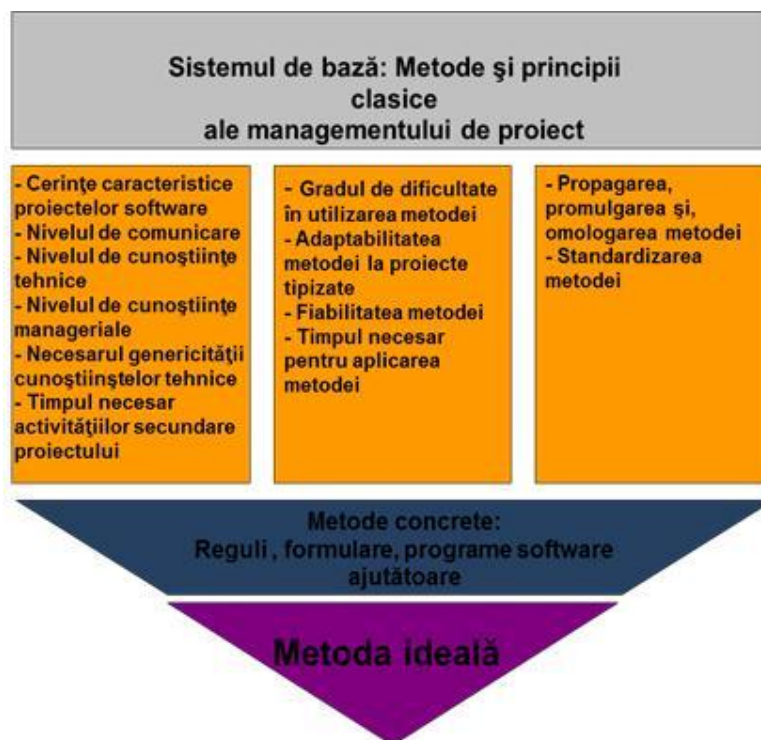


Figura 4 Domenii de îmbunătățire

Capitolul-3

Sisteme de management al calității și securității aplicațiilor software

Sistemele de calitate precum ISO (International Organization for Standardization) 9001 respectiv ISO 90003 (corespunzător proceselor) și ISO 25000 (sau 9126 corespunzător produselor) pentru aplicațiile software, reprezintă expresia cea mai clară a principiilor de gestiune a calității muncii, produselor și serviciilor. Politica de asigurare a calității este una formală la nivel de management, strâns legată de planul de afaceri și (cel mai important) de nevoile clienților, în cadrul căreia fiecare angajat lucrează cu obiective măsurabile. Deciziile legate de calitate se iau doar pe baza datelor înregistrate, iar sistemul este în continuu auditat pentru a asigura conformitate și eficiență. Înregistrările arată clar cum informațiile brute ajung să fie prelucrate, permițând urmărirea până la nivel de sursă a informației, iar modul în care se desfășoară comunicarea cu clienții este unul optim, permițând înțelegerea și înregistrarea ușoară a cerințelor, întrebărilor și a feedback-ului clientului.

3.3 SISTEME ALE CALITĂȚII ȘI SECURITĂȚII APLICAȚIILOR SOFTWARE

Calitatea și securitatea aplicațiilor software s-au arătat de cele mai multe ori simțite în practică, aceste două caracteristici esențiale având urmări majore asupra produselor controlate de software, fie că aceste produse sunt servicii (servicii de piață, servicii interne, și servicii publice etc.), produse soft (regulamente, proceduri, software), produse hard (care sunt tangibile: mașini, produse industriale, bunuri de larg consum) sau produse procesate (care sunt materiale ce au suferit prelucrări).

Calitatea a devenit, de-a lungul timpului, un factor foarte important într-un mediu economic tot mai competitiv. Această constatare se face tot mai mult resimțită și pe piața economică a aplicațiilor software. Încă din faza de proiectare și până la livrarea acestora spre consumatorii finali, aplicațiile software au nevoie de o atenție și o tratare foarte detaliată, dat fiind nivelul de complexitate și dificultatea de înțelegere a acestora, fără o pregătire de specialitate corespușzătoare.

3.3.1.1 Standardul ISO 9001 în dezvoltarea de aplicații software

După cum am specificat, în capitolul precedent, ISO 9001 reprezintă ansamblul de cerințe pentru un sistem de management al calității. Astfel, în mod teoretic, el poate fi aplicat și pentru dezvoltări de proiecte software. După cum reiese și din această normă, ea se aplică în asigurarea calității în proiectare, dezvoltare, producție, instalare și alte servicii. ISO 9001 este dedicată și scrisă pentru industria prelucrătoare, iar acest aspect ridică unele probleme atunci când se aplică la dezvoltarea și întreținerea software-ului.

Întrebarea care se pune în urma afirmației de mai sus este: care este cauza care face ca aplicarea normei în proiectele software să fie de cele mai multe ori dificilă ?

Diferențele majore dintre industria prelucrătoare și cea a domeniului software, este expusă într-un mod foarte elocvent în figura 5 (Oskarsson, Ö., Glass, R., 1995):

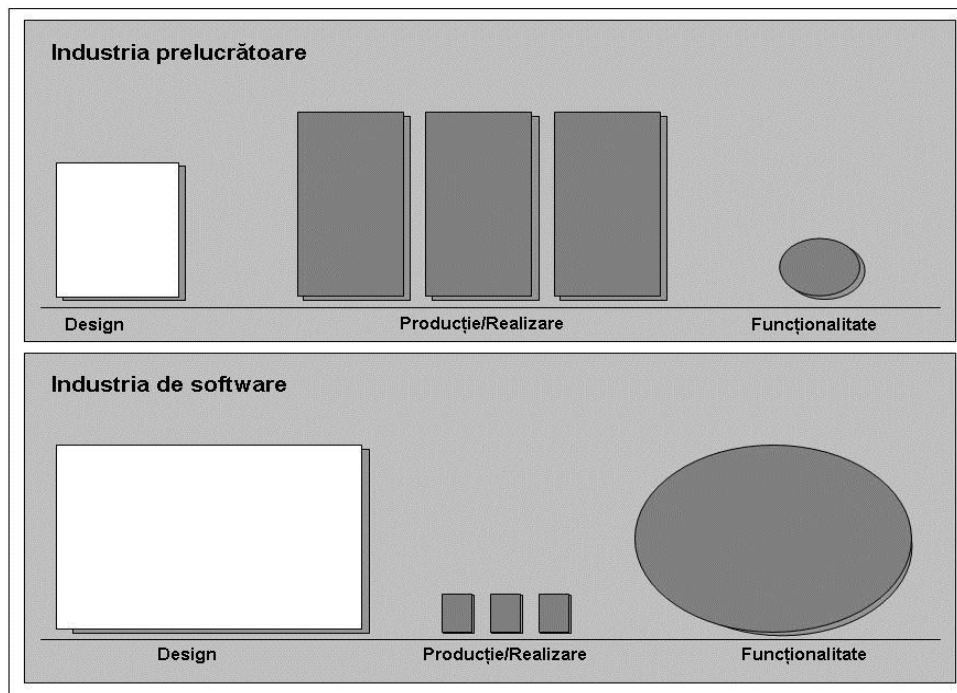


Figura 5 *Industria prelucrătoare versus industria de software*

Din această figură se poate deduce relativ ușor dificultatea aplicării normei ISO 9001 în industria de software. Dreptunghiurile din imagine reprezintă costurile sau efortul de obținere. Analizând activitățile din industria prelucrătoare se observă cum că *design-ul* este o activitate relativ redusă. Pe de altă parte, faza sau activitatea de prelucrare, de producție reprezintă partea majoră în realizarea de produse.

Standardul ISO 9001 cuprinde și elemente de design, dar se concentrează mai mult pe producție. De aici necesitatea unei norme precum: ISO 9000-3, TickIT, CMM, IEEE 730, AQAP-110, AQAP-150, ISO/IEC 12207 și altele care să ajute la controlul și suportul proceselor produselor din industria software.

3.3.1.1.3 Standardul ISO 9000-3

Acest subcapitol al normei ISO 9000 reprezintă un ghid de aplicare a cerințelor normei ISO 9001, acolo unde design-ul aplicațiilor software, dezvoltarea, instalarea și mentenanța, reprezintă un element al afacerii cu furnizorii (EN ISO 9000-3:1997):

- ca parte a unui contract comercial cu o organizație externă;
- ca un produs disponibil unui sector al pieței;
- în suportul proceselor unei afaceri a furnizorului;
- ca software încapsulat într-un produs hardware.

Standardul ISO 9000-3 identifică acele misiuni ce trebuie îndeplinite fiind independent de tehnologie, cicluri de viață ale produselor, procese de dezvoltare, sau structuri organizaționale folosite de furnizor.

Necesitatea unei interpretări a normei ISO 9001 pentru software a fost semnalizată încă din anii 1998 când ISO publică prima variantă în acest scop, pe care o numește ISO 9000-3 cu titlul "*Quality management and quality assurance standards - Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software (ISO 9000-3:1997)*" (Oskarsson,Ö., Glass,R.,1995).

Plecând de la aceeași idee, de a îndruma aplicarea cerințelor specificate de ISO 9001, au existat și alte inițiative de interpretare și suport alături de ISO 9000-3 precum TickIT.

3.3.1.1.4 Inițiativa TickIT

La sfârșitul anilor 1980, standardele calității au început să devină tot mai populare și în Europa, astfel numărul organizațiilor industriale certificate conform ISO 9000 devine tot mai mare (Oskarsson,Ö., Glass,R.,1995).

TickIT nu este altceva decât un alt sistem pentru suportul certificării ISO 9001 în dezvoltarea și întreținerea de software, asemenea cu ISO 9000-3 și este alcătuit din 6 elemente (Oskarsson,Ö., Glass,R.,1995):

- o interpretare a ISO 9001 pentru software;
- un set standard de cerințe pentru competență și comportament al auditorilor;
- o pregătire standardizată pentru certificarea auditorilor;
- un sistem de înregistrare pentru auditorii certificați acceptați;
- un sistem de acreditare a instituției acreditate pentru realizarea certificării;
- un logotip pe certificate pentru a justifica certificarea TickIT.

În zilele noastre, acest sistem însă nu mai este acceptat peste tot. Spre exemplu, RAB (Registrar Accreditation Board) ale autorității de acreditare SUA, nu admite sisteme de certificare specifice software, solicitând în continuare certificare cu ISO 9001

3.3.1.1.5 Sistemul SEI (Software Engineering Institute) de maturitate a capabilității CMM (Capability Maturity Model)

CMM a fost dezvoltat de institutul de dezvoltare software în Pittsburg și a apărut în concurență cu ISO 9001 ca un rival demn de luat în considerare. Acest sistem reprezintă mai mult un criteriu ajutător de clasificare a procesului de dezvoltare software în concordanță cu capabilitatea acestuia (Bamford RC, Deibler WJ, 1993).

Astfel putem identifica 4 diferențe principale între ISO 9001 și CMM (Paulk MC, Bamford RC, Deibler WJ, 1994):

- ISO 9001 se adresează mai mult industriei prelucrătoare, pe când CMM este specific pentru industria software;
- CMM este mai detaliat și mult mai specific;
- ISO 9001 presupune acceptarea unui singur nivel de management al furnizorilor și proceselor, pe când CMM este un sistem de evaluare a performanței și capacității software-ului furnizorului pe o scală de nivele de la 1 la 5;
- ISO 9001 se concentrează pe relația client – furnizor, iar CMM se concentrează asupra procesului de realizare a aplicațiilor software.

CMM implică următoarele aspecte:

Nivele de maturitate: CMM este dispus pe 5 nivele, în care cel mai înalt nivel este nivelul 5 și reprezintă un stadiu ideal în care procesele sunt controlate prin intermediul combinației între procesele de optimizare și cele de îmbunătățire continuă.

Zone cheie ale procesului (KPA- Key Process Area): ansamblu de activități, care acționând împreună, ating scopuri considerate importante.

Scopuri: reprezintă stadiul care trebuie să existe pentru zonele cheie ale procesului pentru ca acestea să poată fi implementate într-un mod eficient și de durată. Cu ajutorul scopurilor se semnalizează domeniul de aplicare, limitele și intenția fiecărei zone de proces.

Caracteristici comune: sunt reprezentate de practicile de implementare și instituționalizare a zonelor cheie ale procesului. Astfel putem diferenția 5 tipuri de caracteristici comune:

- angajamentul de a efectua;
- capacitatea de a efectua;
- activitățile efectuate;
- măsurare și analiză, și
- verificare a implementării.
- practici cheie: descriu elementele unei infrastructuri și contribuie la implementarea și instituționalizarea zonelor KPA.

O reprezentare grafică a nivelelor CMM și modalitatea/condiția de trecere dintr-un nivel în altul se poate observa în figura 6:

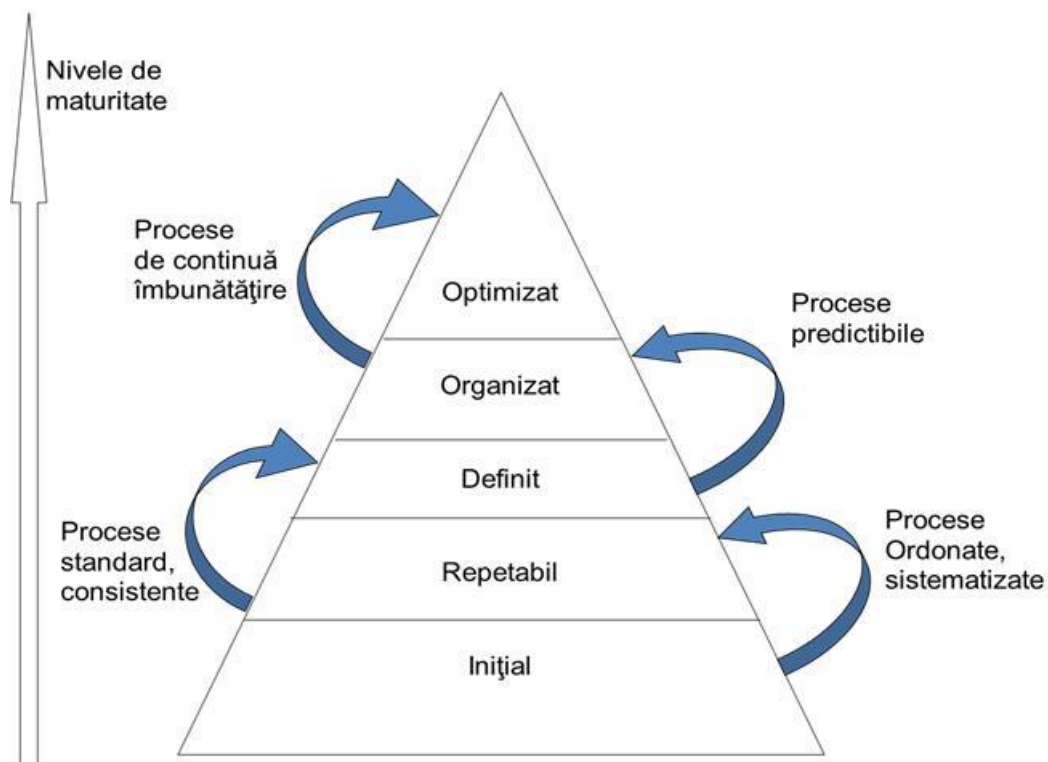


Figura 6 Nivelurile CMM

Arhitectura CMM este destinată pentru a ghida pe cei care doresc evaluarea unei organizații / unor proiecte în concordanță cu CMM.

Multe publicații științifice au comentat și discutat relația între ISO 9001 și CMM. În urma unor studii a rezultat că o organizație certificată cu ISO 9001 poate satisface cerințele nivelului 2 al sistemului CMM (F Coallier, 1994).

3.3.1.1.6 CMMI (*Capability Maturity Model Integration*)

Modelul CMM s-a dovedit util pentru numeroase organizații, dar aplicarea sa în dezvoltarea de software

a fost uneori problematică. Aplicarea unor modele multiple care nu sunt integrate în cadrul unei organizații ar putea fi costisitoare în ceea ce privește formarea profesională, evaluările și activitățile de îmbunătățire.

„Capability Maturity Model Integration” (CMMI), a fost conceput pentru a rezolva problema folosirii de sisteme multiple CMM. Astfel, pentru procesele de dezvoltare software, CMM a fost înlocuit de către Capability Maturity Model Integration (CMMI). CMM continuă însă să fie un model general, teoretic, utilizat în domeniul public (CMMI Guidebook, 2007).

După cum am amintit în capitolul precedent, primul sistem CMM a fost dezvoltat și conceput pentru proiectele software, la sfârșitul anilor 1990. Deoarece și în alte domenii a existat necesitatea unui asemenea sistem, CMM a fost folosit mai departe în domenii precum:

- integrarea și dezvoltarea produselor;
- resurse umane;
- tehnologia/ingineria sistemelor;
- achiziții (aplicații software);
- asigurarea calității aplicațiilor software;
- testare.

Datorită acestui fapt, organizațiile dețineau pentru fiecare domeniu enumerat mai sus un CMM independent. Aceste CMM-uri independente s-au dovedit a fi însă neproductive din următoarele motive: se suprapuneau, se contraziceau, aveau nivele diferite, lipsuri în interfețe și în standardizare.

3.3.1.1.7 Normele IEEE (*Institute of Electrical and Electronics Engineers*)

Alte organizații precum IEEE (*Institute of Electrical and Electronics Engineers*) au publicat o serie de standarde care se referă la dezvoltarea de software și la calitatea acestora. Acestea pot fi observate în tabelul 13:

Normele IEEE	
IEEE Std. 730-1989	Software Quality Assurance Plans.
IEEE Std. 829-1983	Software Test Documentation (bestätigt 1991.)
IEEE Std. 830-1984	Guide for Software Requirements Specification
IEEE Std. 983-1986	Software Quality Assurance Planning
IEEE Std. 1008-1987	Standard for Software Unit Testing
IEEE Std. 1012-1986	Software Verification and Validation Plans
IEEE Std. 1016-1988	Recommended Practice for Software Design Descriptions
IEEE Std. 1028-1988	Standard for Software Reviews and Audits
IEEE Std. 1042-1987	Guide to Software Configuration Management
IEEE Std. 1058.1-1984	Standard for Software Project Management Plans
IEEE Std. 1061-1992	Standard for Software-Quality Metrics Methodology

Tabelul 5 Normele IEEE

IEEE 730 a fost intitulat „IEEE Standard for Software Quality Assurance Plans – Standardul IEEE pentru planurile de asigurare a calității software”(SQAP). Ultima versiune a acestui standard datează din 1989. Așa cum o spune și titlul, acest standard se concentrează asupra conținutului planului de asigurare a

calității dar și asupra unor cerințe de management, pentru dezvoltarea de software, precum necesarul minim de documentație a reviziilor și auditurilor de aplicații software. IEEE 730 tratează aproximativ capitolul 4.4 „Design control” al normei ISO 9001.

Un sistem SQAP, descrie pentru un proiect sau produs, activitățile ce trebuie parcurse în cadrul managementului calității și nu conținutul acestora.

În general, standardele IEEE sunt interpretate ca blocuri menite să reunească elemente și activități în cadrul unui sistem de management al calității. Acestea includ obiective pentru componentele individuale ale unui sistem QMS (Quality Management System, software-specific), dar nu pentru întregul sistem al calității.

3.3.1.1.8 AQAP-110 și AQAP-150

În domeniul militar, se achiziționează o gamă variată de produse. Din acest motiv NATO (North Atlantic Treaty Organization) publică standarde de folosință la clienți în achiziția în masă.

AQAP-110(*Allied Quality Assurance Publication*), publicată în februarie 1995 sub numele „NATO Quality Assurance Requirements for Design, Development and Production”, nu este altceva decât ISO 9001 cu unele adăugări. Are aceeași structură și de multe ori observația regăsită în această normă este ”se aplică cerințele ISO”. Datorită faptului că această normă nu include cerințe software, asemenea cu ISO 9001, NATO publică norma AQAP-150 în martie 1993 și o actualizează mai târziu în Septembrie 1997. (Oskarsson,Ö., Glass,R.,1995), sub numele ” NATO Quality Assurance Requirements for Software Development”

Diferența de bază a acestei norme față de ISO 9000-3 privește formularea. În ISO 9000-3 regăsim formularea „ar trebui” pe când în AQAP-150 formularea „trebuie”. APQP-150 este un standard pe când ISO 9000-3 este un ghid orientativ.

AQAP-150 se deosebește de ISO 9000-3 prin faptul că este orientată mult pe probleme specifice proiectului, în esență conținând cerințe de conținut al SQAP („Software Quality Assurance Plan”) și al activităților controlate de acest plan. De asemenea, AQAP-150 nu conține reguli de calitate, analize manageriale, analize contractuale, audituri interne și instruirii sau mentenanță.

3.3.1.1.9. ISO/IEC 12207

În 1995, două organizații reprezentative, ISO (International Standard Organisation) și IEC (the International Electrotechnical Commission), publică standardul ISO/IEC 12207 sub denumirea „*Information technology – Software life cycle processes*” (Oskarsson,Ö., Glass,R.,1995). Acest standard se bazează pe standardul MIL-STD-498 (Military Standard-498) folosit în domeniul militar, pe care îl și înlocuiește în 1998.

ISO/IEC 12207 conține descrieri de procese, activități și sarcini implicate în procesul de achiziție, de furnizare, de operare și de întreținere a sistemelor și a aplicațiilor software.

Unul dintre dezavantajele acestei norme este dificultatea introducerii lui într-o organizație care deja are un alt sistem datorită complexității și dificultății de înțelegere și interpretare a acestei norme, atunci când este folosită pentru prima dată.

Cerințele în ISO / IEC 12207 sunt destinate a fi adaptate de către client înainte de a le impune unui

furnizor. Acest lucru este important mai ales pentru proiecte mici, caz în care standardul poate deveni ușor apăsător.

O modalitate bună de a folosi ISO / IEC 12207 este aceea de a fi folosit de către client. El include indicii pentru cerințele specifice din standard, pe care furnizorul trebuie să le îndeplinească.

Acest standard internațional oferă de asemenea un proces care poate fi utilizat în definirea, controlul, și îmbunătățirea ciclului de viață al proceselor software. Procesele, activitățile și sarcinile acestui standard internațional, fie singur, fie în colaborare cu ISO / IEC 15288 (*Systems and software engineering — System life cycle Processes*) ar putea fi, de asemenea, aplicate în timpul achiziționării unui sistem care conține software (ISO/IEC 12207:2008).

3.3.1.3 Exemple practice de aplicare a normelor

În cele ce urmează vom ilustra aplicarea acestor sisteme într-un exemplu din practica unei mari organizații europene din industria de autovehicule.

Studiul de caz, este realizat pe un proiect de transfer și conversie de date din sistemul CAD Proe/ProIntralink în format HPGL cu salvare în baza comună de stocare a datelor. Inițial acest proiect a fost realizat fără a se ține cont de nici o normă a calității sau managementului de proiect, satisfacția clientului fiind de aproximativ 30%, prin versiunea beta livrată totuși la termen.

Aplicația software a fost auditat în urma angajării de către firma contractantă a unui serviciu de consultanță pe care furnizorul a fost nevoit să îl accepte. Auditarea produsului s-a realizat conform ISO 9001, apoi am realizat auditarea conform și celorlalte norme enumerate mai sus. Astfel se poate observa variația rezultatului obținut pe același proiect cu norme și metode diferite.

Punctajul obținut poate fi transpus și interpretat ca și grad de rigurozitate în realizarea analizei. Cu cât punctajul obținut este mai mic cu atât lista de acțiuni corective sau îmbunătățiri este mai mare spre satisfacția clientului. În acest exemplu s-au folosit formulare tipice ale normelor („*checklist*”).

3.3.1.3.1 Implementarea conform ISO 9001

Pentru analizarea proiectului conform cu norma ISO 9001 ne-am folosit de tabelul de verificare tabelul 17, din care se poate observa punctajul obținut. Acest tabel a fost structurat în funcție de caracterul fiecărei cerințe, cerințele fiind grupate în elemente: astfel punctele 0 – Noțiuni generale, 1 – Scopul, 2 – Referințe normative, 3 – Reguli și condiții ale normei ISO 9001 au fost grupate în Elementul 1, punctul 4 – Sistemul Calității în Elementul 2, punctul 5 – Responsabilitatea Managementului și 6 – Managementul Resurselor în Elementul 3, punctul 7 – Realizarea Produsului în Elementul 4, iar punctul 8 – Măsurători, Analize și Îmbunătățiri în Elementul 5.

Fiecare element este calculat în procente și contribuie la rezultatul final. Diferența dintre aceste elemente nu are voie să fie semnificativă. Împărțirea în elemente ne oferă posibilitatea să recunoaștem cazul în care o evaluare a unui element nu este făcută în mod corespunzător. Interdependența dintre aceste elemente face ca evaluările lor să fie apropiate. Un alt avantaj al acestei împărțiri îl constituie faptul că un audit poate fi realizat de mai mulți auditori concomitent, elementele putând fi astfel distribuite corespunzător. Evaluarea s-a efectuat pentru fiecare subpunct calificativele fiind următoarele (Tabelul 6):

Tabela calificativelor	
Punctaj	Semnificația
10	Conformitate deplină cu cerințele
8	Respectare predominantă cu cerințele - neconformități minore
6	Respectare parțială cu cerințele - neconformități mai severe
4	Respectare nesatisfăcătoare cu cerințele - neconformități majore
0	Nici o conformitate cu cerințele

Tabelul 6 Tabela calificativelor

Interpretarea rezultatului a fost realizată conform tabelului 7. De remarcat că în cazul de față doar rezultatele A și AB sunt acceptate de organizație, B și C fiind considerate erori de sistem:

Interpretarea rezultatelor		
Clasa de calitate	Gradul general al nivelului de conformitate	Clasificarea procesului
A	90 - 100	Conformitate deplină
AB	> 90 <= 80	Conformitate predominantă
B	> 80 <= 60	Conformitate parțială
C	< 60	Fără conformități

Tabelul 7 Interpretarea rezultatelor

Comportamentul, respectiv efectul fiecărui element asupra evaluării finale poate fi observat în graficul din figura 7. Diferențele mari între elemente, dacă acestea există, reprezintă un indiciu că evaluarea nu s-a făcut îndeajuns de detaliat, sau corespunzător realității:

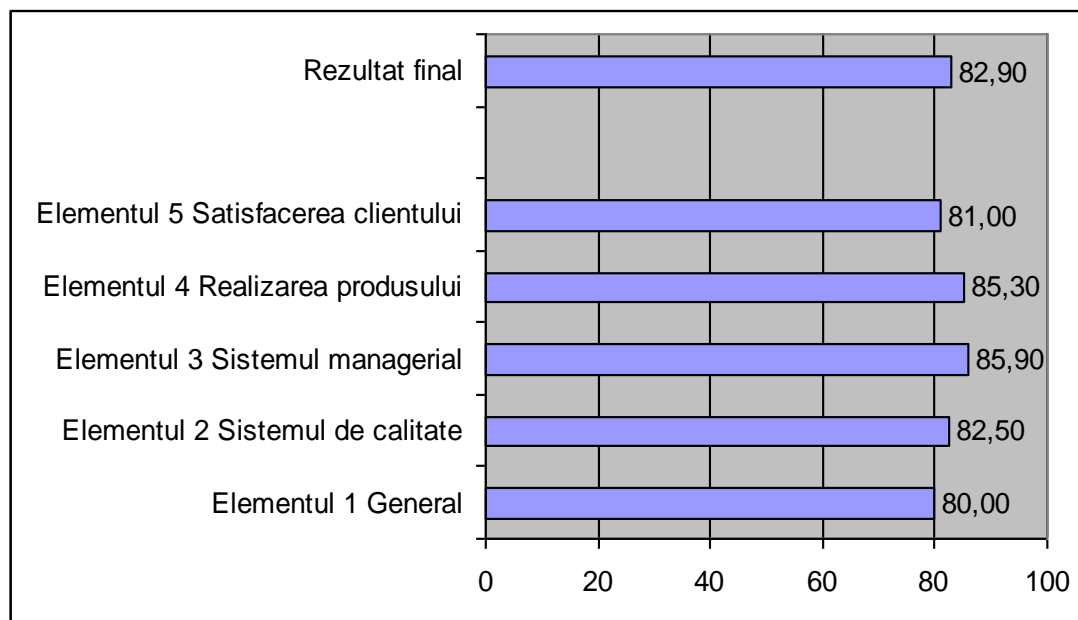


Figura 7 Realizarea și auditarea proiectului conform ISO9001

Analog s-a efectuat proiectul, respectiv auditarea acestuia, folosindu-ne de celelalte tehnici ale calității amintite mai sus. Rezultatele se pot observa după cum urmează în subcapitolele următoare.

3.3.1.3.2 Implementarea și auditarea conform normei ISO 9000-3

Dat fiind faptul că această normă nu este altceva decât un îndrumător pentru aplicarea normei ISO 9001, s-a obținut un rezultat și mai bun, unele aspecte fiind astfel mai bine înțelese obținând un punctaj mai ridicat. Elementele sunt astfel aceleași ca și la ISO 9001. Rezultatul obținut, precum și comportamentul fiecărui element se pot observa în figura 8:

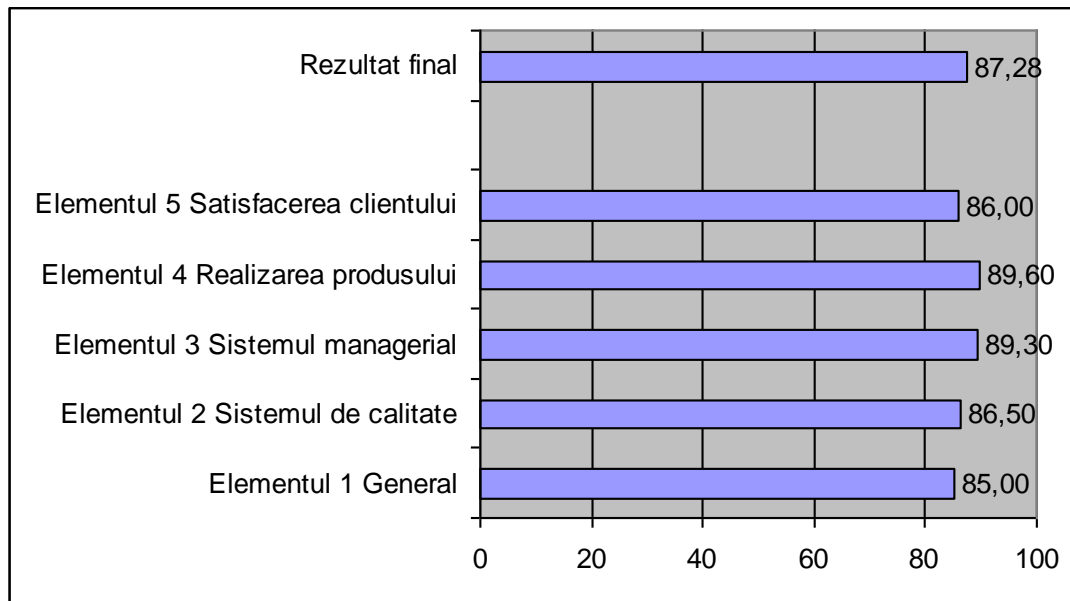


Figura 8 Realizarea și auditarea proiectului conform ISO90003

3.3.1.3.3 Realizarea în conformitate cu CMMI

În acest caz, cerințele au fost foarte exigente, cerându-se alinierea proiectului la nivelul 3 al CMMI, cu toate că ISO 9001 corespunde nivelului 2. Rezultatul obținut este unul mai rău decât celelalte două metode, tocmai datorită acestui fapt. Elementele acestui sistem se împart astfel:

- Elementul 1, **General** :
 - decizii și soluții în urma analizelor;
- Elementul 2, **Organizațional**:
 - definirea procesului organizațional;
 - instruire periodice în cadrul organizației;
 - focusarea pe procesul organizațional(dacă este aplicat corect, înțeles...);
- Elementul 3, **Managerial**:
 - managementul de proiect integrat;
 - dezvoltarea cerințelor;
 - managementul de risc;
- Elementul 4, **Realizarea produsului**:
 - integrarea produsului;
 - soluția tehnică;
- Elementul 5, **Validarea**:

- validarea produsului;
- teste, verificări conform specificațiilor;

Rezultatul obținut în urma evaluării proiectului se poate observa în figura 9:

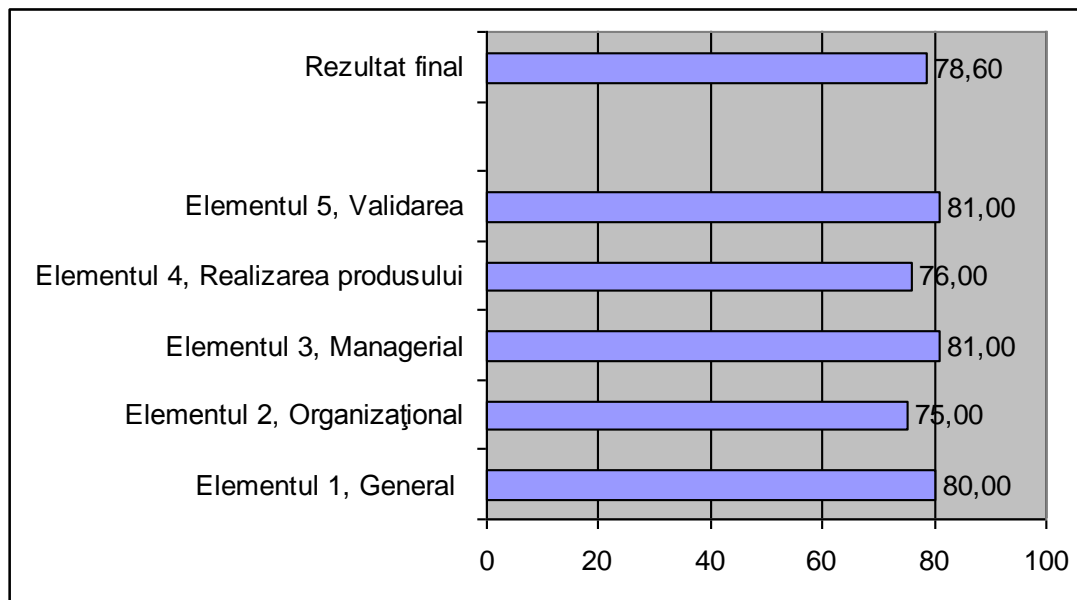


Figura 9 Realizarea și auditarea proiectului conform CMMI

3.3.1.3.4 Realizarea în conformitate cu AQAP 100/ AQAP150

Aceste norme, fiind foarte rigide/exacte în formulare și în ceea ce se cere, au obținut un punctaj foarte slab din proiectul exemplificat mai sus. Elementele sunt împărțite analog cu ISO 9001 și ISO 9000-3, diferența majoră constând în faptul că în ISO9001/ISO 9000-3 regăsim formularea „**ar trebui**” pe când în AQAP-100/AQAP-150 formularea „**trebuie**”. Elementele sunt identice cu cele de la ISO9001 respectiv ISO9000-3. Rezultatul obținut este redat grafic în figura 10:

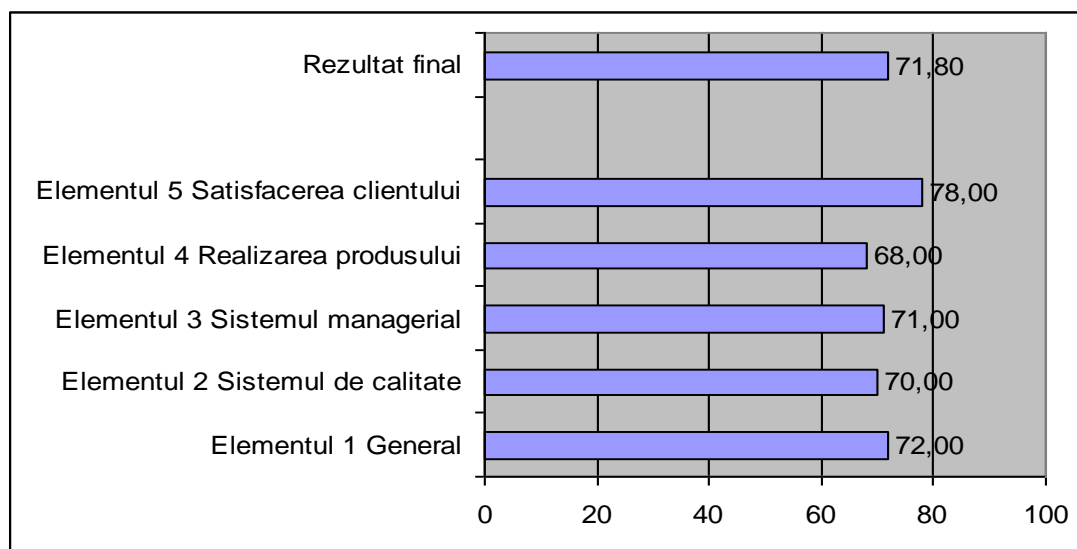


Figura 10 Realizarea și auditarea proiectului conform AQAP100/150

3.3.1.3.5 Realizarea în conformitate cu IEEE 730

Elementele în realizarea, iar apoi evaluarea proiectului au fost definite astfel:

- Elementul 1, **General:**
 - scopul;
 - documente de referință;
- Elementul 2, **Managerial:**
 - standarde, metode, convenții , metrice și
 - monitorizări de control a acestor cerințe;
 - utilități, tehnici și metode;
 - controlul sistemului;
 - controlul furnizorilor;
 - management de risc;
- Elementul 3, **Organizațional:**
 - organizația managerială;
 - instrucții interne;
- Elementul 4, **Realizarea produsului:**
 - documentație;
 - glosar;
 - colecția de date, mentenanța și stocarea acestora;
 - metodele și istoria modificărilor;
 - soluția tehnică;
 - introducerea produsului;
- Elementul 5, **Validare:**
 - revizuirii și audituri de software;
 - teste;
 - raportări de nonconformități și acțiuni corective;

Rezultatul obținut se remarcă în figura 11:

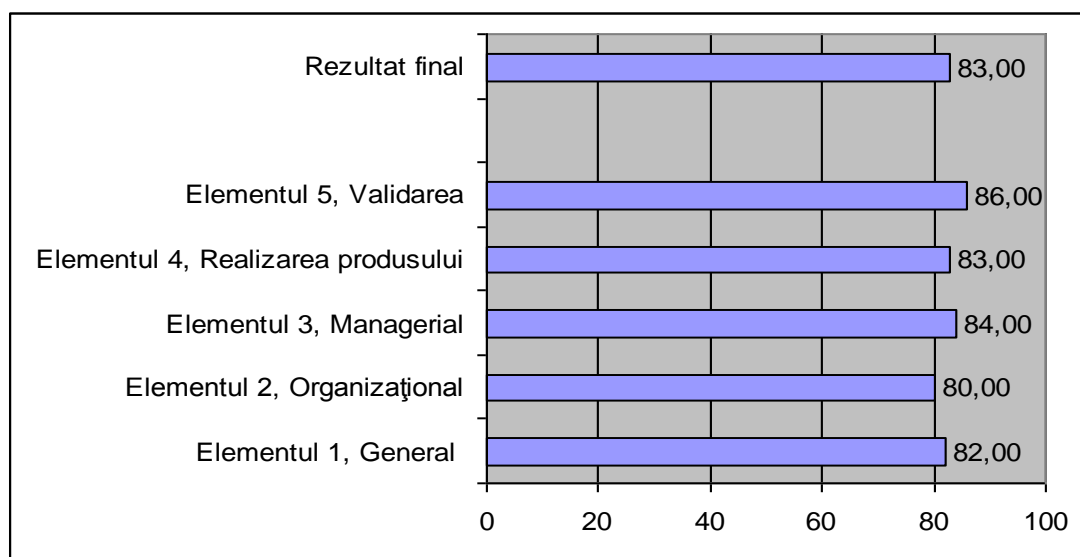


Figura 11 Realizarea și auditarea proiectului conform IEEE730

3.3.1.3.6 Interpretarea rezultatelor

Iată cum prin aplicarea unor metode diferite asupra aceluiași proiect, se obțin rezultate diferite, uneori chiar eșuări în calificarea acestuia. Totuși realizarea proiectelor având drept premiză oricare din aceste sisteme, este mult mai bună decât realizarea acestora fără nici un fel de sistem al calității, când satisfacția clientului este foarte scăzută. Figura 12 ne ajută să vizualizăm mai bine efectul benefic al acestor metode în detrimentul executării proiectului fără a se ține cont de nici o normă sau sistem cunoscut. Îmbunătățirea acestor rezultate se poate obține în urma optimizării acestor procese (folosindu-ne de tehnici sau norme precum ISO 15504). Alegerea metodei potrivite în efectuarea, respectiv auditarea produsului, rămâne la latitudinea managerului de proiect, care trebuie să ia decizia în aplicarea uneia dintre metode în funcție de politica firmei, necesitățile firmei, domeniul și mediul de aplicare, client, factori financiari precum și alți factori cunoscuți încă de la începutul proiectului. Metode și tehnici în alegerea adecvată a mecanismelor de management, au fost tratate în detaliu în capitolul 2.6 „Managementul aplicațiilor software”.

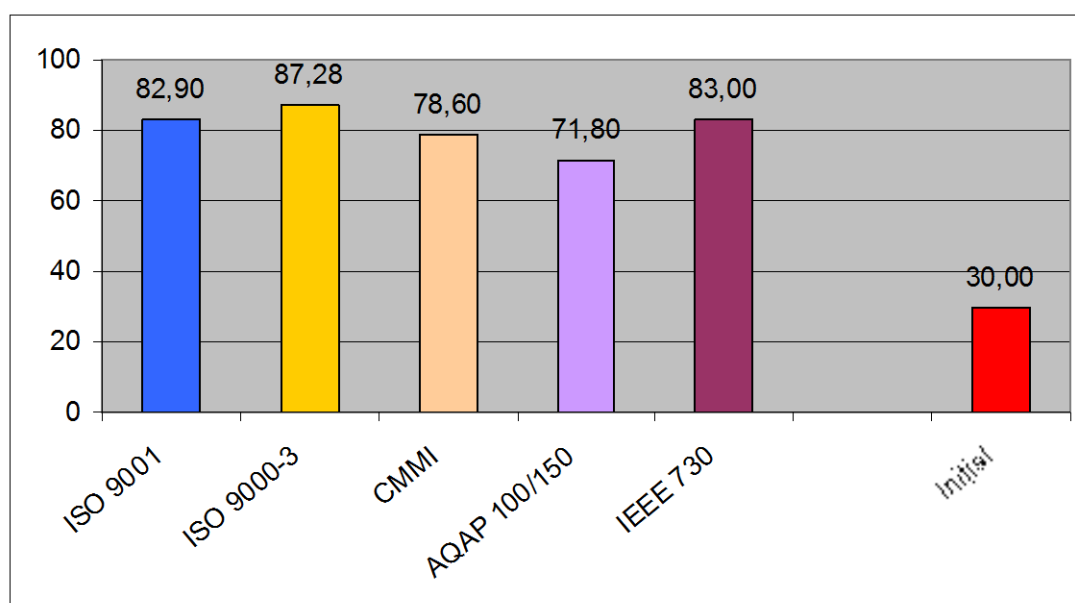


Figura 12 Rezultatele în urma auditului de software cu aplicarea de norme diferite

3.3.2 SECURITATEA APLICAȚIILOR SOFTWARE

Securitatea aplicațiilor software constă în realizarea acestora astfel încât ele să funcționeze corect chiar și în urma atacurilor dăunătoare a programelor virus asupra sistemelor computerizate în care acestea rulează (Mcgraw, G., 2004).

Domeniul securității aplicațiilor software este relativ unul nou. Primele cărți și studii în acest domeniu au apărut în 2001, prin care se arată conștientizarea tot mai aprofundată a dezvoltatorilor în crearea aplicațiilor software protejate. Datorită apariției recente al acestui domeniu este de înțeles de ce chiar și cele mai bune practici în realizarea aplicațiilor software protejate nu sunt folosite și uzuale.

3.3.2.4 Modelul de securitate SAMM din cadrul OWASP

Modelul de securitate software SAMM (Software Assurance Maturity Model), este o arhitectură de tip „open framework” a proiectului OWASP (Open Web Application Security Project) pentru a ajuta

organizațiile să formuleze și să implementeze strategii pentru securitatea aplicațiilor software, potrivit riscului specific cu care se confruntă organizația (OWASP, 2009).

Resursele oferite de acest model sprijină în:

- evaluarea practicilor existente de securitate software dintr-o organizație;
- realizarea unui program de asigurare a securității software în etape bine definite;
- demonstrarea îmbunătățiri concrete printr-un program de asigurare a securității;
- definirea și evaluarea activității de securitate dintr-o organizație.

SAMM a fost conceput a fi foarte flexibil, pentru a putea fi folosit de organizații mici, mari sau mijlocii cu orice stil de dezvoltare. În plus, acest model se poate folosi și în extensiile unei organizații sau chiar doar pentru un proiect individual.

3.3.2.5 Modelul de securitate Microsoft SDL

Microsoft SDL (*Security Development Lifecycle*) este un proces de securitate software practicat în industria de vârf; inițiativa pe scară largă a companiei Microsoft precum și normele obligatorii apărute în anii 2004, au determinat un rol critic de bază al modelului SDL în încorporarea normelor de securitate și confidențialitate în aplicațiile software și în cultura organizației Microsoft. Combinând o abordare holistică și practică, SDL introduce norme de securitate și confidențialitate în faze precoce și în toate etapele procesului de dezvoltare (Mead, N. R., Allen, J.H., 2010).

Livrarea de software sigur necesită un proces amplu, astfel Microsoft definește: „*Secure by Design, Secure by Default, Secure in Deployment, and Communications*” (Microsoft 2010) astfel:

- **secure by Design** – Securizat prin design/arhitectură:
 - asigură arhitectura, design-ul și structura. Dezvoltatorii trebuie să ia în considerare aspectele legate de securitate ca fiind aspecte de bază a design-ului arhitectural în dezvoltarea de software. Ei analizează posibilele modele detaliate pentru probleme de securitate, și ei proiectează și dezvoltă contramăsuri pentru toate amenințările;
 - modele de amenințare și atenuare. Modelele de amenințare sunt create și prezente în toate specificațiile de funcționare și design;
 - eliminarea vulnerabilităților. Nici o vulnerabilitate cunoscută a securității, care reprezintă un risc semnificativ în anticiparea funcționării aplicației software nu are voie să existe în cod după revizuire. Această revizuire include folosirea utilităților de eliminare a claselor de vulnerabilități;
 - îmbunătățiri de securitate. Protocoalele moștenite mai puțin sigure al căror cod este depreciat, trebuie prevăzute cu alternative sigure în concordanță cu standardele industriei.
- **secure by Default** – Securizat în mod implicit:
 - cel mai puțin privilegiat. Toate componentele rulează cu cele mai puține permisiuni posibile
 - apărare în profunzime. Componente nu se bazează pe o singură soluție de atenuare a amenințării, altfel utilizatorii rămân expuși în cazul în care aceasta nu reușește;
 - setările implicite conservative. Echipa de dezvoltare este conștientă de aria de atac pentru aplicația software și minimizează această arie în configurația implicită;
 - evitarea modificărilor implicit riscante. Aplicațiile nu fac nici un fel de modificări în configurația implicită a sistemului de operare sau setări de securitate care reduc gradul de securitate al calculatorului gazdă. În unele cazuri, cum ar fi pentru produsele de securitate, este acceptabil unui program software de a consolida (crește) setările de securitate pentru calculatorul gazdă. Cele mai frecvente încălcări ale acestui principiu sunt jocurile, pentru care porturile firewall trebuie să fie deschise, fără a informa utilizatorul sau a instrui utilizatorii să își deschidă

- porturile firewall și fără a informa utilizatorii asupra riscurilor posibile;
- servicii mai puțin folosite, trebuie închise în mod implicit. Acele servicii sau părți de program care sunt mai puțin de 80% folosite, trebuie închise în mod implicit.
- **secure in Deployment – Securizat în utilizare:**
 - ghiduri de implementare. Ghidurile de implementare prescriptivă trebuie să evidențieze modul de a disloca fiecare caracteristică a unui program de siguranță, inclusiv oferind utilizatorilor informații care să le permită să evalueze riscul de securitate prin activarea de opțiuni non-default (sporind astfel aria de atac);
 - analiză și instrumente de management. Instrumentele de securitate și de analiză de management permit administratorilor să stabilească și să configureze nivelul de securitate optim pentru o lansare de software;
 - utilități de implementare a patch-urilor. Utilitara de lansare ajută în implementare de patch-uri.
- **securizat în comunicare:**
 - reacția securității. Echipele de dezvoltare trebuie să răspundă prompt la rapoarte de vulnerabilități de securitate și să comunice informații despre actualizările de securitate;
 - angajamentul comunitar. Echipele de dezvoltare trebuie în mod proactiv să comunice cu utilizatorii, să răspundă la întrebări despre vulnerabilități de securitate, actualizări de securitate, sau schimbări în domeniul de securitate.

3.3.2.6 Modelul de securitate BSIMM

Modelul BSIMM (*Building security in maturity model*) a fost proiectat pentru a ajuta organizațiile să înțeleagă, să măsoare și să planifice securizarea aplicațiilor software (McGraw, G., Chess, B., Miguez, S., 2010). Acest model a fost creat de-a lungul unui proces de înțelegere și analiză a datelor din lumea reală. Cu toate că multe metode precum OWASP (Open Web Application Security Project), LASP (Lightweight Application Security Process), metoda Microsoft SDL (Security Development Lifecycle) sau metoda SST (Software Security Touchpoints) amintite mai sus, diferă între ele, aceste metode au totuși o bază comună. Această bază comună este identificată și capturată în metoda BSIMM. Ea este predestinată organizațiilor pregătite să adopte principiul de securizare al produselor software proprii și al căror program de management include:

- decizii conștientizate al managementului de risc;
- claritate asupra corectitudinii a ceea ce trebuie făcut de fiecare persoană implicată în securizarea aplicației software;
- reducerea de costuri prin procese standard repetabile;
- creșterea calității codului de programare.

BSIMM nu este un ghid complet pentru securizarea aplicațiilor software, sau un model aplicabil în orice situație, ci este o colecție de cele mai bune practici și activități folosite în ziua de astăzi. Modelul de maturitate este cel mai apropiat model pentru realizarea aplicațiilor software cât mai securizate, o componentă cheie pentru sistemele asigurate, deoarece prin îmbunătățirea securizării aplicațiilor software se schimbă filozofia și modul prin care o companie realizează aplicațiile software (Mead, N. R., Allen, J.H., 2010).

Capitolul-4

Model de sistem de calitate pentru aplicațiile software din industria construcției componentelor de autovehicule

4.1. Introducere

Managementul calității aplicațiilor software câștigă tot mai mult în importanță în toate domeniile economiei, datorită următorilor factori (Kneuper, R., Sollmann, F., 1995):

- pe de o parte, **calitatea aplicațiilor software** reprezintă un factor exponențial în domeniul concurenței, determinat de conștientizarea tot mai mare pe ramură calitativă a beneficiarilor;
- pe de altă parte, corecția de erori software s-a dovedit a fi foarte costisitoare; aceste costuri pot fi însă evitate prin introducerea și folosirea din timp a unui sistem de management al calității.

În mod tradițional managementul calității (atât pentru software cât și pentru produse de orice natură) se bazează pe faptul că un produs finit trebuie verificat/testat sau examinat, astfel încât să îndeplinească toate necesitățile/cerințele, iar în cazul defectelor identificate să le înlăture (ISO 9000). Această abordare s-a dovedit însă a fi insuficientă datorită costurilor ridicate pentru testarea unui produs și pentru procedeul de eliminare a neconformităților apărute la produsul finit. Din aceste motive, se preconizează o calitate a aplicațiilor software în mod indirect, prin îmbunătățirea calității **proceselor de obținere** a aplicațiilor software. Ipoteza care stă la baza acestei abordări, este că printr-un proces cu un înalt nivel de calitate, se pot obține în mod predictibil produse foarte calitative. Este real însă și faptul că și cu procese nestructurate se pot obține produse de înaltă calitate, dar rezultatele acestor procese variază foarte des între produse bune și produse slabe calitativ.

Ca urmare a acestei abordări, se pot diferenția două categorii de norme:

- norme de proces și norme de sistem ale managementului calității;
- norme de produs.

Acest capitol descrie realizarea aplicațiilor software care însoțesc producția, folosind diferite modele de calitate de produs, identifică deficiențele acestor modele și **propune un model de calitate nou (SCSP/QSPS – Sistem de Calitate pentru Software de Producție/Quality System for Production Software)** dedicat realizării aplicațiilor software utilizate în producția componentelor de autovehicule, model ce include norme de proces, norme de produs și recomandări de utilizare. Lucrarea se bazează pe studii de caz în aplicarea celor mai cunoscute norme/modele de calitate în realizarea unei aplicații software pentru producție, identifică deficiențele modelelor folosite și propune un model care are la bază cerințele de calitate din industria prelucrătoare pentru autovehicule precum ISO / TS 16949 într-o structură optimizată pentru utilizarea în producție. În capitolul „*Sisteme de management al calității și securității aplicațiilor software*” am analizat metode ale calității proceselor de obținere a aplicațiilor software precum ISO 9001, ISO 9000-3, TickIT, CMM și CMMI, AQAP-110/AQAP-150, IEEE 730/983 cu rezultatele obținute din figura 13.

Pentru a identifica cerințele industriei construcției componentelor de autovehicule am continuat studiul

prin analiza și aplicarea normei ISO / TS 16949 asupra proiectului software exemplificat și am analizat metode ale calității precum Automotive SPICE, ISO/IEC 25000/9126/14598. Alegerea modelelor de calitate pentru studiul de caz a fost realizată pe criterii ca:

- cele mai reprezentative și/sau
- cele mai frecvent utilizate în practică:

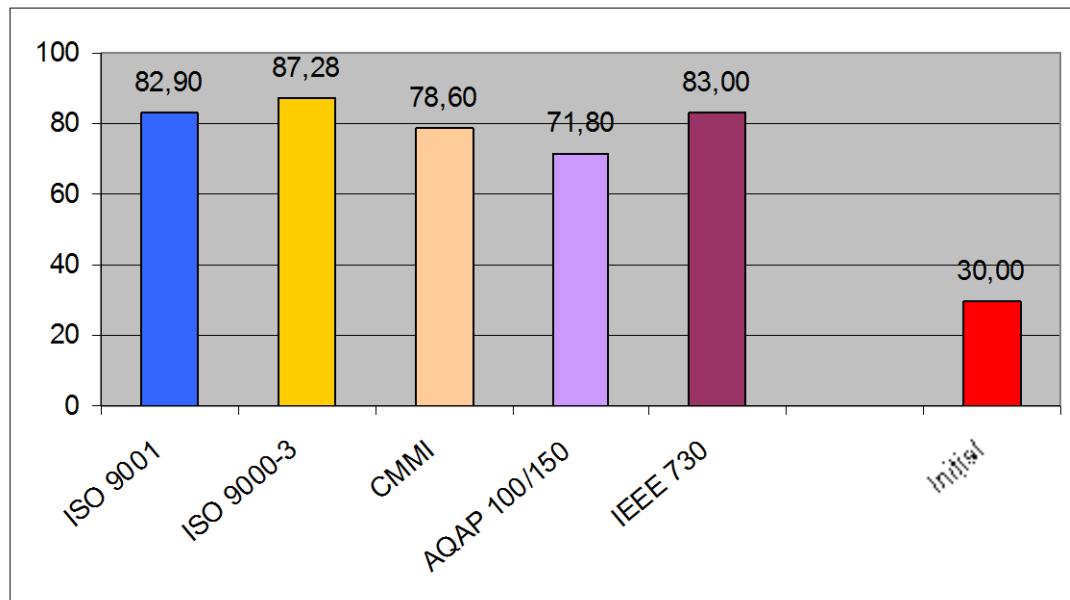


Figura 13 Rezultatele metodelor calității de proces pentru software (în %)

4.2. Modele de calitate de produs pentru software și aplicabilitatea lor în practică

Pentru a identifica gradul de eficiență al modelelor existente în aplicarea lor pentru proiectele software de producție, am folosit un exemplu practic, asupra căruia am aplicat consecutiv metodele enumerate mai sus și am analizat în mod comparativ rezultatele obținute. Analiza comparativă constituie baza modelului nou propus, model ce acoperă deficiențele metodelor existente în aplicarea lor pentru proiectele software de producție din industria componentelor de autovehicule. Exemplul folosit este proiectul software pentru trasabilitatea proceselor unei linii de producție.

4.3. Cercetări privind realizarea unui model de calitate pentru aplicațiile software de producție: reguli, cerințe și repere îndrumătoare

Datorită caracteristicii aparte a aplicațiilor software care însoțesc producția (dificultatea în testare, impactul asupra calității produselor finite), precum și importanței acestora în asigurarea continuității mediului de afaceri al unei organizații (controlul producției), aceste tipuri de aplicații software trebuie tratate în mod special și diferit față de alte tipuri de software a căror norme se regăsesc în literatura de specialitate. Aceste diferențe ale aplicațiilor software de producție se pot observa în figura 14:

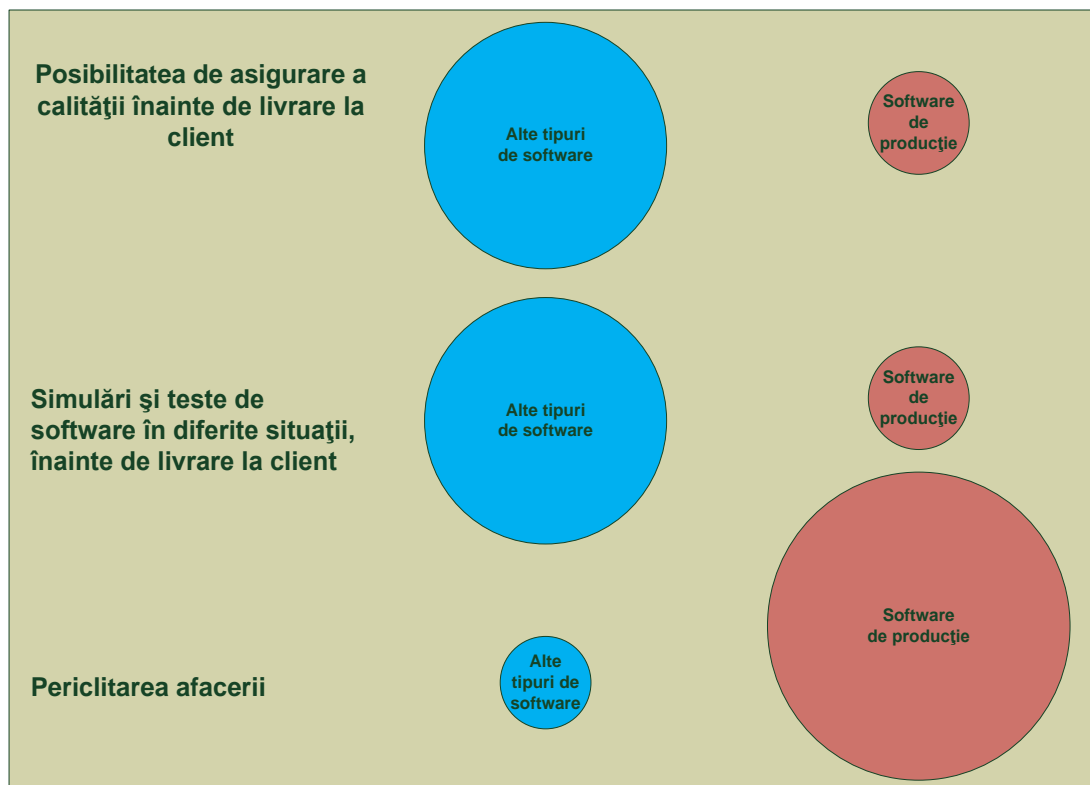


Figura 14 Diferențele majore ale aplicațiilor software de producție

Ținând cont de normele de calitate, de management, risc și securitate pentru software, studiate și analizate până în prezent am realizat **modelul de calitate QSPS** (Quality System for Production Software – Sistem de calitate pentru software de producție) pentru aplicațiile software folosite în producția componentelor de autovehicule.

4.3.1 MODELUL CONCEPTUAL (CADRU) QSPS

În prima etapă de realizare a modelului au fost identificate cerințele și conjuncturile de folosire actuală în producția modernă în care următoarele caracteristici sunt decisive pentru utilizarea și aplicarea modelului:

- timpul și gradul de dificultate necesare utilizării acestui model;
- eficiența sistemului în a salva resurse, costuri (sau a evita potențialele costuri suplimentare), de a controla proiectul și a asigura finalizarea lui;
- transparența ridicată a stadiului și dificultăților proiectului oferită prin folosirea metodei;
- asigurarea calității aplicației software, grija de a nu periclita bunul demers al producției;
- adaptabilitatea modelului la situații extreme și la diferite tipuri de software de acest gen;
- fiabilitatea sistemului;
- ușurința managerilor de a înțelege astfel aplicația software, chiar și cu un nivel de cunoștințe tehnice scăzut și
- îmbunătățirea comunicării între departamentele implicate.

Pornind de la metodele clasice și incluzând caracteristicile mai sus enumerate am conturat cadrul modelului propus care poate fi observat în figura 15:

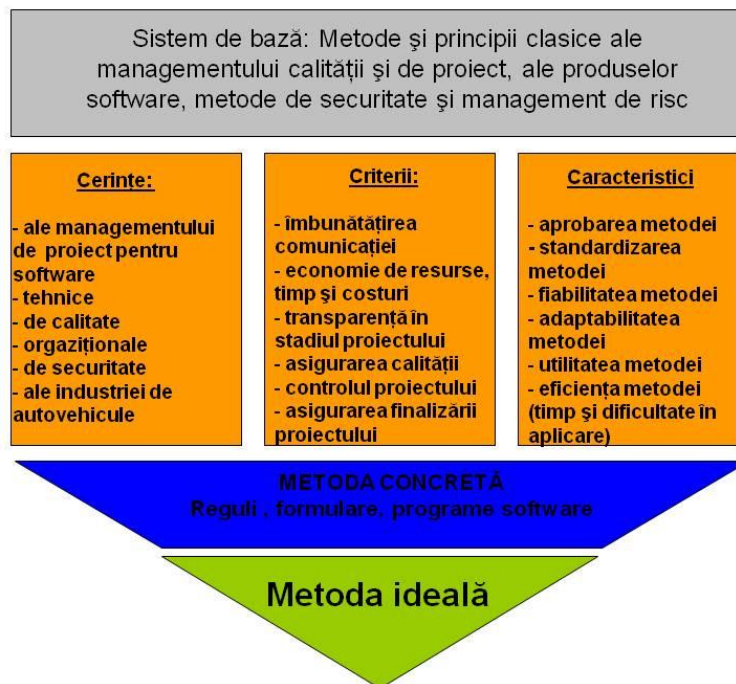


Figura 15 Model cadru de calitate pentru software de producție

4.3.2 PROCESUL ȘI MEDIILE DE APLICARE ALE METODEI QSPS

Luând în considerare cerințele de aplicare ale acestei metode în industria componentelor de autovehicule, am definit procesul de utilizare al metodei astfel: în scopul realizării unei siguranțe sporite în rularea aplicațiilor software pe liniile de producție, se recomandă ca metoda propusă să fie aplicată în trei medii diferite, medii rezultate de altfel, din interiorul cerințelor acestei metode. Deoarece aceste cerințe se referă la faze de pe parcursul dezvoltării, de după dezvoltare, în fazele de testare și la faze din timpul execuției și rulării aplicațiilor software de producție, mediile de aplicare a metodei sunt:

- **mediul de dezvoltare:** reprezintă un mediu tehnic foarte detaliat care oferă dezvoltatorilor de aplicații software toate facilitățile necesare dezvoltării și testării respectiv simulării aplicațiilor software implementate, precum și executării modificărilor planificate și neplanificate ce pot apărea pe parcursul dezvoltării, în fazele de testare și chiar mai târziu pe parcursul rulării în producție;
- **mediul de integrare:** este un mediu de execuție al aplicațiilor software foarte apropiat de mediul de rulare productiv care oferă module și facilități de simulare a aplicațiilor software implementate de către potențiali utilizatori selectați după anumite criterii strategice precum: tipul de producție, modul de utilizare, frecvența și volumul de utilizare. Cu cât mediul de integrare este mai apropiat/identific cu mediul productiv (de rulare) intensitatea și amplexul testului în mediul productiv scad.

Atât mediul de dezvoltare cât și mediul de integrare nu ar trebui să aibă influențe directe asupra mediului productiv de rulare al producției, pentru a nu perturba demersul normal al producției curente.

- **mediul de rulare (productiv):** reprezintă mediul de care se va folosi producția și în care aplicațiile software trebuie să ruleze cât mai corect, astfel încât, impactul negativ asupra producției să fie cât mai redus.

Figura 16 descrie mediile de aplicare a metodei propuse pentru aplicațiile software de producție.

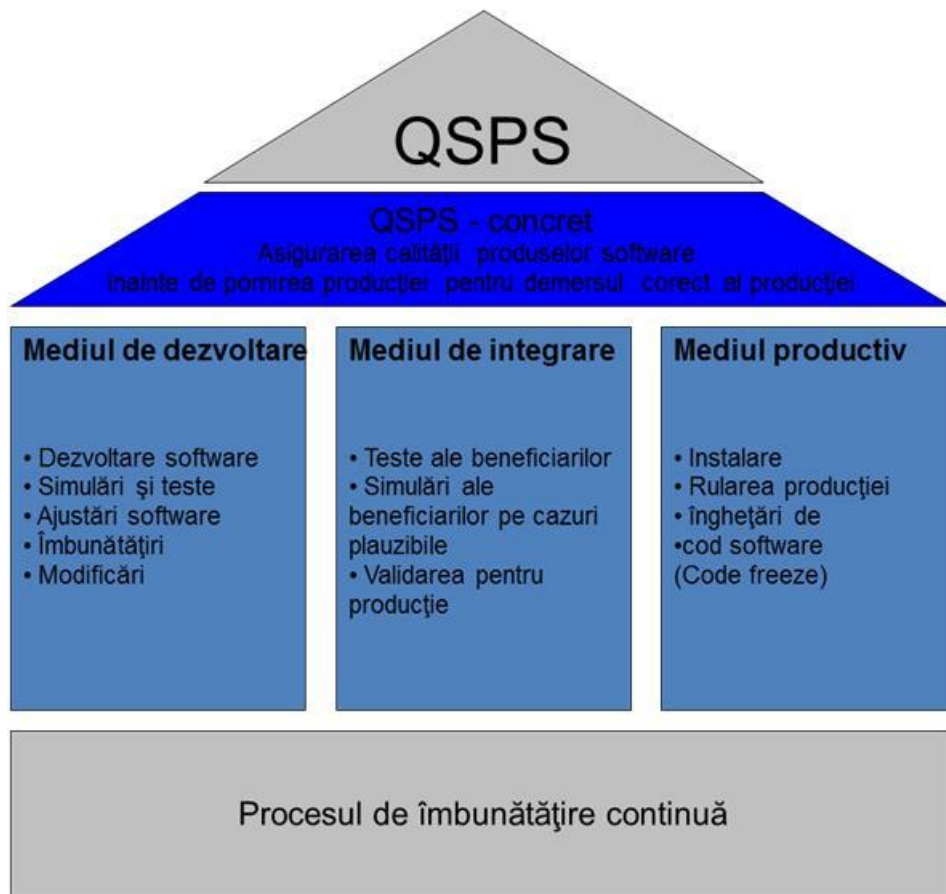


Figura 16 Mediile ciclului de viață recomandate de metoda QSPS

Procesul recomandat al metodei propuse QSPS în ciclul de viață al produsului este redat în figura 17:

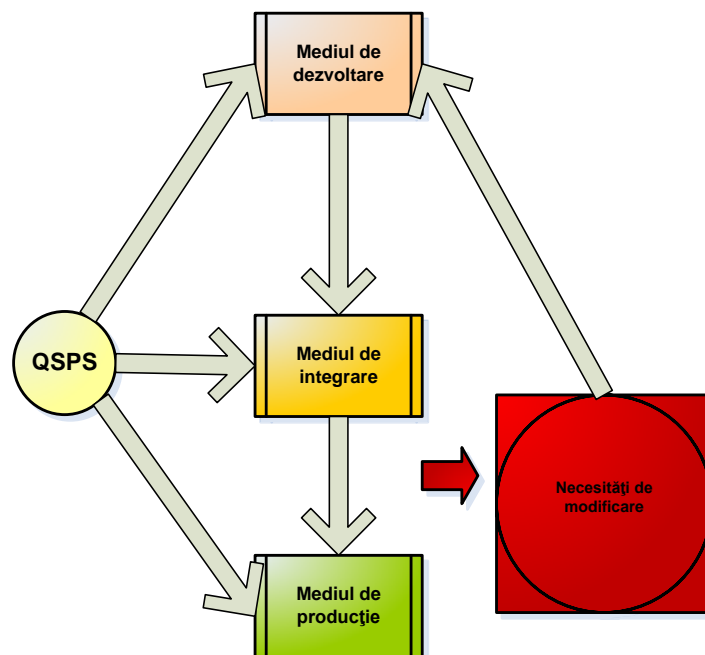


Figura 17 Procesul metodei QSPS

După cum se poate observa din figura 17, sistemul QSPS controlează toate activitățile întreprinse în mediile propuse și de asemenea și cronologia acestora în situații normale sau chiar divergente (tratarea de modificări) de la ciclul normal de viață al produsului.

4.3.3 ARHITECTURA MODELULUI DE CALITATE QSPS

Metoda propusă este constituită din 7 Elemente, atât din perspectiva dezvoltatorilor cât și din perspectiva evaluatorilor. Dacă metoda din perspectiva evaluatorilor conține cerințe, din perspectiva dezvoltatorilor ea conține template-uri și tabele ajutătoare pentru dezvoltatori astfel încât cerințele să fie îndeplinite. Totodată scopul metodei este de a diminua riscul opririi producției din cauza unor probleme de software, și astfel de a elimina și evita pierderile economice ce pot apărea în aceste cazuri.

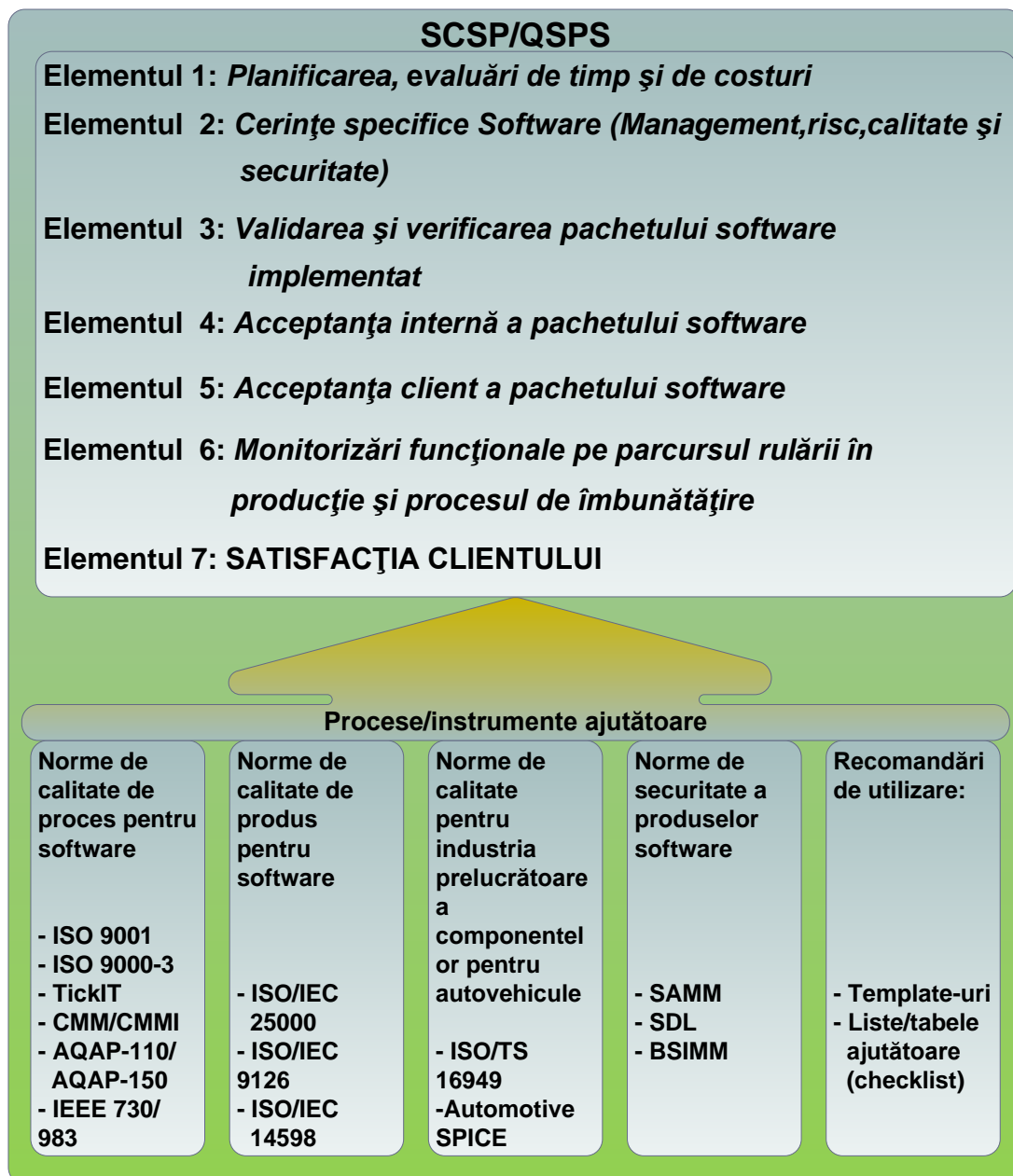


Figura 18 Arhitectura QSPS

4.3.4 ETAPELE ȘI ELEMENTELE MODELULUI DE CALITATE PROPUȘ QSPS

Astfel *Elementul 1, Planificarea, evaluări de timp și de costuri* conține cerințe și recomandări ale managementului de proiect cu privire la: planificare, evaluări de timp, de costuri precum și la riscurile în nefinalizarea proiectului software la data planificată pentru aplicația software de implementat.

Elementul 2, Cerințe specifice Software (Management, risc, calitate și securitate) conține cerințe și recomandări pentru aplicația software astfel încât funcționalitatea cerută să fie corect interpretată și implementată, riscurile să fie din timp identificate și calitatea aplicației software să corespundă cu cerințele produsului finit.

Elementul 3, Validarea și verificarea aplicației software implementate conține cerințe și recomandări privind validarea, calificarea și verificarea funcționalității aplicației software în conformitate cu cerințele definite.

Elementul 4, Acceptanță internă a aplicației software conține criteriile de acceptanță și aprobare internă a aplicației software implementate de către dezvoltatori înainte de livrare la client.

Elementul 5, Acceptanță client a aplicației software conține criteriile de acceptanță și aprobare a aplicației software implementate de către client.

Elementul 6, Monitorizări funcționale pe parcursul rulării în producție și procesul de îmbunătățire conține cerințe și recomandări de monitorizare funcțională a aplicației software pe parcursul funcționării în producție și de asemenea pentru procesul de îmbunătățire continuă.

Elementul 7, Reclamații de la client și serviciul de suport conține cerințe și recomandări pentru modul de tratare al reclamațiilor și de realizare a suportului tehnic pentru client.

Rezultatele obținute prin aplicarea experimentală succesivă a metodelor analizate pot fi observate în tabelul 8:

Metoda QSPS - Valori experimentale pentru modelarea matematică								
	Planificarea, evaluări de timp și de costuri	Cerințe specifice Software (Management, risc, calitate și securitate)	Validarea și verificarea aplicației software implementat	Acceptanță internă a aplicației software	Acceptanță client a aplicației software	Monitorizări funcționale pe parcursul rulării în producție și procesul de îmbunătățire	Satisfacția clientului	Total Procentaj obținut
Fără Metodă	77,7	30	20	90	50	40	30	48,24
ISO 9001(M1)	66,6	85,3	85	87	83,5	89	82,9	82,76
ISO 9000-3(M2)	98	89,6	92	93	89	92	87,28	91,55
CMMI(M3)	92,9	76	80	88	81	87	78,6	83,36
AQAP 100/150 (M4)	67,1	68	70	70	67	72	71,8	69,41
IEEE 730(M5)	108,2	83	82	81	84	86	83	86,74
ISO/IEC 25000 – SQUARE (M6)	100	78	81	87	89	89	88	87,43
ISO/IEC 9126 (M7)	77,7	83	85	88	82	78	80	81,96
ISO/IEC14598 (M8)	77,7	82	83	85	82	67	81	79,67
Automotive SPICE	77,7	89	91	93	87	83	94	87,81
ISO/TS 16949	109,2	56	56	70	66	92	90	77,03
QSPS	126,2	94	98	96	94	98	94	100,03

Tabelul 8 Metoda QSPS - Valori experimentale pentru modelarea matematică

În reprezentarea grafică din figura 19, se poate observa creșterea semnificativă a valorii calitative obținută de produs prin aplicare metodei QSPS:

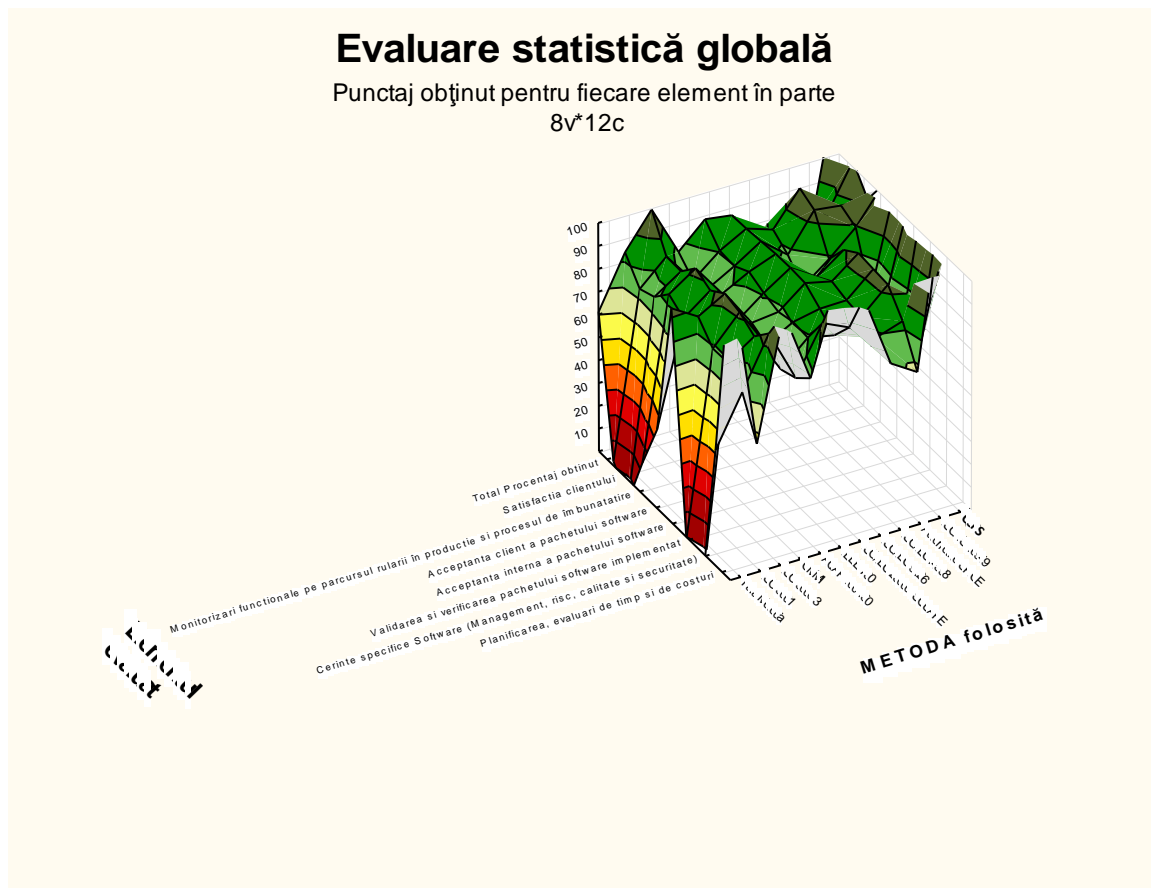


Figura 19 Evaluare comparativă a metodei QSPS

4.4. MODELAREA MATEMATICĂ A METODEI QSPS

Simulările din figura 19, scot în evidență deficiențele fiecărui element în parte, indicându-le prin nuanța verde, care reprezintă mediul cu potențial de îmbunătățire al elementelor.

Sistemul QSPS poate fi interpretat ca fiind suma combinațiilor elementelor sistemului de calitate din industria componentelor de autovehicule ISO/TS 16949 și cerințelor software din această industrie Automotive SPICE, cu toate celelalte elemente a metodelor studiate în parte, asemenea cu formula 1:

$$QSPS = \{E_1, E_2, E_3, E_4, E_5, E_6, E_7\} \quad (1)$$

Elementele sistemului QSPS $E_1 \div E_7$ au fost obținute prin compunerea elementelor sistemelor studiate cu elementele sistemului de calitate din industria componentelor de autovehicule și cu cerințele pentru software ale acestei industrii, asemenea cu formulele 2,3,4,5,6,7 și 8:

$$\begin{aligned}
 E_1 &= \sum_{i=1}^8 E_{1\text{ ISOTS16949}} \circ E_{1\text{ AutomotiveSPICE}} \circ E_{1M_i} = \\
 &= E_{1\text{ ISOTS16949}} \circ E_{1\text{ AutomotiveSPICE}} \circ \sum_{i=1}^8 E_{1M_i} \quad (2)
 \end{aligned}$$

$$E_2 = E_{2\text{ ISOTS16949}} \circ E_{2\text{ AutomotiveSPICE}} \circ \sum_{i=1}^8 E_{2M_i} \quad (3)$$

$$E_3 = E_{3\text{ ISOTS16949}} \circ E_{3\text{ AutomotiveSPICE}} \circ \sum_{i=1}^8 E_{3M_i} \quad (4)$$

$$E_4 = E_{4\text{ ISOTS16949}} \circ E_{4\text{ AutomotiveSPICE}} \circ \sum_{i=1}^8 E_{4M_i} \quad (5)$$

$$E_5 = E_{5\text{ ISOTS16949}} \circ E_{5\text{ AutomotiveSPICE}} \circ \sum_{i=1}^8 E_{5M_i} \quad (6)$$

$$E_6 = E_{6\text{ ISOTS16949}} \circ E_{6\text{ AutomotiveSPICE}} \circ \sum_{i=1}^8 E_{6M_i} \quad (7)$$

$$E_7 = E_{7\text{ ISOTS16949}} \circ E_{7\text{ AutomotiveSPICE}} \circ \sum_{i=1}^8 E_{7M_i} \quad (8)$$

Astfel putem spune că sistemul QSPS poate fi reprezentat astfel:

$$\begin{aligned} QSPS &= \{E_1, E_2, E_3, E_4, E_5, E_6, E_7\} = \\ &= \sum_{i=1}^7 E_{i\text{ ISOTS16949}} \circ E_{i\text{ AutomotiveSPICE}} \circ \sum_{j=1}^8 E_{iM_j} \end{aligned} \quad (9)$$

unde M_i este metoda studiată din exemplul practic observat în tabelul 26 .

Calificativul obținut al produsului software este o funcție direct proporțională cu media aritmetică al acestor elemente conform cu formula 10.

$$\text{Calificativ} = f(QSPS) = \frac{\sum_{i=1}^7 E_i}{7} \quad (10)$$

Pentru a demonstra cât de aproape de cazul real sunt aceste valori am realizat simularea din figura 20 în care am calculat valoarea medie ce poate fi obținută de fiecare element indiferent de metoda aplicată, urmată de o simulare a elementelor pentru a arăta impactul acestora asupra rezultatului final și satisfacția clientului.

În urma acestei simulări putem concluziona faptul că metodele existente pentru produsele software se concentrează cu predilecție pe elementele 2-Cerințe specifice Software (Management, risc, calitate și securitate) și 3- Validarea și verificarea pachetului software implementat, astfel încât pentru celelalte elemente am arătat o atenție sporită în implementarea sistemului QSPS.

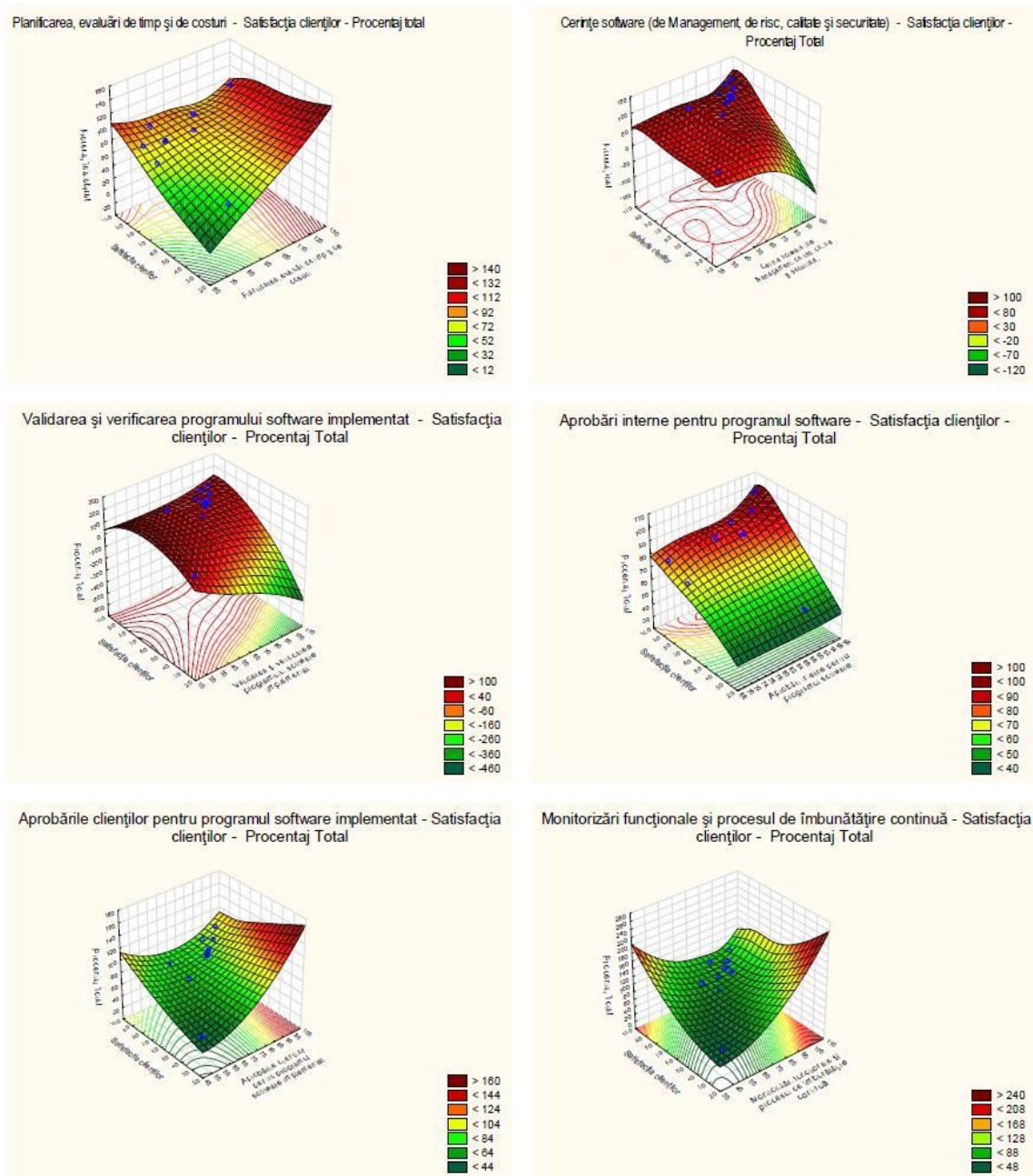


Figura 20 Modelări matematice ale QSPS

Proiectarea acestui sistem a fost realizată în urma unui studiu îndelungat rezultat în urma necesității unei metode în sprijinul realizării aplicațiilor software care însoțesc producția din industria componentelor de autovehicule astfel încât acestea să îndeplinească cerințele normelor de calitate din acest domeniu și să asigure un demers normal al liniilor de producție, în scopul de a evita pierderi economice la nivelul organizațiilor.

Principalele avantaje și caracteristici proprii ale acestei metode sunt:

- este o metodă de asigurare a calității aplicațiilor software dedicată producției;
- ajută dezvoltorii în implementarea aplicațiilor software conform cu cerințele din industria

- componentelor de autovehicule;
- asigură o evaluare reală conformă cu cerințele acestei industrii;
- evită potențialele costuri și pierderi economice ale companiei, ce pot apărea datorită întreruperii producției din motive de software.

Metoda propusă este doar în faza de proiect, ea nefiind dovedită în totalitate din punctul de vedere al eficacității. Astfel studiile viitoare sprijină punerea în aplicare a metodei QSPS pentru un exemplu practic aplicat pe un caz real. Acest exemplu practic ne va oferi posibilitatea de a măsura totodată eficacitatea metodei QSPS.

Capitolul-5

Cercetări privind realizarea unui model experimental de sistem de calitate pentru aplicațiile software din industria componentelor de autovehicule

5.4. SISTEMUL QSMA – MODEL EXPERIMENTAL ȘI PRACTIC

Cu toate că sistemul PM@IT și regulile sistemului informațional sunt foarte detaliate și riguros aplicate, în mediul informațional de producție s-au identificat lipsuri precum și necesitatea unui sistem care să asigure calitatea dezvoltării aplicațiilor software pentru producție.

Cerințele și necesitățile principale se concretizează în interiorul departamentelor de dezvoltare software de producție, care dispun de un proces managerial de coordonare a întregului proiect, dar nu dispun de un model care să îndrume modul de dezvoltare, astfel încât calitatea produselor obținute să fie asigurată.

Cerințele care apar din interiorul departamentelor de dezvoltare sunt următoarele:

Strategice:

- verificarea sistemului protejat QSPS și adaptarea acestuia la cerințele și politicile companiei în concordanță cu PM@IT;
- utilizarea rezultatelor cercetării pentru implementarea unui sistem de asigurare a calității în dezvoltarea de software de producție;
- asigurarea și documentarea preluării și instalării a acestor aplicații software de către beneficiari finali;
- definirea concretă a serviciului de suport cu evaluarea resurselor necesare;
- creșterea satisfacției beneficiarilor finali;
- diminuarea și chiar eliminarea pierderilor din producție cauzate de acest tip de aplicații de software;
- suportul tehnic pentru implementarea practică a acestui sistem până la punerea în aplicare dar și după punerea lui în aplicare.

Tehnice:

- alinierea și încadrarea elementelor sistemului QSPS la etapele sistemului PM@IT;
- realizarea unor check-listuri, algoritmi de calcul de estimare a resurselor;
- grafice pentru definirea responsabilităților;
- template-uri pentru predări de software, definiții pentru eliberare de activități finale sau intermediare;
- template-uri pentru definirea organizației proiectului de dezvoltare;
- definirea procesului pentru dezvoltatori;
- procesul de dezvoltare – reguli, descriere;

- definirea mediului de programare pentru producție;
- verificarea îndeplinirii cerințelor funcționale;
- controlul documentelor și a datelor;
- lista riscurilor;
- lista cerințelor de securitate IT;
- cereri specifice serviciului IT – de tip cod, nefuncționale;
- cerințe și recomandări pentru verificarea de software;
- descrierea procesului de verificare și validare;
- criterii interne de acceptare și aprobare a aplicației software implementat;
- protocolul de acceptanță;
- manualul utilizatorului;
- raportul deviațiilor de la specificații;
- audit intern asupra funcționalității și procesului de obținere a aplicației software (*score list*);
- evaluarea acurateței și clasificarea maturității;
- utilitatea;
- studii de eficiență (rapoarte statistice: iGATE, studii de capabilitate;
- software CPK, studii de fiabilitate) (*checklist*);
- verificare și aprobarea aplicației software implementate de către client;
- procesul de manipulare a erorilor de software;
- studiul și procesul de îmbunătățire;
- scenariu de test;
- verificarea elementelor legale (licențe, legi locale);
- integrarea sistemului;
- procedura de instalare;
- criterii acceptanță și aprobare & procesul de aprobare și protocolul acestuia;
- recomandări și cerințe pentru monitorizarea funcțională;
- rezultatele în urma monitorizării și soluțiile propuse;
- procesul de validare și verificare software;
- calculul ipm (incidente software per 1 million execuții);
- procesul acțiunilor corective și preventive (grafic);
- procesul acțiunilor predictive (grafic);
- evaluări calitative de software;
- controlul și înregistrarea datelor și rezultatele evaluărilor;
- plan de control software;
- înregistrarea experiențelor;
- mediul de înregistrare a erorilor de software.

Astfel, am dezvoltat sistemul QSMA (Quality System for Manufacturing application) care este destinat pentru asigurarea calității aplicațiilor care însoțesc producția.

În cele ce urmează vom descrie conținutul sistemului practic QSMA.

5.4.2 MODUL1-Evaluări de timp și costuri (PM) – CA IT Gate ITG:30->40

- checklist pentru serviciul de dezvoltare – criterii necesare pentru demarare (specificația tehnică - instruire, specificarea și definirea serviciului de suport (SLA), caracteristici pentru determinarea și

- identificarea închiderii proiectului);
- evaluări de timp și de costuri (pentru fiecare departament în parte);
- organigrama membrilor proiectului de dezvoltare a aplicației software MES de producție;
- RASI Chart (cine e responsabil pentru ce);
- definirea criteriilor de eliberare și predare a produsului, definirea etapelor (ce trebuie făcut și când).

5.4.3 MODUL2-Cerințe Software (SW) – CAS IT Gate ITG 40->60

- procesul de dezvoltare – reguli, descriere;
- cereri specifice serviciului IT – de tip cod, nefuncționale:
 - îndrumător de cod (adaptabilitate la infrastructura / adaptabilitate la testare / stabilitate / portabilitate / coexistență / adaptabilitate la modificări / studiu de reutilizare;
 - îndrumător pentru interfața grafică (GUI);
 - îndrumător de performanță;
 - manual de testare;
- verificarea îndeplinirii cerințelor funcționale (în ce măsură și cât de deplin sunt acestea îndeplinite) – definirea perioadelor și a frecvențelor de verificare;
- verificarea utilizării experienței celor mai bune practici (*Best practices*);
- controlul documentelor și a datelor (CVS/Enterprise);
- lista riscurilor;
- lista cerințelor de securitate IT;
- definirea mediului de programare pentru producție (de exemplu configurația;
- companiei pentru implementarea modulelor camline);

5.4.4 MODUL3-Validare și Verificare (VaVe) – CA IT Gate 40->60

- cerințe și recomandări pentru verificarea de software:
 - reverificare/validare de design și dezvoltare
 - test de cod/Securitate și vulnerabilitate
 - clasificare și verificare a funcționalității aplicației software (test de performanță)
 - clasificarea și verificarea funcționalității de software în conformitate cu cerințele definite de departamentul de specialitate.
- descrierea procesului de verificare și validare: template pentru criteriile de verificare și înregistrarea rezultatelor.

5.4.5 MODUL4-Aprobări Interne (IR) – CAS IT Gate 60->70

- criterii interne de acceptare și aprobare a aplicației software implementate, înainte de livrare la client (software PPAP);
- protocolul de acceptanță (însoțit de semnăturile responsabililor) (RASI de la Elementul 2);
- manualul utilizatorului (instrucțiuni de conținut);
- raportul deviațiilor de la specificații (platformă de înregistrare: sharepoint...etc. pentru documentarea cerințelor și a soluțiilor și pentru testarea rezultatelor de după implementare);
- audit intern asupra funcționalității și procesului de obținere a aplicației software;

- evaluarea acurateții și clasificarea maturității (criterii de clasificare alfa,beta, release... version, definirea regulilor acestor criterii);
- utilitatea: - Toleranța la erori;
- gradul de înțelegere/gradul de ușurință în învățare;
- studiul de eficiență (rapoarte statistice: iGATE?, studii de capabilitate, software CPK, studii de fiabilitate).

5.4.6 MODUL5-Aprobările Clientului(CR) – CAS IT Gate 70->90

- verificare și aprobarea aplicației software implementate de către client.(test funcțional-Run @ Rate);
- procesul de manipulare a erorilor de software (în toate mediile sistemului QSMA :de test, de integrare și productiv) și
- mediul de înregistrare a erorilor de software și soluțiile propuse (cea mai bună experiență & Urmărirea evoluției prelucrării erorilor-Bug Tracking);
- scenariu de test bazat pe specificația cerințelor și pe implementarea de software (acțiune -> rezultat așteptat -> rezultat obținut);
- verificarea elementelor legale (licențe, legi locale);
- integrarea sistemului (procedura de instalare: sistemul de integrare -> sistemul productiv);
- criterii acceptanță și aprobare & procesul de aprobare și protocolul acestuia.

5.4.7 MODUL6-Monitorizări Funcționale (FM) – CAS IT Gate 90->100

- recomandări și cerințe pentru monitorizarea funcțională și măsurători calitative ale aplicației software pe parcursul rulării în producție;
- rezultatele în urma monitorizării și soluțiile propuse de implementare de scurtă și lungă durată;
- procesul de validare și verificare software – extensie;
- calculul ipm (incidente software per 1 million execuții);
- procesul acțiunilor corective și preventive;
- procesul acțiunilor predictive;
- evaluări calitative de software și studii de capabilitate;
- controlul și înregistrarea datelor și rezultatele evaluărilor; (proponerea unei structuri de date : sharepoint, iGate Reports);
- plan de control software și manipularea erorilor de software conform cu IT Incident Mng;
- înregistrarea experiențelor (sharepoint etc.);
- studiul și procesul de îmbunătățire a sistemului de calitate QSMA.

5.4.8 MODUL7-Satisfacția Clientului (CS) – CAS IT Gate 90->100

- cerințe și recomandări pentru manipularea plângerilor de la client;
- acordul de support tehnic;
- raportarea/monitorizarea și înregistrarea problemelor și procedura de finalizare a problemelor (sistem propus: remedy, HP Service Management);
- procesul managementului schimbării (template pentru Change Request, template pentru

incidente:8D... etc.);

- instruirii (instrucțiuni, procesul de instruire, conținut);
- procesul de comunicare cu clientul (metode, mediu de comunicare: e-mail, sharepoint, Portal ...);
- evaluări periodice de la client (rapoarte de satisfacție);
- statistici pentru: problem și soluții, timp de reacție, și eficiența soluției : propunere: diagrame PERT);

Pentru a reliefa cerințele companiei (reprezentate prin sistemul QSMA) ne vom folosi de tabelul 9 prin care se pot observa elementele ce trebuie implementate pentru această companie:

Element	QSMA (cerințe de implementat)
Elementul 1	<ul style="list-style-type: none"> • Checklist pentru serviciul de dezvoltare – criterii necesare pentru demarare (specificația tehnică - template, instruirii, specificarea și definirea serviciului de suport (SLA), caracteristici pentru determinarea și identificarea închiderii proiectului) • Algoritmi pentru Evaluări de timp și de costuri (pentru fiecare departament în parte) • Organigrama membrilor proiectului de dezvoltare a aplicației software MES de producție • RASI Chart (cine e responsabil pentru ce) • Definiția criteriilor de eliberare și predare a produsului, definirea etapelor (ce trebuie făcut și când)
Elementul 2	<ul style="list-style-type: none"> • Descrierea și regulile Procesului de dezvoltare • Cereri specifice serviciului IT – de tip cod, nefuncționale: • îndrumător de cod (adaptabilitate la infrastructura / adaptabilitate la testare / stabilitate / portabilitate / coexistență / • îndrumător pentru interfața grafică (GUI) • manual de testare • checklist pentru verificarea îndeplinirii cerințelor funcționale (în ce măsură și cât de deplin sunt acestea îndeplinite) – definirea perioadelor și a frecvențelor de verificare • inițierea de platforme pentru controlul documentelor și a datelor(CVS/Enterprise) • lista riscurilor (template) • Lista cerințelor de securitate IT (checklist) • Definirea mediului de programare pentru producție (de exemplu configurația companiei pentru implementarea modulelor camline)
Elementul 3	<ul style="list-style-type: none"> • Checklist pentru Cerințe și recomandări pentru verificarea de software • Reverificare/validare de design și dezvoltare (checklist) • Test de cod(instrucțiuni)/Securitate și vulnerabilitate • Clasificare și verificare a funcționalității aplicației software (test de performanță) (checklist) • Descrierea procesului de verificare și validare: template pentru criteriile de verificare și înregistrarea rezultatelor (instrucțiuni)

<p>Elementul 4</p>	<ul style="list-style-type: none"> • Criterii interne de acceptare și aprobare a aplicației software implementate înainte de livrare la client (check list) (software PPAP) • Protocolul de acceptanță (însoțit de semnăturile responsabililor) (template) (RASI de la Elementul 2) • Manualul utilizatorului (template, instrucțiuni de conținut) • Raportul deviațiilor de la specificații (document template/platformă de înregistrare: sharepoint...etc. pentru documentarea cerințelor și a soluțiilor și a testa rezultatele de după implementare) • Audit intern asupra funcționalității și procesului de obținere a aplicației software (score list) • Evaluarea acurateței și clasificarea maturității (criterii de clasificare alfa, beta, release... version, definirea regulilor acestor criterii) (instrucțiuni) • Utilitatea: - Toleranța la erori • gradul de înțelegere / gradul de ușurință în învățare (instrucțiuni) • studiul de eficiență (rapoarte statistice: iGATE?, studii de capabilitate, software CPK, studii de fiabilitate) (checklist – instrucțiuni, template-uri)
<p>Elementul 5</p>	<ul style="list-style-type: none"> • Verificare și aprobarea aplicației software implementate de către client.(test funcțional-Run @ Rate, instrucțiuni și conținut) • Procesul de manipulare a erorilor de software (descriere)(în toate mediile sistemului QSMA :de test, de integrare și productiv) și • Mediul de înregistrare a erorilor de software și soluțiile propuse (cea mai bună experiență & Urmărirea evoluției prelucrării erorilor-Bug Tracking) (definiție, checklist, chestionar) • Scenariu de test bazat pe specificația cerințelor și implementării de software (acțiune ->rezultat așteptat -> rezultat obținut)(template) • Verificarea elementelor legale(licențe, legi locale)(checklist) • Integrarea sistemului (procedura de instalare: sistemul de integrare -> sistemul productiv) • Criterii acceptanță și aprobare & procesul de aprobare și protocolul acestuia (checklist)
<p>Elementul 6</p>	<ul style="list-style-type: none"> • Recomandări și cerințe pentru monitorizarea funcțională și măsurători calitative ale aplicației software pe parcursul rulării în producție (instrucțiuni) • Rezultatele în urma monitorizării și soluțiile propuse de implementare de scurtă și lungă durată.(template) • Procesul de validare și verificare software - extensie (template) • Calculul ipm (incidente software per 1 million execuții)(algoritm și metode de calcul) • Procesul acțiunilor corective și preventive (grafic) • Procesul acțiunilor predictive (grafic) • Evaluări calitative de software și studii de capabilitate (check /score list)

	<ul style="list-style-type: none"> • Controlul și înregistrarea datelor și rezultatele evaluărilor (inițierea și realizarea unui astfel de mediu) (propunerea unei structuri de date : sharepoint, iGate Reports) • Plan de control software și manipularea erorilor de software conform cu IT Incident Mng. • Înregistrarea experiențelor (sharepoint etc.) • Studiul și procesul de îmbunătățire a sistemului de calitate QSMA • (modelări matematice)
Elementul 7	<ul style="list-style-type: none"> • Fără necesitate de implementare. Se va considera doar în chestionarul de evaluare a aplicației software.

Tablel 9 Cerințele sistemului QSMA

Adițional acestor liste, chestionare, metode, definiții, procese, reguli și algoritmi sistemul QSMA conține și un chestionar de evaluare a calității aplicațiilor software de producție (Anexa 7).

5.6. CONCLUZII

Sistemul QSMA conduce la asigurarea calității aplicațiilor software de producție și astfel la evitarea costurilor ce pot apărea în producție din cauza aplicațiilor software. Alte avantaje directe aduse prin utilizarea acestui sistem, dovedite de altfel și în practică sunt:

- asigurarea calității aplicațiilor software MES pentru producția constructoare de componente de autovehicule;
- ajută la dezvoltarea de aplicații software MES robuste;
- previne și evită întreruperea producției din motive de software;
- diminuează potențiale costuri ce pot apărea din cauza erorilor de software și astfel a întreruperii liniei de producție;
- oferă o transparență ridicată în clasificarea și identificarea problemelor de producție (probleme de proces versus probleme de software MES);
- predare și acceptarea a soluțiilor MES de către client cu verificarea îndeplinirii cerințelor;
- crește satisfacția clientului;

Sistemul experimental poate fi în continuare îmbunătățit și adaptat la nevoile actuale ce apar pe parcurs din situații practice. Sistemul de punctaj al sistemului QSMA constituie o bază solidă de date de intrare în modelările matematice, astfel sistemul putând fi monitorizat și supus procesului de îmbunătățire continuă.

Capitolul-6

Contribuții proprii și arii potențiale de cercetare

Sistemul derivat QSMA, obținut pe baza sistemului conglomerat QSPS, este rezultatul experimental al studiului în compania Continental și reprezintă contribuția proprie a autorului pentru un studiu de caz. În sistemul global QSPS, se poate remarca proprietatea de adaptare a acestuia la mediul de aplicare, astfel facilitându-se dezvoltarea de alți derivați, în funcție de domeniul de aplicabilitate și strategia companiei.

După cum se poate observa **Elementul 1** aduce cu sine, pe lângă chestionare manageriale și un algoritm de calcul pentru evaluarea duratei de implementare, bazat pe filozofia PERT, dar adaptat pentru aplicațiile software de producție.

Elementul 2 prezintă un proces de dezvoltare pentru programatorii de aplicații software de producție și alte instrucțiuni din domeniul programării pentru acest tip de programe, observate și optimizate de-a lungul experienței autorului cu aceste aplicații software. Totodată aici se fac recomandări pentru interfața grafică a programelor, mediu de interacțiune directă cu utilizatorii echipamentelor industriale, cu scopul de a identifica în timp optim problemele ce apar în producție.

Pe de altă parte **Elementul 3** vine cu recomandări și reguli reprezentative de verificare a aplicațiilor software, care să ușureze și să diminueze eforturile de verificare, dar care să crească aria asupra căreia se exercită verificarea.

Elementul 4 descrie reguli și repere în urma cărora un program software se poate declara funcțional și poate fi deliberat spre punerea în funcțiune.

Elementele 5,6 și 7 sunt orientate spre clientul acestor programe și descriu reguli de acceptanță, de monitorizare și de îmbunătățire continuă.

Dintre principalele **contribuții practice** ale cercetării reprezentate prin chestionare, instrucțiuni și checklisturi pot fi amintite:

- Checklist pentru condiții de demarare (Anexa 1)
- Calcul estimativ de resurse (Anexa 2)
- Graficul RASI (Anexa 3)
- Definirea etapelor de calitate (*Q-Gate*) pentru proiect (Anexa 4)
- Checklist de maturitate - tehnologie software (Anexa 5)
- Chestionar responsabilități pe timpul ciclului de viață (Anexa 6)
- Chestionar de evaluare (Anexa 7)

Dintre principalele **contribuții teoretice** originale ale cercetării pot fi amintite:

- identificarea necesităților unei metode ale calității pentru aplicațiile software de producție;
- definirea și sistematizarea terminologiei privind termenii cheie și conceptele utilizate în teză;
- analiza bibliografică și acumularea sistematică a datelor și informațiilor;
- modelarea conceptului de model de calitate pentru aplicațiile software de producție;
- sinteză cu privire la modele utilizate în dezvoltarea aplicațiilor software;
- sinteză cu privire la modele utilizate în dezvoltarea de produse de autovehicule
- descrierea originală a tehnicilor și instrumentelor calității utilizate în proiectarea și dezvoltarea

- aplicațiilor software;
- descrierea originală a sistemului MES și a arhitecturii acestui sistem;
- modelarea și dezvoltarea unui model de calitate pentru aplicațiile MES;
- conceperea de checklisturi și templaturi pentru aplicarea acestui model;
- aplicarea teoretică și experimentativă a modelului QSPS obținut;
- crearea derivatului QSMA pentru aplicarea practică;
- analiză sistematică a beneficiilor acestui sistem, criteriu de obținere și aprobare a bugetului pentru implementarea practică;
- integrarea modelului în infrastructura informațională a unei cele mai mari companii europene din industria auto
- modelarea matematică modelului integrat conceput;
- analiza impactului preconizat al modelului integrat conceput;
- simularea matematică a modelului integrat;
- analiza rezultatelor obținute în urma simulării modelului;
- validarea modelului integrat.

Aceste contribuții au fost publicate și în nenumărate articole de reviste (Anexa 10) după cum urmează: 2 lucrări indexate ISI, o lucrare indexată BDI, 2 lucrări indexate Ulrichs Web, 5 lucrări în curs de publicare.

De asemenea printre contribuțiile proprii se numără și participarea la proiecte cu relevanță pentru acest studiu precum: proiectul DQ200 al companiei Continental, proiectul în curs de implementare QUALiPSO propunere de proiect PN-II-PT-PCCA-2011-3.2-1093, și proiectul QSMA la compania Continental

Principala direcție de viitor a acestui studiu este de a aplica și adapta metoda la cât mai multe organizații industriale, și deci de a crea cât mai mulți derivați ai metodei dezvoltate QSPS.

Rezultatele obținute în urma aplicărilor practice a acestor sisteme derivate din metoda QSPS, vor facilita îmbunătățirea continuă a metodei, fapt observabil cu ajutorul simulărilor matematice.

Aceste aspecte de îmbunătățire, precum și combinarea cu noi strategii de management și calitate sau alte norme care nu au fost cuprinse în acest studiu, constituie ariile viitoare de cercetare a acestei lucrări. Crearea de alți derivați ai sistemului QSPS (de ex.QUALiPSO propunere de proiect PN-II-PT-PCCA-2011-3.2-1093), reprezintă totodată o arie nouă și foarte largă de cercetare pentru aplicarea sistemului în diferite ramuri ale industriei componentelor de autovehicule.

De asemenea setul de instrucțiuni, chestionare, procese și instrumente ajutătoare pot fi reunite sub forma unei platforme software (de tip cockpit, aplicații software care comunică între ele), cu scopul de a optimiza aplicarea calității în industria cu acest profil. Realizarea unui astfel de aplicație software al calității poate reprezenta o direcție viitoare a studiului.

Rezultatele acestei lucrări vor putea facilita cercetarea acestui domeniu într-un mod și mai detaliat, ele reprezentând un punct de plecare pentru concepte noi. Totodată acest studiu poate reprezenta demararea de inițiative noi de implementare a calității în domeniul industrial, aducând instituțiile universitare mai aproape de mediul practic, industrial.

" Calitatea nu este niciodată un accident, ci este întotdeauna rezultatul unui efort inteligent."

– John Ruskin

REFERINȚE BIBLIOGRAFICE:

- [1.] Ahlemann, F., Teuteberg, F. & Vogelsang, K. (2008) *Project management standards – Diffusion and application in Germany and Switzerland*, International Journal of Project Management 27 (2009) 292–303, Retrieved from www.sciencedirect.com
- [2.] Ahlemann, F., (2009) Towards a conceptual reference model for project management information systems, International Journal of Project Management 27 (2009) 19–30 din www.sciencedirect.com.
- [3.] Alba, E., Chicano, J.F., 2007, Software project management with Gas, Information Sciences 177 (2007) 2380–2401, din www.elsevier.com/locate/ins
- [4.] Albrecht, D.: Management von Stakeholderbeziehungen mit dem EFQM-Modell. Centre for Sustainability Management, 2008 CSM Lüneburg
- [5.] Amza, C., Reggio, G., A Notation for Component-Based Design of Java Applications, book chapter, Lecture Notes in Computer Science, 2003, Volume 2604, Scientific Engineering for Distributed Java Applications, Pages 155-164
- [6.] ASRO (2006). Sisteme de management al calității. Principii fundamentale și vocabular, SR EN ISO 9000:2006
- [7.] Baccarini, D., 1999. The logical framework method for defining project success. Project Management Journal 30, 25–32.
- [8.] Bamford, R., C., Deibler WJ: Comparing, contrasting ISO 9001 and the SEI capability maturity model. IEEE Computer, October 1993
- [9.] Banciu, D. Sisteme automatizate de informare și documentare, București, Editura Tehnică, 1997.
- [10.] BANCIU, Doina; COARDOS, Dora. Pilot System for Information – Documentation Based on Web Platform. In: Conferința internațională “Information Literacy”. Sibiu, 20 – 22 aprilie 2010.
- [11.] Banermann, P.L., 2008, Risk and risk management in software projects: A reassessment, The Journal of Systems and Software 81 (2008) 2118–2133, din www.elsevier.com/locate/jss
- [12.] Basili, V.R., (1985). Quantitative evaluation of software methodology. Tech. Rep. TR-1519, University of Maryland.
- [13.] Basili, V.R., Green, S., (1994). Software process evolution at the sel., IEEE Software 11 (4), 58–66.

- [14.] Beck,K., Andres,C.: Extreme Programming Explained. Embrace Change. 2nd Edition, Addison Wesley, Dezember 2004, ISBN 0-321-27865-8
- [15.] Boehm, B.W., 1981. Software Engineering Economics. Prentice Hall, Englewood Cliffs, NJ.
- [16.] Booz,A. & Hamilton Earned Value Management Tutorial Module 2: Work Breakdown Structure, Office of Project Assessment, doe.gov. Accessed 01. Dec 2008
- [17.] Brennan, K.,2009. A Guide to the Business Analysis Body of Knowledge (Babok Guide). International Institute of Business Analysis. pp. 29. ISBN 0981129218
- [18.] Budgen, D., Software Design, second ed., Addison-Wesley, Harlow, England, 2003
- [19.] Bullinger, H-J., Warschat, J., Schumacher, O., Slama, A.& Peter Ohlhausen, P., (2005), Ontology-Based Project Management for Acceleration of Innovation Projects, M. Hemmje et al. (Eds.): E.J. Neuhold Festschrift, LNCS 3379, pp. 280 . 288, 2005. © Springer-Verlag Berlin Heidelberg 2005, din www.springerlink.com
- [20.] Burghardt, M.: Projektmanagement, Ediția 6, München, Publicis Corporate Publishing, 2002, ISBN 3-89578-199-1
- [21.] CamLine, 2004, LineWorks EMaC, Equipment Main Data Configuration, Users Guide, Revision 1.1 03 / 2004
- [22.] CamLine, 2011, <http://www.camline.com/products/>
- [23.] Chatfield, C., 2007 A short course in project management, din <http://office.microsoft.com/en-us/project/HA102354821033.aspx>
- [24.] Chen, C-B., Lin,C-T., Wang,C-H., Chang,C-W.,2005, Model for measuring quality of software in DVRS using the gap concept and fuzzy schemes with GA, Information and Software Technology 48 (2006) 187–203, <http://www.sciencedirect.com>
- [25.] Chen,C-T.,Huang,S-F., 2007, Applying fuzzy method for measuring criticality in project network, Information Sciences 177 (2007) 2448–2458, din www.elsevier.com/locate/ins
- [26.] Cioffi,D.,F., 2006, Designing project management: A scientific notation and an improved formalism for earned value calculations, International Journal of Project Management 24 (2006) 136–144, din www.sciencedirect.com
- [27.] Cleland, D.,I., Ireland, Lewis R. (eds.) (2008), Project manager’s handbook: applying best practices across global industries. New York: McGraw-Hill

- [28.] Cleland,D.,I., Gareis,R., (2006). Global project management handbook. McGraw-Hill Professional, 2006. ISBN 0071460454. p.1-4
- [29.] CMMI Guidebook Acquirer Team (2007). Understanding and Leveraging a Supplier's CMMI Efforts: A Guidebook for Acquirers (pdf). CMU/SEI-2007-TR-Software Engineering Institute. <http://www.sei.cmu.edu/library/abstracts/reports/07tr004.cfm>.
- [30.] CMMI/SEI, CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD/SS, V1.1, Staged), March 2002 edition of CMMI from SEI, chapter 2 page 11, Pittsburgh, PA 15213-3890, CMU/SEI-2002-TR-012
- [31.] Coallier, F., How ISO 9001 fits into the software world. IEEE Software, January 1994.
- [32.] Covrig,M,2006, Dezvoltarea de produs prin proiect, Editura MATRIX ROM Bucuresti 2006 ISBN: 973-755-123-8
- [33.] Crosby, Ph.B.(1979). Quality Is Free. McGraw-Hill, New York
- [34.] Dan Florin NIȚOI, Gheorghe AMZA – Sandwik Coromant Software for the Milling and Turning Tools Selection , pag. 323 – 326 - Annalis of MTeM for 2007 PROCEEDINGS of the 8 th International Conference Modern Technologies in Manufacturing Cluj Napoca, 4 th 5 th October 2007. ISBN 973-9087-83-3.
- [35.] Deming, W.E.: Out of the Crisis. Massachusetts Institute of Technology, Cambridge 1982, ISBN 0-911379-01-0, P. 88
- [36.] DGQ- Deutsche Gesellschaft für Qualität, (Societatea Germană pentru Calitate)
- [37.] Draft.R.,L.,2010, Management, 9th Edition, ISBN-10:0324595840, ISBN-13:9780324595840, publicată de Casebound © 2010
- [38.] Ebert,C., 2007, The impacts of software product management, The Journal of Systems and Software 80 (2007) 850–861, din www.elsevier.com/locate/jss
- [39.] ECSS-M-30A,1996: Raumfahrt-Projektmanagement - Projektphaseneinteilung und -planung, 19. April 1996, ESC-TR-2002-012
- [40.] Extreme Programming (lecture paper), USFCA.edu, webpage: USFCA-edu-601-lecture <http://www.cs.usfca.edu/~parrt/course/601/lectures/xp.html>
- [41.] Fabian, B., Seda,G., & Maritt.H., —A Comparison of Security Requirements Engineering Methods. Requirements Engineering 15, 1 (March 2010): 7-40.

- [42.] Floyd, C., Software development as reality construction, in: C. Floyd, H. Zullighoven, R. Budde, R. Keil-Slawik (Eds.), *Software Development and Reality Construction*, Springer, Berlin, 1992, pp. 86–100.
- [43.] Foley, K.J. Hermel, Ph. (ed.)(2008) *The Theories and Practices of Organizational Excellence: New Perspectives*. SAI Global Ltd, Sydney, Australia
- [44.] Geraldi,J.G., Turner,J.R. , Maylor, H., Anders So derholm, Hobday,M., Brady,T., *Innovation in project management: Voices of researchers*, *International Journal of Project Management* 26 (2008) 586–589, preluat din www.sciencedirect.com
- [45.] Gheorghe AMZA, Constantin PETRICEANU (University POLITEHNICA of Bucharest), *The implementation of specialized software for the ultrasonic non-destructive control of composite materials*, SISOM 2007 and HOMAGIAL SESSION OF THE COMMISSION OF ACOUSTICS, BUCHAREST, 29-31 May.
- [46.] Graham,P.,1995, *Mary Parker Follett - Prophet of Management. A Celebration of Writings from the 1920s*. Bear Books, Washington D.C. 1995: S. 13 f., ISBN 1-58798-213-7.
- [47.] Gulick, L. H. (1936). *Notes on the Theory of Organization*. L. Gulick & L. Urwick (Eds.), *Papers on the Science of Administration* (pp. 3–35). New York: Institute of Public Administration.
- [48.] Harrison, F.,L., Lock,D., (2004). *Advanced project management: a structured approach*. Gower Publishing, Ltd., 2004. ISBN 0566078228. p.34
- [49.] Herrmann, J., W., *History of Decision-Making Tools for Production Scheduling*, *Proceedings of the 2005 Multidisciplinary Conference on Scheduling: Theory and Applications*, New York, July 18-21, 2005
- [50.] Hoglund, G., McGraw, G.,2004, *Exploiting Software: How to Break Code*, Addison-Wesley, 2004.
- [51.] Howard, M., Lipner,S.,2003, *Inside the Windows Security Push*, *IEEE Security & Privacy*, vol. 1, no.1, 2003, pp. 57–61.
- [52.] Humphrey, Watts (1989). *Managing the Software Process*. Addison Wesley. ISBN 0201180952.
- [53.] Hyväri, I., (2006), *Project management effectiveness in project-oriented business organizations*, *International Journal of Project Management* 24 (2006) 216–225, retrieved from www.sciencedirect.com
- [54.] IEEE P1363-2000: *Standard Specifications for Public-Key Cryptography*

- [55.] IEEE STD 1058-1998 Standard for Software Project Management Plans
- [56.] IEEE STD 1233-1998 Guide for Developing System Requirements Specifications
- [57.] IEEE STD 1471-2000 Recommended Practice for Architectural Description
- [58.] IEEE STD 829-2008 Standard for Software and System Test Documentation
- [59.] IEEE STD 830-1998 Recommended Practice for Software Requirements Specifications
- [60.] IEEE Std. 1016-1998 IEEE Recommended Practice for Software Design Descriptions
- [61.] IEEE Std. 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology
- [62.] IEEE Std. 730-1998 IEEE Standard for Software Quality Assurance Plans
- [63.] IEEE Std. 828-1998 IEEE Standard for Software Configuration Management Plans
- [64.] INTERNATIONAL STANDARD ISO 9000, (2005) Quality management systems - Fundamentals and vocabulary (ISO 9000:2005), EN ISO 9000:2005
- [65.] INTERNATIONAL STANDARD ISO 9000-3, (1997) Guidelines for the application of ISO 9001, to the development, supply, installation and maintenance of computer software, EN ISO 9000-3:1997
- [66.] INTERNATIONAL STANDARD ISO 9000-3, (1997) Guidelines for the application of ISO 9001, to the development, supply, installation and maintenance of computer software, EN ISO 9000-3:1997
- [67.] INTERNATIONAL STANDARD ISO 9000-3, Guidelines for the application of ISO 9001, to the development, supply, installation and maintenance of computer software, EN ISO 9000-3:1997
- [68.] INTERNATIONAL STANDARD ISO/IEC 12207, Systems and software engineering — Software life cycle processes, ISBN 0-7381-5664-7
- [69.] INTERNATIONAL STANDARD ISO/IEC 14598, Software engineering – Product evaluation, Part 1 - General Overview(1999), Part 2 - Planning and Management(2000), Part 3 – Process for Developers(2000), Part 4 – Process for acquirers(1999), Part 5 - Process for evaluators (1998) and Part 6 – Documentation of evaluations modules(2001)

- [70.] INTERNATIONAL STANDARD ISO/IEC 14598, Software engineering – Product evaluation, Part 1 - General Overview(1999), Part 2 - Planning and Management(2000), Part 3 – Process for Developers(2000), Part 4 – Process for acquirers(1999),Part 5 - Process for evaluators (1998) and Part 6 – Documentation of evaluations modules(2001)
- [71.] INTERNATIONAL STANDARD ISO/IEC 17799 Second edition 2005-06-15, Information technology — Security techniques — Code of practice for information security management, Reference number ISO/IEC 17799:2005(E) © ISO/IEC 2005
- [72.] INTERNATIONAL STANDARD ISO/IEC 25000, (2005) Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE
- [73.] INTERNATIONAL STANDARD ISO/IEC 9126, Software engineering — Product quality —Part 1:Quality model(2001), Part 2 (External metrics, 2003); Part 3 (Internal metrics, 2003); and Part 4 (Quality in use metrics, 2004), Reference number ISO/IEC 9126:2001(E)
- [74.] 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
- [75.] Isik,Z., Arditi, D., Dikmen,I., Birgonul, M.,T., 2009, Impact of corporate strengths/weaknesses on projectmanagement competencies, International Journal of Project Management 27 (2009) 629–637,
din www.elsevier.com/locate/ijproman
- [76.] ISO 10007:2003 Quality management systems – Guidelines for configuration management
- [77.] ISO 15489-1:2001 Information and documentation – Records management – Part 1: General
- [78.] ISO 19011:2002 Guidelines for quality and /or environmental management systems auditing
- [79.] ISO 19011:2002, Guidelines for quality and/or environmental management systems auditing
- [80.] ISO 9000:2000, Quality management systems — Fundamentals and vocabulary
- [81.] ISO 9001:2000, Quality management systems — Requirements

- [82.] ISO 9004:2000, Quality management systems — Guidelines for performance improvements
- [83.] ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability
- [84.] ISO/IEC 11770-1:1996 Information technology – Security techniques – Key management – Part 1: Framework
- [85.] ISO/IEC 12207:1995 Information technology – Software life cycle processes
- [86.] ISO/IEC 12207:1995, Information technology — Software life cycle processes
- [87.] ISO/IEC 12207:1995/Amd.1:2002, Information technology — Software life cycle processes
- [88.] ISO/IEC 13335-1:2004, Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management
- [89.] ISO/IEC 13335-1:2004, IT security techniques — Management of information and communications technology security — Part 1: Concepts and models for information and communications technology security management
- [90.] ISO/IEC 13407:1999, Human-centered design processes for interactive systems
- [91.] ISO/IEC 13888-1: 1997, Information technology – Security techniques – Non-repudiation – Part 1: General
- [92.] ISO/IEC 14516:2002 Information technology – Security techniques – Guidelines for the use and management of Trusted Third Party services
- [93.] ISO/IEC 14598-1:1999, Information technology — Software product evaluation — Part 1: General overview
- [94.] ISO/IEC 14598-2:2000, Software engineering — Product evaluation — Part 2: Planning and management
- [95.] ISO/IEC 14598-3:2000, Software engineering — Product evaluation — Part 3: Process for developers
- [96.] ISO/IEC 14598-4:1999, Software engineering — Product evaluation — Part 4: Process for acquirers
- [97.] ISO/IEC 14598-5:1998, Information technology — Software product evaluation — Part 5: Process for evaluators

- [98.] ISO/IEC 14598-6:2001, Software engineering — Product evaluation — Part 6: Documentation of evaluation modules
- [99.] ISO/IEC 14888-1:1998 Information technology – Security techniques – Digital signatures with appendix – Part 1: General
- [100.] ISO/IEC 15288:2002, System Engineering — System Life Cycle Processes
- [101.] ISO/IEC 15408-1:1999 Information technology – Security techniques – Evaluation Criteria for IT security – Part 1: Introduction and general model
- [102.] ISO/IEC 15504:2003, Information technology — Software process assessment
- [103.] ISO/IEC 15504-2:2003, Information technology — Process assessment — Part 2: Performing an assessment
- [104.] ISO/IEC 15504-3:2004, Information technology — Process assessment — Part 3: Guidance on performing an assessment
- [105.] ISO/IEC 15504-4:2004, Information technology — Process assessment — Part 4: Guidance on use for process improvement and process capability determination
- [106.] ISO/IEC 15939:2002, Software engineering — Software measurement process
- [107.] ISO/IEC 17799:2000, Information technology — Code of practice for information security management
- [108.] ISO/IEC 17799:2005(E)
- [109.] ISO/IEC 18028-4 Information technology – Security techniques – IT Network security – Part 4: Securing remote access
- [110.] ISO/IEC 2005 – All rights reserved 107 Management
- [111.] ISO/IEC 25001(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Planning and management
- [112.] ISO/IEC 25010(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Quality model
- [113.] ISO/IEC 25021(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Quality measure elements
- [114.] ISO/IEC 25022(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Measurement of internal quality

- [115.] ISO/IEC 25023(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Measurement of external quality
- [116.] ISO/IEC 25024(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Measurement of quality in use
- [117.] ISO/IEC 25030(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements
- [118.] ISO/IEC 25040(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Evaluation reference model and guide
- [119.] ISO/IEC 25041(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Evaluation modules
- [120.] ISO/IEC 25042(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Evaluation process for developers
- [121.] ISO/IEC 25043(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Evaluation process for acquirers
- [122.] ISO/IEC 25044(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Evaluation process for evaluators
- [123.] ISO/IEC 25062(new) Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry format for usability test reports
- [124.] ISO/IEC 9126-1:2001, Software engineering — Product quality — Part 1: Quality model
- [125.] ISO/IEC 9796-2:2002 Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms
- [126.] ISO/IEC 9796-3:2000 Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms
- [127.] ISO/IEC FDIS 15504-5, Information technology — Process assessment — Part 5: An exemplar process assessment model
- [128.] ISO/IEC Guide 2:1996, Standardization and related activities – General vocabulary
- [129.] ISO/IEC Guide 73:2002, Risk management – Vocabulary – Guidelines for use in standards

- [130.] ISO/IEC TR 13335-3:1998, Information technology – Guidelines for the Management of IT Security – Part 3: Techniques for the management of IT Security
- [131.] ISO/IEC TR 18044 Information technology – Security techniques – Information security incident
- [132.] ISO/IEC TR 9126-2:2003, Software engineering — Product quality — Part 2: External metrics
- [133.] ISO/IEC TR 9126-3:2003, Software engineering — Product quality — Part 3: Internal metrics
- [134.] ISO/IEC TR 9126-4:2004, Software engineering — Product quality — Part 4: Quality in use metrics
- [135.] J.-W. Li, Modeling a quality assurance information system for product development projects with the UML approach, International Journal of Computer Integrated Manufacturing, Vol.20, Nr.4, 2007.
- [136.] Jones, C., 1995. Patterns of large software systems: failure and success. IEEE Computer 28, 86–87.
- [137.] Juran, J.M., Gryna jr., F.M. (1973).Calitatea produselor.Tratat de planificare, proiectare, realizare și control. Trad.din l.engl. Editura Tehnică, București
- [138.] Kai-Yuan Cai, Bey-Bey Yin, Software execution processes as an evolving complex network, Information Science, 2008.
- [139.] Kerr,J.,M., Hunter,R.,1993, Inside RAD: How to Build a Fully-Functional System in 90 Days or Less, McGraw-Hill, ISBN-10: 0070342237
- [140.] 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, Volume 18, Issue 3, pag. 247 – 254, 2009.
- [141.] Kifor, C.V., Oprean, C., Lobonț, L., Cîndea, C., Zerbese., M., Decision Support System (DSS) for higher education institutions. Part 1. System implementation at LBUS. International Journal of technology and engineering education, Association of Taiwan Engineering Education Management, pag. 13 – 18, ISSN 1816 – 9325, 2009.
- [142.] Kifor, C.V., Tudor, N., Oprean, C., A PRACTICAL APPROACH to a Quality System for Production Software for managing technological changes, Proceeding of the Conference: Management of technological Change, Alexandroupolis, Grecia, 2011.
- [143.] Kneuper, R., Sollmann,F.,1995: Normen zum Qualitätsmanagement bei der Softwareentwicklung Informatik Spektrum, Band 18 (1995), S. 314-323

- [144.] Kotler, P., Keller, K. L., Bliemel, F., (2007) Marketing Management - Strategien für wertschaffendes Handeln, 12. Auflage, Person Education Deutschland GmbH, München ISBN 978-3-8273-7229-1 Seite 12 Kapitel 1.2.2 Waren und andere Austauschobjekte
- [145.] Kousholt, B., 2007. Project Management – Theory and practice., Nyt Teknisk Forlag. ISBN 8757126038. p.59
- [146.] Kouskouras, K., G., Georgiou, A., K., 2007, A discrete event simulation model in the case of managing a software project, European Journal of Operational Research 181 (2007) 374–389, din www.elsevier.com/locate/ejor
- [147.] Krainz, E., 2009, Sozialkompetenz im Projektmanagement: Eine unterschätzte Dimension, Plädoyer für einen gruppendynamischen Paradigmenwechsel, 40:235–256. DOI: 10.1007/s11612-009-0088-5
- [148.] Kühner, G., Torsten, B., Employing industrial standards in software engineering for W7X, Fusion Engineering and Design, 2009.
- [149.] Lee, M., Miller, J., 2004, Multi-Project Management in Software Engineering Using Simulation Modelling, Software Quality Journal, 12, 59–82, preluat din www.springerlink.com
- [150.] Linberg, K.R., 1999. Software developer perceptions about software project failure: a case study. Journal of Systems and Software 49, 177–192.
- [151.] Liu, J., Chen, H., H., Jiang, J., J., Klein, G., 2010, Task completion competency and project management performance: The influence of control and user contribution, International Journal of Project Management 28 (2010) 220–227, preluat din www.sciencedirect.com
- [152.] Mabin, V., Balderstone, S. 1998, A Review of Goldratt's Theory of Constraints - Lessons from the International Literature, Operational Research Society of New Zealand 33rd Annual Conference, Auckland, pp. 205–214
- [153.] Manuela Dittmann, 2011, Project Management @ Automotive IT, Handbook 2.0
- [154.] Mari'a N. Moreno Garcí'a , Isabel Ramos Roma'n , Francisco J. Garcia Penalvo , Miguel Toro Bonilla , 2008, An association rule mining method for estimating the impact of project management policies on software quality, development time and effort, Expert Systems with Applications 34 (2008) 522–529, din www.sciencedirect.com
- [155.] Mărăscu-Klein Vl., Toma, V. - Maintenance – an improvement factor of the production systems' performances – Review of Management and Economical Engineering, Cluj-Napoca, nr.1/2007, pag. 107-118 (ISSN 1583-624X)

- [156.] Mărăscu-Klein Vladimir - Utilization of relational data base for material selection. International Conference "Mechanics and Machine Elements", Sofia, 4-6 nov. 2005, pag. 289-294 (ISBN 954-323-181-8).
- [157.] McBride, T., 2008, The mechanisms of project management of software development, The Journal of Systems and Software 81 (2008) 2386–2395, preluat din www.elsevier.com/locate/jss
- [158.] McGraw, G., From the Ground Up: The DIMACS Software Security Workshop, IEEE Security & Privacy, vol. 1, no. 2, 2003, pp. 59–66.
- [159.] McGraw, G., 2004, Software security, PUBLISHED BY THE IEEE COMPUTER SOCIETY 1540-7993/04 © IEEE SECURITY & PRIVACY
- [160.] McGraw, G., Chess, B., Miguez, S.. Building Security In Maturity Model BSIMM v2.0. <http://www.bsimm2.com/> (Accessed July 2010)
- [161.] McGraw, G., Chess, B., Miguez, S.. Building Security In Maturity Model BSIMM v2.0. <http://www.bsimm2.com/> (Accessed July 2010)
- [162.] Mead, N. R., Allen, J.H., Building Assured Systems Framework, September 2010 TECHNICAL REPORT CMU/SEI-2010-TR-025 ESC-TR-2010-025 CERT® Program, <http://www.sei.cmu.edu>
- [163.] Menon, A. et al. (1999). Antecedents and Consequences of Marketing Strategy Making. Journal of Marketing (American Marketing Association) 63(2) (2): 18–40. doi:10.2307/1251943. <http://jstor.org/stable/1251943>
- [164.] Method123 Ltd, 2003, Project management Guidebook, ISBN 0-473-10445-8 retrieved from www.sciencedirect.com.
- [165.] Microsoft, 2010, Microsoft Security Development Lifecycle Version 5.0. http://download.microsoft.com/download/F/2/0/F205C451-C59C-4DC7-8377-9535D0A208EC/Microsoft%20SDL_Version%205.0.docx (Updated March 31, 2010).
- [166.] Microsoft, 2010, Microsoft Security Development Lifecycle Version 5.0. http://download.microsoft.com/download/F/2/0/F205C451-C59C-4DC7-8377-9535D0A208EC/Microsoft%20SDL_Version%205.0.docx (Updated March 31, 2010).
- [167.] Mincă, D.G., Popescu, A., Țereanu, C., - 1. Introducere în management, Conferința națională pentru management medical modern, 2006, cap 7
- [168.] Mordechai Ben-Menachem, 2008, Towards management of software as assets: A literature review with additional sources, Information and Software Technology 50 (2008) 241–258, preluat din www.sciencedirect.com

- [169.] N.Tudor, C.Kifor, C.Oprean, (2009), QUALITY SYSTEM FOR PRODUCTION SOFTWARE – QSPS, THE 6TH INTERNATIONAL CONFERENCE ON ADVANCED MANUFACTURING TECHNOLOGIES ICAMaT 2009, October 8-10, 2009, Cluj-Napoca, Romania
- [170.] N.Tudor, C.Kifor, C.Oprean, (2010), Using QSPS in developing and realization of a production line in automotive industry, International Conference on Computers, Communications & Control, ICCCC 2010, May 12-16, Baile Felix (Spa), Oradea, Romania
- [171.] Negulescu, S.C., Kifor C.V., Oprean, C. Ant Colony, Solving Multiple Constraints Problem: Vehicle Route Allocation. International Journal of Communication, Computer and Control, Vol. 3, nr. 4, Decembrie, ISSN 1841 – 9836, pag. 367 – 374, 2008.
- [172.] NeuroShell 2 Help, Ward Systems Group, Inc. <http://www.wardsystems.com>.
- [173.] Newell, M; Grashina, M (2003). The Project Management Question and Answer Book. American Management Association. p. 98
- [174.] Nokes, Sebastian, The Definitive Guide to Project Management.. 2nd Ed.n. London (Financial Times / Prentice Hall): 2007. ISBN 978 0 273 71097 4 Keider, S.P., 1974. Why projects fail. Datamation 20, 53–55.
- [175.] OECD Guidelines for Cryptography Policy, 1997
- [176.] OECD Guidelines for the Security of Information Systems and Networks: ‘Towards a Culture of Security’, 2002
- [177.] OGC (Office of Government Commerce) (2005). Managing Successful Projects with PRINCE2. TSO (The Stationery Office). ISBN 9780113309467
- [178.] Open Web Application Security Project (OWASP). Software Assurance Maturity Model (SAMM) v1.0. http://www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model (2009).
- [179.] Oprean, C., Kifor, C.V, O.Suciu, Managementul integrat al calității, Editura Universității “Lucian Blaga” din Sibiu, 2005, Bibliogr. ISBN 973 -739 -034 -2
- [180.] Oprean, C., Kifor,C.V, Managementul calității, Editura Universității “Lucian Blaga” din Sibiu, 2002, Bibliogr. ISBN 973 -651 -310 -6
- [181.] Oskarsson, Ö., Glass,R., Prentice Hall 1995, An ISO 9000 Approach to Building Quality Software

- [182.] Oskarsson, Ö., Glass,R., Prentice Hall 1995, An ISO 9000 Approach to Building Quality Software
- [183.] Oskarsson, Ö., Glass,R., Prentice Hall 1995, An ISO 9000 Approach to Building Quality Software
- [184.] Ourdev, I., Xie, H., AbouRizk,S., 2008, An Intelligent Agent Approach to Adaptive Project Management, TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 20/67 pp121-125 Volume 13, Number S1, October 2008, din www.springerlink.com
- [185.] Paulk, M. C., Weber, C. V., Curtis, B., Chrissis, M. B. (1995). The Capability Maturity Model: Guidelines for Improving the Software Process. Boston: Addison Wesley. ISBN 0201546647
- [186.] Paulk, M.,C., Bamford RC, Deibler WJ: Basis of contrast between ISO 9001 and SEI Capability Maturity Model challenged. IEEE Computer, February 1994.
- [187.] Peterson, K., Wohlin,C.,(2010), Software process improvement through the Lean Measurement (SPI-LEAM) method, The Journal of Systems and Software
- [188.] Pinto, J.K., Slevin, D.P., 1988. Project success: definitions and measurementtechniques. Project Management Journal 19, 67–72.
- [189.] Pires, J.N., (2005), Semi-autonomous manufacturing systems:The role of the human–machine interface software and of the manufacturingtracking softwar, Mechatronics 15 (2005) 1191–1205
- [190.] Procaccino,J.D., Verner,J.M., 2006, Software project managers and project success: An exploratory study, The Journal of Systems and Software 79 (2006) 1541–1551, din www.sciencedirect.com
- [191.] Prof. Dr. Ahlemann, F., Prof. Dr. Riempp,G.,(2007), RefMod: A Conceptual Reference Model for Project Management Information Systems, DOI: 10.1365/s11576-008-0028-y, din www.springerlink.com
- [192.] Raffo, D. M., Kellner., M.I. , Field study results using the process tradeoff analysis method, Proceedings of the 1996 Software Engineering Process Group Conference, Held in Atlantic City, New Jersey, May 20–23, 1996.
- [193.] Raffo,D.M., 2005, Software project management using PROMPT: A hybrid metrics, modeling and utility framework, Information and Software Technology 47 (2005) 1009–1017, din www.elsevier.com/locate/infso
- [194.] Sauer,C., Reich,B.H., 2009, Rethinking IT project management: Evidence of a new mindset and its implications, International Journal of Project Management 27 (2009) 182–193, din www.elsevier.com/locate/ijproman

- [195.] Schwaber,K., Beedle,M.: Agile Software Development with Scrum. Prentice Hall, 18. Februar 2002, ISBN 978-0130676344
- [196.] Sharp,H., Baddoo,N., Beecham,S., Hall,T., Robinson,H., 2009, Models of motivation in software engineering, *Information and Software Technology* 51 (2009) 219–233, din www.elsevier.com/locate/infsof
- [197.] Simon Collyer,S., Warren, S.M.J., 2009, Project management approaches for dynamic environments, *International Journal of Project Management* 27 (2009) 355–364, preluat din www.sciencedirect.com
- [198.] Software Engineer Standard Comitee of the IEEE Computer Society, IEEE Standard for Software Quality Assurance Plan, IEEE Std 730TM- 2002
- [199.] Software Engineering Institute. (2006) Standard CMMI Appraisal Method for Process Improvement (SCAMPISM) A, Version 1.2: Method Definition Document.CMU/SEI-2006-HB-002. <http://www.sei.cmu.edu/library/abstracts/reports/06hb002.cfm>.
- [200.] Stacie,P., Vijay,V.,2008, Facilitating experience reuse among software project managers, *Information Sciences* 178 (2008) 1783–1802, din www.elsevier.com/locate/ins
- [201.] Staehle,W.,H.: Management. 8. Aufl. München: Vahlen, 1999, S. 71. - ISBN 3800623447
- [202.] Stamelos,I., 2010, Software project management anti-patterns, *The Journal of Systems and Software* 83 (2010) 52–59, din www.elsevier.com/locate/jss
- [203.] Steen, O., 2007, Practical knowledge and its importance for software product quality, *Information and Software Technology* 49 (2007) 625–636, <http://www.sciencedirect.com>
- [204.] Stefan Rosskopf, 2008, MES Electronics Plants Current Implementation (camLine)
- [205.] Stefan Rosskopf, 2010, CC-MES Development & Interfaces
- [206.] Stephen Leybourne, S., Sadler-Smith,E., 2006, The role of intuition and improvisation in project management, *International Journal of Project Management* 24 (2006) 483–492, preluat din www.sciencedirect.com
- [207.] Stevens,M. (2002). Project Management Pathways. Association for Project Management. APM Publishing Limited, 2002 ISBN 190349401X p.xxii

- [208.] Stumpf, M.: Erfolgskontrolle der Integrierten Kommunikation. Messung des Entwicklungsstandes integrierter Kommunikationsarbeit in Unternehmen. Wiesbaden, 2005
- [209.] Suikki,R., Tromstedt, R., Haapasalo, H., 2006, Project management competence development framework in turbulent business environment, Technovation 26 (2006) 723–738, din www.elsevier.com/locate/technovation
- [210.] Sutherland,J., 2005, Future of Scrum: Parallel Pipelining of Sprints in Complex Projects. Brighton, MA 2005
- [211.] Taylor,F.W.,1911, The principles of scientific management, publicată inițial de Harper&Broders 1991, republicată de Elibron Classics series. © 2005 Adamant Media Corporation. ISBN 1-4021-7299-0 (paperback) ISBN 1-4212-8459-6 (hardcover)
- [212.] The SPICE User Group 2005-2010 Automotive SPICE® Process Reference Model, Automotive SIG 2010-05-10
- [213.] Thomas Boos, 2011, camLine MES System Overview
- [214.] Thwin, M.M.T., Quah, T-S.,2004, Application of neural networks for software quality prediction using object-oriented metrics, The Journal of Systems and Software 76 (2005) 147–156, <http://www.sciencedirect.com>
- [215.] Tudor, N., Kifor C. V., Oprean, C., Quality system for production software QSPS. Academic Journal of Manufacturing Engineering, editura Politehnica, ISSN 1583-7904, pag. 135 – 140, 2009.
- [216.] Tudor, N., Kifor C.V., Oprean, C. Quality Management in Software Implementation and Design for Automotive Industry. Acta Universitatis CIBINIENSIS, 2011 .
- [217.] Tudor, N., Kifor C.V., Oprean, C., Using QSPS in developing and realization of a production line in automotive industry. Proceedings International Conference on Computers, Communications & Control, ICCCC 2010.
- [218.] Tudor, N., Kifor,C., Oprean,C. (2009), QUALITY SYSTEM FOR PRODUCTION SOFTWARE – QSPS, THE 6TH INTERNATIONAL CONFERENCE ON ADVANCED MANUFACTURING TECHNOLOGIES ICAMaT 2009, October 8-10, 2009, Cluj-Napoca, Romania
- [219.] Tudor,N.,Dumitrașcu,D., 2008, ADVANCE ESTIMATE EXPENSES FOR PROJECT EXECUTION TIME, ANNALS of the ORADEA UNIVERSITY, Fascicle of Management and Technological Engineering, Volume VII (XVII), 2008, din,http://imtuoradea.ro/auo.fmte/files-2008/MIE_files/TUDOR%20NICOLAE%201.pdf

- [220.] Tudor,N.,Dumitrașcu,D., 2008, THE BENEFITS OF PROJECT STRUCTURING IN SUB-PROJECTS AND WORKPACKAGES, Fascicle of Management and Technological Engineering, Volume VII (XVII), 2008 din http://imtuoradea.ro/auo.fmte/files-2008/MIE_files/TUDOR%20NICOLAE%201.pdf
- [221.] Van Truong,L., Soo-Yong, K., Tuan-Anh, H., 2008, Improving project management performance of large contractors using benchmarking approach, *International Journal of Project Management* 26 (2008) 758–769, din www.sciencedirect.com
- [222.] Viega, J., McGraw,G., *Building Secure Software*, Addison-Wesley,2001; www.buildingseuresoftware.com.
- [223.] Virine, L., Trumper M., *Project Decisions: The Art and Science* (2007). *Management Concepts*. Vienna, VA, ISBN 978-1567262179
- [224.] Wallace,L., Keil,M., Rai,A., (2004), Understanding software project risk: a cluster analysis, *Information & Management* 42 (2004) 115–125, preluat din www.sciencedirect.com
- [225.] Webster F.M, Knutson J., What is project management. *Project management concepts and methodologies*. In: Dinsmore PC, Cabanis- Brewin J, editors. *The AMA handbook of project management*. New York, NY: AMACOM; 2006. p. 1–10.
- [226.] Whitty,S.,Harvey Maylor, H.,2009, And then came Complex Project Management (revised), *International Journal of Project Management* 27 (2009) 304–310, din www.sciencedirect.com
- [227.] William, D., *A Guide to the Project Management Body of Knowledge*, copyright page, edition 2 ISBN 1-880410-12-5 (free .pdf edition), and edition 3 2004 ISBN 1-930699-45-8, and edition 4 2008 ISBN 1933890517
- [228.] Winter, M., Smith, C., Morris,P., Cicmil,S., (2006), Directions for future research in project management: The main findings of a UK government-funded research network, *International Journal of Project Management* 24 (2006) 638–649, din www.elsevier.com/locate/ijproman
- [229.] Zika-Viktorsson, A., Ingelga, A., (2006), Reflecting activities in product developing teams: conditions for improved project management processes, *Res Eng Design* (2006) 17:103–111, DOI: 10.1007/s00163-006-0019-1
- [230.] Zorțelan,T., Burduș,E., Căprărescu,G., *Managementul organizației*, Editura Economică, București, 1999

Anexe:

ANEXA 2 – CALCUL ESTIMATIV DE TIMP ȘI COSTURI (RESURSE)

Resource Evaluation										
Project name				Time evaluation						
Version date				15.10.2007						
Work Package (WP)	Number		Time [h]					WP-Type	Theor. time [d]	
	Empl./MA	Par.	opt.	Probably	pe.ss.	expected	Total			Variance
SubProject 1										
Workpackage 1										
Workpackage 2										
Workpackage 3										
Subproject 2										
Workpackage 1										
Workpackage 2										
			0	0	0	0,0	0,0	0,0	0,0	

Anexa 2(continuare)

Evaluation/Auswertung						
Table 1/Tabelle 1						
Time [h]/Aufwand [h]					Varianz	Standard-abweichung
opt.	wahr	pess.	erwarte	gesamt		
0	0	0	0	0	0	0
Table 2/Tabelle 2						
Berechnung Eintrittswahrscheinliche			Stunden	Tagen		
Aufwand			1.134	142		
Eintrittswahrscheinlichkeit			70%	70%		
Table 3/Tabelle 3						
Berechnung Aufwand			Stunden	Tagen		
Gewünschte Eintrittswahrscheinlichkeit			70%			
Fertigstellungsaufwand			#ZAHL!	#ZAHL!		
Table 4/Tabelle 4						
Eintrittswahrscheinlichkeiten der Aufwandsbereiche		Stunden		Days		
		von	bis	von	bis	
Aufwand bei 70% Eintrittswahrscheinliche		0	0	0	0	
Aufwand bei 95% Eintrittswahrscheinliche		0	0	0	0	
Aufwand bei 99% Eintrittswahrscheinliche		0	0	0	0	

ANEXA 6 – CHESTIONAR RESPONSABILITĂȚI PE TIMPUL CICLULUI DE VIAȚĂ

Lifecycle	Process category	Domain of activity	Responsible (Name)	Implemented (%)
	Primary	Acquisition		
		Supply		
		Development		
		Operations		
		Maintenance		
	Supporting	Documentation		
		Change management		
		Quality assurance		
		Verification		
		Validation		
		Joint review		
		Audit		
	Organizational	Problem Resolution		
		Management		
		Infrastructure		
Improvement				
	Training			

ANEXA 7 – CHESTIONAR DE EVALUARE

Software evaluation according to QSMA

Software Product

Product port folio / Beneficiary Plant or department
Audit subject / software package name
MES Software
Main customers

Date of audit:
04.05.2011
Date of report:
04.05.2011

Overall Degree of Conformity:	57	[%]
Automated Calculated Rating:	C	
Deviate Rating by Evaluator(see remarks):		

Evaluator's Name
Department:

Distribution list:		
Name	Department	Contact

Software Development	
Company Name	Continental Automotive
Developer Name	Nicolae Tudor
Department	MES Department
Certification	

Participants			
Continental Automotive MES Department		CA Customer	
Name	Organisation / Function	Name	Organisation / Function
Management:			
Production:			
Project management:			
Quality:			
Software developer:			

Anexa 7(continuare)

Software Project Name:	MES Software
Date:	04.05.2011

Overall Degree of Conformity:	57
Automated Calculated Rating:	C
Deviate Rating by Evaluator(see remarks):	

Individual assessment of the questions

Each questions is assessed on the basis of the individual demands and their subsequent compliance as well as the assurance of this process.

Points	Assessment
10	Full compliance with requirements
8,9	Predominant compliance with requirements minor nonconformities
6,7	Partial compliance with requirements more severe nonconformities
4,5	Unsatisfactory compliance with requirements major nonconformities
0,1,2,3	No compliance with requirements

Grade	Overall degree of conformity level (%)	Grading of the process
A*	90-100	Full compliance
AB*	80 to less than 90	Predominant compliance
B*	60 to less than 80	Partial compliance
C	less than 60	No compliance

- Final Audit result can be downgrading in case of:
- one ore more elements rating under 75%
 - one ore more requirements scored with 0
- * The downgrading reason must be specified in the remark Field of the Frontpage

So an A result could be downgraded from A to AB, AB -> B or B -> C

Anexa 7(continuare)

Software Project Name:	MES Software
Date:	04.05.2011

Requirements

1 Element 1 - Project management - Planning, time and cost evaluation (PM)		Rating
1.1.	Checklist Service development – criteria to start (Functional Spec. template, Training, Service level Agreement, Development closure characteristic etc.)	<input type="text"/>
1.2.	Time and cost evaluation (on Component level: reporting, Equipment software...etc.)	<input type="text"/>
1.3.	Project organization / Software development	<input type="text"/>
1.4.	RASI Chart (who is responsible for what)	<input type="text"/>
1.5.	Handover /Q-gates definition (what to do when)	<input type="text"/>
2 Element 2 - Software - Requirements (SW)		
2.1.	Process for developer – processing in application (presentation)	<input type="text"/>
2.2.	IT requirements – non functional:	<input type="text"/>
2.3.	IT Requirements compare cross check to Business requirements	<input type="text"/>
2.4.	Best practices (check list)	<input type="text"/>
2.5.	Document & source code control & process (version control)	<input type="text"/>
2.6.	IT list of risks (template)	<input type="text"/>
2.7.	IT security requirements (checklist)	<input type="text"/>
2.8.	MA Dev. Environment checklist.for example Conti configuration model for camline	<input type="text"/>
3 Element 3 -		
3.1.	Requirements and recommendations for verification of the software(validation checklist)	<input type="text"/>
3.2.	•Design and development review/ verification/validation (checklist)	<input type="text"/>
3.3.	•Code Review/Security & Vulnerability Testing	<input type="text"/>
3.4.	•Classification and verification of the software functionality in accordance to the defined IT requirements (performance test)(checklist)	<input type="text"/>
3.5.	•Classification and verification of the software functionality in accordance to the defined business requirements.(requirements test)	<input type="text"/>
3.6.	Process for verification and validation: template for verification criteria and result	<input type="text"/>
4 Element 4 -		
4.1.	Internal criteria of acceptance and approval of the implemented software kit by the developers before delivering it to the client. (check list) (software PPAP)	<input type="text"/>
4.2.	Acceptance protocol (with approval signatures) (template)(RASI from Step2)	<input type="text"/>
4.3.	User manual (form and template, content instructions)	<input type="text"/>
4.4.	Deviations from specifications (document template/record medium: sharepoint...etc. to document requests/solutions and testing results after implementation)	<input type="text"/>
4.5.	Internal functional & process audit for software (score list)	<input type="text"/>
4.6.	Accuracy/maturity evaluation and classification (notification criteria for alfa, beta, release... version, definition of software maturity classification rules) (instructions)	<input type="text"/>
4.7.	Usability	<input type="text"/>
4.8.	Efficiency compliance (statistical reports: iGATE?, capability studies, software CPK, lifetime study) (checklist)	<input type="text"/>

Anexa 7(continuare):

Software Project Name:	MES Software
Date:	04.05.2011

Requirements

5 Element 5 -		Rating
5.1.	Verification and approval of the implemented software kit by the customer.(functionality test-Run @ Rate)	<input type="text"/>
5.2.	Process of BUGs handling (environment definition test-integration- production)	<input type="text"/>
5.3.	Recording of BUGs situations and solution (lesson learned for developer & Bug Tracking)	<input type="text"/>
5.4.	Test scenario based on requirements-specification & implementation (action->expected result-> run result)	<input type="text"/>
5.5.	Statutory and regulatory conformity check(licenses, local law regulation)	<input type="text"/>
5.6.	System integration (rollout procedure: integration system -> production system)	<input type="text"/>
5.7.	Acceptance and release criteria & Release process and protocol	<input type="text"/>
6 Element 6 -		Rating
6.1.	Recommendations and requirements for functional monitoring and measurement of the software product during the production (instructions)	<input type="text"/>
6.2.	Monitor results and long term improvements.(template)	<input type="text"/>
6.3.	Process of software validation and verification extension (template proposal)	<input type="text"/>
6.4.	Calculation of ipm (number of software incidents per 1 million executing)	<input type="text"/>
6.5.	Process of corrective and preventive action * (chart)	<input type="text"/>
6.6.	Process of predictive actions ** (chart)	<input type="text"/>
6.7.	Software quality assessment and capability (check /score list)	<input type="text"/>
6.8.	Control and record of quality data collection and results (proposal for data structure : sharepoint, iGate Reports)	<input type="text"/>
6.9.	Software control plan and handle of bugs in alignment with IT Incident Mng.	<input type="text"/>
6.10.	Lesson learned, collection medium (sharepoint etc.)	<input type="text"/>
6.11.	QSMA dynamic adjustment and improvement. Potential Improvement measurement.	<input type="text"/>
7 Element 7 - Customer service / Satisfaction		Rating
7.1.	Requirements and recommendations for handling customer complains (instructions)	<input type="text"/>
7.2.	Technical support agreement (contract template)	<input type="text"/>
7.3.	Problem reporting/monitoring/recording and closing procedure (by customer acceptance of the solution) (system proposal: remedy, HP Service Management)	<input type="text"/>
7.4.	Process of change management and problem/incidents resolution (templates for Change Request, templates for incidents:8D...etc)	<input type="text"/>
7.5.	Training (process, methods and rules) -> template with content recommendation	<input type="text"/>
7.6.	Customer communication process (methods, communication medium: e-mail, sharepoint, Portal ...)	<input type="text"/>
7.7.	Periodical customer feedback evaluation /customer satisfaction reports(template)	<input type="text"/>
7.8.	Statistics: over problems and solutions, reaction times, and solution efficiency :statistic proposal like PERT diagram)	<input type="text"/>

Anexa 7(continuare):

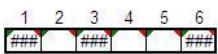
Process audit according to QSMA

Developed by:	Nicolae Tudor	Continental Automotive / MES Department	Date of evaluation:	04.05.2011
Evaluator's Name:			Software Project Name:	MES Software

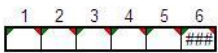
1 Element 1 - Project management - Time and cost evaluation (PM)

1 Project management  PM %


2 Element 2 - Software requirements (SW)

2 Software requirements  SW %

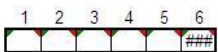
3 Element 3 - Validation and verification of the implemented software package (VaVe)

3 Validation and verification  VaVe %


4 Element 4 - Internal software release (IR)

4 Internal release  IR %

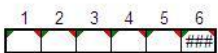
5 Element 5 - Customer software release (CR)

5 Customer release  CR %

6 Element 6 - Functional monitoring during production running (FM)

6 Functional monitoring  FM %

7 Element 7 - Customer satisfaction (CS)

7 Customer satisfaction  CS %

Overall degree of conformity

$$\frac{\text{PM} + \text{SW} + \text{VaVe} + \text{IR} + \text{CR} + \text{FM} + \text{SM}}{\text{No. of evaluated elements}} = \boxed{57}$$

Anexa 7(continuare):

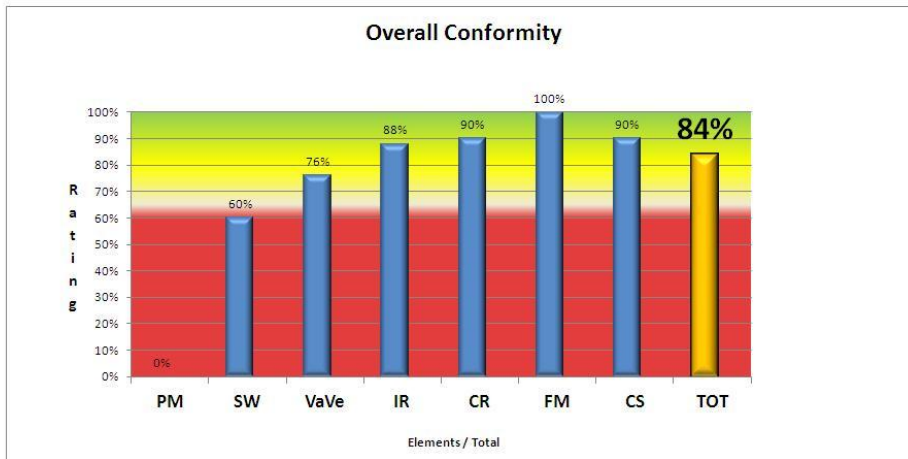
Process audit according to QSMA

Developed by:	Nicolae Tudor	Date of evaluation:	04.05.2011
Evaluator's Name:		Software Project Name:	MES Software

Overall conformity

Evaluated Processes

P roject M anagement	Element 1	PM	
S oftware requirements	Element 2	SW	60%
V alidation and V erification	Element 3	VaVe	76%
I nternal release	Element 4	IR	88%
C ustomer release	Element 5	CR	90%
F unctional m onitoring and improvement	Element 6	FM	100%
C ustomer s atisfaction	Element 7	CS	90%
Overall Degree of Conformity	TOTAL	TOT	84%



ANEXA 8 – CONFIRMARE APLICARE PRACTICĂ



Continental Automotive IT
Manufacturing Applications / Development & Interfaces

Nuremberg, 07. February, 2012

We hereby confirm that Mister Nicolae Tudor, post-graduate engineer, has performed research activities in the period 2008-2012, within the doctoral training, with the aim of developing a **Quality Management System for Software Applications in Automotive – QSPS**. The research activities were supervised by Prof. Claudiu Vasile Kifor, PhD. Eng, the scientific coordinator, from “Lucian Blaga” University of Sibiu and by Stefan Roßkopf, Eng. Dipl., from Continental Automotive, Head of Automotive IT Manufacturing Applications / Development & Interfaces

The technical and the economical departments from Continental Automotive have analyzed, in the period 2010-2011, the QSPS model proposed and have confirmed its feasibility and 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 Continental Automotive: Core System, Reporting Solutions, Integration (Client) Solutions, Communication Middleware and Integration Frameworks.

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 indicators such as:

- 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 indicators could change during the integration, if needed.

In the next stage (aprox. the second semester of 2012) the Quality System for Manufacturing Applications, developed on the QSPS structure, will be integrated in MES - Manufacturing Execution System, in the 92 plants of Continental Automotive. 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:

Anexa 8(continuare):



Continental Automotive IT
Manufacturing Applications / Development & Interfaces

Step 1:

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

Step 2:

1. Process for developer – processing in application (presentation)
2. IT requirements – non functional:
 - Coding guideline (Adaptability / test adaptability / Stability / portability/ coexistence /changeability / re-use study)
 - GUI guideline
 - Performance guide line
 - Test Guideline
3. IT Requirements compare cross check to Business requirements (cover ability check) – Functionality compliance check frequency and timing
4. Best practices (check list) -> plugin teal frames.net
5. Document & source code control & process (version control-"Subversion")
6. IT list of risks (template), Technical risk (FMEA)
7. IT security requirements (checklist), see IT at conti.
8. MA Dev. Environment checklist-for example Conti configuration model for camline

Step 3:

1. 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)
2. Process for verification and validation: template for verification criteria and result

Anexa 8(continuare):



Continental Automotive IT
Manufacturing Applications / Development & Interfaces

Step 4:

1. Internal criteria of acceptance and approval of the implemented software kit by the developers before delivering it to the client. (check list) (software PPAP)
2. Acceptance protocol (with approval signatures) (template)(RASI from Step2)
3. User manual (form and template, content instructions)
4. Deviations from specifications (document template/record medium: sharepoint...etc. to document requests/solutions and testing results after implementation)
5. Internal functional & process audit for software (score list)
6. Accuracy/maturity evaluation and classification (notification criteria for alfa, beta, release...version, definition of software maturity classification rules) (instructions)
7. Usability: - Fault tolerance
understandability /
learn ability analysis (instructions)
8. Efficiency compliance
(statistical reports: iGATE?, capability studies, software CPK, lifetime study) (checklist)

Step 5:

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

Step 6:

1. Recommendations and requirements for functional monitoring and measurement of the software product during the production (instructions)
2. Monitor results and long term improvements (template)
3. Process of software validation and verification extension (template proposal)
4. Calculation of ipm (number of software incidents per 1 million executing)
5. Process of corrective and *preventive* action * (chart)
6. Process of *predictive* actions ** (chart)
7. Software quality assessment and capability (check /score list)
8. Control and record of quality data collection and results
(proposal for data structure : sharepoint, iGate Reports)

Anexa 8(continuare):



Continental Automotive IT
Manufacturing Applications / Development & Interfaces

9. Software control plan and handle of bugs in alignment with IT Incident Mng.
10. Lesson learned, collection medium (sharepoint etc.)
11. QSMA dynamic adjustment and improvement. Potential Improvement measurement.
(Mathematical simulation of known MES software: use software like Statistica)

Step 7:

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

15.2.12 

Stefan Roßkopf, Eng. Dipl.,
Continental Automotive,
Head of Automotive IT Manufacturing Applications / Development & Interfaces

ANEXA 9 – CURRICULUM VITAE



Europass Curriculum Vitae



Personal information

First name(s) / Surname(s) **Nicolae TUDOR**
Address(es) Str.Podului Nr. 30, 557260 Selimbar-Sibiu, Romania
Telephone(s) +49(0)911 9526 3273 Mobile: +49(0)151 126 73493
Fax(es) +49(0)911 9526 13203
E-mail nicolae.tudor@continental-corporation.com
Nationality Romanian
Date of birth 13.March.1975

Occupational field **Project Manager Systems**

Work experience

Dates	
01.2011 - :	Project Manager Systems, Supplier Management, Continental Germany
12.2009- 01.2001:	Supplier Quality Manager (SQM-Lead Auditor VDA 6.3), Continental Germany
01.2007-12.2009:	Team Leader Test and Traceability, Department for Production, Continental, Sibiu- Romania
01.2002-12.2006:	CAD Data exchange and conversion – Methods Concept and Data exchange software Development, BMW Munich

Occupation or position held Dipl. Ing. Computer science / Project Manager Systems
Main activities and responsibilities Supplier & Process Management / SPM
Name and address of employer
Type of business or sector Purchasing Automotive

Education and training

Dates	
10.2009 – 10.2011	PhD
	Thesis Title: "CONTRIBUȚII PRIVIND REALIZAREA UNUI SISTEM DE MANAGEMENT AL CALITĂȚII PENTRU APLICAȚIILE SOFTWARE DIN INDUSTRIA COMPONENTELOR PENTRU AUTOVEHICULE"
Name and type of organisation providing education and training	UNIVERSITY „LUCIAN BLAGA” of SIBIU
Level in national or international classification	ISED 6

Anexa 9 (continuare):

Dates 11.2009 – 12.2010
 Title of qualification awarded SQM –Supplier Quality Management - Continental Lead Auditor (VDA6.3 -ISO/TS 16949)
 Principal subjects/occupational skills covered Quality Automotive Industry
 Name and type of organisation providing education and training Continental Temic GmbH
 Level in national or international classification ISCED 3

Dates 03.2000-09.2002
 Title of qualification awarded
 Principal subjects/occupational skills covered Mathematics
 Name and type of organisation providing education and training University of Regensburg, Germany
 Level in national or international classification ISED 5A

Dates 10.1994 – 06.1999
 Title of qualification awarded Dipl. Ing
 Principal subjects/occupational skills covered Computer systems science
 Name and type of organisation providing education and training UNIVERSITY „LUCIAN BLAGA” of SIBIU
 Level in national or international classification ISED 5A

Mother tongue(s) **Romanian**

Other language(s)

Self-assessment
 European level (*)

German

English

Polish

Understanding				Speaking				Writing	
Listening		Reading		Spoken interaction		Spoken production			
C2	proficient	C2	proficient	C2	proficient	C2	proficient	C1	proficient
C1	independent	B2	independent	B2	independent	B2	independent	B2	independent
B1	Basic user	A1	Basic user	A2	Basic user	A2	Basic user		


(*) [Common European Framework of Reference for Languages](#)

Social skills and competences

I have worked with different kind of professionals in different areas like software development, software support and customizing, product engineering and coordination of production, supplier management, quality and processes.

Anexa 9 (continuare):

Organisational skills and competences	<p>More than 6 Years leading experience of small teams of 10-20 employees. Experience in teambuilding and team structure organization. Project leading experience for over 8 Years. I work in the Concept Phase and integration of the production lines into MES in plant Nürnberg, was responsible for product line transfer to Sibiu and integration in MES. I build up the MES/Test Team in Sibiu, coordinate this Team and been the interface to the MES core team and process Team in Nürnberg. As part of my promotion, currently I develop the quality system QSMA (Quality system for manufacturing application) for MES together with MES core team. I work on the borders between young people, policies and different locations of the company, but also between different departments in the area of activity, bringing them together for getting the desired results. I am experienced at working in a Worldwide dimension, have very good intercultural skills, obtained from the communication and work dimension in a multicultural company distributed worldwide from Europe to Asia, Oceania or America and south America. 2008 I organize a football Team and manage organizational issues and playing strategies as coach.</p>
Computer skills and competences	<p>Programming language: Pascal, C, C++, Java, Assembler, Sybase SQL, Oracle SQL, Access, HTML, Korn-, c/t-shell, awk, sed, Python, Perl, C-API, CA-API</p> <p>Tools: MsOffice, Internet browsers, CAD (CATIA V4, V5, Proe), remedy, ProeToolkit (Programming), LabWindows CVI</p> <p>Operating Systems: Windows (OS), DOS, Linux, UNIX</p>
Other skills and competences	<p>Management: Tools and strategies, Team building and team task management, Quality management, Software Project Management</p> <p>Quality: Lead Auditor VDA 6.3</p>
Driving licence	Cl.B
Additional information	<p>References: Dr. Willibald Neumayer, CA/PDM Dataexchange BMW, willi.neumayer@bmw.de Eynollah Rahideh, Vice president Operations, Continental, eynollah.rahideh@continental-corporation.com</p> <p>Personnel Interests: Playing piano, join classical music concerts. Enjoy all sports particularly football and running. Love to travel and experience different cultures.</p>

NICOLAE TUDOR


ANEXA 10 – LISTA PUBLICAȚIILOR ȘTIINȚIFICE

<ul style="list-style-type: none"> • <u>Lucrări științifice publicate în reviste cotate ISI Web of science:</u>
<ul style="list-style-type: none"> – Tudor, N., Kifor C.V., Oprean, C., <i>Using QSPS in developing and realization of a production line in automotive industry</i>. Proceedings International Conference on Computers, Communications & Control, ICCCC 2010,
<ul style="list-style-type: none"> • <u>Lucrări științifice publicate în proceedings ale conferintelor internaționale indexate ISI Web of Knowledge:</u>
<ul style="list-style-type: none"> – Kifor, C.V., Tudor, N., Oprean, C., <i>A PRACTICAL APPROACH to a Quality System for Production Software for managing technological changes</i>, Proceeding of the Conference: Management of technological Change, Alexandroupolis, Grecia, pag. 29 – 32, ISBN 978 – 960 99486 – 2 – 3, 2011.
<ul style="list-style-type: none"> • <u>Lucrări publicate în jurnale internaționale indexate BDI:</u>
<ul style="list-style-type: none"> – Tudor, N., Kifor C. V., Oprean, C., <i>Quality system for production software QSPS</i>. Academic Journal of Manufacturing Engineering, editura Politehnica, ISSN 1583-7904, pag. 135 – 140, 2009.
<ul style="list-style-type: none"> • <u>Lucrări publicate în jurnale internaționale indexate Ulrichs Web:</u>
<ul style="list-style-type: none"> – Tudor,N.,Dumitrașcu,D., 2008, <i>ADVANCE ESTIMATE EXPENSES FOR PROJECT EXECUTION TIME</i>, ANNALS of the ORADEA UNIVERSITY, Fascicle of Management and Technological Engineering, Volume VII (XVII), 2008 – Tudor,N.,Dumitrașcu,D., 2008, <i>THE BENEFITS OF PROJECT STRUCTURING IN SUB-PROJECTS AND WORKPACKAGES</i>, Fascicle of Management and Technological Engineering, Volume VII (XVII), 2008
<ul style="list-style-type: none"> • <u>Alte lucrări: (în curs de publicare):</u>
<ul style="list-style-type: none"> – Tudor, N., Kifor C. V., <i>A practical approach system to QSPS: templates, procedures and techniques for using in practice</i> – Tudor, N., Kifor C. V., <i>ISO/TS 16949 aplicable for production software?:Analytical study of the norm ISO/TS 16949 for using it for production software</i> – Kifor, C.V., Tudor, N., <i>Quality requirements for production software</i> – Tudor, N., Kifor C.V., Oprean, C. <i>Quality Management in Software Implementation and Design for Automotive Industry</i>. Acta Universitatis CIBINIENSIS, 2011 – C.V. KIFOR¹, N.TUDOR², LAL MOHAN BARAL, <i>QSPS: An innovative approach to improve the efficiency and effectiveness of the quality systems for production software.</i>
<ul style="list-style-type: none"> • Proiecte
<ul style="list-style-type: none"> – Proiectul QSMA, Compania Continental – QUALiPSO propunere de proiect PN-II-PT-PCCA-2011-3.2-1093 – Proiectul DQ200 al companiei Continental