Quality Considerations in SAP Architectures [1]

Wolfgang Theilmann*, Roger Kilian-Kehr*

* SAP Research, CEC Karlsruhe, Vincenz-Prießnitz-Str. 1,
76131 Karlsruhe, Germany
{wolfgang.theilmann, roger.kilian-kehr}@sap,com

Abstract. This paper provides an industrial perspective on the relationship between architectural issues and quality concerns in SAP. We show that quality concerns matter for different kinds of architectures addressing different perspectives on SAP systems such as technology, business logic and business view. Furthermore, the system lifecycle also plays an important role as architectures and kind of quality questions evolve along this lifecycle. We derive 3 key requirements for future system architectures and the modeling of quality characteristics: (1) the ability to deal with underspecified environments, (2) the embedding into the development process and (3) joining of programming models with architectures.

Keywords: SAP architectures, system lifecycle, quality concerns, performance, programming models

1 Introduction

Quality concerns have always played an important role in the design of business applications at SAP [1]. However, current cost pressure in IT industries has lead to an even more important role of quality considerations in the development process and the design of SAP architectures.

The costs of IT systems are mainly determined by 3 factors: cost of engineering, cost of provisioning and cost of operation. Exploiting economies of scale significant further cost reduction can be achieved by increasing the effort in developing solutions of high quality which then allow for provisioning and operation at lower costs. Of course, this only works out if there is a positive tradeoff between the additional effort and the saved provisioning/operation costs.

This paper provides an industrial perspective on the relationship between architectural issues and quality concerns in SAP. This is done in the following steps. Section 2 introduces the solution lifecycle and shows when and by whom quality concerns are dealt with. Section 3 explains some key characteristics of SAP architectures and how these address quality requirements. Section 4 presents a complexity assessment for different quality and sketches specific requirements for

[1] The research leading to these results is partially supported by the European Community's Seventh Framework Programme ([FP7/2001-2013]) under grant agreement n° 216556

efficiency issues. Last, Section 5 concludes with a summary and some key requirements for future system architectures and the modeling of quality characteristics.

2 Solution Lifecycle

Quality concerns need to be dealt with along the complete lifecycle of a solution. The following figure addresses the main phases and steps of this lifecycle at SAP.

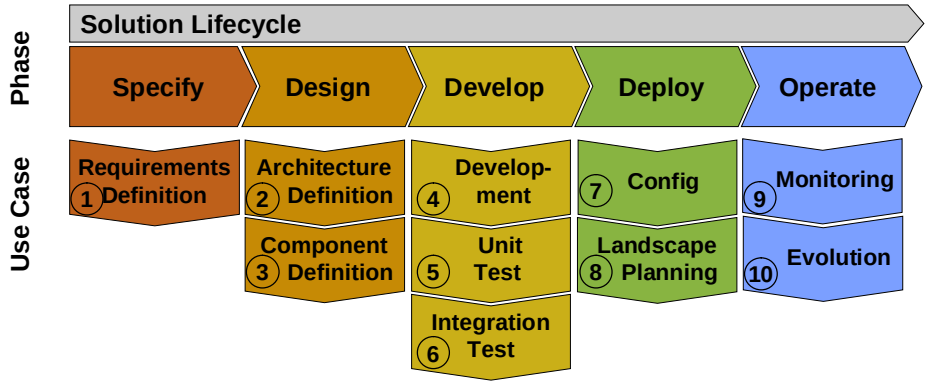

Fig. 1. Basic *solution lifecycle* at SAP (cycles, feedback loops and partial new development not shown in this figure).

Along this lifecycle the following main roles deal with the following issues:

1. Solution Manager provides estimated quality KPIs.
2. System architect specifies assumed characteristics of building blocks.
3. Component Designer relates requested quality with characteristics of underlying components.
4. Developer refines previously assumed quality characteristics,
5. Developer derives actual quality characteristics from unit tests.
6. Tester validates requirements from solution manager.
7. Consultant/Customer specifies actual artifacts to be used (=> first TCO estimate).
8. Consultant/Customer specifies actual landscape (=> final TCO determination).
9. Administrator observes system whether quality targets are met,
10. Consultant/Customer explores impact of planned changes.

Noteworthy, that this lifecycle is a quite simplified abstraction in 2 ways: First, the actual solution lifecycle includes various cycles and feedback loops which allow for an iterative development process that can benefit from early prototypes and tests and feed their results back into earlier stages of the lifecycle. Second, actual development almost never starts from scratch but faces an already existing system which needs to be modified or extended. This obviously limits the freedom but also the uncertainty for all the involved stakeholders.

3 Characteristics of SAP Systems

SAP systems range from midsize to very large size systems, the latter consisting of several hundred million lines of code and deployed on hundreds of distributed compute nodes. In addition to this sheer size, there are a couple of other relevant characteristics which are briefly sketched below.

Several kinds of Architectures. There is no single architecture of an SAP system that covers all perspectives of the relevant stakeholders. Specific architectures are

- the *technology platform* [2], providing generic IT platform and integration functionality,
- the *business process platform*, providing general business logic which can be flexibly used, combined and composed,
- *applications architectures*, representing actual and possibly customer or industry specific solutions,
- *service architectures*, embedding an IT solution into a bigger business service solution, and
- *system landscapes*, describing the actual infrastructure that operates an IT solution and its configuration.

All these architectures provide a different view on different quality concerns for different stakeholders and at different granularity. A proper development process must ensure that information flows correctly between these different perspectives and overall quality concerns can be properly managed and achieved.

System (Development) Paradigms. SAP has adopted the paradigm of service-orientation for developing and providing business functionality. The so-called Enterprise SOA approach [3] goes far beyond regular Web services as Enterprise services feature clear business semantics (they are structured according to a harmonized enterprise model based on business objects, process components, and global data types), quality and stability (they safeguard a stable interface for future versions) and adherence to standards (they are based on open standards such as WSDL and UN/CEFACT CCTS).

Furthermore, a sound model-based development approach has been chosen which provides multiple rich models for both business (integration scenarios, process components) and IT perspective (business logic, integration logic, configuration).

Customer Engagement. Customers are involved in the specification of new SAP solutions already in the very first phases of a solution lifecycle. This co-development serves for creating solutions which eventually meet market needs in terms of functionality but also quality requirements. However, the (quality) requirements and environment of specific customers are largely unknown at design time. This significantly increases the difficulty at design time to predict quality properties and costs of only vaguely known target environments. Consequently, traditional SAP systems require thorough go-live check at a customer site before they can become operational.

Quality Concerns @SAP. Following, we sketch how the most important quality aspects are addressed in SAP.

Scalability is clearly a cross-cutting concern that spans across all kinds of architectures. However, scalability is largely solved by the technology platform which allows for linear scalability of the application server cluster.

Availability is largely achieved in the technology platform by a fault tolerant setup of application server & database.

Responsiveness (response time) is again a cross-cutting concern. There is a global requirement of 1-2s maximal response time for interactive applications. This requirement is broken down via budgeting to architectural layers and components.

Efficiency (resource consumption) is another cross-cutting concern. A sizing formula [4] allows relating usage profile per applications with resource demands. There are currently running efforts to build a sizing repository (for components). However, actual customer requirements for efficiency can be very specific and SMEs in particular are very cost-sensitive. This motivates current research efforts for multi tenancy support, i.e. resource sharing between different customers/tenants.

Extensibility (as part of maintainability) is mainly addressed at the business process platform. Here a dedicated framework part of the architecture assures various degrees of seamless extensibility.

Portability is solved by the technology platform by avoiding usage of any hardware/OS/DB-specific features.

Other quality aspects such as usability and security are mainly addressed by development guidelines and do not directly reflect in any architecture.

4 Complexity & Challenges

The following figure qualitatively summarizes the complexity of the various quality concerns at SAP in terms of architectural complexity (i.e. how many architectures are involved) and process complexity (i.e. how many stakeholders and lifecycle phases are involved).

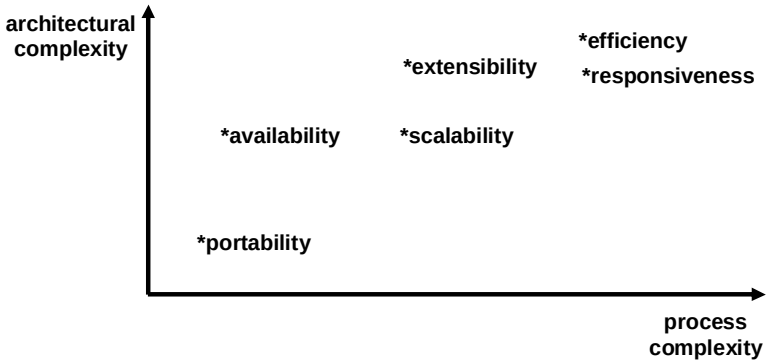

Fig. 2. *Architecural and process complexity* of quality concerns at SAP.

As illustrating example, we sketch some specific challenges for managing efficiency.

First, there is the underspecified environments which means that (a) concrete deployment and infrastructure (hardware, DB, OS) are unknown at design time, (b) customer requirements/behaviour unknown at design time and still underspecified at go-live time, (c) actual control flow is known vaguely (at design time) and slightly better (at testing time – scenario-based testing) again better after business configuration and even better at run-time, (d) component developers are focussed on one architectural layer while non-functional characteristics of lower layers are only vaguely specified and subject to change, (e) the number of configuration & usage variants prevents from exhaustive testing and (f) scenarios of dynamic service composition are even harder to predict.

Second, the various architectures and programming models are just loosely coupled which means that no formal/provable relationship between architecture models and programming artefacts exists. It is currently unclear whether a closer coupling is feasible at all with general purpose programming languages.

Third, technical expertise on non-functional behaviour of artefacts is widely spread and poorly formalized, so it's hard from an overall perspective to say who knows/does what and when.

5 Conclusions

Quality issues play in increasing role for SAP systems in order to allow for more cost effective provisioning and operation of these. Facing the size and complexity of SAP systems they are extremely hard to properly manage. Future software/system architectures and their associated models could play an important role for better management of these quality issues. In order to realize that the following key requirements need to be solved:

- Specification/prediction of quality aspects must be supported in underspecified environments (e.g. infrastructure, service composition, usage profile unknown). The decoupling of the roles of software and service provider (only latter one knows actual execution environment, customer requirements and service wiring) must be acknowledged. Last, quality characteristics in flexible service composition need better support.
- Adequate quality management requires joining programming models and architectures where abstractions on the one side meet with abstractions on the other.
- Quality management requires deep embedding into the development process with clear specification who knows/does what and when. Adhoc solutions do not scale for large organisations/systems.

However, all these requirements and possible related measures need a careful assessment of the tradeoff between required additional engineering effort vs. saved provisioning and operation costs. For having a business case a clear return on investment (ROI) needs to be specified.

References

1. SAP Web-Site, http://www.sap.com
2. SAP NetWeaver platform, https://www.sdn.sap.com/irj/sdn/nw-products
3. SAP Enterprise SOA, https://www.sdn.sap.com/irj/sdn/enterprisesoa
4. Susanne Janssen and Ulrich Marquard: Sizing SAP Systems, SAP Press, ISBN 978-1-59229-156-4, http://www.sap-press.com/product.cfm?account=&product=H2904