Performance Management Work

DOI 10.1007/s12599-014-0323-7

The Authors

Andreas Brunnert, M.Sc. (✉)
Christian Vögele, M.Sc.
Alexandru Danciu, M.Sc.
Dipl.-Inf. Matthias Pfaff
fortiss GmbH, An-Institut Technische Universität München
Guerickestr. 25
80805 München
Germany
brunnert@fortiss.org
voegele@fortiss.org
danciu@fortiss.org
pfaff@fortiss.org

Dr. Manuel Mayer
Prof. Dr. Helmut Krcmar
Chair for Information Systems
Technische Universität München
Boltzmannstr. 3
85748 Garching
Germany
mayer@in.tum.de
krcmar@in.tum.de

Received: 2014-02-11
Accepted: 2014-03-14
Published online: 2014-04-11

This article is also available in German in print and via http://www.wirtschaftsinformatik.de: Brunnert A, Vögele C, Danciu A, Pfaff M, Mayer M, Krcmar H (2014) Performance Management Work. WIRTSCHAFTSINFORMATIK. doi: 10.1007/s11576-014-0414-6.

1 Performance Management Work as a Continuous Task

Performance is a key quality factor of application systems (AS). AS performance is quantified by the metrics response time, resource utilization, and throughput (Becker et al. 2013). To guarantee AS performance, it is important to define quantifiable performance goals using performance metrics. These metrics have to be continuously measured and evaluated. Based on these metrics, activities can be defined to ensure that performance goals are met. The coordination and execution of all activities required to achieve performance goals during system development are described by the term software performance engineering (SPE) (Woodside et al. 2007). Corresponding activities during operation are typically referred to as application performance management (APM) (Menascé 2002). An isolated consideration of SPE and APM neglects their interrelation. The combination of SPE and APM activities is therefore summarized by the term performance management work (PMW). PMW is becoming a growing challenge due to developments in the areas of AS architecture, IT governance, and system life cycle.

AS architectures have evolved over time from monolithic to distributed to system of systems architectures (Jamshidi 2011). The spatial, organizational, cultural and technical diversity of system of systems architectures increases the difficulty of PMW activities (Grabski et al. 2007). Since different AS subsystems[1] are associated with different organizations, this architectural style also implies a change from a uni- to a multilateral IT governance, thereby making it necessary to coordinate PMW activities across multiple subsystems and organizations.

The subsystem life cycle is driven by functional enhancements and maintenance efforts. A subsystem life cycle is defined by a continuous iteration from system development to operation. Subsystems can be in different life cycle phases at any one point in time. However, life cycles need to be synchronized and a key challenge to achieving this synchronization is the pursuit of different goals by development and operation teams. Development teams try to realize new functionalities with high quality requirements as quickly as possible. Operation teams, on the other hand, are more interested in keeping their environments in a stable state. The term “DevOps” denotes concepts to better combine and integrate efforts in both life cycle phases (Humble and Molesky 2011). DevOps concepts can only ensure that performance goals are met if corresponding activities are closely interlinked.

Thus, there is necessity to coordinate PMW activities across organizations and life cycles. The current state of the art of PMW activities does not support such global coordination. Whenever new PMW activities or tools are introduced, they are usually concerned with ensuring the performance for certain AS architectures or within specific life cycle phases (Becker et al. 2013). The business and information systems engineering community should extend the existing research by a process view that supports a comprehensive coordination of PMW activities.

2 Performance Management Work Activities

PMW activities can be categorized according to the performance goals they support during system development and operation.

2.1 Performance Management Work During System Development

During system development, performance goals are to ensure that given non-functional requirements, such as the scalability of an AS architecture, are met. Non-functional performance requirements are often specified by maximum response times for specific transactions. The scalability, in particular, is specified by the flexible adaption of an AS architecture to different user counts and the required throughput. In order to ensure that these performance goals are met, different activities are combined to collect the required metrics and to derive and realize optimizations based on these metrics.

Load and performance (L&P) tests are often executed at the end of the system development process. The resulting performance metrics describe an AS in its current state. The representativeness

[1] For reasons of readability, we will hereafter refer to AS subsystems as subsystems.

of the collected performance metrics depends on whether or not a test system is comparable to a corresponding production environment. Executing representative L&P tests is a huge challenge in practice because the organizational separation of subsystems makes it difficult to access representative instances of dependent systems.

For detailed performance analysis in the early phases of the system development process, activities such as code analysis, profiling, or an instrumentation of the source code are used. The validity of performance metrics collected using these activities is often limited because only subsystems can be analyzed which often have different configurations compared to their target environments.

The activities presented so far are combined in the SPE methodology (Woodside et al. 2007). Additionally, SPE supports system development by introducing performance models. Performance models can predict the performance of a system based on its software designs. To improve the predictions, these models can be enhanced with performance metrics collected during the system development process. Performance models are not yet in widespread use in industrial practice (Koziolek 2010) because the effort of modeling currently outweighs its benefits.

2.2 Performance Management Work During Operation

The primary performance goal during the operation phase of an AS is to ensure that service-level agreements (SLA) are met. SLAs can be specified by any combination of the performance metrics response time, throughput, and resource utilization. Monitoring systems are used to continuously collect these metrics. These systems allow operations staff to get an up-to-date view of the current situation and to evaluate if SLAs are met.

Furthermore, new systems are introduced that automatically analyze performance metrics collected by monitoring systems to reconfigure AS before SLA violations occur. An example for such systems is the dynamic resource allocation in virtualized environments (i.e. cloud infrastructures). The use of such systems can increase the flexibly of organizations while providing new applications and services. Should SLA violations occur, new soft- and/or hardware can be added to an AS.

Not all AS architectures can be scaled elastically (Vaquero et al. 2011). Moreover, virtualization cannot guarantee that an AS behaves consistently over a period of use. The reason for these behavior differences is the concurrent access of multiple AS to shared IT resources. Therefore, one of the main research directions in the performance field is to explore approaches that improve the dynamic resource allocation. Other important topics in this research area are scalable AS architectures and runtime prediction models (Becker et al. 2013).

Another goal of PMW activities in the operation phase is the coordination and control of continuous changes introduced into production systems. It is essential to evaluate the performance impact of any alternations (i.e. hard- or software changes) before they go into production. Since larger changes are often carried out in separate change projects, all activities mentioned in the system development phase are of relevance.

3 Future Developments, Capabilities, and Application Areas

A look at existing approaches reveals that these individual activities need to be integrated to meet performance goals. If performance is not considered during system development, it can also not be guaranteed during operation. Additionally, experience from the operation phase is necessary for making informed performance predictions. This is especially the case in early system development phases. A process-oriented view, which combines all activities required to fulfill performance goals, is still missing. The following sections, therefore, present integration options of PMW activities from the AS architecture, IT governance, and system life cycle perspectives.

3.1 Integrating Individual Activities

To integrate PMW activities from the AS architecture and IT governance perspectives a mapping of subsystems and PMW activities to organizational units is necessary. In the context of cross-organizational IT value-added chains, possibilities need to be investigated to coordinate and integrate PMW activities of different organizations. A basic requirement for such integration is to enable the interchange of performance metrics across subsystems (Schmietendorf 2001). To simplify this exchange, independent methods and tools need to be combined from a technical as well as an organizational perspective. The results of research in this area are environments and process models for monitoring, analysis, and optimization of system of systems architectures.

Integrating the system life cycle and AS architecture perspectives supports PMW activities from the requirements phase to the operation phase. In order to achieve this goal, approaches for designing the transition between life cycle phases need to be identified. Storing and transferring information between different life cycle phases is a considerable challenge; the feedback cycle between these phases should be automated in an effort to address this challenge.

The integration of the IT governance and system life cycle perspectives addresses the organizational framework for PMW activities. It is important to determine which competences are required for this integration in organizations. A new competence profile should be defined that addresses the processes and tools to ensure that performance goals can be met. As performance is a key quality factor of AS, an integration of this competence profile into the European e-Competence Framework should be attempted (EU 2013). Additionally, an investigation should be undertaken as to how the rights and responsibilities of different organizational units can be represented throughout the system life cycle and how PMW activities can be integrated into the IT service management of an organization.

3.2 Capabilities and Application Areas

An increased integration of PMW activities creates new application areas. An example is the description of the resource requirements of AS. Such resource descriptions help to refine accounting models for internal and external IT providers (Brandl et al. 2007). Thus, hardware, energy, licensing, and administration costs can be allocated to the organizational units creating these costs. Additionally, transparency of the resource demands helps to reduce these costs in total. Hence, integrated PMW activities support cross-organizational investment and purchasing decisions for complex system of systems architectures. A transparency of performance metrics for different vendors also simplifies the selection of cloud and other service providers.

Overall, a better integration of PMW activities increases the transparency of bottlenecks in the IT value-added chain. As soon as performance metrics are available across organizations, the local optimization of subsystem performance can be replaced by a global optimization of an AS. AS planning should be dealt with in a cross-organizational manner, as is the case in traditional value-added chains.

Increasing energy costs will further strengthen green IT initiatives. From an energy perspective the current focus is on increasing the efficiency of hardware and cooling. As the software running on the hardware influences the IT resource and the resulting energy demand, a stronger focus on the energy efficiency of software is inevitable and needs to be integrated into the acceptance process. The transparency of performance metrics due to an increased integration of PMW activities therefore contributes to a reduction of the energy demand in data centers. As a result, PMW ensures the environmental friendliness of AS and prepares the IT for its way into a more efficient future.

References

Becker S, Hasselbring W, van Hoorn A, Reussner R (eds) (2013) Proceedings of the symposium on software performance: joint Kieker/Palladio days. CEUR workshop proceedings, vol 1083, Karlsruhe, CEUR-WS.org

Brandl R, Bichler M, Ströbel M (2007) Cost accounting for shared it infrastructures – estimating resource utilization in distributed it architectures. WIRTSCHAFTSINFORMATIK 49:83–94

EU (2013) European e-competence framework version 2.0. http://www.ecompetences.eu/

Grabski B, Günther S, Herden S, Krüger L, Rautenstrauch C, Zwanziger A (2007) Very large business applications. Informatik-Spektrum 30(4):259–263

Humble J, Molesky J (2011) Why enterprises must adopt DevOps to enable continuous delivery. Cutter IT Journal 24(8):6–12

Jamshidi M (2011) System of systems engineering: innovations for the twenty-first century. Wiley series in systems engineering and management, vol 58. Wiley, New York

Koziolek H (2010) Performance evaluation of component-based software systems: a survey. Performance Evaluation 67(8):634–658

Menascé DA (2002) Load testing, benchmarking, and application performance management for the web. In: Proceedings of the computer measurement group conference (CMG), Reno, pp 271–282

Schmietendorf A (2001) Prozess-Konzepte zur Gewährleistung des Software-Performance-Engineerings in großen IT-Organisationen. In: Dumke R (ed) Magdeburger Schriften zum empirischen Software Engineering. Shaker Verlag, Aachen

Vaquero LM, Rodero-Merino L, Buyya R (2011) Dynamically scaling applications in the cloud. Computer Communication Review 41(1):45–52

Woodside M, Franks G, Petriu DC (2007) The future of software performance engineering. In: Future of software engineering (FOSE). IEEE Computer Society, Washington, pp 171–187