

CASE STUDY

Open Access

From project-oriented to service-oriented software development: an industrial experience guided by a service reference model

Marcos Kalinowski^{1*}, Stefan Biffel², Rodrigo Oliveira Spínola³ and Sheila Reinehr⁴

* Correspondence:

kalinowski@ice.ufjf.br

¹Federal University of Juiz de Fora,
Rua José Kelmer s/n, Juiz de Fora
36.036-330, Brazil

Full list of author information is
available at the end of the article

Abstract

Introduction: In organizations with software systems in production, new and often unexpected requirements for development come up due to strategic, tactical, and operational customer needs. In this context, it is a strategic advantage for software suppliers to be able to provide software services that meet these demands faster and with less overhead than negotiating traditional value-neutral project-oriented software deliveries.

Case description: This article reports on the industrial experience of restructuring the supplier-side software development process into a value-based service-oriented format, guided by a service reference model. A service level agreement (SLA) was established between supplier and customer reflecting the business needs and values. The report describes the contractual aspects and internal managerial controls employed to facilitate the compliance of the provided services with the SLA, including the integrated use of a managerial spreadsheet, an issue-tracking system, and a Kanban chart.

Discussion and evaluation: The feasibility and results of restructuring software development into a service-oriented format are evaluated. Major results were that only moderate effort was required, around one person month, due to the support of the service reference model and a sufficient level of previously installed capabilities, and that the goals regarding improved quality, productivity, and customer satisfaction were successfully achieved. Additionally we discuss stakeholder needs, the support from the service reference model, the lessons learned, and the success factors for such restructuring.

Conclusions: Restructuring software development in the format of continuous service delivery, guided by a service reference model, is feasible and for suitable contexts can provide significant benefits concerning quality, productivity, and customer satisfaction.

Keywords: Service reference model; Software development as a service; Software process; Software project management; Software quality

Background

In dynamic organizational environments with software systems in production, it is not always possible to forecast and formalize in a contract the requirements for development that will arise over time (Barney *et al.* 2008). Collecting upcoming requirements in formal projects can incur significant overhead and delay to evolving the software that supports mission-critical business processes and analyses.

Therefore, it is common for customer organizations to seek for Information Technology (IT) suppliers who can provide services to efficiently and quickly handle demands according to their business needs, respecting the varying volumes and priority of these demands (Khan *et al.* 2011). However, from the point of view of the supplier, restructuring the software development process to meet such customer expectations is not always an easy task, in particular, if it is not clear whether the supplier software organization process is sufficiently mature to drive towards a service delivery format (Lehman and Sharma 2011).

Treating a customer demand for software development as a request for an IT service is a promising way for addressing the customer value expectations, in line with the definition of services by ISO/IEC (2011), which defines a service as a “means of delivering value to customers by facilitating outcomes customers want to achieve”. In practical terms this means migrating software development from a traditional project management format (PMI 2013) to a continuous service delivery management format (TSO 2011).

IT service management can be defined along the lines of the ITIL (Information Technology Infrastructure Library), a model conceived by the British government with a view to provide a consensus on the best IT service management practices, as “a set of specialized organizational capabilities for providing value to customers in the form of services” (TSO 2011).

In the context of providing IT services, it is important to make an effort to establish efficient service management processes (TSO 2011), preferably based on a reference model that supports the best practices for improving service processes and, consequently, increasing productivity and effectiveness of the services provided.

One of the programs available to meet this type of need is the Brazilian nationwide MPS.BR program (Santos *et al.* 2012). In Brazil about 73% of the software industry is constituted of small and medium-sized enterprises (SMEs) (MCTI 2013). Therefore, an effort was made to develop national reference models for software development and IT service delivery, which are compliant to well-established international standards and reference models, in order to provide their suppliers, including SMEs, with more fine-grained steps to define and achieve an appropriate level of software process maturity. Software regulators in other countries with a high share of SME software suppliers can benefit from the lessons learned in Brazil to better support their small-scale software suppliers.

Thus, the main MPS.BR objective is to develop and disseminate reference models that meet the requirements of the Brazilian software and IT services industries, allowing software suppliers, including SMEs, to deliver to customers according to internationally recognized quality standards (Santos *et al.* 2012). The MPS.BR family of reference models currently consists of the MPS-SW for Software (SOFTEX 2012a) and the MPS-SV for IT Services (SOFTEX 2012b). The MPS-SV reference model is a promising basis for establishing service management processes. Another similar and compatible alternative (SOFTEX 2012b) would be following the guidelines of the international CMMI-SVC (CMMI for Services) reference model (SEI 2010).

While the MPS-SW has been established in 2003 and has been widely adopted in Brazil, with 548 official assessments (over 70% of them in SMEs) published by April 2014 (SOFTEX 2014), the MPS-SV model is still very recent and saw its first evaluation published in September 2012. The MPS-SW has already helped on the adoption of good software engineering practices in the Brazilian industry (Kalinowski *et al.* 2010).

There is also objective evidence of positive impacts on the performance of the software suppliers that adopted this model (Kalinowski *et al.* 2008a) (Travassos and Kalinowski 2013), which could be transferred to other countries with a similar software development structure. The MPS-SV model has a broader application scope than the MPS-SW model, since MPS-SV can be applied to support the structuring and improvement of IT service processes in general. These services might include Help Desk and support services (SOFTEX 2012b) or even software development services, as in the experience reported in this paper.

The MPS-SV was developed in conformance to the international standards ISO/IEC 20000:2011 (ISO/IEC 2011), ISO/IEC 15504 (ISO/IEC 2004), being compatible with the CMMI-SVC model (SEI 2010) (SOFTEX 2012b). Therefore, in the context of this paper the MPS-SV can be seen as a representative for international standards. Similar to the MPS-SW model, the MPS-SV is structured in seven maturity levels for assessment (G to A, where A is the highest maturity level), while the CMMI-SVC reference models is structured in four maturity levels (2 to 5, where 5 is the highest maturity level). The compatibility between the MPS-SV and CMMI-SVC reference models is given by the maturity level mapping shown in Table 1.

Software development as a service (SDaaS) started being discussed recently (Lehman and Sharma 2011) and the thematic of applying service reference models to software development has been informally presented at the SEPG North America 2011 (Penn 2011) and SEPG North America 2013 (Penn 2013). However, to the best of our knowledge, there are no published experience reports available on this topic. Therefore, the lack of published work related to moving from a project-oriented to a service-oriented process, and in particular based on a service reference model, leads to uncertainties concerning feasibility, effects, and success and risk factors (pitfalls). An initial effort to bridge this gap was reported in (Kalinowski and Reinehr 2013), the paper we are herein extending. Given this scenario, we investigate the following two research issues (RIs) to shed light on applying service reference models to re-structure software development from a project-oriented to a service-oriented management format from an industry perspective.

RI-1. Survey on perceived utility of structuring software development guided by service reference models

Do software engineering consulting experts see significant utility in adopting service delivery practices for software development? Would these experts consider using a service reference model as a basis for adopting service delivery practices?

Table 1 MPS-SV and CMMI-SVC maturity level compatibility

MPS-SV maturity levels	CMMI-SVC maturity levels
A – In Optimization	5 – In Optimization
B – Quantitatively Managed	4 – Quantitatively Managed
C – Defined	3 – Defined
D – Largely Defined	
E – Partially Defined	
F – Managed	2 – Managed
G – Partially Managed	

To address this research issue, a survey was conducted in the MPS-SV context with software engineering consultants (19 certified MPS-SW implementation consultants from 11 different MPS.BR accredited implementation institutions), who work with software development suppliers helping them to organize their production processes (Jordão and Kalinowski 2013). The survey was structured following the Technology Acceptance Model (TAM) (Davis 1989) to gather the perception on the utility, ease of adoption and intention to adopt from the point of view of those consultants. In a systematic review concerning the TAM, conducted by Turner *et al.* (2010), a correlation between the intention to adopt, as stated in the studies that used the TAM, and actual adoption could be identified, which reinforced the decision of following this model.

Results indicated that the MPS-SW software engineering consultants consider service reference models useful and that they would consider adopting them for providing continuous delivery capabilities to their software industry customers. A summary of these results, which reinforce the motivation for investigating the following research issue, are compiled in the Discussion and Evaluation Section. Further details on the survey planning, operation and limitations are published in a separate paper (Jordão and Kalinowski 2013).

RI-2. Industry experience report on the feasibility and results on restructuring software development guided by a service reference model

What are typical stakeholder needs that trigger software development restructuring? Can a service reference model be helpful to meet restructuring needs in software development? What effort is to be expected? What are the potential effects on quality and productivity? What are the main lessons learned? What are the involved success and risk factors?

To address this research issue a real experience of restructuring software development guided by a service reference model was conducted, analyzed, and an initial report (in Portuguese) was produced (Kalinowski and Reinehr 2013). Main reported results were: (i) the MPS-SV model was found helpful to guide the restructuring; (ii) restructuring required only modest effort; and (iii) significant benefits were perceived in terms of quality, productivity, and customer satisfaction.

However, this initial report only provided an overview of the restructuring and brief indications on effort and on the produced effects on quality and productivity. Therefore, many questions raised in RI-2, to which answers would provide additional insights into structuring software development as a service, remained unanswered. For instance, stakeholder needs that typically trigger such restructuring and how the MPS-SV can support meeting these needs were not described. Lessons learned were also not discussed in depth to allow further understanding the assumptions for such restructuring and possible improvements. Finally, major success and risk factors were not identified.

This paper extends the initial report (Kalinowski and Reinehr 2013) by providing further details on the context and on how managerial skills for handling service requests were established aiming at complying to a Service Level Agreement (SLA) with the customer to satisfy his business needs. Contractual aspects of the restructuring and internal controls used to ensure compliance with the SLA, are described in details. Those internal controls encompass the integrated use of managerial spreadsheets, issue-tracking systems and Kanban charts (Anderson 2010) to monitor the demand prioritization.

Moreover, further analyses of the experience allowed identifying the stakeholder needs and how the MPS-SV supported meeting them. The discussion on lessons learned was extended and success and risk factors were identified. We assume that these additional details, analyses and extensions can provide further insights into the feasibility of applying service reference models to software development, which are relevant for SME software suppliers in Brazil and in comparable international contexts.

The remainder of this paper is organized as follows. Case description describes the case: context of the experience, including the identification of the stakeholder needs, and how the development process was re-structured. Discussion and evaluation presents the discussion and evaluation, including obtained results, lessons learned, success and risk factors. Conclusions concludes and discusses issues for further research.

Case description

This section describes the experience related to investigating RI-2 by restructuring the software development process guided by a service reference model. Therefore, the context is described and further details on how the development was structured as a service are provided.

Context and stakeholder needs

The experience described herein occurred in the context of the companies Kali Software (KS), as the software supplier, and Tranship Transportes Marítimos (TTM), as the customer.

The supplier, KS, can be characterized as a small-sized Brazilian software supplier (20 employees – including the board of directors) that had worked on custom development for almost a decade, providing services for national and international customers in a range of business areas (e.g., naval industry, health insurances, and financial). Despite of its small size the development followed distributed processes, with the development team and their local management were located in the city of Juiz de Fora (Brazil), and directors and requirement analysts located in the city of Rio de Janeiro (Brazil). The previously adopted development process followed a traditional interactive and incremental development life cycle, where increments were included in plan-driven projects (Boehm and Turner 2003), with each project negotiated as a separate development contract.

The customer, TTM, is a medium-sized Brazilian sea freight company (about 400 employees) that provides strategic services to the country. These services include coastal transport and naval support to oil rigs working at the pre-salt layer of the continental shelf. TTM had several software suppliers and internal IT support. The internal support was provided by an IT manager (responsible for supplier agreement management and conducting acceptance tests) and an IT support analyst (responsible for the installed server infrastructure and help desk support).

The software development partnership between the two companies was established in February 2011. The main developed software in this context concerned an Enterprise Resource Planning (ERP) system, involving several modules, such as administrative, operations, allocation of ship crew members, finance, and human resources, which were gradually put into production. In total, 87 use cases were implemented in an overall effort of more than 4000 hours, resulting in a management system of around

83,000 lines of Java code and 167 data tables. At the time of the herein reported experience, 10 of KS employees (1 director, 1 project manager, 1 requirements analyst, 6 developers and 1 testing analyst) were involved – not all exclusively –with the TTM ERP project.

With the modules entering production, the stakeholder's needs started to change. TTM's directors (executive, operations, financial, and human resources), for instance, began needing information and new functions very quickly according to their immediate strategic, tactical, and operational business goals. The contractual model for new development requests was previously structured as separate projects (characterized according to the definition of the PMI (2013) by having a manageable scope, a beginning and an end), in which a new contract had to be negotiated in order to develop new functions. This contractual approach was not adequate for the new scenario anymore. The main problems of this value-neutral approach (Boehm 2006) were the effort of negotiating new contracts and not considering added business value for prioritizing individual demands. As a consequence, fast deployment of the most important new system capabilities from a business point of view, was not achieved.

Table 2 shows an overview on the different stakeholder interests after the system entered production. The interests mainly conflicted with the negotiation of demands to be included in a value-neutral project context. This negotiation effort and not sufficiently considering added business value of individual development requests hindered delivering optimal business value to the customer.

Therefore, the directors of both companies met to define a new and more flexible contractual model to satisfy the new stakeholder needs. They decided on a contractual framework based on the provision of services, in which demands were treated as separate requests for services with different priorities and the billing would be linked to compliance with a SLA (in traditional project-oriented development, contracts usually include terms for incremental delivery and related penalties). By considering a SLA based on the customer's business needs, this contractual framework helps integrating value considerations into software development, as suggested by the Value-Based Software Engineering (VBSE) discipline (Biffl *et al.* 2006).

The decision of establishing a provision-of-services contract meant that the supplier had to adopt a strategy for delivering such services with internal controls to facilitate the management of individual requests in order to comply with the SLA. The following subsection describes this strategy.

Table 2 Stakeholder interests conflicting with the value-neutral project approach

Stakeholder	Stakeholder interests
TTM - Directors, managers and operational level employees of the different business areas.	Simple contract negotiation; fast definition and deployment of system capability changes or increments; delivery priority according to current added business value.
TTM - IT Manager & IT support analyst	Fast, predictable, and high quality deliveries.
KS – Directors and Project Manager	Simple contract negotiation; steady inflow of system capability changes or increments; translate technical productivity into higher net gains; visibility of team productivity.
KS – Developer& Quality Assurance	Quick customer feedback.

Restructuring software development as a service

Although an official assessment was not in the plans of the supplier, it was decided that the structuring of software development as a service should follow the guidelines of the reference model MPS-SV maturity level G (SOFTEX 2012b). This decision was taken to ensure that service delivery best practices would be incorporated into the new development process. The MPS-SW has contributed to the adoption of good practices by the Brazilian software industry (Kalinowski *et al.* 2010), and the national experts expect that the MPS-SV can do the same for the IT services sector.

Level G of the MPS-SV reference model (SOFTEX 2012b) contemplates five processes: Work Management, Requirements Management, Service Delivery, Service Level Management, and Incident Management. During the experience described in this paper, the practices of Work Management and Requirements Management, already in place at KS in the software development context, were complemented with practices of the remaining three processes directly related to service management. A brief description of these three processes (Service Delivery, Service Level Management, and Incident Management) follows.

The Service Delivery process aims to use a strategy for service delivery in line with the established service agreements. Service Level Management process aims to ensure that the customer's SLAs are fulfilled. Finally, the purpose of the Incident Management process involves capabilities for managing incidents and service requests (SOFTEX 2012b).

MPS-SV's Service Delivery precepts are handled in Contractual aspects Section, in which the contractual aspects that made the service delivery strategy possible and the SLA established in order to meet the customer's needs are described. Service Level Management and Incident Management are addressed in Supplier-side managerial controls Section, in which the internal managerial controls employed for monitoring the SLA and how to operate service requests are detailed.

Contractual aspects

The contractual model had to undergo changes to enable the new service delivery strategy. In this new model, demands for development came to be treated as service requests. Previously, demands had been grouped into increments handled as development projects, billed accordingly as the projects progressed (20% at the outset, 30% after functional specification, and 50% following final delivery).

With the new desired dynamics, requests would no longer be grouped into development projects, but rather handled as work associated with isolated services that should comply with a SLA. The CMMI-SVC reference model (SEI 2010) defines work as "a managed set of people and other resources that delivers one or more products or services to a customer or end user". Thus, in the case of this experience, switching from project management to work management would result in a finer granularity of items to manage.

The strategy that was defined to enable service delivery involved dividing the billing of the contract into two parts, one fixed and one variable. The former included the payment of a full-time requirements analyst in charge of receiving requests from users and specifying them as service requests. In addition to the analyst, the fixed part also included a fee for corrective maintenance services (development effort related to fixing

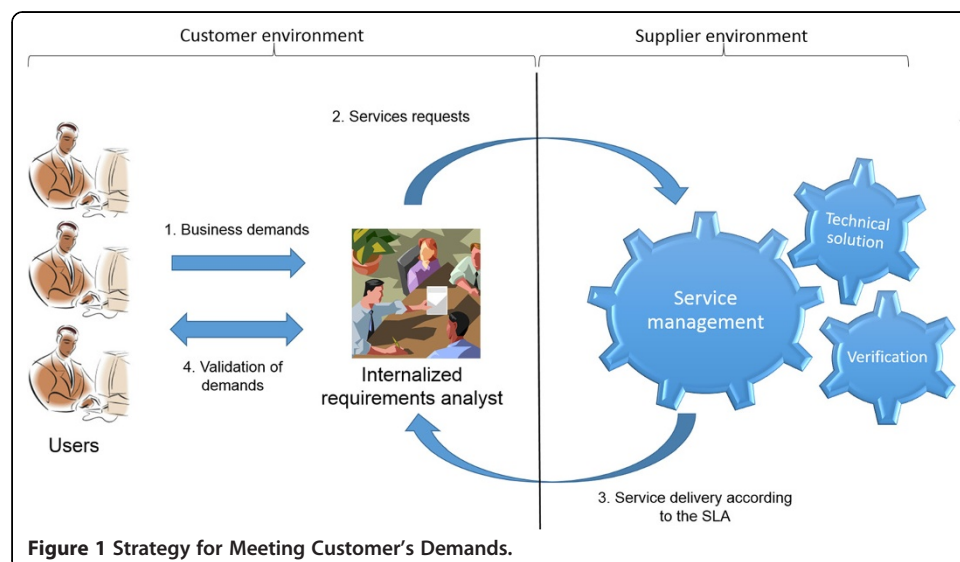
system failures). The variable monthly part, on the other hand, was calculated on a fixed day of the month, based on the implemented requests, according to the SLA. Therefore, higher productivity would directly and intuitively result in a higher monthly net gain for the software supplier.

The strategy for satisfying customer requests is outlined in Figure 1. As shown in this figure, the requirements analyst receives business requests from the system's users. These requests get prioritized according to their added business value and written in the form of requirements to be implemented, in order to become service requests. The priorities are defined together with TTM's IT Manager and other appropriate stakeholders.

The treatment of the service requests was managed to ensure that each request was attended complying with the established SLA. As soon as the developer concludes the technical solution of a request, before it is considered ready for validation, the solution is sent to a test by an independent team. Finally, the delivered requests get validated with the system's users.

The 'Service Level Agreement and Penalties' subsection shows the SLA part of the contract. In this SLA, requests have different deadlines, defined as numbers of days, according to their priority. The priority should reflect the added business value, so that higher added business value has to be treated faster. The following prioritization criteria was adopted:

- *Critical Requests*: these impede the use of the system for the business activities of some sectors or relate to new functionalities that can significantly improve business results.
- *High Priority Requests*: these do not impede the use of the system for the business activities of entire sectors, but hinder the conclusion of some specific business operations;
- *Medium Priority Requests*: these relate to new functionalities that can allow the speedier execution of some business operations;
- *Low Priority Requests*: these relate to new functionalities that enable features of the system to be improved.



According to this SLA, if a critical request of a new functionality with estimated effort of 40 hours is fully answered on the tenth working day after the request and with a total effort of 80 hours, then only 42 of the 80 worked hours would be billable. This amount corresponds to the maximum billable amount of 60 hours with a penalty of 30%, since it should have been answered at the eighth workday and was only answered in the tenth workday. In cases the supplier did not agree with classifying a new functionality as critical, the customer had to explain how the new functionality would improve business results (e.g., avoiding losses or enabling higher gains).

On the other hand, if a critical correction with estimated effort of 40 hours is fully answered on the tenth working day after the request and with a total effort of 80 hours, then 18 billable hours would be deducted from the total monthly billable hours. In this case, corresponding to a 30% penalty for the delay over the 60 maximum billable hours.

Given this scenario, in which delays have direct financial impact, monitoring the request against the SLA becomes of critical importance. Therefore, to monitor compliance with the SLA, requests were recorded on a spreadsheet, one of the internal managerial controls used to facilitate management of services. More information concerning managerial controls is provided in subsection Supplier-side managerial controls.

Service level agreement and penalties

This service level agreement is valid for corrections, as well as for changes and new developments, requested from the entry date of this agreement.

Agreement

The requests will have the following deadlines to be fully met (with no defects)

- *Critical Requests*
Deadline in hours = (1.5 x estimated effort) Hours.
Deadline in days = (Deadline in hours/8) Working Days.
- *High Priority Requests*
Deadline in hours = (1.5 x estimated effort) Hours.
Deadline in days = (Deadline in hours/8) + 1 Working Days.
- *Medium Priority Requests*
Deadline in hours = (1.5 x estimated effort) Hours.
Deadline in days = (Deadline in hours/8) + 2 Working Days.
- *Low Priority Requests*
Deadline in hours = (1.5 x estimated effort) Hours.
Deadline in days = (Deadline in hours/8) + 4 Working Days.

The deadline in hours will be used as a limit on the number of billable hours. The deadline in days will be used for penalties related to the deadline. Whenever this number is odd it should be rounded up, i.e., a period of 8.32 days should be met within the ninth day.

Penalties

Change requests or new developments that are not answered within the service level agreement will automatically be penalized by 30%.

Correction requests, included in the monthly amount, that are not answered within the service level agreement, will be automatically converted into a penalty of 30% of the deadline in hours, taking into account the hourly cost established in this agreement. This amount will be deduced from the variable cost to be billed monthly.

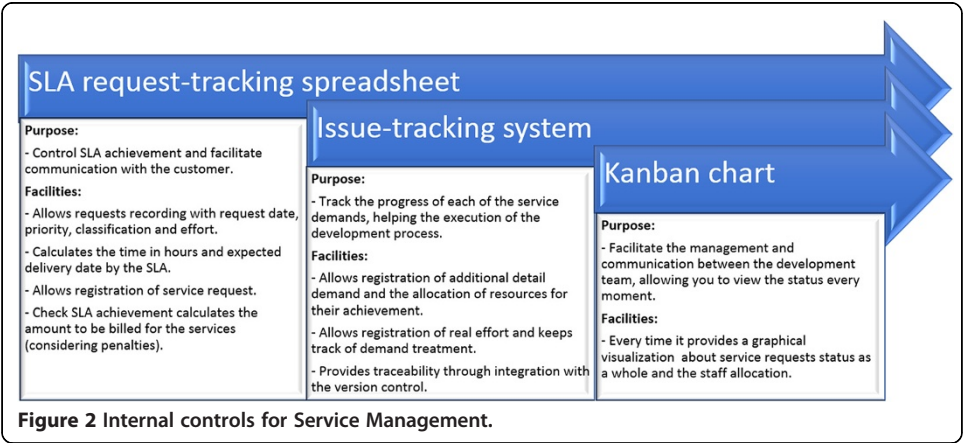
Supplier-side managerial controls

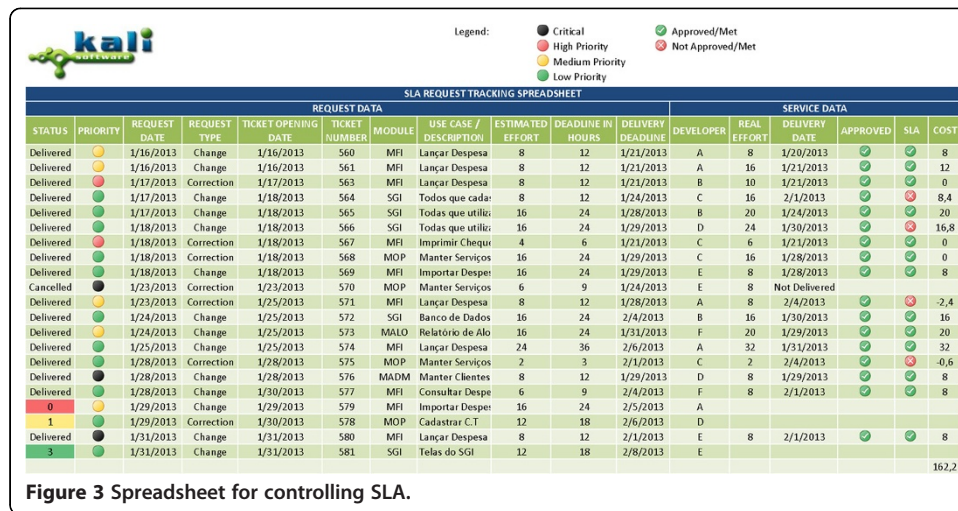
The following three internal managerial controls were used for supporting service management at the software supplier:

- a *request-tracking spreadsheet* for monitoring the requests and their conformance to the SLA;
- an *issue-tracking system* (integrated with the version control), in which a ticket is registered for each service request; and
- a *Kanban chart* to show the service status at any time to the whole team involved in providing the overall continuous development service.

Kanban charts have been used in agile development approaches such as Lean and SCRUM, for more information on the use of Kanban charts in the technology business refer to (Anderson 2010). Figure 2 shows the sequence in which these controls are applied for handling service requests according to the precepts of the MPS-SV Service Level Management and Incident Management processes. This figure also highlights the purpose and facilities offered by each of these controls.

Initially, a new service request is recorded in the request-tracking spreadsheet. The spreadsheet containing the requests made between 01/16/2013 and 01/31/2013 is shown in Figure 3. The columns for the developer and the cost have been deliberately modified to avoid revealing individual performance and cost information. The cost is represented by the number of billable hours rather than the expected cash value of the services. This spreadsheet allows incident management and monitoring the conformance of the provided services to the SLA, as required for compliance with the MPS-SV Incident Management and Service Level Management processes. This enables services delivery according to their business value and priority. It is important to state that this spreadsheet was also shared with the customer for billing purposes, thus communicating the status to



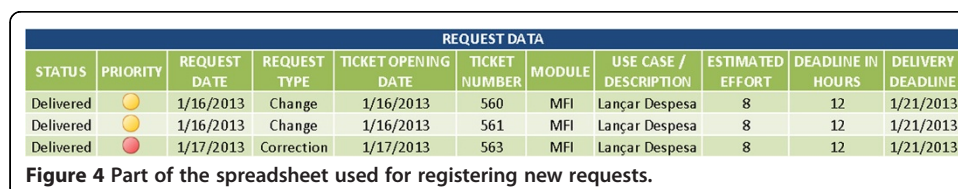


the interested stakeholders, as also required by the MPS-SV Service Level Management process, is also achieved. Figures 4 and 5 present a zoomed view of the parts containing data related to registering and processing the new service request.

Figure 4 shows the data registered for each request: (i) the date of the request, (ii) the request type (“correction” or “change” – in this case “change” also includes new development requests), (iii) the opening date of the ticket, (iv) the ticket number (to enable monitoring in the issue-tracking system), (v) the system module (e.g., administrative, financial), (vi) the related use case, and (vii) the estimated effort. Using this information, the deadline in work hours is calculated and the intended delivery date is set in accordance with the SLA. The first column highlights the overall managerial status, showing whether each request has been delivered, cancelled or how many days are left before delivery ought to be accomplished.

Figure 5 shows the data registered in the spreadsheet for the service being processed for each request, including the chosen developer, real development effort (obtained from the issue-tracking system), actual delivery date and customer approval (validation). With this information, the spreadsheet calculates the extent of compliance of the work performance with the SLA and the amount to be billed for the service, taking into account contractual SLA penalties for late delivery, when applicable. In the fifteen day period shown in Figure 3, for instance, besides the fixed cost (regarding the internalized requirements analyst and the corrections), there are 162.2 billable hours concerning the provided services of implementing changes and eventual penalties.

The issue-tracking system used was *Assembla* (Assembla 2013), a cloud-based service system, already including the integration of tickets (requests) with version control. The integration with version control provides traceability on how the tickets were handled.



SERVICE DATA					
DEVELOPER	REAL EFFORT	DELIVERY DATE	APPROVED	SLA	COST
A	8	1/20/2013	✓	✓	8
A	16	1/21/2013	✓	✓	12
B	10	1/21/2013	✓	✓	0

Figure 5 Part of the spreadsheet used for registering the treatment of the requests.

For each ticket all the files added or changed for providing the technical solution can be explicitly identified, as well as the modifications done in each file. Therefore, if a change request on a given ticket is received, the files to be modified can be traced, supporting the involved impact analysis and effort estimation.

This system was also used for allocating the developers and registering the real effort. As soon as a developer concludes the technical solution of a request, he registers the effort and passes corresponding ticket to be tested by an independent team. If the ticket passes testing then its status is changed and it is considered ready to be deployed for customer validation, with the support from the internalized requirements analyst at the customer side.

Finally, monitoring the service status for each request was facilitated by the Kanban chart, which was physically displayed in the room where the development team was located. Figures 6 and 7 show a real example of a request in Assembla and a Kanban chart with the progress status of service requests, respectively. In this Kanban chart the status of each ticket can be easily identified. Each ticket becomes a separate post-it, using different colors reflecting the SLA priorities. To illustrate the use of the Kanban chart, Figure 7 highlights the status of ticket #560, shown in Figure 6, as being in the queue for supplier side testing.

The importance of tests in this context is obvious, since a service request is only accepted for billing if it does not show any failures during validation. Moreover, if an

The screenshot shows the Assembla web interface. At the top, there's a navigation bar with 'Assembla', 'My Start', and 'Tranship'. Below that, a header for 'Tranship' includes a search bar and user information 'Edmilson Silva'. A secondary navigation bar contains links like 'Wiki', 'Tickets', 'Milestones', 'Team', 'Stream', 'Messages', 'Files', and 'More'. Below this, there's a 'New Ticket' button and a list of tabs: 'List', 'Planner', 'Cardwall', 'Outline', 'Metrics', and 'Batch Update'. The main content area displays a ticket titled '#560 MFI - Lançar Despesas'. On the left, a sidebar shows ticket details: 'Created on 2013-01-16', 'Reported by Vagner Lopes Cunha', 'Worked hours 7.0 hours', 'Status Fixed', 'Priority Normal (3)', 'Assigned to -', 'Component MFI', 'Milestone Manutenção Evolutiva', 'Permission type None', and 'Estimate Small'. The main content area contains the ticket description, which includes a list of bullet points and a paragraph. At the bottom, there's a section for 'Activity' with links for 'Attachments' (1), 'Related Tickets' (0), 'Followers' (0), and 'Time', along with a 'Leave a comment' link.

Figure 6 Real request in the Assembla issue-tracking system.

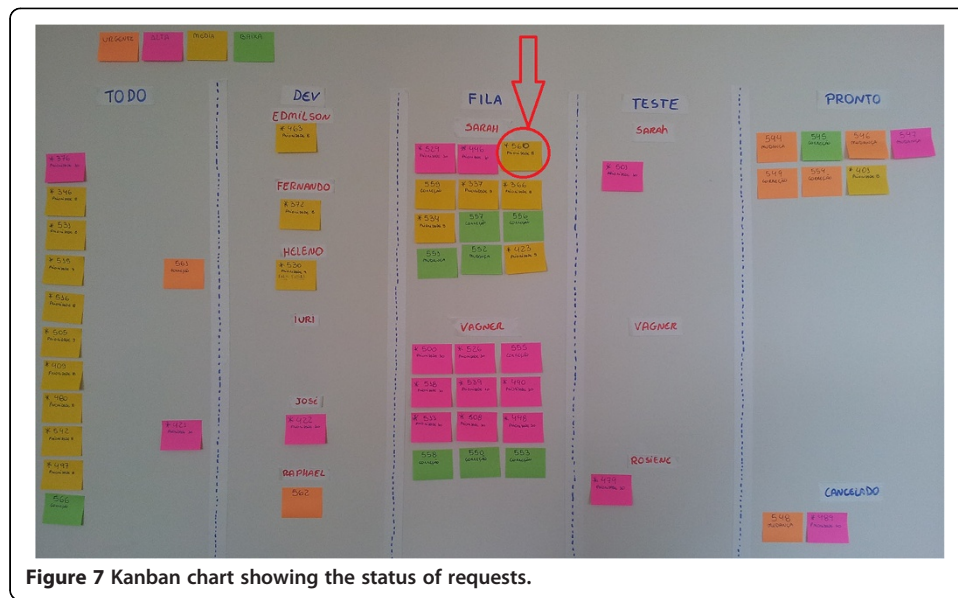


Figure 7 Kanban chart showing the status of requests.

acceptance test fails, the delivery data remains unchanged and the SLA will probably not be met, resulting in penalties. Considering this critical quality issue, requirements inspections (Kalinowski *et al.* 2007) and code peer reviews (Kemerer and Paulk 2009) were also adopted.

Having presented the contractual aspects and the internal managerial controls defined in this industrial experience, based on the guidelines of the MPS-SV service reference model, to allow switching from a project-oriented to a service-oriented software development approach, the next section presents the discussion and evaluation of the overall experience.

Discussion and evaluation

Aiming at understanding the feasibility and effects of restructuring software development from a project-oriented to a service-oriented approach guided by a service reference model, two research issues were identified in the Background Section. RI-1, related to the perceived utility, was investigated through a survey reported in detail in (Jordão and Kalinowski 2013).

Results indicated that the MPS-SW software engineering consultants consider service reference models, such as the MPS-SV, useful for providing continuous delivery capabilities to their software industry customers (93% totally or partially agreed). The consultants also indicated an expected ease of use (76% totally or partially agreed) and intention to adopt (63% totally or partially agreed). The survey's overall confidence level was of 78.5%.

These survey results can be seen as an additional motivation for investigating the second identified research issue, RI-2, which is the main contribution of this article and concerns the feasibility and results or restructuring software development guided by a service reference model. The discussion and evaluation of questions that provide further insights into RI-2, based on the reported experience, follows.

What are typical stakeholder needs that trigger the restructuring from a project-oriented approach to a service-oriented approach?

Table 2 provides an overview on the main stakeholders and their interests. From the customer's point of view, the main interests on software development were facilitating contract negotiations and integrating value considerations into the development process. Therefore, allowing faster deployment of new system capabilities prioritized according to the added business value. From the supplier's point of view, on the other hand, the main interests were related to simple contract negotiation, a steady (or growing) inflow of development requests, and translating productivity directly into higher monthly net gains. It is noteworthy that there were no specific expectations regarding changes in the cost of developing individual functionalities. Actually, given the need to comply with the SLA (and potential penalties) the customer and supplier agreed to increase the hourly cost rate.

We assume that the restructuring allowed properly addressing the stakeholder's main interests, by (i) providing a contractual framework that avoids excessive negotiation effort for new development requests, and (ii) allowing the new service requests to be directly addressed according to the added business value, as defined in the SLA. From the supplier's point of view, the variable monthly net gain also accounted for the direct return of investment of efforts to improve productivity.

The general perception from the customer and the supplier was that a more flexible model had been adopted to meet the customer's business needs and to live up to his expectations in terms of supply, as expressed in the SLA. In fact, the contract model considers the business priority of individual requests and does not have to be reconsidered due to a variation in the volume of requests, for instance.

Can a service reference model be helpful to meet those needs?

In this experience, the Brazilian MPS-SV service reference model, which is representative for similar international standards, directly supported the restructuring to the service-oriented approach. The resulting process implementation followed the guidelines of MPS-SV maturity level G (SOFTEX 2012b). The Service Delivery, Service Level Management, and Incident Management reference processes, directly related to providing services, were considered helpful to structure basic service delivery capabilities in order to meet the identified stakeholder needs and to increase the confidence in the provided solutions. The main purpose of each of those processes, their expected results according to the MPS-SV reference model, and how they were implemented during this experience are shown in Table 3. This implementation does also meet (SOFTEX 2012b) the specific goals of the CMMI-SVC (SEI 2010) Service Delivery process area (maturity level 2), and of the related ISO/IEC 20000 processes (ISO/IEC 2011).

What is the effort to be expected?

The restructuring took 160 person hours and happened within the timeframe of one month. As the restructuring included aspects of all processes at Level G of the MPS-SV model, we assert that implementing this model for software development requires only moderate effort, when compared for instance to the average duration of over 12 month for implementing MPS-SW maturity levels (Travassos and Kalinowski 2013). Especially if work management and requirements management capabilities, including requirements

Table 3 MPS-SV level G processes and how they were implemented

MPS-SV process purpose and expected results	How the expected results were implemented
Service Delivery (SD)	
<i>Purpose:</i>	
The purpose of this process is defining the strategy and establishing the service system to deliver services in conformance with the service agreements.	
<i>Required Results:</i>	
SD 1. A service delivery and operation strategy is established and maintained;	SD 1. The defined service delivery and operation strategy is shown in Figure 1.
SD 2. The availability of the needed elements for providing the service is confirmed;	SD 2. The availability of needed elements was assured by the contractual framework, which included an internalized requirements analyst at the customer side.
SD 3. The service system is put into operation to deliver the agreed services;	SD 3. The service system was put into operation.
SD 4. The service system is maintained to assure continuous service delivery.	SD 4. The service system was maintained operating.
Service Level Management (SLM)	
<i>Purpose:</i>	
The purpose of this process is to assure that the SLA goals for each customer are met.	
<i>Required Results:</i>	
SLM 1. Services and their dependencies are identified;	SLM 1. The identified service was software development, including corrective and evolutionary (new functionalities) maintenance.
SLM 2. Service level goals are defined in an SLA;	SLM 2. The defined SLA is shown in subsection 'Service Level Agreement and Penalties'.
SLM 3. Services are monitored against the SLA;	SLM 3. Monitoring was achieved by defining the service management controls depicted in Figure 2.
SLM 4. Service level performance is communicated to relevant stakeholders;	SLM 4. The request-tracking spreadsheet (Figure 3) was shared with the customer and sent monthly for billing purposes. It provides an overview of the service performance against the SLA.
SLM 5. Changes in service requirements reflect in the SLA.	SLM 5. Those changes did not happen in the context of this experience, but would imply in changing the SLA appendix of the contractual framework and both sides, supplier and customer) were open to discuss such changes.
Incident Management (IM)	
<i>Purpose:</i>	
The purpose of this process is to handle individual incidents and service requests within a SLA.	
<i>Required Results:</i>	
IM 1. A strategy for incident and service request management is established and maintained;	IM 1. The defined strategy involved fine-grained request monitoring by using the internal managerial controls shown in Figure 2.
IM 2. An incident and service request management system is established and maintained;	IM 2. The internal managerial controls were established and maintained.
IM 3. Incidents and service requests are registered and classified;	IM 3. Each request was registered into the request-tracking spreadsheet (Figure 3) and classified as an incident (correction) or a new service request (change or new functionality).
IM 4. Incidents and service requests are prioritized and analyzed;	IM 4. Each request had its priority defined and impact analysis performed, using the traceability provided by the issue-tracking system, to estimate the required effort.

Table 3 MPS-SV level G processes and how they were implemented (Continued)

IM 5. Incidents and service requests are resolved and concluded;	IM 5. The request-tracking spreadsheet shows the managerial status of each request, allowing managing them until conclusion.
IM 6. Incidents and service requests that did not progress according the SLA are communicated to higher level management;	IM 6. The request-tracking spreadsheet was shared and monthly sent to directors of both companies. It explicitly identifies the request that did not meet the SLA.
IM 7. Status information on incidents or service requests is communicated to relevant stakeholders.	IM 7. Status communication to higher management and customer stakeholders was achieved through the spreadsheet. Communication to the internal development team by using the Kanban chart.

traceability, are previously installed, as in the experience here described. Besides the moderate effort, interesting results were obtained, including increases in the productive capacity, as pointed out in the answer to the next question.

What are the potential effects on quality and productivity?

Quantitative data concerning this question was obtained by comparing the deliveries in the new service format to the most recent delivery in the previously established project format. For this comparison, only new use case developments of comparable medium complexity (according to the criteria for counting use case points: from 4 to 7 transactions including alternative steps) were considered. Table 4 summarizes these quantitative results.

Regarding quality, during customer acceptance tests of medium complexity cases, the number of failures was notably lower, falling from 0.33 failures per use case to 0.18 failures per use case. Concerning productivity, the real development effort spent per use case was also slightly lower, falling from 24 person hours per use case to 21 person hours per use case. Note that these productivity improvements directly translate into saved costs as the supplier pays for the overall development effort in person hours.

Of course, these differences are not statistically significant since only 23 medium complex use cases served as the basis for the comparison (12 implemented in the project format and 11 implemented in the IT service format). Moreover, there are several factors that can influence these results (e.g., low precision in measuring use case complexity, inherent variations of individual productivity, learning factor). However, feedback from the customer also allowed observing a perceived improvement in quality (fewer problems during acceptance tests) and productivity, especially regarding the fact that the new service-oriented value-based approach resulted in faster delivery of demands of higher added business value.

An informal causal analysis session (Kalinowski *et al.* 2008b) (Kalinowski *et al.* 2012), aiming at investigating the causes of these changes in quality and productivity, was conducted with the suppliers development team and showed that the existence of the SLA and closer management resulted in greater commitment on the part of the developers.

Table 4 Comparison of the project and IT service delivery approaches

Approach	# of use cases	# Failures (Acceptance)	Effort (Person-hours)
Project	12	4	284
IT Service	11	2	232

Some developers mentioned that, knowing of the individual penalties that the company would be liable to pay in each request, they saw that it was much more important to meet deadlines than they did when the project format was in use, in which the tasks were scheduled but the overall deadline was scoped to the entire project. Indeed, closer management and finer granularity are suggestions included in Humphrey's reflections on efficient team management (Humphrey 2010).

What are the main lessons learned?

In the reported experience, applying practices of service management to software development resulted in a flexible contractual framework, allowing addressing the customer's business needs according to their priorities without additional contract negotiation delays. New requests could be directly Among the lessons learned, that could be passed on to international SME companies that seek to supply development as an IT service guided by a service reference model, the following stand out:

- **Contractual framework.** In this supply model, it is important to have some headroom in the budget for the fixed monthly part of the contractual framework, compensating for possible variations in the volume of delivered requests to be billed in the variable part. Otherwise, financial constraints may lead to excessive pressure, on management and development levels, at the supplier's side. After all, the development team has to be paid by the supplier anyway, although if the supplier has different customers following similar contractual service frameworks he might be able to allocate developers to his project portfolio according to the individual project's volume of received demands.
- **Prioritization criteria.** The priority criteria for customer requests should be very clear, as it might be applied when it comes to avoiding undue financial penalties, as specified in the SLA. After all, ambiguities involving financial aspects may lead to potential relationship problems between the contractor and the supplier.
- **SLA and team capacity.** It is important to assess the SLA carefully, checking whether the company has the installed productive capacity of actually satisfying the requested service level. If not, this would result in deliveries not meeting the SLA and the application of penalties with direct impact on the monthly net gain.
- **Managerial controls.** The managerial controls were fundamental to allow handling the demands as separate service requests, by facilitate monitoring the compliance with the SLA. Not having such controls established may imply in several request passing the SLA's deadline and, as a consequence, in penalties.
- **Traceability.** Traceability plays a key role when handling request as separate services to be provided, by supporting effective impact analysis, effort estimation and risk assessments for handling each new request.
- **Build, Test and Deployment Automation.** Aiming at continuous delivery, efficient build, test and deployment automation is strongly desired (Humble and Farley 2010). In the case of our experience the build and deployment were fully automated. Test automation however could be improved to reduce regression testing effort. So far, only superficial smoke tests had been recorded to allow identifying major side effects of new handled request, by using the Selenium IDE plugin (Selenium 2013).

- **Team and individual benefits.** Some employees felt pressured by the SLA and the close and quantitative monitoring. Therefore, it might be interesting to establish a complementary policy for awarding employees with bonuses for improved productivity. Note that this is feasible, since an improvement in productivity also implies in a higher monthly net gain.
- **Overall restructuring effort and benefits.** Around one person month effort was required for the restructuring and benefits were perceived in terms of improved quality, productivity, and customer satisfaction. We take it that the increased customer satisfaction was mainly related to meeting their main interests by considering the integration of added business value into the software engineering practices.

What are the involved success and risk factors?

Another fruitful consequence of the experience was the opportunity to identify success (and risk) factors related to restructuring software development in the format of service delivery. Based on the lessons learned from the experience report, the main success factors were:

- **Success factor service reference model.** The adoption of a service reference model allowed benchmarking the new service format against IT service delivery best practices and improved the confidence in the solutions provided by the restructuring approach. These solutions include the internal managerial controls to facilitate monitoring the compliance with the SLA.
- **Success factor senior management support.** The idea of the restructuring addressed specific interests of stakeholders including the directors of both companies. Therefore, direct support from senior management was obtained for the process improvement initiative. We saw on several occasions that this support has considerably facilitated and accelerated the restructuring.
- **Success factor relationship of trust.** The previous period of over one year providing development services successfully allowed establishing a relationship of trust between the contractor and the supplier. This relationship provided the basis to discuss the new contractual framework for continuous delivery with varying monthly net payments and gains.
- **Success factor previously established capabilities.** Some previously established software engineering capabilities were crucial in the successful transition. Concrete requirement traceability capacity, for instance, helps to handle request separately by allowing effective impact analysis, effort estimation and risk assessments for handling each new request. Moreover, build, test and deployment automation was also extremely helpful for implementing the continuous delivery strategy.

Concerning the risk factors, we see a direct mapping, in the sense that the absence of any of the success factors represents a significant risk. Additional risks can also be identified from not addressing major issues in the the lessons learned.

Conclusions

This article reported on an experience of restructuring software development in the form of providing IT services in the context of an SME software supplier. The

restructuring involved, in addition to the technical changes, important management changes, i.e., establishing a contractual framework and capacities for managing service requests in order to comply with a Service Level Agreement (SLA) made between the supplier and the customer to satisfy the business needs of the latter.

The restructuring project was guided by the MPS-SV service reference model, which is compatible to the CMMI-SVC reference model, complementing already established software engineering practices for the provision of development as a continuous IT service. The contractual aspects and internal managerial controls employed to facilitate the compliance with the SLA were described. Those controls included the integrated use of a managerial spreadsheet, an issue-tracking system, and a Kanban chart to monitor how demands were prioritized to be met.

Further insight into the feasibility and results on such structuring were provided by discussing valuable and experience-grounded answers to the following core questions: What are typical stakeholder needs that trigger such a restructuring? Can a service reference model be helpful to meet those needs? What effort is to be expected? What are the potential effects on development quality and productivity? What are the main lessons learned? What are the involved success and risk factors?

The overall analysis of the experience showed that only moderate effort, around one person month, was required for structuring software development as a service guided by a reference model and that perceived benefits were obtained in terms of quality, productivity, and customer satisfaction. The increased customer satisfaction was mainly related to meeting the customer's business needs by integrating value considerations into software engineering practices. Therefore, using a service reference model for restructuring software development can represent an alternative path towards value-based software engineering. Nevertheless, we would like to reinforce that, as expected in an experience report, the obtained results relate to a specific scenario and industrial environment. However, those results provide preliminary indications into feasibility and potential benefits, further motivating the conduct of more rigorous studies (e.g., case studies or controlled experiments) on the impact of applying service reference models to re-structure software development from a project-oriented to a service-oriented management format.

A key benefit of the Brazilian standards is the provision of smaller steps in the lower range of the process maturity levels, which allows, in particular, small companies with limited resources to systematically define and achieve software and service maturity levels that suit their needs and means. This report and the identified lessons learned can serve as a reference for SME companies that operate in the context of an international software maturity reference model and wish to restructure in order to supply software development in the format of an IT service. These companies can benefit from adopting a service reference model, such as the MPS-SV or CMMI-SVC models, in synergy with their already established software engineering practices.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MK was the executive director of the supplier company and worked directly with the customer on establishing the contractual framework and the service level agreement. He also worked with the supplier team to enable the restructuring in the format of the service-oriented approach. He was the lead author of the initial report and of this extended paper. SB helped to extend the initial report, discussing the overall solution and argumentation line and

enriching it with his strong scientific stakeholder analysis and value-based software engineering background. ROS was the operations director of the supplier and helped extending the report with additional details of the overall experience. SR is the main responsible person for the MPS-SV reference model and used her service quality background to help writing the initial report and to support the extension with the argumentation on the international standards and reference models. All authors read and approved the final manuscript.

Authors' information

MK holds a MS and PhD in Computer and Systems Engineering from the Federal University of Rio de Janeiro (UFRJ) and is currently professor of Software Engineering at the Federal University of Juiz de Fora (UFJF) in transference to Fluminense Federal University (UFF). He is part of the MPS.BR project team and a certified MPS.BR lead appraiser and process implementation consultant.

SB holds a MS and PhD in Computer Science from the Vienna University of Technology (TUW) and a MS in Social and Economic Sciences from the University of Vienna. He also received a Habilitation degree (Venia Docendi) in "Praktische Informatik" for his contributions on empirical software engineering. Currently he is professor of Software Engineering at TUW and the head of the Christian Doppler research laboratory – Software Engineering Integration for Flexible Automation Systems. He was lead editor for the book "Value-Based Software Engineering" in collaboration with co-editors A. Aurum, B.W. Boehm, H. Erdogmus, and P. Grünbacher.

ROS holds a MS and PhD in Computer and Systems Engineering from the Federal University of Rio de Janeiro (UFRJ) and is currently professor of Software Engineering at the University of Salvador (UNIFACS). He is a certified MPS.BR process implementation consultant.

SR holds a MS in Informatics from the Catholic University of the State of Paraná (PUCPR) and a PhD in Production Engineering from the University of São Paulo (USP) and is currently professor of Software Engineering at PUCPR. She is the coordinator of the service area of the MPS.BR project team and a certified MPS.BR lead appraiser and process implementation consultant.

Acknowledgements

We would like to thank CNPq (Brazilian Research Council) for financial support. Thanks also to everyone at Tranship Transportes Marítimos and Kali Software directly involved in the experience reported in this article, especially Rosiene Dilly (the project manager) and Vagner Lopes (the internalized requirements analyst). This research was in part supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria.

Author details

¹Federal University of Juiz de Fora, Rua José Kelmer s/n, Juiz de Fora 36.036-330, Brazil. ²Institute of Software Technology and Interactive Systems, CDL-Flex-, Vienna University of Technology, Favoritenstr. 9/188, Vienna 1040, Austria. ³University of Salvador, Rua Doutor José Peroba 251, Salvador 41.770-235, Brazil. ⁴Catholic University of the State of Paraná, Rua Imaculada Conceição 1155, Curitiba 80.215-901, Brazil.

Received: 9 October 2013 Accepted: 5 July 2014

Published: 30 August 2014

References

- Anderson DJ (2010) Kanban: successful evolutionary change for your technology business. Blue Hole Press, Sequim, WA
- Assembla (2013) Assembla task & issue management., <http://www.assembla.com>. Accessed 29 Sep 2013
- Barney S, Aurum A, Wohlin C (2008) A product management challenge: creating software product value through requirements selection. *J Syst Architecture* 54(6):576–593
- Biffi S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (eds) (2006) Value-based software engineering. Springer, Heidelberg
- Boehm B (2006) Value-based software engineering: overview and agenda. In: Biffi S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (eds) Value-based software engineering. Springer, Heidelberg
- Boehm B, Turner R (2003) Balancing agility and discipline: a guide for the perplexed. Wesley, Addison
- Davis F (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly* 13(3):319–340
- Humble J, Farley D (2010) Continuous delivery: reliable software releases through build, test, and deployment automation. Addison-Wesley, Boston, MA
- Humphrey WS (2010) Reflections on management: How to manage your software projects, your teams, your boss, and yourself. Addison-Wesley, Boston, MA
- ISO/IEC (2004) ISO/IEC 15504–1: information technology - process assessment – part 1 - concepts and vocabulary. ISO, Geneve
- ISO/IEC (2011) ISO/IEC 20000–1:2011 – information technology service management. ISO, Geneve
- Jordão L, Kalinowski M (2013) Investigando a Aplicabilidade do MPS-SV na Melhoria de Serviços de Desenvolvimento e Manutenção de Software. In: IX Workshop Anual do MPS.BR (WAMPS). Brazilian Computer Society (in Portuguese), Campinas, Brazil
- Kalinowski M, Reinehr S (2013) Estruturando Desenvolvimento de Software como um Serviço de TI: Uma Experiência Prática. In: XII Simpósio Brasileiro de Qualidade Software (SBQS). Brazilian Computer Society (in Portuguese), Salvador, Brazil
- Kalinowski M, Spinola RO, Dias-Neto AC, Bott A, Travassos GH (2007) Inspeções de requisitos de software em desenvolvimento incremental: Uma experiência prática. In: VI simpósio brasileiro de qualidade software (SBQS). Brazilian Computer Society (in Portuguese), Porto de Galinhas, Brazil
- Kalinowski M, Weber KC, Travassos GH (2008a) IMPS: an experimentation based investigation of a nationwide software development reference model. In: International symposium on empirical software engineering and measurement (ESEM). ACM and IEEE, Kaiserslautern, Germany

- Kalinowski M, Travassos GH, Card DN (2008b) Guidance for efficiently implementing defect causal analysis. In: VII Brazilian symposium on software quality (SBQS). Brazilian Computer Society (in Portuguese), Florianópolis, Brazil
- Kalinowski M, Santos G, Reinehr S, Montoni M, Rocha AR, Weber KC, Travassos GH (2010) MPS.BR: Promovendo a Adoção de Boas Práticas de Engenharia de Software pela Indústria Brasileira. In: XIII Congreso Iberoamericano en "Software Engineering" (CIBSE). Universidad del Azuay (in Portuguese), Cuenca, Ecuador, ISBN 978-9978-325-10-0
- Kalinowski M, Card DN, Travassos GH (2012) Evidence-based guidelines to defect causal analysis. *IEEE Software* 29(4):16–18, doi:10.1109/MS.2012.72
- Kemerer CF, Paulk MC (2009) The impact of design and code reviews on software quality: an empirical study based on PSP data. *IEEE Trans Softw Eng* 35(4):534–550
- Khan SU, Niazi M, Ahmad R (2011) Factors influencing clients in the selection of offshore software outsourcing vendors: an exploratory study using a systematic literature review. *J Syst Softw* 84(4):686–699
- Lehman TJ, Sharma A (2011) Software development as a service: agile experiences. In: Annual SRIL global conference (SRIL). IEEE, San Diego, USA
- MCTI (2013) Brazilian ministry of science, technology and innovation., <http://www.mcti.gov.br>. Accessed 29 Sep 2013
- Penn ML (2011) Applying CMMI-SVC process areas to CMMI-DEV projects. Presentation at SEPG North America 2011, available via CMMI Institute., <http://cmmiinstitute.com/resources>. Accessed 29 Sep 2013
- Penn ML (2013) Harmonization of CMMI-SVC and CMMI-DEV, Presentation at SEPG North America 2013, available via SEPG., <http://sepgconference.org/harmonization-of-cmmi-svc-and-cmmi-dev>. Accessed 29 Sep 2013
- PMI (2013) A guide to the project management body of knowledge, 5th edn. Project Management Institute, Newtown Square, PA
- Santos G, Kalinowski M, Rocha AR, Travassos GH, Weber KC, Antonioni JA (2012) MPS.BR program and MPS model: main results, benefits and beneficiaries of software process improvement in Brazil. In: International Conference on the Quality in Information and Communications Technology (QUATIC). IEEE, Lisbon, Portugal
- SEI (2010) CMMI for services, version 1.3 (CMU/SEI-2010-TR-034), available via Software Engineering Institute, Carnegie Mellon University., <http://www.sei.cmu.edu/library/abstracts/reports/10tr034.cfm>. Accessed 29 Sep 2013
- Selenium (2013) Selenium IDE., <http://docs.seleniumhq.org/projects/ide>. Accessed 29 Sep 2013
- SOFTEX (2012a) MR-MPS-SW – guia geral MPS de software., Available via SOFTEX. <http://www.softex.br/mpsbr>. Accessed 29 Sep 2013
- SOFTEX (2012b) MR-MPS-SV – guia geral MPS de serviços., Available via SOFTEX. <http://www.softex.br/mpsbr>. Accessed 29 Sep 2013
- SOFTEX (2014) SOFTEX MPS.BR program website., <http://www.softex.br/mpsbr>. Accessed 06 May 2014
- Travassos GH, Kalinowski M (2013) IMPS 2012: evidências sobre o desempenho das empresas que adotaram o modelo MPS-SW desde 2008. SOFTEX, Campinas, English edition: Travassos GH, Kalinowski M (2013) iMPS 2012: Evidence on Performance of Organizations that Adopted the MPS-SW Model since 2008 (trans: Kalinowski M). SOFTEX, Campinas
- TSO (2011) ITIL – information technology infrastructure library v2011, Available via TSO – the stationery office., <http://www.itil.org.uk>. Accessed 29 Sep 2013
- Turner M, Kitchenham B, Brereton P, Charters S, Budgen D (2010) Does the technology acceptance model predict actual use? a systematic literature review. *Inf Softw Technol* 52:463–479

doi:10.1186/s40411-014-0010-x

Cite this article as: Kalinowski et al.: From project-oriented to service-oriented software development: an industrial experience guided by a service reference model. *Journal of Software Engineering Research and Development* 2014 2:10.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com