



A Structured Approach to Reducing SAP Complexity

Ad van Dijke and Rob Scheele

The complexity of SAP environments is increasing. With every SAP country roll-out, upgrade, additional CRM/SRM application and custom development, the SAP environment is becoming more complex and difficult to manage in terms of cost effectiveness and ability to change. Organizations that make use of the SAP platform should beware of the consequences of complexity, and they should try to reduce complexity as much as possible.



A.A. van Dijke
is a senior manager at KPMG Advisory N.V.
vandijke.ad@kpmg.nl



R. Scheele
is a management consultant at KPMG Advisory N.V.
scheele.rob@kpmg.nl

Introduction

Organizations are focused on keeping their SAP environment cost effective and agile to respond to changes in business processes. In the last decade organizations have taken initiatives in an attempt to reduce the total cost ownership, such as system consolidation (“one SAP”), SAP in the cloud and outsourcing of functional and technical support.

However, these initiatives have not led to sustainable and significant cost savings. In this article we will substantiate that the main cause for this is *complexity*. Complexity in the SAP environment hurts: changes take a long time from initiation to implementation, there is a lot of custom ABAP code to maintain, many (country specific) configuration changes and a relatively high number of manual postings, corrections and rework by business users.

The main causes of the increasing level of SAP complexity are diverse in nature and include, amongst others, the following:

- the increasing use of additional SAP modules and the need for integration between them, including SAP Financial Supply Chain Management, Environment Health & Safety, Foreign Trade and the Closing Cockpit
- the increasing number of separate SAP application servers for CRM, SRM, BW, APO, GRC, as well as the interfacing between them
- the increasing number of legacy interfaces and SAP servers resulting from mergers and acquisitions
- legacy SAP systems that have been migrated several times to ECC 6.0 and that have never been cleaned up
- the expanding number of roll-ins where operating companies make use of a global SAP environment with local (non-harmonized) configuration, interfaces and custom ABAP code
- the outsourcing of crucial knowledge on the SAP environment to external parties, making it increasingly difficult to maintain an overall view and centralized insight into the complexity.

The main objective is to reduce the level of SAP complexity. Organizations that do so can improve their ability to change their SAP environment and increase their cost effectiveness.

This article covers a structured approach on how to reduce SAP complexity. The foundations of complexity will be explained as well as our approach on how to analyze and reduce SAP complexity.

Complexity and SAP

To understand SAP complexity it is important to lay out some fundamental concepts and definitions of complexity. The word “complex” can be defined as “consisting of interconnected and interwoven parts”. It is essential to understand not only the behavior of the parts but how they act together to form the behavior of the whole ([Bar97]). In this article we are talking about complex SAP environments,

Organizations that manage complexity in their SAP environment adequately can become more agile and cost effective

which are essentially complex systems. On a conceptual level complex systems have the following characteristics ([Bar97]):

- the number of elements (or subsystems)
- the interactions between them
- the operation between the subsystems
- the diversity/variability
- the environment (and its demands)
- the activities and their objectives

All these elements contribute to the level of complexity of the whole (complex) system. The more subsystems (or modules), interactions and diversity (e.g. different types of postings), the more complex the system is.

It helps to explain these concepts in the context of SAP. SAP is complex in itself, so how can we define complexity in the context of SAP environments? A typical SAP environment has a lot of transactions, modules, interfaces and diversity in its operations. This type of complexity is defined as *controlled* (and manageable) complexity, or in other words, “rewarded complexity”. Rewarded complexity is complexity that adds business value. “Rewarded” as opposed to “unrewarded” complexity is a known, stable and controllable factor in the SAP environments design and operation. An example of rewarded complexity is programming a user exit for Available to Promise (ATP) to support the business requirement for real-time inventory checking in the Webshop.

Unrewarded complexity in an SAP environment may have the following symptoms:

- time consuming implementation of individual change requests
- many different types of changes originating from custom (ABAP) developments
- many country specific (operating company) change requests
- high reliance on external consultants with specific expert knowledge
- complex authorization role structure with many single role changes
- unpredictable system behavior (affecting both performance and reliability)
- a relatively high number of manual process steps
- a relatively high number of unused master data and customizing
- many deviations from the SAP standard (many custom-made developments)
- a high number of errors in master data, transactional & interface processing

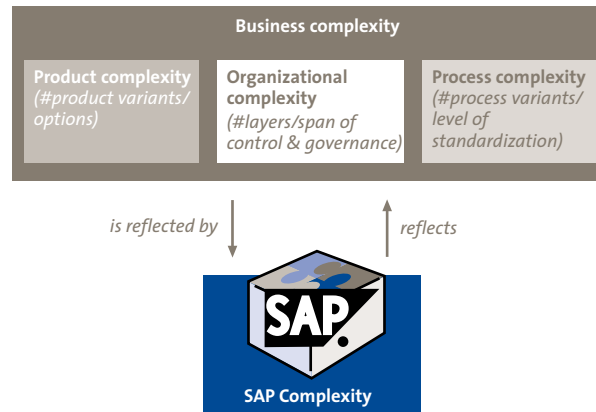


Figure 1.

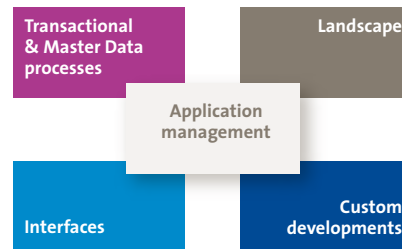


Figure 2.

In our approach we make a distinction between SAP complexity and business complexity. The business complexity consists of product, organizational and process complexity. These categories are reflected in the SAP environment in several ways: the complexity of products and services impacts the product master data in SAP, the organizational structure complexity impacts the enterprise structure (i.e., company code and profit center structure) in SAP and the level of process standardization impacts the number of sales process flows (refer to Figure 1).

Furthermore we distinguish five categories of SAP complexity (see also Figure 2):

- transactional & master data processes: complexity encountered during the execution of transactions in SAP
- interfaces: complexity caused by the number and distinct types of interfacing
- landscape: complexity caused by a heterogeneous and diverse aggregate of interconnected applications and subsystems
- custom developments: complexity caused by the extent to which custom code developments have been applied
- application management: all complexities impacting the functional and technical maintenance of the SAP environment

Rewarded as opposed to unrewarded complexity is a known, stable and controllable factor in the SAP environments design and operation

Analyzing SAP complexity

Our approach to complexity reduction consists of two phases, namely analyzing SAP complexity and, based on the outcome of this analysis, designing a complexity reduction program.

Analyzing SAP complexity consists of three steps:

1. formulating complexity hypotheses
2. measurement of SAP complexity
3. determination of main complexity drivers

The objective of the analysis phase is to determine the main complexity drivers, i.e. the factors that contribute significantly to the complexity of the SAP environment and the business it supports. These complexity drivers can be used as a starting point to simplify the SAP environment and identify sources of overly complex business design, such as a relatively high number of process and product variants.

The first step is exploratory in nature and consists of developing hypotheses as to what factors influence SAP complexity. This step is conducted based on interviews with stakeholders, inspecting documentation and performing observations within the SAP system. Special attention is given to the difference and relationship between SAP and business complexity. For example, the results of this step may include identifying the SAP sales order pricing mechanism as complex (SAP complexity), which may relate to a relatively high number of manual interventions in sales discounts (business complexity). On the other hand, extensive VAT regulation (business complexity) may result in a complex VAT setup in the SAP system (SAP complexity). The first step results in an overview of factors that add to SAP complexity.

The goal of the second step is to factually measure SAP complexity based on data extracted from the SAP applications. This step benchmarks the current system against plain SAP, assuming minimum customizing and the absence of custom developments and modifications to the standard SAP. This approach also takes into account exceptions in business process execution such as manual interventions, cancellations, errors, disharmonized usage, and so forth. Using KPMG's Facts2Value™,

a comprehensive approach to data analytics, the complexity of the SAP landscape is measured along the SAP complexity categories.

Example analyses performed include:

- necessity of custom made developments (user exits, append structures, custom tables, custom classes, custom reports & transactions)
- unused custom developments
- feasibility of replacing custom made developments with standard SAP AG software components
- processes with a relatively high number of manual process steps & corrections
- the use of different interface types
- necessity of interfaces
- the interfaces with the most errors
- complex customizing (especially workflow and sales pricing)
- frequently changed customizing, master data & custom developments
- unused customizing and master data
- disharmonized use of customizing and transactional processes

The result of the second step is a benchmark of the SAP environment compared to vanilla SAP. This benchmark provides clear insights into the main areas of complexity.

In the third and final step the primary sources of complexity, i.e. the main complexity drivers, are determined. In this step the vanilla benchmark is compared against the potential sources of complexity that were identified during the first step. This results in areas that require further detailed investigation. For example, in the first step it is identified that the VAT environment involves complex legislation across multiple countries. The benchmark indicates that the VAT codes are frequently changed, which may indicate that the VAT environment is not reflected adequately in the system. The other way around, it could be determined that 50% of the G/L accounts are not used in a harmonized way, while the financial reporting structure allows a harmonized usage.

Based on further detailed investigation, the main complexity drivers, i.e. the primary sources of complexity, can be established. The result of the third step is an overview of opportunities to reduce complexity.

Reducing SAP complexity

Reducing SAP complexity consists of the design and execution of a complexity reduction program.

The design starts with determining the scope of the complexity reduction program. The scope is determined by offsetting the level of complexity against the level of business reward. A comprehensive product portfolio with many product variants implies intensive material maintenance and usage of complex bills of materials. This type of complexity is rewarded in the sense that it supports its counterpart of complexity in business requirements. On the other hand, the usage of different document types and other supporting customizing elements across a company that has harmonized its business processes should be defined as unrewarded complexity, as it lacks its business counterpart, and thus lacks value.

An example of plotting business reward/value against SAP complexity is depicted in Figure 3, whereby each of the circles represents a complexity driver.

The main focus area of the design of the complexity reduction program is in the lower right: in this area SAP complexity is relatively high, while the level of reward is relatively low. SAP environments with the majority of complexity drivers in this area can be described as “technically complex” with little or no justification in business terms. Common origins of this area of complexity include:

- *Multiple (technical) solutions to meet the same business requirements across the organization (“not invented here syndrome”)*, e.g. the usage of distinct customizing elements to support the same process and product variants across business units. These types of complexity are commonly caused by suboptimal IT governance and lack of central oversight across custom developments.
- *Technical solutions that are not used (“dead wood”)*, such as high amounts of unused custom developments. These types of complexity can be caused by changing business requirements and heritage from adopted SAP systems (as a result of mergers and acquisitions, for example). While these technical solutions do not add any business value, these types of solutions need to be technically maintained.
- *Custom developments instead of using standard (SAP) solutions (“reinventing the wheel”)*. These types of complexity originate from not using standard SAP solutions to solve a problem, commonly caused by limited understanding of available solutions, an outdated SAP environment and a technically oriented approach to SAP deployments.

While the area of complexity in the upper right of the figure (rewarded complexity) is not the main focus area, it may provide valuable insights into overly complex busi-

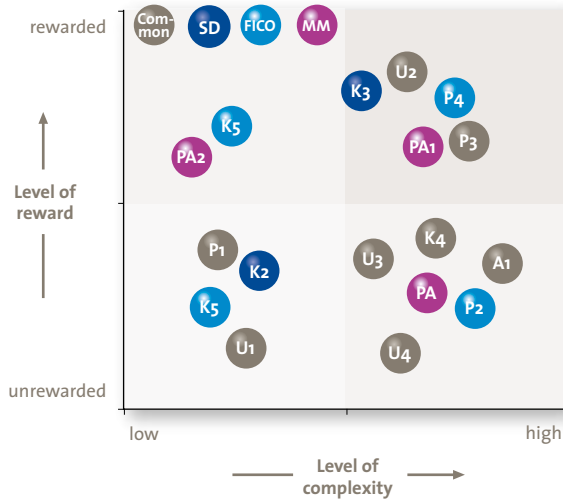


Figure 3.

ness design. If the main complexity drivers originate from this area, management could consider limiting the number of products and process variants, and simplifying the structure of the organization.

After the scope of the complexity reduction program has been determined, complexity reduction initiatives should be designed for each of the unrewarded complexity drivers. Depending on the nature and level of complexity involved, a complexity reduction program may involve both technical and organizational initiatives as depicted in the following figure. On one end, re-implementing a vanilla SAP removes technically unrewarded complexity in full. However, if the SAP complexity reduction program does not address its business complexity counterparts, the unrewarded SAP complexity will be re-introduced. On the other end, eliminating unrewarded SAP complexity without business complexity counterparts does not require organizational initiatives in the complexity reduction program.

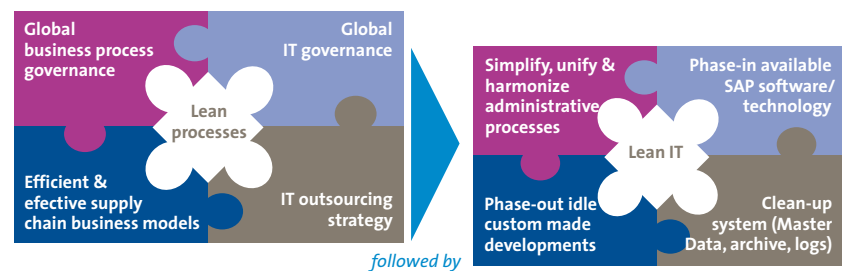


Figure 4.

Three complexity reduction scenarios can be chosen from, varying from a pure technical redesign to a full strategic business improvement scenario. A scope that is mainly focused on reducing technically unrewarded complexity requires a technical redesign, and a strategic redesign of business processes is suitable for addressing organizational complexity as depicted in Figure 5.

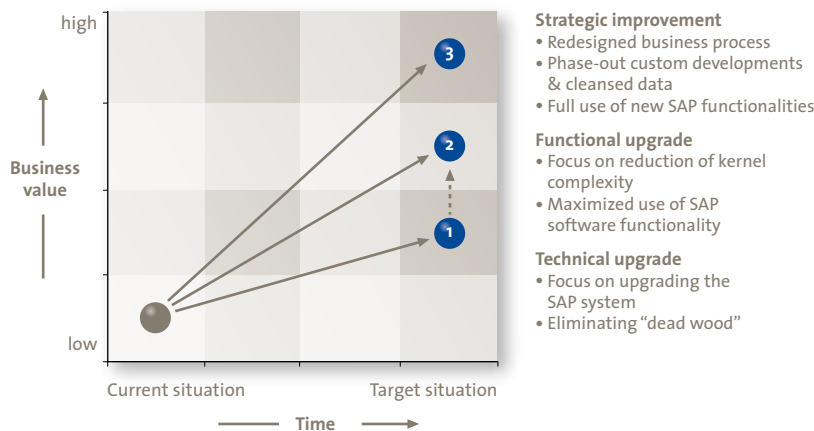


Figure 5.

References

- [Bar97] Y. Bar-Yam, Dynamics of Complex Systems, 1997.
- [Koc12] M. Koch, ABAP Development for Sales and Distribution in SAP: Exits, BADIs, and Enhancements.

About the Authors

A.A. van Dijke is a senior manager at KPMG Advisory N.V. He specializes in the optimization of SAP environments, working capital reduction and the analysis and optimization of the SAP technical layer.

R. Scheele is a management consultant at KPMG Advisory N.V. He focuses on business process improvement and control using SAP, data analytics and business intelligence applications.

As a good practice, a SAP complexity reduction program could involve the following initiatives:

1. Simplifying the SAP landscape by rationalizing interfaced applications:
 - reducing the number of components by, e.g., consolidating multiple SAP systems into a single client system, and phasing out / replacing (legacy) components with the latest software components of SAP AG
 - reducing the required interactions between the components by evaluating the possibility of overall removal of the interfaces through component integration
 - making use of a standardizing interfacing middleware such as SAP Process Integration software or other integration software that standardizes the messaging between components that require interaction in the SAP landscape
2. Cleanup of “dead wood” by:
 - cleanup of the system by removal of the unused customizing and custom ABAP developments
 - archiving unused master data and transactional data
3. Simplification of overly complex, but rewarded functional areas and IT maintenance processes such as:
 - reducing the number of sales pricing variants and as a result, harmonizing condition tables and access sequences across sales organizations
 - simplifying material master maintenance by harmonized usage of material types, SAP maintenance views and required material master data fields
 - simplifying the user maintenance process by limiting the number of roles and by overall improvement of the SAP authorization design
 - simplifying the profit and cost center setup, including hierarchies
 - reducing the complexity of SAP account determination by using a harmonized chart of accounts and reducing the number of G/L accounts
 - redesign of the IT change management process by reducing the approval steps, unnecessary bureaucracy and performing a root cause analysis on frequently changed customizing

4. Harmonization of processes by providing a standardized SAP solution across business units. Harmonization, as opposed to standardization, means limiting the degree of freedom in processes. In practical SAP terms, this means that the business units make use of common customizing elements for each process and product variant, such as common material types, document types, movement types, chart of accounts and so forth
5. Making improvement / updates to programs and interfaces by performing a root case analysis on frequently occurring program and interface errors
6. Standardization of exceptions by redesigning the processes to model the frequently occurring exceptions (manual interventions, errors, cancellations etc.) in process execution, which are discovered during the assessment phase
7. Phase out of custom developments by evaluating the phase-in of the latest technology, upgrades and enhancement packages.
8. Implement / mandate a central process governance board that promotes the use of standardized SAP solutions across business units and prevents local developments, except those responding to local fiscal and regulatory requirements.

Conclusion

Complexity in your SAP environment has significant consequences in terms of cost effectiveness and ability to change. While organizations have taken various initiatives in the last decade in efforts to increase efficiency and reduce complexity, these initiatives have not all led to sustainable or significant improvements. Organizations that manage complexity in the SAP environment adequately can gain significant and sustainable cost benefits and improve their ability to change. Therefore we propose a structured approach to managing SAP complexity.

The starting point of reducing SAP complexity is a thorough analysis of business and SAP complexity, applying a fact-based analysis of the SAP environment setup and usage. The goal is to identify unrewarded SAP complexity that does not support business requirements. Based on this analysis, a complexity reduction program can be initiated with the objective of removing the unrewarded complexity.