Supporting Software Services' Trustworthiness in Collaborative Networks

Maiara Heil Cancian, Ricardo J. Rabelo, Christiane Gresse von Wangenheim

Supporting Software Services' Trustworthiness in Collaborative Networks

Maiara Heil Cancian[1], Ricardo J. Rabelo[1], Christiane Gresse von Wangenheim[2]

[1] Department of Automation and Systems - Federal University of Santa Catarina – Brazil
[2] Department of Computer Science - Federal University of Santa Catarina – Brazil
{maiara, rabelo}@das.ufsc.br; gresse@inf.ufsc.br

Abstract. Collaborative Networks realization fundamentally relies on the need of *collaboration* among involved partners. From the supporting ICT infrastructures point of view, an increasing movement has been observed in direction of the development of applications based on the Service Oriented Architecture paradigm as a way to maximize reuse, to minimize integration efforts and to augment processes' adaptability. The underlying motivation of this paper is the possibility of CN members to enlarge their collaboration sharing software services. This means allowing that available services can be accessed by any member so enlarging and reinforcing collaboration while development and hosting costs are decreased. This paper presents an approach, in the form of a reference guide and based on Software-as-a-Service (SaaS) model, that helps both CN clients to be more confident when accessing services from CN members, and to CN members to have a support on how their services should be properly developed and made available to CN clients.

Keywords: Software-as-a-Service; trustworthiness; Quality reference guide.

1 Introduction

The adoption of the Collaborative Networks (CN) paradigm by organizations imposes enormous pressure on the companies´ competitive matrix, affecting their market positioning in terms of general quality, diversity and innovation of processes and products, prices, delivery dates, and level of relationship with suppliers and customers.

CN realization relies, however, on three essential pre-conditions. The first one is that it requires *collaboration* among involved partners. The second one is the existence of *trust*, considering that partners shall trust to one another (at variable levels). The third pre-condition is the *digital transactions*, i.e. most of the activities carried out within a CN should be made via computer networks [1]. The ultimate goal is to enable networked organizations to agilely define and set up relations with other organizations as well as to be adaptive according to the business environment conditions and current organizations' autonomy levels [2]. This requires, therefore, very high effectiveness and trustworthiness in businesses and collaborations.

From the ICT point of view, a number of approaches have been proposed to cope with those requirements in terms of supporting ICT infrastructures devoted to CNs [1]. The most relevant aspect is the shift from traditional monolithic software packages and

licenses to Service Oriented Architecture (SOA) and Utility paradigms [3]. SOA can be generally defined as an architectural paradigm for components of a system and interactions between them, where its functions are viewed as software modules called *services* [4]. *Utility* associates the idea that a service should be somehow permanently available, discoverable and usable anytime, anywhere, like electricity.

The underlying motivation of this paper is the possibility of CN members - VBE, VE, VO, PVC[1] members and other type of companies, as logistics operators and software providers - to enlarge their collaboration towards a full-fledged, fully interoperable and transparent collaboration "cloud". Several works have been developed having this future CN scenario in mind, such as virtual machining [5], knowledge search and sharing over CN's information repositories [6], virtual shop-floor [7] and CN supporting ICT infrastructures [1]. This paper intends to contribute within this wider collaborative scenario towards sharing *software services*, i.e. to see services under the Utility paradigm point of view.

The relevance of this issue is related to the fact that, in practice, CN members who have SOA-based solutions keep the involved services (typically implemented using web-services technology) in their local silos. This means that the SOA potential in terms of reuse could be extended and highly increased if such silos could be shared among CN members, i.e. if all services could be accessed by any member so enlarging and reinforcing collaboration while development and hosting costs are decreased. This seems particularly relevant as CN members are mostly composed of MSMEs[2], without many conditions to maintain IT infrastructures and costly staff. In other words, this means that CN members can be both clients and service providers [1].

Providing this level of CN collaboration bumps, however, into plenty of obstacles, of several levels of complexity. One problem refers to the access mode and business models that can be aligned to this collaborative scenario [8]. In the SOA context, SaaS (*Software-as-a-Service*) [9, 10] has arisen as one of the most prominent models. Using SaaS, clients (i.e. CN members) can flexibly build their services portfolio according to their needs [11]. Services are accessed remotely, upon request, paid-per-use, based on contractual rules specified in SLAs (*Service Level Agreement*) [12, 13] for hosting, managing, providing access to them following QoS levels, no matter where the services providers are and how services have been deployed [10]. This model seems suitable to accommodate CN intrinsic specificities, namely members' independence, autonomy and heterogeneity, especially when they also act as service providers. A second problem comes as a consequence of this envisaged collaboration scenario: how trusting in a certain service whose CN member (provider) is not necessarily known or whose service quality is not trustworthy. Therefore, clients must feel confident to access and to use a given service as it is going to be used or integrated into their composite SOA applications, so they want to minimize the risk of a crash in the system due to the lack in the service's quality. The problem is that there are not very concrete guidelines and practices about how to develop highly reusable services under the SaaS business and architectural models. Existing quality reference models are huge and complex hence far from been easily adopted by MSMEs.

---

[1] VBE (Virtual Organization Breeding Environment), VE (Virtual Enterprises), VO (Virtual Organization), PVC (Professional Virtual Communities).

[2] MSME – Micro, Small and Medium Enterprises.

This is the underlying research question of this paper. Based on software engineering theories and quality models, this ongoing work approaches the problem of service trustworthiness in a qualitative and partially exploratory way, proposing a reference guide devoted to software services for SaaS model in way CN members can be more confident in sharing services among them. The research hypothesis is that a reference guide can act as an extremely useful enabler to support that envisaged collaboration at software services level. CN providers can be assisted on how to better develop and make available their services on one hand, and clients can be assisted on how to evaluate services trustworthiness on the other hand.

This paper is organized as following. Section I has motivated the issue of services trustworthiness. Section II gives an overview on the researches about trustworthiness. Section III resumes the methodology used to devise the proposed guide. Section VI presents the preliminary version of the Reference Guide for SaaS applications. Section V provides an assessment of the work and some conclusions.

2 A Revision on Software Trustworthiness and Quality

Trustworthiness in loosely coupled computational systems – which is the case of SOA/SaaS – has received great attention in the last recent years [14]. Actually, there is neither a largely adopted definition for it nor for the best way to manage it [15].

A current view of trustworthiness defines it as a measurable property with different entities at different levels. In the software services context, the concept of trustworthiness is extended to include a view of a mechanism for the realization and maintenance of the service quality and its interactions [14]. Trustworthiness can be determined based on some information, like personal experiences, observations, recommendations and global reputation [15].

In some works, the concepts of trustworthiness and reputation seem similar. Trustworthiness is gained by the act of meeting policies, code of ethics, rights and agreed promises [16]. Reputation is gained throughout the time, based on the general service performance and on users' evaluations [17]. Malik and Bouguettaya [18] mention that the reputation of web-services-based systems depends on the information received to establish trustworthiness between unknown participants. This work also includes the assessment of reputations of newly deployed web services, which is an important matter in SOA environments, as historical information is not always available. This work presented a hierarchical ontology, but does not present any technique for the verification of this reputation.

Msanjila and Afsarmanesh [19] approaches the automation of processes related to the trustworthiness level assessment in CNs. They formalized an analytical model and developed a system to evaluate the trustworthiness level of CN members [20]. As it focused on companies, it does not offer comprehensive support to software services.

Supporters of 'SOA architecture as Global Information Grid' (GIG), 'Department of Defense' (DoD) and the 'Intelligence Community' (IC) state that there is an explicit relation among security, flexibility and cost when trustworthiness is the focus [12]. They developed a trustworthiness management architecture based on their experiences,

predicting a unified management of the specification and interpretation of the security policies, credentials e relationships. This architecture is however very directed to the GIG specific model, hard to use by other companies, besides not providing supporting methodologies for its implementation. Grandison e Sloman [21] present a work related to trustworthiness in web applications based on a research on available methods and tools, such as *PolicyMaker* e *KeyNote*. However, the scope of this work does not cover software services, but Internet pages only, in order to control their content.

Within the software arena, the most considered and recognized way to get trustworthiness is via certifications [21]. Nowadays, there are institutions that certify the software development process of companies applying some reference models, such as ISO/IEC 15504 [22], CMMI Development [23], MPS.br [24]. There are also some initiatives in terms of reference models devoted to services, but they provide a different contextualization when SaaS is considered. In these models (e.g. ITIL [25], CMMI for Services [26] and the Capacity Models for outsourcing [27, 28]), service is understood just as an outsource or a post-sales provision (such as *helpdesk*, for instance).

Progress Software and its partners have jointly developed a SOA Maturity Model (SOA-MM), which is based on CMMI and ISO standards [29, 30]. It provides IT decision makers with a framework for benchmarking the strategic value of their SOA implementation. Based on CMMI, IBM has also developed a maturity model for SOA, called SIMM - Service Integration Maturity Model, composed of six levels of capability [23, 31]. This model includes elements that impact services' maturity, namely: coupling, standards usage, service identification, business models, goals, and metrics that need to be supported by services, technologies, governance and infrastructure. However, it is very generic and do not present any support for SaaS.

CobiT (Control Objectives for Information and related Technology) is another reference work [17]. It is a framework of best practices directed to the management of information technology with the focus on governance. This framework is platform independent as well as is not bound to any business model [32]. Despite of its large acceptance by companies, it has no specific recommendations for SaaS scenarios.

In fact, there are many other works which deal with software trustworthiness and quality in the literature. This overview tried to highlight the ones considered as the most relevant for the scope of this paper. As a conclusion of this revision, it was realized that there is no quality reference models for software services devoted to SaaS. Most of models cope with the traditional software acquisition model and even the ones related to SOA are still too generic. As such, they are models that cannot be readily applied by CN members towards the envisaged collaborative scenario. Yet, there are not models which take the particularities of the CN environment (i.e. a "club" of actors who are, per definition, willing to collaborate and to share assets) into account.

Building a totally new reference quality model for SaaS from scratch and that can comprise CN features is evidently too complex. In this sense, the approach was considering consolidated existing practices (like CMMI and ISO' practices) and adapting them to the context of SaaS and CN. Next section describes the methodology in general.

3 Methodology for the Reference Guide development

This chapter resumes the methodology used to devise the proposed guide development. The detailed description of the methodology can be found in Cancian et al 2010 [33].

The first general step of the methodology was related to know which requirements were necessary to be assessed from CN services providers regarding trustworthiness when CN clients would be playing the role of a SaaS client. A research was performed to gather information related to services and processes' quality criteria that should be analyzed or required from services providers.

A group of 280 professionals from the software services area were duly selected, from different countries, from academia / R&D and from companies. This step was carried out by using a free tool called *LimeSurvey* [34], which allows the creation of researches with several questionnaires. The research was available in the Internet during 60 days and 84 answers were received. A list of 25 criteria has been identified at the end. A formal description was assigned to each quality criteria, normalizing the different definitions and scope given by the different interviewed based on formal definitions existing in the reference literature. After this, quality requirements were grouped into three categories (as suggested in [35, 36]) with the aim of facilitating their understanding and awareness about their scope of use:

- *Organization* requirements: requirements that the organization must have anyway no matter the product or process dealt with;
- *Product* requirements: requirements to allow a verification in the product (i.e. the service itself) as a whole;
- *Process* requirements: requirements needed to follow in terms of services design and implementation.

The professionals were also asked about the importance - Essential, Very Important, Important or Unnecessary - of each criterion when looking at SaaS scenario. Figures 1 to 3 show the final compilation of the results, per category.

ITEM	QUALITY CRITERIA RELATED TO THE PRODUCT OCCURRENCES			
	Essential	Very important	Not critical	Unnecessary
Accessibility	59.26%	25.93%	14.81%	null
Reliability	58.02%	37.04%	3.70%	1.23%
Performance	27.16%	58.02%	13.58%	1.23%
Availability	33.33%	41.98%	19.75%	4.94%
Scalability	20.99%	50.62%	27.16%	1.23%
Integrity	69.14%	25.93%	4.94%	null
Interoperability	27.16%	51.85%	19.75%	1.23%
Robustness	29.63%	48.15%	20.99%	1.23%

Fig. 1. Quality requirements related to Product.

The answers were further and deeply discussed and assessed by three PMP (Project Management Professional) specialists on software services. The occurrences classified as "essential" and "very important" were considered as relevant criteria for SaaS (i.e.

included in the guide) when their sum were above 75%. This average was considered as representative based on the experience of those PMPs.

At this stage of the methodology the goal was to have a list of requirements considered relevant to be considered by both clients and providers when dealing with SaaS. Next stage was then related to identifying which processes and practices from software quality reference models that CN members should apply to indeed prepare themselves to adopt SaaS. A mapping was done in order to make this analysis.

ITEM	QUALITY CRITERIA RELATED TO THE ORGANIZATION OCCURRENCES			
	Essential	Very important	Not critical	Unnecessary
Infrastructure capability	32.10%	59.26%	7.41%	1.23%
Technically competent employees	29.63%	37.04%	33.33%	null
Prevision of continuity of service	29.63%	41.98%	23.46%	4.94%
Technically competent in business	22.22%	50.62%	25.93%	1.23%
Utilization of standards	35.80%	41.98%	19.75%	2.47%
Governance	complement	complement	complement	complement
Reputation	complement	complement	complement	complement

Fig. 2. Quality requirements related to Organization.

ITEM	QUALITY CRITERIA RELATED TO THE PROCESS OCCURRENCES			
	Essential	Very important	Not critical	Unnecessary
Acquisition	34.57%	48.15%	17.28%	null
Change control	22.22%	56.79%	19.75%	1.23%
Quality control on software process	38.27%	45.68%	16.05%	null
Version control	30.86%	48.15%	19.75%	1.23%
Development and requirement management	48.15%	35.80%	16.05%	null
Maintenance	29.63%	45.68%	22.22%	2.47%
Certification of process quality	6.17%	37.04%	45.68%	11.11%
Security (of sent and stored data)	54.32%	38.27%	7.41%	null
Help desk	19.75%	49.38%	29.63%	1.23%
Tests	30.86%	54.32%	13.58%	1.23%

Fig. 3. Quality requirements related to Process.

QFD was used (*Quality Function Deployment*) as the supporting methodology for this mapping (requirements against models' practices). QFD considers the point of views of the clients and of the organization, according to the technological needs [37-39]. The mapping was performed by those same three PMP specialists on software process improvement. An example (on *accessibility*) of this is given below.

After a very comprehensive analysis it was observed that ISO/IEC 15504 copes with most of the elicited requirements. Other processes more directed related to 'services' came from CMMI and CobiT due to the governance requirement.

Next step was to identify the relevance of each models' process, i.e. which processes are indeed important to be considered in a SaaS scenario. Those PMP specialists went through every single process and they classified each process as Essential ("E"), Very important: ("V"), Important ("I"), Weakly important ("W"), and Unnecessary ("U"). Let´s take the Accessibility quality criterion (see Figure 1) as an example. Accessibility within ISO/IEC 15504 is dealt by a number of processes, e.g. acquisition. According to this model, there are five issues (subprocesses) that should be observed within acquisition: acquisition preparation, supplier selection, contract agreement, supplier monitoring, and customer acceptance. In theory, targeting traditional business models, the acquisition process is very relevant and the reference models indicate a number of practices to support all of them. However, looking at SaaS perspective, it is observed that while e.g. *supplier selection* subprocess is not critical (i.e. is it unnecessary), the *contract agreement* subprocess is seen as essential regarding the importance of SLAs. Another example is the *infrastructure* subprocess (within the *resource and infrastructure* process), which is considered as essential. It is because a SaaS-driven service should naturally be available permanently and with a very high-level of performance as it tends to be accessed intensively and by several CN members, perhaps even simultaneously.

The subprocesses classified as "V" and "I" were also considered relevant for being considered by CN SaaS providers. For example, the subprocess *software requirements analysis* (within the *engineer* process) was classified as "I". As such, the practices associated to it should be applied as well (see Table 1 as an example of a subset of practices for this subprocess).

This all means that a CN member that wants to share its software services under the SaaS model should apply the recommended model's practices that are associated to each subprocess that has been classified as E, V or I as a way to "prove" to clients that it supports the subprocesses according to the norm. In this way, CN clients will be more confident to access the given service / provider as there would be a "guarantee" that the subprocesses would had been properly implemented.

As the final result of this mapping a correlation matrix was created. It contains the relevance identification of all elicited quality criteria against every subprocess[3].

[3] The complete matrix is available at www.gsigma.ufsc.br/~cancian/msc/mapping.pdf

4 Preliminary Reference Guide

The ultimate but general goal of a reference guide is to serve as a quick and easy source of information to users [40]. In this case, it serves as a guide to help both CN clients to have a more solid basis on how to select services from CN providers, and to CN providers to know which quality requirements (from the elaborated list) should be supported, besides which practices should be used for each process.

The Reference Guide can be either visualized on-line or be downloaded[4]. Figure 4 shows an excerpt of the guide. The Guide is composed of quality requirements and of best practices suggested for each requirement. For each best practice, it was suggested a level of importance according to its relevance: (i) Essential; (ii) Very important; (iii) Important; (iv) Weakly important; or (v) Unnecessary. For the Guide utilization, one should implement the best practices suggested for each requirement. The set of requirements with their descriptions can be found at label "Quality Requirements"

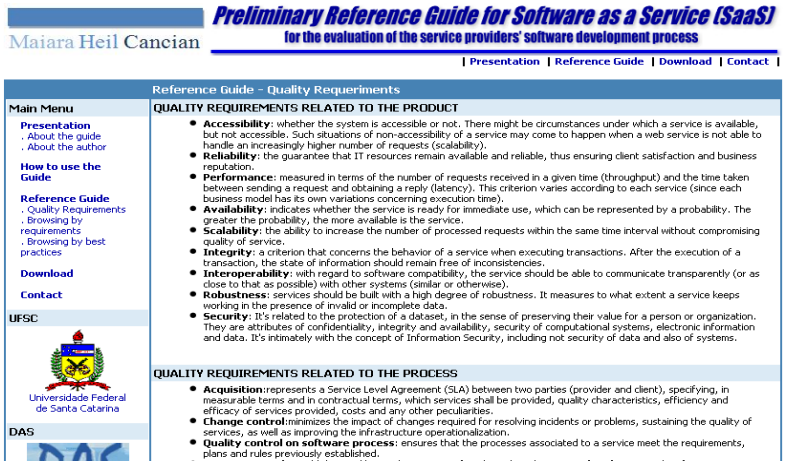

Fig. 4. Elicited SaaS software quality requirements.

The Guide offers two options for browsing over requirements and their best practices:

- Browsing by requirements: requirements are grouped by their classification. Clicking on the classification, requirements for that classification are shown (figure 5);
- Browsing by processes: best practices' titles are shown firstly. Clicking on the given practice, the respective requirements are shown.

Table 1 presents a partial view of the description of a process' practice. This exemplifies the Requirements elicitation. The information of this practice are: Process ID, Process Name, Process Purpose, Process Outcomes, Base Practices.

[4] The complete Reference Guide is available at http://www.gsigma.ufsc.br/~cancian/guide

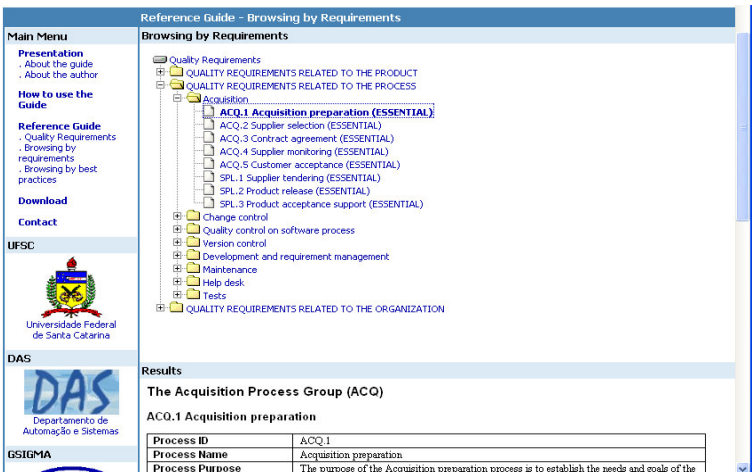

Fig. 5. Navigation through Requirements.

Table 1. Eng.1 Requirements Elicitation Description.

Process ID	**ENG.1**
Process Name	**Requirements elicitation**
Process Purpose	Requirements elicitation process aims to gather, process, and track evolving customer needs and requirements throughout the life of the product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed work products. Requirements elicitation may be performed by the acquirer or the developer of the system.
Process Outcomes	As a result of successful implementation of Requirements elicitation process: 1) continuing communication with the customer is established; 2) agreed customer requirements are defined and baseline; 3) a change mechanism is established to evaluate and incorporate changes to customer requirements into the baseline requirements based on changing customer needs; 4) …
Base Practices	**ENG.1.BP1: Obtain customer requirements and requests.** Obtain and define customer requirements and requests through direct and continuous solicitation of customer and users. NOTE 1: Requirements may also be obtained through review of customer business proposals, target operating and hardware environment, and other documents bearing on customer requirements. **ENG.1.BP2: Understand customer expectations.** Ensure that both supplier and customer understand each requirement in the same way. Review with customers their requirements and requests to better understand their needs and expectations and to check the feasibility and appropriateness of their requirements. [Outcome: 6] NOTE 2: Environmental, legal and other constraints that may be external to the customer need to be considered. **ENG.1.BP3: Agree on requirements.** Obtain agreement across teams on the customer requirements, obtaining the appropriate sign-offs by representatives of all teams and other parties contractually bound to work to these requirements. [Outcome: 2] **ENG.1.BP4: ...**

5 Final Considerations

This work has presented an approach on how services' trustworthiness in SOA-based SaaS applications can be supported within a CN. This approach is represented by a reference guide, which is seen as a list of quality criteria to be used by CN members (services clients and providers) when sharing services, no matter their size or CN type.

The elaboration of the guide has followed a methodology, which was fundamentally grounded on the rationale of existing and best-of-breed quality reference models of

software engineering. However, the guide cannot be seen as definitive. SaaS in an emergent area and there is not very solid reference theoretical foundations about it yet. In spite of this, it is believed that the guide has a structure that can be applied as an useful instrument to assess CN members services' trustworthiness, keeping their autonomy and heterogeneity while collaboration among them is enlarged.

The guide integrates some reference models and best practices into a more specialized, single, relatively simple and free-of-access model. This is important as long as small companies have tremendous difficulties to go through each (huge) reference model and to evaluate which one is better aligned to their strategy. Therefore, from the exploratory perspective of this research, it can be said that a guide like that has the potential to solve the "theoretical" part of the problem, which refers to how CN members – the guide's clients and beneficiaries – can be more confident when sharing their services. It is important to point out that a guide should be seen as *a* common instrument, among many others (e.g. security, infrastructures, service discovery, etc.), necessary to support the envisaged collaborative scenario.

However, from the applied perspective of this research, a guide like this, even being simpler and devoted to SaaS, is not easy to implement in practice as it impacts companies at several levels. As such, the usage of a guide may be not so feasible for CN members regarding their usual small size and limitations. As a consequence, CN members might rather prefer maintaining their services in their isolated silos or even developing their services without sharing purposes as a possible attitude regarding that development of software services are not their core business.

On the other hand, the own VBE can adopt an institutional position creating a task-force or hiring a specialized company to help members in implementing the guide in all members in way to also share costs. This kind of joint or shareable activities is a common practice in VBEs. As a matter of fact, this can be considered as one additional preparedness level for a member getting into a VBE. Moreover, companies can make some profit out of this since the access to their services (considering that they would have to maintain them anyway) would be on-demand and paid-per-use. Yet, strategic benefits can be a good argument for the guide adoption as companies can become more agile and feel better prepared to enter in a new VO as new required software services may be, in fact, available at the VBE. This would prevent them from new services development or acquisition, in opposite to the case without any sharing. Therefore, this "new" collaboration *bus* can add some value to CN members' business.

From the SOA perspective, a guide like this can also be a powerful mechanism for helping in services discovery and SLA establishment.

Future short-term improvements in the guide include the definition of more detailed practices to cope with SaaS for every single criterion as well as the creation of different capacity levels (e.g. A, B and C) allowing a better assessment by CN members. After this, a more solid and larger validation of the guide intends to be carried out using methodologies like IEEE [41] in way its accuracy, completeness and consistency can be better evaluated. Finally, new more CN-related criteria may be added to the guide, such as 'collaboration frequency', for instance.

ACKNOWLEDGMENTS
This work has been partially supported by the Brazilian agencies CAPES and CNPq.

References

[1] Rabelo, R., Advanced Collaborative Business ICT Infrastructures, in Methods and Tools for Collaborative Networked Organizations, Springer, Editor. 2008. p. 337-370.
[2] Camarinha-Matos, L. and H. Afsarmanesh, The Emerging Discipline of Collaborative Networks, in Virtual Enterprises and Collaborative Network, Springer, Editor 2004 p 3-16.
[3] NESSI, _Strategic_Research_Agenda , Framing the future of the Service Oriented Economy. Version 2006-2-13 - Revision 3.1 (http://www.nessi-europe.com/documents/NESSI_SRA_VOL_1_20060213.pdf). 2006: ICT for Enterprise Networking (http://cordis.europa.eu/ist/directorate_d/en_intro.htm).
[4] Singh, M.P. and M.N. Huhns, Service-Oriented Computing Semantics, Processes, Agents. 2004: John Wiley & Sons, Ltd.
[5] de_Souza_Jr., J.L.N.J., et al., An Internet-oriented Management and Control System in a Distributed Manufacturing Environment. International Journal of Manufacturing Research, 2009. 5: p. 5-25. International Journal of Manufacturing Research.
[6] Tramontin_Junior, R.J., R.J. Rabelo, and H. Chirab, Customizing Knowledge Search in CNOs through Context-based Query Expansion. Production Planning & Control, 2010. 21(2): p. 229-246. Production Planning & Control.
[7] Ribeiro, L., J. Barata, and A. Colombo, Supporting agile supply chains using a service-oriented shop floor. Eng. Application Artificial Intelligence, 2009. 22(6): p. 950-960.
[8] Borst, I., et al., Technical Report D62.2 ICT-I Business Models,www.ecolead.org. 2005.
[9] Candan, K.S., et al. Frontiers in Information and Software as Services. in Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on. 2009: p. 1761-1768.
[10] Ma, D. The Business Model of "Software-As-A-Service". in Services Computing, 2007. SCC 2007. IEEE International Conference on. 2007: p. 701-702.
[11] Tsai, W.T. Service-oriented system engineering: a new paradigm. in Service-Oriented System Engineering, 2005. SOSE 2005. IEEE International Workshop. 2005: p. 3-6.
[12] Eliadis, H. and A. Rand. Setting Expectations In SaaS: The Importance of the Service Level Agreement to SaaS Providers and Consumers. in SIIA Software as a Service Working Group. 2007: Software & Information Industry Association.
[13] Cancian, M.H., R.J. Rabelo, and C.G.v. Wangenheim. A proposal for elaborating SLAs for Software-as-a-Service (SaaS) [in Portuguese]. in 8th International Information and Telecommunication Technologies Symposium. 2009.
[14] Terzis, S. (2009) Trust Management. Guest Editor's Introduction - The many faces of trust, Available from: http://www.computer.org/portal/web/computingnow/archive/april2009.
[15] Bhargava, B., et al., The pudding of trust. Intelligent Systems, IEEE, 2004. 19(5): p. 74-88.
[16] Blaze, M., et al., Dynamic Trust Manag. Computer, 2009. 42(2): p. 44-52. Computer.
[17] ISACA. Serving IT Governance Professionals - COBIT Control Objectives for Information and related Technology. 2008 10/12/2008]; Available from: http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=31519.
[18] Malik, Z. and A. Bouguettaya, Reputation Bootstrapping for Trust Establishment among Web Services. Internet Computing, IEEE, 2009. 13(1): p. 40-47. Internet Computing.
[19] Msanjila, S. and H. Afsarmanesh, Automating Trust Assessment for Configuration of Temporary Partnerships, in Innovation in Manufacturing Networks. 2008. p. 95-104.
[20] Msanjila, S.S. and H. Afsarmanesh. HICI: An Approach for Identifying Trust Elements The case of technological trust perspective in VBEs. in Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on. 2007: p. 757-764.

[21] Grandison, T. and M. Sloman, A Survey of Trust in Internet Applications. IEEE Communications Surveys and Tutorials, 2001. 3(4).
[22] ISO/IEC, International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 15504-5: Information Technology - Process Assessment, in Part 5: An exemplar Process Assessment Model. 2006: Genebra.
[23] Team, C.P., CMMI for Development (CMMI-DEV), in Technical Report CMU/SEI-2006-TR-008, V. 1.2, 2006, Carnegie Mellon University / Software Engineering Institute.
[24] Softex, Associação para Promoção da Excelência do Software Brasileiro, MPS.BR – Guia Geral, versão 1.2, http://www.softex.br/mpsbr/_guias/default.asp, Editor. 2007.
[25] Taylor, S., ITIL Service Management Practices - V3 Qualification scheme, http://www.itil-officialsite.com/Qualifications/Examiners/SharonTaylor.asp, Editor. 2007. p. 31.
[26] Team, C.P., CMMI for Services (CMMI-SVC) in Technical Report CMU/SEI- PA 15213-3890. 2008, Carnegie Mellon University / Software Engineering Institute: Pittsburgh.
[27] Hefley, W.E. and E.A. Loesche, The eSCM-CL v1.1: Model Overview The eSourcing Capability Model for Client Organizations, C. Mellon. 2006, Carnegie Mellon University.
[28] Hyder, E.B., K.M. Heston, and M.C. Paulk, The eSCM-SP v2.01: Model Overview The eSourcing Capability Model for Service Providers, C. Mellon, Editor. 2006, Carnegie Mellon University (http://www.ae-scm.fr/bibliotheque-escm/eSourcing%20Capability%20Models%20Overview.pdf).
[29] Sonic, Progress Software Corporation, A new Service Oriented Architecture (SOA) Maturity Model. 2007: Bedford, MA 01730 USA. p. 27.
[30] Josuttis, N.M., SOA in practice: ed. O'Reilly. 2008.
[31] Arsanjani, A. and K. Holley. Increase flexibility with the Service Integration Maturity Model (SIMM). 2005; Available from: http://www.ibm.com/developerworks/webservices/library/ws-soa-simm/.
[32] Sahibudin, S., M. Sharifi, and M. Ayat. Combining ITIL, COBIT and ISO/IEC 27002 in Order to Design a Comprehensive IT Framework in Organizations. in Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on. 2008: p. 749-753.
[33] Cancian, M.H., et al. Discovering Software Process and Product Quality Criteria in Software as a Service. in The 11th International Conference on Product Focused Software Development and Process Improvement (PROFES 2010). 2010. Limerick, Ireland.
[34] LimeSurvey. The open source survey application 2008 [cited 2008 02/04/2008]; Available from: http://www.limesurvey.org/.
[35] Nightingale, P., The product-process-organisation relationship in complex development projects Research Policy, 2000. 9(7): p. 913-930. Research Policy.
[36] Noël, F., A product-process-organisation integrative model for collaborative design, in Innovation in Life Cycle Engineering and Sustainable Development. 2006. p. 407-418.
[37] Liu, X., et al., QFD application in software process management and improvement based on CMM. SIGSOFT Software Engineering Notes, 2005. 30(4): p. 1-6.
[38] Richardson, I. Quality Function Deployment - A Software Process tool? in Third Annual International QFD Symposium 1997. Linkoping, Sweden: p. 58-66.
[39] Herzwurm, G., S. Schockert, and W. Pietsch, QFD for Customer-Focused Requirements Engineering, in Proceedings of the 11th IEEE International Conference on Requirements Engineering. 2003, IEEE Computer Society.
[40] Steehouder, M. and H. van der Meij. Designing and evaluating procedural instructions with four components model. in Professional Communic. Conference, 2005: p. 797-801.
[41] IEEE, S., IEEE Recommended Practice for Software Requirements Specifications, in The Institute of Electrical and Electronics Engineers. 1998: New York, NY.