Communications of the Association for Information Systems

Volume 26 Article 3

1-2010

Application Portfolio Management—An Integrated Framework and a Software Tool Evaluation Approach

Daniel Simon
act! consulting

Kai Fischbach
Department of Information Systems and Information Management, University of Cologne, fischbach@wim.uni-koeln.de

Detlef Schoder
Department of Information Systems and Information Management, University of Cologne

Communications of the Association for Information Systems

CAIS

Application Portfolio Management—An Integrated Framework and a Software Tool Evaluation Approach

Daniel Simon

act! consulting

Kai Fischbach

Department of Information Systems and Information Management, University of Cologne

fischbach@wim.uni-koeln.de

Detlef Schoder

Department of Information Systems and Information Management, University of Cologne

Abstract:

Despite the growing number of organizations that have lost track of their application landscape and have suffered from a sharp increase in application portfolio complexity, a comprehensive and systematic approach to Application Portfolio Management (APM) still appears far from being adopted. To move the adoption process along, this paper develops a comprehensive framework assimilating and extending previous research and presents an APM process comprising data collection, analysis, decision-making, and optimization phases. This paper also presents an approach for evaluating software tools for APM and identifies which software tool families are best able to provide support for specific purposes. With this integrated conceptual guideline for APM and its translation into a model for measuring appropriate practical support, this paper not only allows for a move more deeply into the research area but also offers advice for both researchers and practitioners.

Keywords: enterprise architecture, IT planning, IT architecture, information technology management, tool

Volume 26, Article 3, pp. 35-56, January 2010

I. INTRODUCTION

Today many organizations face the challenges that come with a heterogeneous, nontransparent and vast application landscape, mainly due to mergers and acquisitions and rapid growth [Betz 2007; Caruso 2007; Keller 2007; Molter 2005]. This complexity of the application landscape often entails several "points of pain" with which a company must cope, such as soaring application costs and insufficient support of business processes [Caruso 2007; Hafner and Winter 2008]. As organizations realize their need to overcome these challenges, the concept of Application Portfolio Management (APM) gains significance. Using APM, companies aim to reduce the application landscape's complexity, especially through simplification and harmonization [Betz 2007; Kersten and Verhoef 2003; Molter 2005; Swanson and Dans 2000]. The considerable savings on maintenance expenditures promised by APM have helped spur awareness of its need and of the benefits that could accrue through its implementation [Kersten and Verhoef 2003].

While APM appears to have come to the forefront of many firms' attention, theoretical substantiation of APM has not followed. For one, APM is not defined consistently, although discussion of APM is hardly new to the academic literature. Moreover, publications dealing with APM have made little progress toward achieving an integrated view that encompasses the concept as a whole. Hence, overall APM lacks a systematic and comprehensive approach.

Against this background, the paper's scope is the concept of APM and its integral elements, presented in a comprehensive framework constructed according to the design science paradigm [Hevner et al. 2004]. The framework includes these factors: drivers, objectives, risk factors, maturity, and, particularly, the APM process itself. Our aim is to reduce the gap between APM in theory and practice. Furthermore, we aim to provide a bridge to the APM software tools market by building a model, based on our framework, for evaluating software tools in terms of their APM capabilities.

The remainder of this paper is organized as follows: Section II surveys research on the APM concept and related issues. Section III provides a definition of APM. Section IV builds the APM framework step by step and, therefore, is the essential part of this work. Section V builds on the framework and introduces the software tool evaluation model, which we then apply to an initial set of eight tools. Finally, Section VI briefly summarizes our findings and discusses potential further research.

II. RELATED WORK

At its heart, the concept of APM stems from matrix-based portfolio approaches, classifying applications along specific dimensions and deriving appropriate management action and investment allocation. A portfolio approach to information systems was first introduced by McFarlan [1981], and there have been several subsequent attempts at classification-based application portfolio analysis [e.g., Kwan and West 2004; Ward 1993; Weill and Vitale 1999]. However, there is little evidence of a broader approach to managing the application landscape. Mårtensson [2006], Hirvonen [2004], and Ward [1988] discuss part of the traditional approaches in detail. A more comprehensive examination of APM is that of Maizlish and Handler [2005], which discusses the concept of APM as one pillar of Information Technology Portfolio Management (IT PM) and offers a wide set of criteria for application analysis. Fabriek et al. [2007] make further contributions, proposing a method for application portfolio rationalization comprising three major steps: assessment, evaluation, and planning. Similarly, Dern [2006] offers an extensive set of criteria for analyzing applications, along with a process for portfolio analysis and planning. Like Dern [2006], Riempp and Gieffers-Ankel [2007] delve into APM from an Enterprise Architecture perspective, emphasizing APM's integrative nature by proposing different APM viewpoints, such as information technology (IT) strategy and IT infrastructure.

Interest in APM has risen in recent years. Kersten and Verhoef [2003] stress the importance of a portfolio approach to IT, with applications playing an integral part. Verhoef [2003] points to the role of source code analysis within a portfolio approach to IT, particularly to estimate the cost of changes. Likewise, Caruso [2007] emphasizes the necessity of APM for future IT, while Bahadur et al. [2006] stress the enormous impacts of application portfolio configuration on business value. Similarly, van Wegen and de Hoog [1996] point out benefits that can accrue from changes within the application portfolio. According to Swanson and Dans [2000], portfolio complexity is related directly and positively to maintenance efforts, and increased maintenance efforts go hand in hand with a greater overall life expectancy. Thus, organizations seeking to reduce portfolio complexity need a balanced approach:

investing both in application maintenance, which prolongs the life expectancy of applications, and in new applications. Groot et al. [2005] describe a “picture approach” for analyzing, redesigning, and combining an organization’s applications to reduce complexity. In addition, Molter [2005] dives into portfolio complexity and emphasizes the need to establish a process-oriented application landscape.

APM has also been a subject of discussion within a broader research context that stretches across a wider set of IT issues. Betz [2007] introduces the concept of the “IT value chain,” in which “Architecture, Portfolio and Service Delivery” is an integral activity. Benson et al. [2004] develop the “Strategy-to-Bottom-Line Value Chain” for integrating different IT practices, such as “Alignment” and “Performance Measurement.” “Portfolio Management” is among the concepts that provide support for these practices, with APM serving as a major component. Bloem et al. [2006] discuss how IT Governance may work in the face of current compliance pressure and identify the principles of IT PM as promising great support. In particular, they present some practical lessons for IT PM.

As already indicated, APM is tied closely to the concept of Enterprise Architecture.[1] Keller [2007] provides further evidence of the utility of approaching APM from an architectural point of view, advocating APM as a major Enterprise Architecture process. Hafner and Winter [2008] are also concerned with an organization’s application architecture and propose a generic process model for its management, though this appears to stop short of offering any comprehensive advice on how to analyze applications along multiple dimensions and then put that analysis to use in making decisions on a portfolio-wide level—advice we consider essential in our approach.

It is interesting that Longépé [2003] promotes an “urbanization” approach to information systems, that is, drawing an analogy between the building, improvement, and maintenance of an organization’s information systems and those of a city. Similarly, Namba and Iijima [2004] advocate a “city planning” approach to information systems. In addition, the field of “Software Cartography,” which deals with systematic visualizations of the application landscape, deserves closer attention. This is detailed, for example, by Buckl et al. [2007b]. To create such visualizations and manage the portfolio effectively, Buckl et al. [2007a] describe a pattern-based approach for constructing organization-specific architecture information models. From a practical point of view, Ernst et al. [2005] and Matthes et al. [2008] present a fine-grained tool evaluation in the field of “Enterprise Architecture Management.”

All in all, though, a systematic assimilation of APM methods, issues, and its practical application still seems to be lacking. This motivates the development of our approach presented in this paper.

III. CONCEPTUAL FOUNDATION

Generally speaking, a portfolio can be considered a collection of items grouped together to facilitate their efficient and effective management [Benson et al. 2004; IT Project and Portfolio Office, University of Utah 2008]. Considering an application as a special class of software that provides direct support for business processes [Maizlish and Handler 2005; Riempp and Gieffers-Ankel 2007; Lankes and Schweda 2008], an application portfolio simply describes the sum of all applications run by a specific organizational body [Riempp and Gieffers-Ankel 2007].[2]

The literature offers a variety of views with respect to a definition of APM itself. According to Maizlish and Handler [2005], for example, the accurate assessment of the value and business benefit, as well as of the costs of an application, can be seen as the cornerstones of APM. Similarly, Caruso [2007] points to the objective of APM as the balancing of expense against value. In fact, many approaches to the management of the application portfolio center largely on matrix-based classification or analysis along certain criteria. However, they do not deal with a more comprehensive application analysis that is followed by concrete management action.

Other researchers take up a more technological position [Betz 2007], associating APM with source code analysis in which business information is additional but nonessential. Riempp and Gieffers-Ankel [2007] offer a more sophisticated definition, stating that APM, in contrast to other application-oriented metaphors such as “application architecture management,” “IT city planning,” and “Software Cartography,” emphasizes “active management and decision-making. APM comprises all models, methods, and guidelines applied by IT decision-makers for the assessment, management, and optimization of an AP. It requires the consideration and integration of different concerns and viewpoints such as IT strategy, business and application needs, IT architecture, IT operation, IT project management, and IT investment.” As our aim is to advance different views of APM to be used as the basis for an integrated APM framework, we offer the following definition for the purposes of this paper:

[1] This is also reflected in well-known frameworks within the field of Enterprise Architecture [e.g., TOGAF—The Open Group Architecture Framework], where the application portfolio is among the subjects of interest.
[2] Instead, one may also use the term “application architecture” [Hafner and Winter 2008] or “application landscape.” This is in line with Buckl et al. [2007a] stating that the application landscape is “the entirety of business applications and their relationships in an organization.”

Application Portfolio Management is the ongoing application of systematic and structured decision-making processes to evaluate an organization's applications along various dimensions (from a business and a technical viewpoint), weigh various actions for the purpose of optimization, and implement appropriate actions to resolve identified issues and meet key enterprise objectives. The promise of Application Portfolio Management lies primarily in reducing the complexity of the application landscape, which is approached from a holistic viewpoint.[3]

In essence, our definition is meant to provide a sharper focus on process- and goal-related aspects and, thus, already points to some important elements that play a key role within our APM framework depicted in Section IV:

- "Ongoing application": APM is not a one-time effort; rather, it needs to be applied continually.
- "Systematic and structured … processes": APM is a process comprising distinct phases that systematically build on each other. Hence, thorough APM embraces much more than matrix-based analyses.
- "Reducing the complexity": The basic rationale behind APM is the reduction of the application landscape's complexity.
- "Holistic viewpoint": APM integrates, in a portfolio-wide view, different application dimensions from a business and a technical perspective, including costs, technical health, and lifecycle, for example.

It should be pointed out that APM is not a stand-alone approach but, rather, that it interacts with related concepts in various ways. In particular, APM can be considered a main pillar of IT PM [Kersten and Verhoef 2003; Maizlish and Handler 2005], which in turn can be seen as a cornerstone of a structured approach to managing IT as a business [Bloem et al. 2006]. Essentially, IT PM is "the application of systematic management to large classes of IT items" [Betz 2007, p.18]. These items, to be quantified through IT PM, include not only applications but also projects, services, and infrastructure. Thus, other pillars of IT PM include Project Portfolio Management (PPM), Service Portfolio Management, and Infrastructure Portfolio Management. In addition, APM is related closely to the concept of Enterprise Architecture Management (EAM) [Riempp and Gieffers-Ankel 2007], but only when EAM is defined as it is in this paper; that is, as going considerably beyond simple modeling techniques to make the Enterprise Architecture actionable in a way that allows a multidimensional decision-making approach—criteria that have not yet permeated every approach to EAM [Johnson et al. 2004]. Considering APM as an application-oriented viewpoint in the context of Enterprise Architecture [Riempp and Gieffers-Ankel 2007] is, however, in line with viewing EAM as:

> "a continuous and iterative process controlling and improving the existing and planned IT support for an organization. The process not only considers the information technology (IT) of the enterprise, also business processes, business goals, strategies etc. are considered in order to build a holistic and integrated view on the enterprise. The goal is a common vision regarding the status quo of business and IT as well as of opportunities and problems arising from these fields, used as a basis for a continually aligned steering of IT and business" [Matthes et al. 2008, p.24].

Actually, considering the support of business processes through applications is of particular importance for the thorough management of the application portfolio [Molter 2005].

IV. A FRAMEWORK FOR APPLICATION PORTFOLIO MANAGEMENT

Drawing upon our definition of APM above, the APM framework developed in this section specifies the following aspects:

- Drivers, objectives, and risks of Application Portfolio Management
- The Application Portfolio Management process
- Application Portfolio Management maturity

Each of these aspects is eventually incorporated within an integrated APM model.

Drivers, Objectives, and Risks of Application Portfolio Management

Taking a structured view of APM drivers brings several points to the surface. The need for APM is driven predominantly by the complexity of the application landscape, which can be traced in particular to past mergers and acquisitions, and rapid growth [Betz 2007; Groot et al. 2005; Keller 2007; Kersten and Verhoef 2003; Molter 2005; Swanson and Dans 2000]. Highly decentralized IT management [Fabriek et al. 2007], or the existence of various

[3] Refer to Betz [2007], IT Project and Portfolio Office, University of Utah [2008], and Riempp and Gieffers-Ankel [2007].

legacy applications that have not been migrated (for whatever reason), but that have been complemented by many other applications over time, compound the complexity [McKeen and Smith 2002a]. Most likely, such applications would involve a high degree of individual solutions and customized software. Worse yet would be an IT staff that has faced reductions and high turnover, leaving the organization with poor knowledge of the application landscape.

Several aspects of the application landscape speak particularly to its complexity, including its heterogeneity [Swanson and Dans 2000], lack of transparency, and vast scope. These may reflect a considerable number of both technical and economic "points of pain." On the one hand, a complex application landscape can raise many technical issues, such as interface problems [Caruso 2007; Molter 2005], bloated source code [Swanson and Dans 2000], a lack of insight into the technical quality of applications, troublesome portfolio maintenance and expansions [Molter 2005], and the portfolio's exposure to failure propagation. This exposure may, in fact, create a domino effect that jeopardizes the robustness and performance of applications because so many are operated and interact in complex ways [Maizlish and Handler 2005; Niemann 2006; Lankes and Schweda 2008]. On the other hand, a complex landscape also gives rise to a variety of economic points of pain, which may not, however, be completely independent from the above technical concerns but, instead, actually originate from them.

As mentioned above, it is all too often the case that organizations lose sight of the entirety of their applications under operation. Given that, how could they possibly know the details of the application costs, benefits, and risks? When they lack such knowledge, they may make application investment decisions that do not draw sufficiently on rational and financial methods [Caruso 2007]. There is, though, a general awareness of soaring application costs due mainly to increased maintenance efforts caused by high portfolio complexity [Caruso 2007; Hafner and Winter 2008; Molter 2005]. With such complexity, costs and benefits are likely to become increasingly imbalanced, and operational inefficiencies [Molter 2005], as well as a delayed time-to-market [Niemann 2006], are a common outcome.

There are other economic issues of concern as well when the application landscape suffers from too much complexity. For instance, there may be a lack of the applications' strategic alignment and insufficient support of business processes [Hafner and Winter 2008]. A complex application landscape that has hindered active management can result in an organization's failure to revise the application portfolio as needed to keep up with business needs that may have changed over time [Molter 2005]. In particular, an organization facing high portfolio complexity is likely to suffer significant limitations with respect to its flexibility, agility, and innovative opportunities, since resources may be heavily weighted toward "lights-on" activities, with only a small amount of resources available for investment in future capabilities [Caruso 2007; Benson et al. 2004]. Therefore, APM aims at a reduction of application portfolio complexity, especially through simplification and harmonization [Kersten and Verhoef 2003; Molter 2005]. By managing out complexity, points of pain, such as those described above, are likely to be mitigated, thereby keeping innovations from being at a constant disadvantage because of the multitude of redundant applications. In addition, any post-merger integration is likely to be significantly facilitated.

Another factor that may speak to an organization's need for APM is the general lack of a permanent approach to the application landscape that draws on structured and rational methods for making decisions about application investments on a portfolio-wide basis. In fact, APM aims at managing the application portfolio as a business [Maizlish and Handler 2005]. Further still, environmental drivers and other internal initiatives may increase the need for APM. Environmental drivers worthy of mention include legal requirements; increasingly capable rivals who have gained a competitive edge through cost advantages; ongoing pressure from customers to reduce prices, thereby forcing the organization to reduce IT costs; changing industry and business requirements; and shorter technical innovation cycles [Bloem et al. 2006; Caruso 2007; Groot et al. 2005; Hafner and Winter 2008; Kwan and West 2004]. Relevant internal initiatives in this context are, among others, business process reorganization projects [Caruso 2007], legacy modernization or innovation initiatives [Dern 2006], internal compliance initiatives, and IT due diligence efforts [Kersten and Verhoef 2003].[4] Where there are plans to outsource applications, APM may also facilitate the selection of applications to be outsourced as well as the request for proposals, since application portfolio transparency could allow potential contractors to obtain a better estimate of whether their capabilities are sufficient. APM may even assist both parties to negotiate and size a contract. Once an outsourcing agreement has been established, APM may serve as a way to monitor the technical quality of the contractor's deliverables.

Of course, APM does not automatically guarantee great success [Bloem et al. 2006; Fabriek et al. 2007]. Certain risk factors [Sherer and Alter 2004] may create obstacles, including inadequate funding [Fabriek et al. 2007], stakeholder resistance, lack of infrastructure for real-time reporting [Maizlish and Handler 2005], poor documentation of business processes [Molter 2005], and the reduction of application analysis to too few dimensions [Rispens and

[4] Another relevant internal initiative could also be a large software implementation project, since the accurate definition of the project scope may require insight into the application portfolio and any relations and dependencies. Also, efforts of project portfolio planning may drive this need.

Vogelezang 2007]. We group these risk factors into the following areas [Sherer and Alter 2004]: Finance, Policy/ Culture, Methodology, Organization, Technology, and Strategy/Governance. The prevalence of a given risk factor has a significant impact on the organization's APM readiness. To avoid failure, certain success factors [Sherer and Alter 2004] need to be considered, such as ensuring top management commitment.

The Application Portfolio Management Process

In this section, we describe the APM process, which is the heart of our framework. The process we have designed builds on previous models developed by Maizlish and Handler [2005], Fabriek et al. [2007], Niemann [2006], and Dern [2006]. Through several step-by-step phases—Data Collection, Analysis, Decision-Making, and Optimization—our approach seeks to demystify the application portfolio and improve its composition and quality. Again, APM is not a one-time effort; rather, these phases form a continuous cycle [Bloem et al. 2006; Keller 2007]. In the Data Collection phase, an application inventory is established. Analysis involves a careful examination of the inventoried applications, which forms the basis for Decision-Making, which ends by having concrete actions implemented during the Optimization phase. This, in turn, feeds Data Collection with updated information.

Data Collection

In the Data Collection phase, the current state of the application portfolio is captured. Ideally, all information on the organization's applications is centrally available. This is rare, however, particularly in the case of highly decentralized IT organizations with a large quantity of individual solutions [Benson et al. 2004]. Therefore, a dedicated Data Collection phase is essential to establish a detailed application inventory [Keller 2007; McKeen and Smith 2002b]] which forms the basis for in-depth analysis [Buckl et al. 2007a]. The inventory of existing applications, which allows greater transparency, is complemented by information on applications both planned and in development [Matthes et al. 2008].

The application inventory should go beyond a simple list of which applications are actually operated or planned to include their general characteristics. These characteristics include, but are not limited to, application name, release version, implementation date, application owner, key capabilities, user groups, number of users, vendor, operating system, enabled business processes, affected business units, lines of code, and technical components [Keller 2007; Maizlish and Handler 2005]. These general characteristics are critical in identifying each application, understanding its role, and ensuring access to important information, such as whom to contact with questions about the application [Maizlish and Handler 2005]. In addition to the general characteristics, there are key attributes to identify, including costs and operational performance (specified in the subsection that follows). These are essential for subsequent portfolio analysis. In particular, it is necessary to obtain a clear picture of the business processes [Molter 2005], reflecting that APM shows close links to Business Process Management (BPM).[5]

In summary, the Data Collection phase encompasses three levels of understanding of the application portfolio:

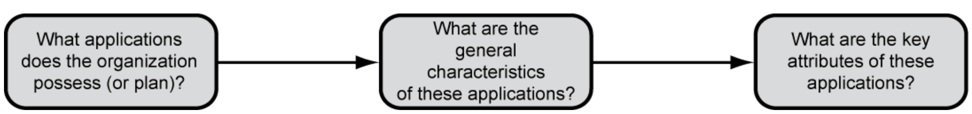

Figure 1. Levels of "Application Portfolio Understanding"

There are three general options for Data Collection. Automatic data collection refers to the capability of reading source code to take stock of existing applications automatically. This likely requires sophisticated Source Code Management[6] that may not be in place in every organization. Moreover, packaged applications may pose a huge challenge to automatic data collection, since their source code is usually not accessible [Keller 2007]. This drives the need for considering further data collection methods. Semi-automatic data collection involves gathering data from other IT systems via dedicated interfaces. Although information is not obtained directly from the source code, no manual user input is required. Finally, manual data collection involves obtaining data about applications directly from stakeholders [Weill and Vitale 1999]. It can be technologically facilitated through online surveys and automated

---

[5] Further business information relevant to APM includes business strategy, goals, constraints, products, capabilities, functions, and so on.

[6] Hung and Kunz [1992] define a source code management system as "a set of tools keeping track of various versions and configurations of programs or applications."

notification mechanisms, and may include application landscape modeling (Software Cartography techniques hold out promise to be very useful for this purpose, as discussed below with sample techniques).[7]

The major deliverable of the Data Collection phase is an accurate application inventory structured according to the specific needs of the enterprise [Buckl et al. 2007a; Kurpjuweit and Winter 2007].

Analysis

Once there is a detailed application inventory, the next step is to analyze the inventory information thoroughly to gain insights into the as-is portfolio. Applications can be analyzed along several attributes. Actually, we suggest considering an analysis along the following dimensions: Business Process Support, Strategic Fit, Value/Benefits, Costs, Risks, Lifecycle, Regulatory Compliance, Functional Wealth, Technical Health, Operational Performance, Relations and Dependencies, and Vendor Information.[8]

The most promising starting point is Business Process Support [Niemann 2006], both in present and future terms. From this detailed examination of the applications' range of support [Maizlish and Handler 2005] functional redundancies, and business areas lacking support, the analysis can then be extended to other dimensions (outlined below). To assess the applications' support of business processes, heat maps [Betz 2007], and process support maps [Buckl et al. 2008; Dern 2006; Keller 2007; Matthes et al. 2008] can be employed. Basically, these instruments from the field of Software Cartography make it possible to plot applications in matrices of processes and organizational units (or products), taking into consideration the criticality (and the related business function) of the illustrated processes.

How an application supports business processes is only one facet of its alignment with the business [Dern 2006; Silvius 2006]. In addition, an application's strategic fit—that is, its compliance with strategic directives and its support of business goals—must be gauged [Dern 2006; Maizlish and Handler 2005; Weill and Vitale 1999, again both in present and future terms.

Closely linked to these dimensions is an application's business value and benefits.[9] Application value derives primarily from automating business processes and thus empowers users to accomplish the tasks for which they are responsible [Maizlish and Handler 2005; Smith and McKeen 2003]. This may allow benefits such as increased revenue, reduced cost, cycle time reductions, risk mitigation, and improved managerial decision-making [Maizlish and Handler 2005; Weill and Vitale 1999; Silvius 2006; Smith and McKeen 2003]. Determining the application's value is essential. It is only when an application is critical to success that weaknesses in dimensions such as Technical Health and Operational Performance may become particularly relevant [Niemann 2006]. In fact, it takes a thorough examination of the current and potential future impact of applications [Keller 2007] before investment decisions should be made [Ward 1993] in the subsequent phase. Because application value has different aspects, a variety of evaluation methods [Silvius 2006] may be employed, particularly financial (e.g., ROI, NPV, EVA) and strategic (resource-based view) instruments.

To analyze applications by costs, IT expenses need to be broken down accurately at the application level. These expenses or cost attributes [Maizlish and Handler 2005] include the ongoing costs required to keep an application running (operations, maintenance, licensing, and depreciation costs), as well as those associated with upgrading or replacing an application. Further differentiation can be made between direct and indirect costs. So-called "Activity Based Costing" can help develop an overview of the indirect costs of an application (e.g., downtime costs) [Fabriek et al. 2007]. Likewise, the "Total Cost Of Ownership" concept, which involves examining indirect costs, can be useful in determining overall application costs [Silvius 2006].

Analysis of application risks should focus on the probability of application failure and the effects this failure would have on other systems and the business overall, as well as on an application's ability to meet certain requirements on an ongoing basis and adapt to changed conditions. A number of attributes need to be considered when evaluating application risks, since they may have a significant impact on the application's long-term use and its future prospects, as well as on the organization's business opportunities, flexibility, and agility. These attributes include, but are not limited to, vendor viability, the extent of required regulatory compliance, the application's technical specificity and conditions, the application's security and privacy capabilities [Maizlish and Handler 2005],

[7] Manual data collection may best be conducted with first-order attention to data that can be collected from current projects (i.e., data about specific applications that were necessary to gather at project initialization). In fact, we consider these data most likely to be of high quality.
[8] This set of dimensions was compiled based on different contributions in literature, all of which is referred to below.
[9] Although these two terms are not considered equivalent, they are closely linked. However, a thorough discussion of these terms is beyond the scope of this paper. Refer to Walter and Spitta [2004] for further information.

connections to other applications [Keller 2007], and the application's compatibility and maintainability [Jordan and Silcock 2005].

Evaluating the current lifecycle state of applications [Dern 2006] requires creating a clear picture of current and planned application versions and releases [Betz 2007; Buckl et al. 2008] that reflect their level of maturity, life expectancy, future role, and any anticipated future efforts to keep the applications running and aligned with business needs.

Analysis of the regulatory compliance of applications is driven by whether applications comply with internal policies and external requirements [Niemann 2006].[10] If they do not, an estimate must be made regarding the effort required to achieve compliance. Unlike with regulatory compliance, the functional wealth of applications becomes particularly important in cases of redundant applications, where it is essential to examine the quality of support through applications [Riempp and Gieffers-Ankel 2007] by assessing both functionality that affects usability (e.g., search functionality) and functionality that is directly related to the fulfillment of user tasks (e.g., reporting capability).

The technical health of applications [Weill and Vitale 1999] comprises, on the one hand, factors related to source code [Dern 2006], such as size, complexity (measured, e.g., by the McCabe metric [cyclomatic complexity, McCabe 1976], Keller 2007; Rispens and Vogelezang 2007; Niemann 2006], and object orientation and, on the other hand, factors such as maintainability [Keller 2007], compatibility, the degree of documentation [Weill and Vitale 1999], and the architectural fit (compliance with reference architectures or other technical standards such as programming constraints) [Buckl et al. 2008; Dern 2006; Maizlish and Handler 2005; Riempp and Gieffers-Ankel 2007; Rispens and Vogelezang 2007]. Calculating source code-related figures has already proven to be quite promising [Verhoef 2003]. Strongly linked to, and mainly determined by, technical health, the operational performance of applications is reflected by factors such as the number of service requests, trouble tickets and failures, response times, and availability [Maizlish and Handler 2005; Weill and Vitale 1999] (for the examination of both technical health and operational performance refer to, e.g., ATAM—the Architecture Tradeoff Analysis Method [Kazman, Klein, and Clements 2000]).

So-called "neighborhood analysis" may aid in determining the relations and dependencies of applications. This approach reveals the effects application disposal or downtimes may have on other applications or on the entire portfolio, and provides a means for evaluating the extent to which the portfolio is prone to failure propagation [Niemann 2006; Lankes and Schweda 2008; Buckl et al. 2008]. From our point of view, dependencies should also be scrutinized in terms of application vendors; numerous relations between applications from certain vendors may point to an opportunity to develop standardized interfaces between these applications. Finally, an analysis of application vendors themselves should include vendor size, financial health, future product plans, the importance of given products within the overall vendor portfolio [Maizlish and Handler 2005], geographic coverage, and support, consulting, and training services.

Ultimately, to gain comprehensive insights into the application portfolio, it is not sufficient to analyze the portfolio dimension by dimension. Techniques such as portfolio matrices [Weill and Vitale 1999], scoring models [Betz 2007], the IT Balanced Scorecard [Bloem et al. 2006], and the SWOT analysis can be used to conduct an assessment along two or more dimensions or to achieve a certain degree of aggregation of several criteria. To prepare for subsequent decision-making, however, it is necessary to differentiate between discretionary (determinable) and non-discretionary (non-determinable) applications [Maizlish and Handler 2005]. Whereas the former are subject to the entire range of options that determine whether applications will be part of the future portfolio and what they might look like (see the following subsection), choices for the latter may be limited, as such applications must be operated according to legal requirements or their critical role in keeping the business running.

Decision-Making

The third step in the APM process is Decision-Making, that is, planning and shaping the to-be portfolio.[11] It is based on detailed application analysis and, in general, involves determining how to allocate investments in existing applications, the purposes of such investments, and which new applications or related infrastructure warrant investments. Specifically, concrete optimization options may need to be weighed against one another. Optimization options fall into three general categories [Keller 2007]: "Create," "Modify," and "Delete." These can affect single applications, a number of applications, or even the entire portfolio.

[10] "Regulatory Compliance" does not cover strategic and technical compliance. These aspects are included in other application dimensions.
[11] The to-be portfolio may be divided further into a series of planned and a target portfolio [Matthes et al. 2008], with the former describing intermediate transitional states.

"Create" comprises the specific options of "Investment" (in entirely new applications) and "Replacement." While investments in new applications are most likely made when business processes have changed significantly and existing applications no longer support these processes, replacements may be appropriate when applications possess insufficient functional or technical quality. Replacements should also be considered if a huge application has become difficult to maintain and might best be replaced with several smaller applications. Conversely, it may be best to replace several small applications that are poorly integrated with a single application that offers better integration [Keller 2007]. Basically, we consider both "Investment" and "Replacement" to be realized through purchasing a packaged application, developing an individual solution, implementing a hosted application, or launching an Open Source solution.

"Modify" encompasses the options "Functional Enhancement" [Keller 2007], "Outsourcing" [Benson et al. 2004], "Service-oriented Architecture" (SOA) [Keller 2007; Ren and Lyytinen 2008], "Integration," "Integration Optimization," and "Reengineering" [Keller 2007]. Functional enhancements can be used to address poorly developed application functionalities. Outsourcing is a potential optimization option for several reasons, including financial, strategic, and technological considerations [Leimeister et al. 2008]. Building a Service-oriented Architecture may be the desired course of action to achieve greater flexibility and enhanced maintainability, and to breathe new life into legacy applications that possess unique capabilities.

To integrate applications or to optimize any integration, "Enterprise Application Integration" [McKeen and Smith 2002a] may provide the best answer. "Integration" here refers to the implementation of new interfaces in response to new opportunities for application collaboration that are uncovered but thus far unexploited, whereas "Integration Optimization" is the improvement of collaboration among existing applications. Reengineering activities such as platform/database migration, language migration, code optimization, and improvement of object orientation should be strongly considered to improve the technical quality of applications.

Finally, "Delete" comprises the two corrective options "Disposal" and "Consolidation." "Disposal" is likely the preferred option when applications are no longer used, are managed specifically for a business unit no longer considered financially viable [Keller 2007], or support business processes that are outsourced to third parties. "Consolidation" is the reduction of various overlapping or redundant applications that support the same process in favor of a single, existing application [Keller 2007].

Of course, an organization may decide to take no corrective actions at all [Keller 2007]. We do not, however, consider this an optimization option, since it does not involve any real changes in the portfolio and the applications that comprise it, and may involve only a modified resource allocation.

Coming to well-grounded decisions requires a holistic view of applications; that is, taking into account the entire range of dimensions for analysis. Although techniques such as portfolio matrices allow a certain degree of aggregation and reduce the decision-making arena to a more manageable number of high-level options, such techniques are still limited in scope to only a few dimensions. To make balanced decisions demands consideration of all dimensions as part of a coherent, highly aggregated picture [Riempp and Gieffers-Ankel 2007]. This picture could potentially be developed through mathematical programming techniques, which may allow either a prioritization or generic characterization of applications and thus determine the urgency and order of change [Weill and Vitale 1999], taking into particular account whether the supported business processes are distinctive and value-adding [Niemann 2006]. For this purpose, both the application and the process owner should be assigned a significant role within decision-making [Maizlish and Handler 2005].

As for the specific optimization options, appropriate decision-making also requires scrutiny of so-called "contingency factors," these will affect which options receive the closest consideration [Mårtensson 2006]. Thus, factors such as the current IT governance model (centralization vs. decentralization), existing IT principles [Weill and Ross 2004] and prevailing strategic circumstances, the uniqueness of business processes, the importance of flexibility, norms and values of realizing benefits, the degree of risk aversion [Niemann 2006], the required investment volume [Fabriek et al. 2007], and relevant market and technology developments [Mårtensson 2006; Niemann 2006] should not be left ignored.

To resolve in the best possible way any portfolio deficiencies that are revealed, one must also determine how these problems came to exist. It is, therefore, necessary to widen the scope to encompass the infrastructure and related business processes [Fabriek et al. 2007; Molter 2005; Buckl et al. 2008]. Furthermore, "underlying patterns" such as culture, communication, and the IT staff's understanding of the business strategy should be examined [Fabriek et al. 2007]. Failure to overcome cultural obstacles [Andriole 2007] when using APM could doom an organization to remain stuck in an environment of redundancies.

Our process suggests further that possible side effects from decisions should be scrutinized. An organization should gauge how portfolio decisions might affect the infrastructure [Andriole 2007], IT services, and projects that may be required to implement individual optimization options. To avoid duplicate efforts, planned applications also need to be considered. Moreover, in terms of side effects, an APM initiative should examine carefully whether portfolio changes may enable the improvement of business processes or the development of innovations, all of which should be done with a particular focus on potential "quick wins" [Niemann 2006].

The outcome of the Decision-Making phase is a concrete action plan, like a "road map," for portfolio optimization [Dern 2006]. Such a plan should also establish the business cases [Maizlish and Handler 2005] that justify the implementation of the selected optimization projects, and should be complemented by a model of the future landscape to illustrate the intended resulting portfolio.

Optimization

The Optimization phase puts the results of the Decision-Making phase into action [Niemann 2006]. Having determined the general course of action, decisions may first need to be made more concrete and detailed in a well-defined solution, e.g., through the selection of certain products or vendors (e.g., middleware product, outsourcing contractor), taking into consideration how these selections might affect the business case and the individual application dimensions and whether there are any prior experiences or standards (e.g., programming languages) that should be drawn upon [Niemann 2006]. Similar to the Decision-Making phase, the Optimization phase is closely linked to PPM efforts, since changes within the application portfolio may require the implementation of huge projects.[12] Each chosen action needs to be carefully managed throughout the entire phase and kept aligned with its specific business case [Fabriek et al. 2007]. Specific experiences that may have been gained during project implementation should be made reusable for future optimization efforts [Niemann 2006].

Note that beyond a modified portfolio, the portfolio optimization may have some particular effects that have not necessarily been intended. For example, changes to the application portfolio itself may unveil business processes that can be improved or even modified completely. Optimization may also unveil opportunities for innovation. We see other possible effects as well. The application of APM may lead to an application outsourcing decision, which in turn raises the need to monitor the chosen contractors. Further, APM may be applied as part of IT due diligence in support of a merger or acquisition under consideration. Should the merger or acquisition go through, there would be a need to integrate two different application portfolios.

Application Portfolio Management Maturity

Based on the depicted APM process, this section presents our APM maturity model. Maturity models have evolved in several areas over the past years. Most are based on the "Capability Maturity Model" (CMM) for processes within the area of software development and are used to "assess the 'maturity' of ... management processes as a means to improve those processes in order to achieve organizational goals" [Benson et al. 2004]. We consider the APM maturity level as a measure of how far the APM process has gone and the extent to which APM, and IT PM as a whole, are institutionalized as ongoing approaches.

In fact, our APM maturity model uses six different levels of maturity (see Figure 2) and considers whether actions are performed once or continually. These levels are as follows: Level 0, "Application Portfolio Obscurity," is the starting point for many organizations that have lost track of their application portfolios and seek greater transparency. Collecting data about the application portfolio may lead to Level 1, "Application Portfolio Understanding," which draws on the application inventory that has been built. Thorough portfolio analysis, with resultant insights into the application portfolio, allows for moving up to Level 2, "Application Portfolio Intelligence." The phases of Decision-Making and Optimization described earlier lead to Level 3, "Application Portfolio Quality," reflecting an optimized application portfolio.

At this point, APM may have been only a one-time project effort. For an organization to avoid dropping back to Level 0, it becomes essential to institutionalize ongoing APM efforts [Maizlish and Handler 2005], i.e., continually collecting data and analyzing applications, which may lead to further optimization action. This also implies that ongoing benefits management, which measures and communicates realized benefits of both individual optimization efforts and APM as a whole, is taking place, in turn ensuring the sustained support of stakeholders [Bloem et al. 2006; Maizlish and Handler 2005]. Establishing a defined APM organization with clear roles and responsibilities and thus institutionalizing the APM process leads to Level 4, "Application Portfolio Excellence," which ensures ongoing portfolio quality and systematic decision-making regarding future investments in applications.

[12] While the previous phases focus on the enterprise level and may thus be considered making up strategic APM, the Optimization phase is carried out on a project level, representing operational APM efforts.

The institutionalization of APM may be taken even further by incorporating IT PM as a whole, on an ongoing basis. It is important to note that achieving Level 4 does not automatically mean that every pillar of IT PM is fully in place and completely integrated, despite the outlined links between APM and other areas of IT PM. An organization may not be able to take full advantage of APM until a holistic institutionalization has taken place that closely aligns the relevant areas of IT PM with business processes and requirements, resulting in ongoing, active management of the IT portfolio according to business needs. It is only then that an organization can reach the highest level of APM maturity—Level 5, "IT Portfolio Excellence."

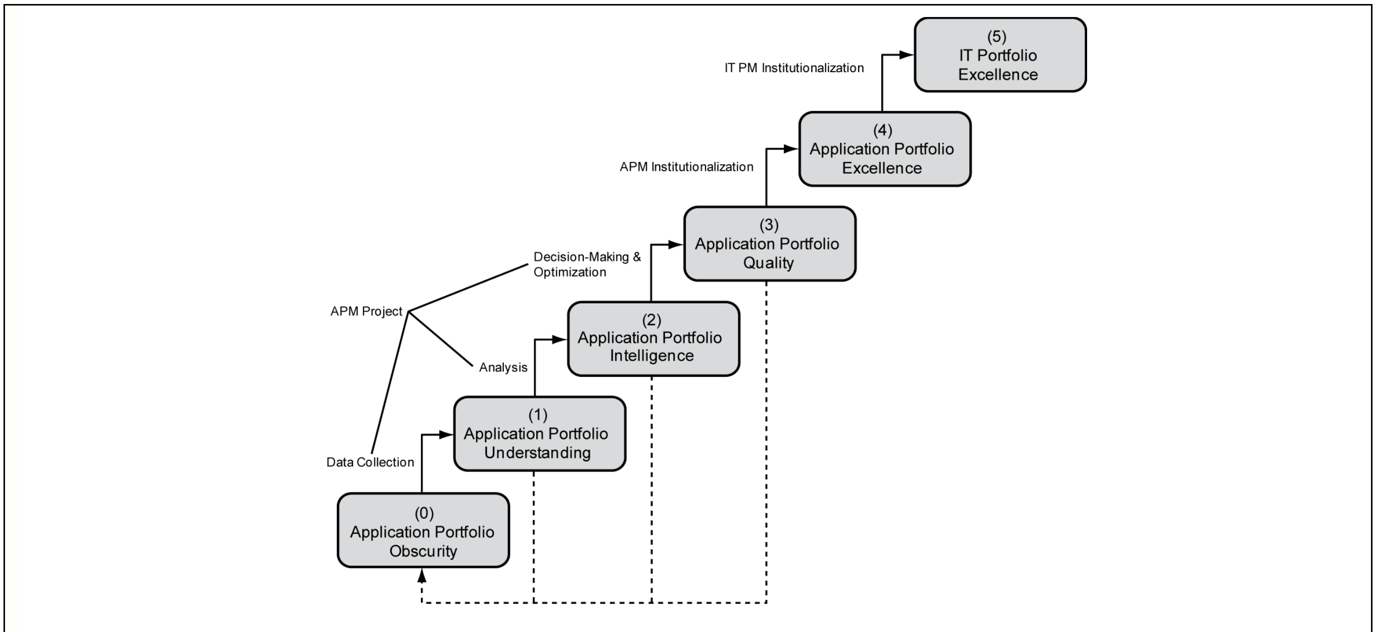

Figure 2. Application Portfolio Management Maturity Model

Figure 3 shows our integrated APM model, based on these APM maturity levels and the other components of our framework. The model lays the foundation for our examination of suitable software tools for APM.

V. EVALUATION OF SOFTWARE TOOLS FOR APPLICATION PORTFOLIO MANAGEMENT

Having established a comprehensive APM framework, the next step is to capture opportunities for practical support from software tools, which we believe to be essential to applying the framework in practice. In this section, we introduce an approach to evaluating software tools, based on the framework established earlier. We then apply this approach to some tools.

Evaluation Model

In the dedicated model we have developed to evaluate software tools from an APM perspective, evaluation dimensions are grouped into two parts corresponding to the characteristics of APM as specified within our framework. The first group, named "Scope," addresses the range and quality to which a study of the application portfolio is subject, basically reflecting the application dimensions introduced in the subsection "Analysis" in Section IV above. The second group involves specific tool capabilities that do not refer directly to the coverage of APM core dimensions, but that determine strengths in those areas and make it possible to capture those strengths to the greatest degree possible.[13]

The first group, the tools' scope, comprises the following dimensions: Business Support, Cost, Risk & Value, Technical Health, Operational Performance, Lifecycle, Functional Wealth, Compliance, Dependency, and Vendor Information. They are evaluated in terms of the extent to which they are covered and integrated by the tool out of the box, for example, through available application attributes and data processing means. An additional dimension, "Scope Flexibility," is aimed at capturing the possibilities and conditions for widening a given tool's scope.

[13] Ernst [2005] and Matthes et al. [2008] offer a similar approach, distinguishing between subject-oriented management tasks and specific tool functionalities.

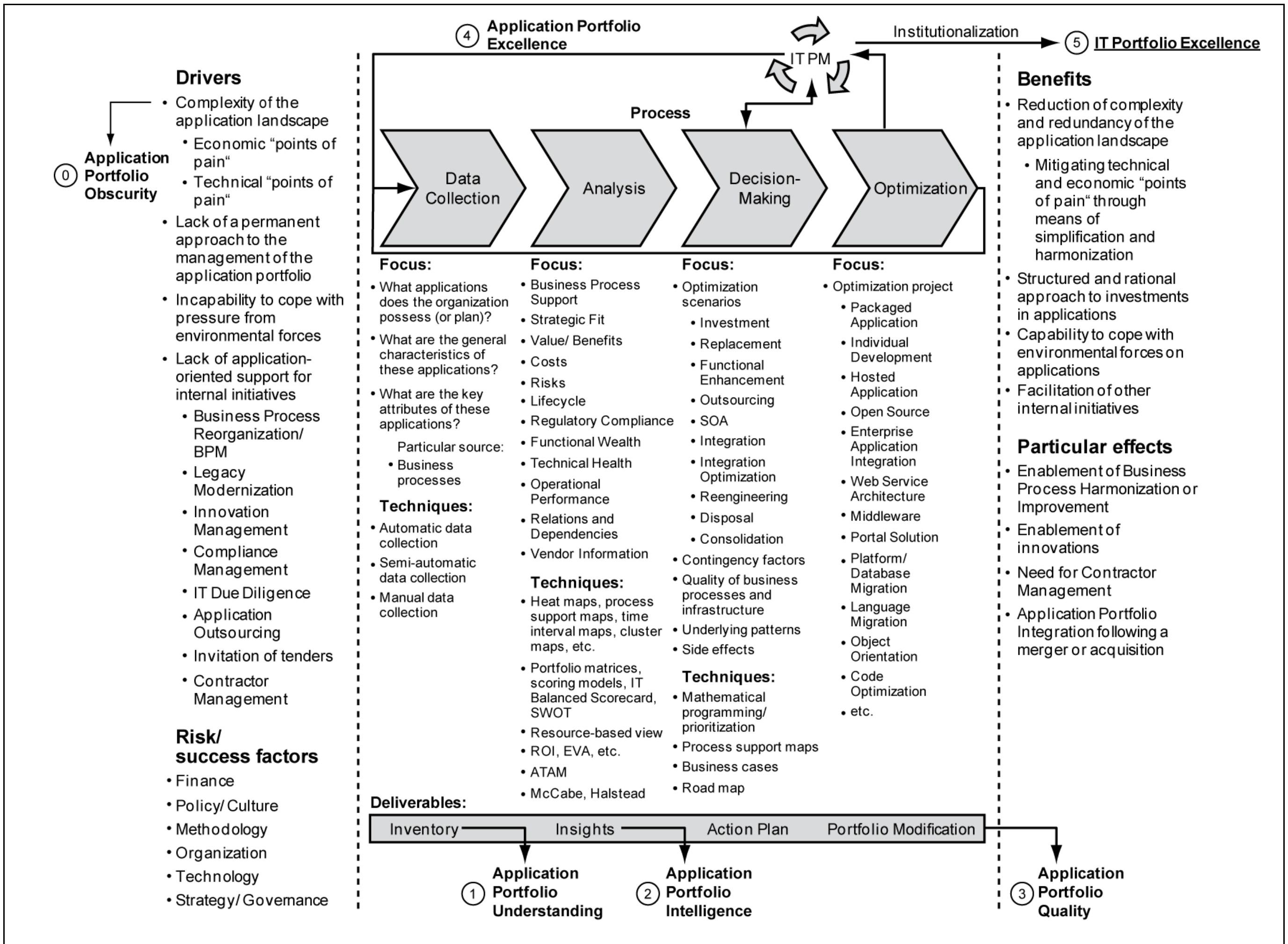

Figure 3. The Integrated APM Model

Table 1. Evaluation Model—Scope

Scope	Criteria
Business Support	To what extent are the areas covered and integrated within the tool, for example, through available attributes and means of data processing?
Cost, Risk, and Value	
Technical Health	
Operational Performance	
Lifecycle	
Functional Wealth	
Compliance	
Dependency	
Vendor Information	
Scope Flexibility	Is it possible to widen the scope, both within a certain area and to other areas? Are changes limited to minor enhancements or can major enhancements also be achieved without great effort?

Our model's second group of evaluation dimensions focuses on tool capabilities that correspond to certain APM phases as follows, omitting the diverse Optimization phase. Reflecting the alternative techniques for data collection and the enormous communication and collaboration efforts that may be required to obtain needed application information, the desired tool capabilities involve, among others, "User Participation" (e.g., online surveys) for gathering data as well as "Quality Assurance" of entered data. Moreover, a tool's ability to read source code ("Source Code Understanding") may be a significant facilitator of understanding the application portfolio, revealing applications whose existence in the portfolio may not even be known. Tools able to read source code can discover applications being operated automatically. In addition, "Modeling" capabilities can aid in capturing the current composition of the portfolio.

We expect, further, that tools can be integrated with other systems to allow data importing and exporting, which accounts for the importance of “Integratability” for both Data Collection and Decision-Making. BPM and PPM are two particular areas with which APM needs to be integrated. Thus, whether a tool can not only be integrated with dedicated BPM or PPM tools, but also handle the relevant functionalities itself, is also evaluated (while our model takes such functionality into account, we do not see it as mandatory in the APM context).

For conducting analyses, tools should offer sophisticated “Reporting” capabilities that include automated calculations, impact analysis, and visualizations such as portfolio matrices. Likewise, we consider it to be of great relevance that a tool may not only be able to gauge the present state of applications and the entire landscape, but also that it allows comparisons to previous states so that changes may be tracked. Hence, our model also includes the “Traceability of Historical Data.” Closely linked to their reporting capabilities, tools should also make it possible to map the application landscape (“Landscape Visualization”) at any given time [Keller 2007]. While this could also be integrated with Reporting, we think it should be regarded as an individual tool capability.

Beyond the ability to read source code, a tool may make it possible to delve deeply into the technical health of applications if it can derive intelligence on that source code (“Source Code Intelligence”) by calculating code quality and complexity indices. A tool that can even offer “advice” regarding specific source code improvements (“Source Code Optimization Intelligence”) is particularly useful and crosses over to tool capabilities for the Decision-Making phase. Ideally, a tool should be able to consider not only each dimension individually, but also provide an integrated picture (“Scope Aggregation”) for comprehensive decision-making. Additionally, our evaluation model considers tool capabilities that represent an “Optimization Intelligence” beyond source code issues (e.g., outsourcing). While this may require sophisticated mathematical capabilities that need to be highly customized, these capabilities should not go unconsidered. We also expect tools to be capable of weighing different optimization scenarios against one another (“Scenario Planning”). The possibility to plan and map the future landscape (and transition states) by creating future versions (“Landscape Planning”) rounds out the desired decision-making capabilities [Keller 2007].

Finally, there are two further, phase-independent capabilities we deem worthy of consideration. The first, “Role Differentiation,” reflects the users’ different needs for communication and collaboration, and addresses the access to certain information or functionality. The second, “Usability,” signifies that criteria that affect usability deserve thorough examination.

Evaluation of Selected Software Tools

In April–May 2008, we applied the model to evaluate an initial set of eight tools identified as market leaders by well-known market analysts and considered leading tools within specific tool families. This evaluation comprised three tools in the Application Portfolio Management family, four tools in the Enterprise Architecture Management family, and one tool in the Business Process Management family.[14] We designed a comprehensive questionnaire for tool vendors requesting information regarding the coverage of the tools in the context of the evaluation dimensions. The questionnaire was complemented by a reference scenario to illustrate the problem domain and reflect the stated APM drivers and objectives.[15]

The questionnaire was structured in six parts: company information, reference scenario, general tool and support information, functional characteristics, nonfunctional characteristics, and, finally, tool focus and self-concept. Most of the information that was relevant for the evaluation related to a tool’s functional characteristics, since these form the primary basis for evaluation. Further input for the evaluation was taken from answers to questions regarding non-functional performance characteristics, the main tool benefits (through a dedicated question in the “general tool and support information” section), and potential software tool support within the scope of the reference scenario. General company information, for example, was not taken into consideration for evaluation. As the information was obtained from vendors themselves, we were certainly aware of the risk of vendor bias, which can jeopardize any objective evaluation. We addressed this by inviting each vendor to present its tool in a live demonstration that was complemented with a semi-structured interview.[16] Through the interview we were able to clarify questionnaire answers and address any issues raised during the demonstration.

[14] Despite the similarity in names, these should not be equated with the similarly named concepts we have detailed earlier. Also note that tools from the same family may in fact have a different origin (e.g., one Enterprise Architecture Management tool may have dictionary roots, while another one may have roots in software engineering [Niemann 2006]).

[15] The evaluation was conducted in Germany, and hence both the questionnaire and reference scenario are in German. The questionnaire is not included here because it would exceed the paper’s adequate length.

[16] Ruling out common methods bias in the information gathering process, this approach allowed for attaining an adequate level of validity [Straub, Boudreau, and Gefen 2004].

Table 2. Evaluation Model—Capabilities

Capabilities	Criteria
User Participation	Is it possible to obtain information and evaluation from users through surveys? Are questionnaires highly preconfigured or highly configurable? What criteria and questions are provided? Are there notification mechanisms for updating out-of-date data? Are notifications initiated manually or automatically, and is the mechanism applicable only for predefined objects or for a broad set of objects?
Quality Assurance	Are auditing mechanisms available (save of meta-data, approval, etc.)? Is auditing restricted to certain predefined objects or attributes?
Modeling	Is it possible to draw models manually? Are modeling capabilities rudimentary, or are sophisticated techniques available? Is it possible to assign attributes to modeled objects?
Source Code Understanding	Is the tool capable of reading source code? Of reading many or only a few languages? Are changes within the source code tracked automatically?
Integratability	Is it possible to import/export data via Microsoft Excel or similar tools? Are interfaces available, including dedicated interfaces to certain tools? Are there mechanisms/components for building interfaces?
Reporting	What types of reports are available? Which means of visualization are provided? How customizable are reports? Is it possible to define new reports manually without great effort?
Traceability of Historical Data	Are versioning mechanisms available? Are versions created manually or automatically? Is versioning restricted to certain objects or attributes?
Landscape Visualization	Is it possible to generate different kinds of (non-technical) maps of the application landscape (similar to software maps that are used in Software Cartography)?
Source Code Intelligence	What does the tool derive from the source code: basic characteristics, basic metrics, and/or sophisticated metrics?
Source Code Optimization Intelligence	Does the tool point to certain code areas for improvement, and does it provide suitable advice? Does it recommend action for improving the source code as a whole?
Scope Aggregation	Are analysis dimensions aggregated in some way? Are sophisticated metrics available for aggregating various dimensions? Can metrics be adapted individually?
Optimization Intelligence	Are action proposals provided based on prior analysis? Is recommended action limited to individual applications or are means for optimizing the entire portfolio also taken into account?
Scenario Planning	Is it possible to compare scenarios for optimization against each other? Are further contingency factors considered within decision-making? How easily are contingency factors integrated with the decision-making process?
Landscape Planning	Is it possible to plan the evolution of the application landscape, particularly by creating future versions of objects? Is it possible to map (non-technically) the landscape's future state?
Role Differentiation	Is it possible to define different roles in terms of access to certain information and analysis? How granular can roles be configured?
Usability	Which kinds of search and help functionality exist? How does one navigate through the tool? Is there any sort of dashboard available?
BPM	Are Business Process Management functionalities provided?
PPM	Are Project Portfolio Management functionalities provided?

Armed with this information from the vendors, we then put several research assumptions and questions at the center of our evaluation. We first assumed that their modeling capabilities make Business Process Management tools [Van der Aalst et al. 2003] useful for data collection, but we questioned whether they offer any further support. Second, we assumed Enterprise Architecture Management tools to be quite suitable for mapping the application landscape, aligned with further layers (e.g., business, infrastructure), but we questioned them in terms of their approach to application dimensions beyond business support. Third, we assumed that Application Portfolio Management tools take into account the widest range of dimensions and are most useful for tackling technical complexity. The research issues raised by these assumptions were based on our expectation that the APM process would be supported by different types of tools for specific purposes that may turn out to demonstrate expertise only in certain areas.

We then analyzed the information obtained through the questionnaire and the subsequent meetings, making use of our software tool evaluation model and a four-point scale ranging from 0 to 3, with 0 indicating “not available,” 1 indicating “weakly developed,” 2 indicating “fairly developed,” and 3 indicating “strongly developed.” Having jointly developed the underlying APM framework, we considered ourselves well-prepared and trained to apply these ratings ourselves, allowing the evaluation to benefit from a high familiarity with the study context and a common understanding of the subjects of interest. Using the above coding scheme, we allocated scores for each evaluation dimension.[17] Coding was first done independently by the three authors. The results provide strong evidence of high inter-rater reliability among the raters, which was calculated in SPSS 17.0.2 as an intra-class correlation coefficient (ICC) with two-way mixed effects [Shrout and Fleiss 1979]. The raters strongly agreed in their judgments on each of the eight products, as can be seen from high inter-rater reliabilities, with the lowest value among the products amounting to ICC = .972 and the highest value even showing a perfect inter-rater agreement of ICC = 1. The few identified discrepancies were eventually resolved through intense joint discussions.[18]

The final values are rated on an ordinal scale; that is, a given tool may be rated as “better” than another tool, but this does not indicate any degree of how much better this tool is than the other one. The evaluation did not lead to a final ranking of tools; that is, the resulting values for each tool were not ranked one against another. Instead, we produced a scorecard with our evaluation model with which organizations may match their individual requirements (see the Appendix).[19]

Based on the evaluated data and the applied ratings, we created graphic overviews of the tools aimed at addressing the research issues raised above. Figure 4 is an overview of the scope of the tools in terms of their completeness and the quality with which they cover the various dimensions. While the “Completeness of Scope” (abscissa) describes how many dimensions the tool covers (number of scores > 0), the “Relative Quality of Scope” (ordinate) shows how tools are ranked according to the quality they deliver, as determined by the tools’ number of strongly developed dimensions as the first criterion and by fairly developed dimensions as the secondary criterion. This graphic illustration shows that no tool covers the entire range of application dimensions.[20]

In contrast to our assumption stated above, it is Enterprise Architecture Management tools and not Application Portfolio Management tools that show up among the top tools in terms of the range and quality of the application dimensions they address. Figure 5 also reflect this, providing an overview of the capabilities of the tools to provide high-quality support in different phases of the APM process.[21] Actually, it illustrates how tools are rated depending on how many phases they serve with at least one strongly developed capability (ordinate, named “Phase Quality”). It is interesting that the Business Process Management tool indeed provides sophisticated capabilities in only one phase, Data Collection. This is based on a strong rating of the tool’s modeling capabilities (again, we refer to the Appendix for exact evaluation scores). However, while the evaluation scores show that support capabilities in further phases are rather limited, the assumption of no support at all beyond the Data Collection phase proves to be wrong. The strong capabilities found for Application Portfolio Management tools, in contrast, stem from the source code expertise of these tools, which allows them to provide the best support for certain issues (i.e., technical), since technical application issues are barely covered by other tool families.

We draw the following conclusions with regard to the research issues raised above.[22] Business Process Management tools may, in fact, be suitable for data collection, particularly because of their modeling capabilities. Although capabilities in further phases appear rather limited, a few valuable capabilities do exist. Hence, the application range of Business Process Management tools concentrates on the Data Collection phase, although minor support is provided within subsequent phases.

---

[17] Thus, the questionnaire did not allow the vendors to provide these numerical ratings, we applied the ratings ourselves.
[18] Further information about inter-rater reliability and corresponding guidelines can be found in, for example, Palvia et al. [2003], and Straub, Boudreau and Gefen [2004].
[19] In the scorecard, tools/ratings are shown in an anonymized form, with tool names replaced by numbered abbreviations for the respective tool families.
[20] Taking into account evaluation dimensions of both “Scope” and “Capabilities,” only 21 percent of the allocated scores equal 3, while 46 percent equal 0.
[21] This positive correlation between ratings in ”Scope“ and in “Capabilities“ points to a satisfactory internal consistency reliability [Straub, Boudreau, and Gefen 2004].
[22] Of course, we do not recommend generalizing these conclusions to each and every tool within respective tool families. Assessment of additional tools will be needed to strengthen the results.

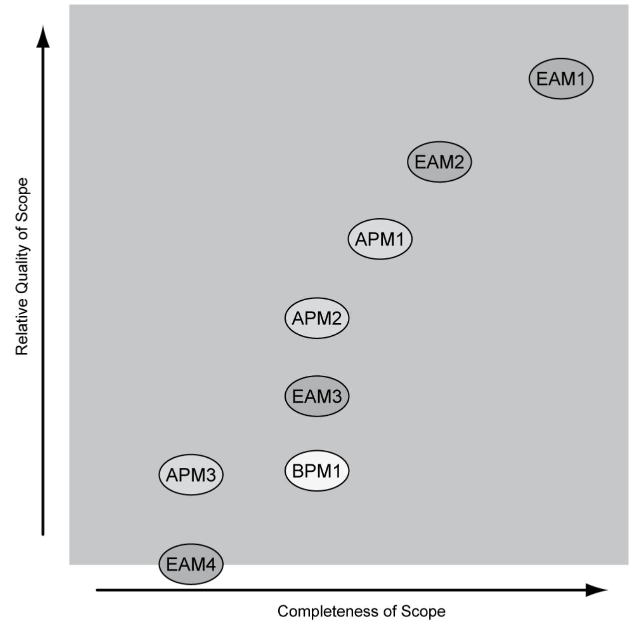

Figure 4. Tool Overview—Scope

Most Enterprise Architecture Management tools are indeed suitable for mapping the application landscape, aligned with further layers (e.g., business, infrastructure), yet they also cover further application dimensions beyond the business support, to a considerable degree, which allows for an active, decision-supportive approach to the application portfolio. Hence, Application Portfolio Management tools are outranked by sophisticated Enterprise Architecture Management with respect to their coverage of application dimensions (see Figure 4). In contrast, Application Portfolio Management tools provide the best support for addressing technical complexity.

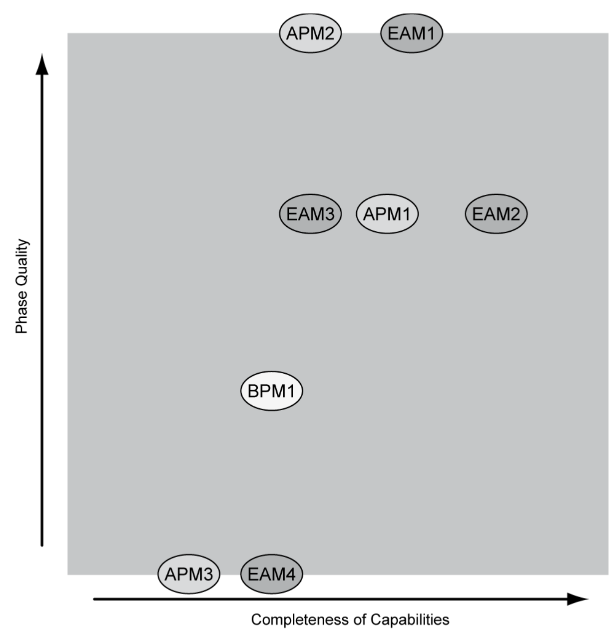

Figure 5. Tool Overview—Qualitative Phase Coverage

VI. SUMMARY AND OUTLOOK

Having examined APM from both a theoretical and, in terms of potential software tool support, a practical point of view, we now offer a brief summary and our outlook regarding potential future research. Essentially, we have developed a comprehensive APM framework where one that embraces the concept as a whole had been lacking. This framework builds primarily on the increased complexity of application landscapes and proposes a straightforward process comprising Data Collection, Analysis, Decision-Making, and Optimization phases to increase the enterprise's APM maturity. An essential future step would be to apply the approach in a number of real-life cases to verify its applicability, robustness, and accuracy and uncover areas for improvement and extension; this would include a closer examination of the Optimization phase and would provide further advice on how to prepare for and establish APM in an organization.

Taking into account that a practical application of APM may require sophisticated software tools, we also introduced an APM software tool evaluation approach, which could be bolstered in the future by adding further evaluation dimensions, or examining the dimensions from different angles or in different aggregates or clusters (e.g., Reporting and Landscape Visualization dimensions may be broken down into several sub-dimensions). Applying the evaluation model to a selection of tools, we were able to show the suitability of certain tool families for different APM purposes, with that of decision-making showing a particular improvement potential. With this difference in expertise, we also consider an integrated and combined use of more than one tool to be promising. Of course, an enterprise's decision about which tool to select remains subject to its specific needs and requirements. Further, our overall evaluation as set out in this paper could be widened to include additional tools and tool families, with a more in-depth investigation using a wider evaluation scale.

All in all, we believe this approach of applying design science to develop an integrated APM framework, and examining the software tool market to evaluate appropriate practical support, is a way to move more deeply into the research area and offers ideas and advice for researchers and practitioners alike. From an academic perspective, though, further research and empirical evidence is required to validate and extend the findings.

All too often, IT managers have been puzzled about the composition and value of their application portfolios and have been at a loss regarding what future investments they should make in the portfolio. The APM framework developed in this paper may, in the future, serve as a conceptual guideline, with a potential to be tailored to the circumstances of the individual enterprise and suit specific needs. It is a rational, decision-oriented approach to the application portfolio, thus assisting IT managers in coming to grips with the portfolio and arriving at well-founded decisions regarding, for example, which applications are worth keeping and which missing pieces must either be built or purchased. Moreover, our software tool evaluation approach can aid IT managers in applying APM means to determine which tools may best satisfy their needs and which tools are appropriate for applying our approach in practice.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Rolf T. Wigand, the Associate Editor, and Scott M. Cooper for many helpful suggestions and comments that significantly improved this paper. The authors also thank Julia Pradel and David Basten for their support and valuable comments on earlier versions of this paper.

REFERENCES

Editor's Note: The following reference list contains hyperlinks to world wide web pages. Readers who have the ability to access the web directly from their word processor or are reading the paper on the web, can gain direct access to these linked references. Readers are warned, however, that:

1. These links existed as of the date of publication but are not guaranteed to be working thereafter.
2. The contents of web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. The author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.
4. The author(s) of this article, not AIS, is (are) responsible for the accuracy of the URL and version information.

Andriole, S.J. (2007). "Mining for Digital Gold: Technology Due Diligence for CIOs," *Communications of the Association for Information Systems* (20), pp. 371–381.

Bahadur, K., D. Desmet, and E. van Bommel (2006). "Smart IT Spending: Insights from European Banks," *McKinsey Quarterly*, January 2006.

Benson, R.J., T.L. Bugnitz, and W.B. Walton (2004). *From Business Strategy to IT Action: Right Decisions for a Better Bottom Line, 1st Edition*. Hoboken, NJ: Wiley.

Betz, C.T. (2007). *Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler's Children, 1st Edition*. San Francisco: Morgan Kaufmann.

Bloem, J., M. van Doorn, and P. Mittal (2006). *Making IT Governance Work in a Sarbanes-Oxley World, 1st Edition*. Hoboken, NJ: Wiley.

Buckl, S., et al. (2007a). "A Pattern-Based Approach for Constructing Enterprise Architecture Management Information Models," *Proceedings of Wirtschaftsinformatik 2007*, pp. 145–162.

Buckl, S., et al. (2007b). "Generating Visualizations of Enterprise Architectures Using Model Transformations," *Enterprise Modelling and Information Systems Architectures—An International Journal* (2)2.

Buckl, S., et al. (2008). *Enterprise Architecture Management Pattern Catalog* (Version 1.0, February 2008), TU München, Chair for Informatics 19 (sebis).

Caruso, D. (2007). "Application Portfolio Management—A Necessity for Future IT," *Manufacturing Business Technology* (25)10, pp. 48–50.

Dern, G. (2006). *Management von IT–Architekturen: Informationssysteme im Fokus von Architekturplanung und -entwicklung, 2nd Edition*, Wiesbaden: Vieweg.

Ernst, A.M., et al. (2006). "Tool Support for Enterprise Architecture Management—Strengths and Weaknesses," *Proceedings of the 10th International Enterprise Distributed Object Computing Conference*, pp. 13–22.

Fabriek, M., S. Brinkkemper, and J. van Dullemen (2007). "A Method for Application Portfolio Rationalization," *Proceedings of the 2nd International Conference on Digital Information Management*, pp. 466–472.

Groot, R., M. Smits, and H. Kuipers (2005). "A Method to Redesign the IS Portfolios in Large Organizations," *Proceedings of the 38th Hawaii International Conference on System Sciences*.

Hafner, M., and R. Winter (2008). "Processes for Enterprise Application Architecture Management," *Proceedings of the 41st Hawaii International Conference on System Sciences*.

Hirvonen, A. (2004). "Application Portfolio Models in Practice—Tools for ICT Planning, Management and Business Value Evaluation," *Proceedings of 11th European Conference on Information Technology Evaluation*, pp. 183–192.

Hevner, A.R., et al. (2004). "Design Science in Information Systems Research," *MIS Quarterly* (28)1, pp. 75–105.

Hung, T., and P.F. Kunz (1992). *UNIX Code Management and Distribution*, Technical Report SLAC-PUB-5923, Stanford Linear Accelerator Center, Stanford University.

IT Project and Portfolio Office, University of Utah (2008), http://www.acs.utah.edu/ppo/portfolio_management.shtml (current 11 October 2009).

Johnson, P., et al. (2004). "Using Enterprise Architecture for CIO Decision-Making: On the Importance of Theory," *Proceedings of the 2nd Annual Conference on Systems Engineering Research*.

Jordan, E., and L. Silcock (2005). *Beating IT Risks, 1st Edition*. Hoboken, NJ: Wiley.

Kazman, R., M. Klein, and P. Clements (2000). *ATAM: Method for Architecture Evaluation*, Technical Report CMU/SEI-2000-TR-004, Carnegie Mellon Software Engineering Institute.

Keller, W. (2007). *IT-Unternehmensarchitektur: Von der Geschäftsstrategie zur ptimalen IT-Unterstützung, 1st Edition*, Heidelberg: dpunkt.

Kersten, B., and C. Verhoef (2003). "IT Portfolio Management: A Banker's Perspective on IT," *Cutter IT Journal* (16)4, pp. 27–33.

Kurpjuweit, S., and R. Winter (2007). "Viewpoint-Based Meta Model Engineering," *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures*, pp. 143–161.

Kwan, S.K., and J. West (2004). "Heterogeneity of IT Importance: Implications for Enterprise IT Portfolio Management," *Proceedings of the 64th Annual Academy of Management Conference*.

Lankes, J., and C.M. Schweda (2008). "Using Metrics to Evaluate Failure Propagation in Application Landscapes," *Proceedings of the Multikonferenz Wirtschaftsinformatik 2008*, pp. 1827–1838.

Leimeister, S., T. Böhmann, and H. Krcmar (2008). "IS Outsourcing Governance in Innovation-Focused Relationships—An Empirical Investigation," *Proceedings of the 16th European Conference on Information Systems*.

Longépé, C. (2003). *The Enterprise Architecture IT Project: The Urbanization Paradigm*, 1st Edition. London: Kogan Page Science.

Maizlish, B., and R. Handler (2005). *IT Portfolio Management Step-by-Step,* 1st Edition. Hoboken, NJ: Wiley.

Mårtensson, A. (2006). "A Resource Allocation Matrix Approach to IT Management," *Information Technology and Management* (7)1, pp. 21–34.

Matthes, F. et al. (2008). *Enterprise Architecture Management Tool Survey 2008*, TU München, Chair for Informatics 19 (sebis).

McCabe, T.J. (1976). "A Complexity Measure," *IEEE Transactions on Software Engineering* (2)4, pp. 308–320.

McFarlan, F.W. (1981). "Portfolio Approach to Information Systems," *Harvard Business Review* (59)5, pp. 142–150.

McKeen, J.D., and H.A. Smith (2002a). "New Developments in Practice II: Enterprise Application Integration," *Communications of the Association for Information Systems* (8), pp. 451–466.

McKeen, J.D., and H.A. Smith (2002b). "New Developments in Practice IV: Managing the Technology Portfolio," *Communications of the Association for Information Systems* (9), pp. 77–89.

Molter, M. (2005). "Die prozessorientierte Applikationslandschaft" in Scheer, A.-W., W. Jost, and K. Wagner (eds.). Von *Prozessmodellen zu lauffähigen Anwendungen*. Berlin: Springer, pp. 151–172.

Namba, Y., and J. Iijima (2004). "City Planning Approach for Enterprise Information System," *Proceedings of the 8th Pacific-Asia Conference on Information Systems*, pp. 169–180.

Niemann, K.D. (2006). *From Enterprise Architecture to IT Governance—Elements of Effective IT Management,* 1st Edition. Wiesbaden: Vieweg+Teubner.

Palvia, P., et al. (2003). "Management Information Systems Research: What's There in a Methodology?" *Communications of the Association for Information Systems* (11), pp. 289–309.

Ren, M., and K.J. Lyytinen (2008). "Building Enterprise Architecture Agility and Sustenance with SOA," *Communications of the Association for Information Systems* (22), pp. 75–86.

Riempp, G., and S. Gieffers-Ankel (2007). "Application Portfolio Management: A Decision-Oriented View of Enterprise Architecture," *Information Systems and e-Business Management* (5)4, pp. 359–378.

Rispens, M., and F. Vogelezang (2007). "Application Portfolio Management, The Basics: How Much Software do I Have?" *Proceedings of the 4th Software Measurement European Forum*.

Sherer, S.A., and S. Alter (2004). "Information Systems Risks and Risk Factors: Are They Mostly about Information Systems?," *Communications of the Association for Information Systems* (14), pp. 29–64.

Shrout, P.E., and J.L. Fleiss (1979). "Intraclass Correlations: Uses in Assessing Rater Reliability," *Psychological Bulletin* 86(2), pp. 420–428.

Silvius, A.J.G. (2006). "Does ROI Matter? Insights into the True Business Value of IT," The *Electronic Journal of Information Systems Evaluation* (9)2, pp. 93–104.

Smith, H.A., and J.D. McKeen (2003). “Developments in Practice VII: Developing and Delivering the IT Value Proposition,” *Communications of the Association for Information Systems* (11), pp. 438–450.

Straub, D., M.-C. Boudreau, and D. Gefen (2004). “Validation Guidelines for IS Positivist Research,” *Communications of the Association for Information Systems* (13), pp. 380–427.

Swanson, E.B., and E. Dans (2000). “System Life Expectancy and the Maintenance Effort: Exploring Their Equilibration,” *MIS Quarterly* (24)2, pp. 277–297.

Van der Aalst, W.M.P, A.H.M. ter Hofstede, and M.Weske (2003). “Business Process Management: A Survey,” *Proceedings of the 1st International Conference on Business Process Management*, pp. 1–12.

Van Wegen, B., and R. de Hoog (1996). “Measuring the Economic Value of Information Systems,” *Journal of Information Technology* (11)3, pp. 247–260.

Verhoef, C. (2003). “Managing Multi-Billion Dollar IT Budgets Using Source Code Analysis,” *Proceedings of the 3rd International Workshop on Source Code Analysis and Manipulation*, pp. 77–78.

Walter, S.G., and T. Spitta (2004). “Approaches to the Ex-ante Evaluation of Investments into Information Systems,” *Wirtschaftsinformatik* (46)3, pp. 171–180.

Ward, J.M. (1988). “Information Systems and Technology Application Portfolio Management: An Assessment of Matrix-Based Analyses,” *Journal of Information Technology* (3)3, pp. 205–215.

Ward, J.M. (1993). *Investment in IT—The Portfolio Approach,* 1st Edition. London: Chartered Institute of Management Accountants.

Weill, P., and M. Vitale (1999). “Assessing the Health of an Information Systems Applications Portfolio: An Example from Process Manufacturing,” *MIS Quarterly* (23)4, pp. 601–624.

Weill, P., and J.W. Ross (2004). *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results,* 1st Edition. Boston: Harvard Business School Press.

APPENDIX: SOFTWARE TOOL EVALUATION—FINAL EVALUATION SCORES

Table 3. Evaluation Model—Scores

Capabilities	Tools							
	EAM1	**APM3**	**EAM4**	**APM2**	**BPM1**	**EAM3**	**APM1**	**EAM2**
User Participation	3	0	0	0	0	0	2	3
Quality Assurance	2	0	1	0	3	3	0	2
Modeling	0	0	0	0	3	3	0	3
Source Code Understanding	0	2	1	3	0	0	3	0
Integratability	3	1	3	2	3	3	3	3
Reporting	3	1	1	2	1	1	2	3
Source Code Intelligence	0	2	1	3	0	0	3	0
Traceability of Historical Data	2	0	2	2	2	2	2	2
Landscape Visualization	3	0	0	0	1	2	0	2
Scope Aggregation	1	0	0	0	0	0	0	1
Source Code Optimization Intelligence	0	1	0	3	0	0	1	0
Optimization Intelligence	0	0	0	1	0	0	0	1
Scenario Planning	2	0	0	0	0	0	1	2
Landscape Planning	3	0	0	0	0	3	0	2
Role Differentiation	3	2	3	3	3	3	3	3
Usability	3	1	3	3	2	2	3	3
BPM	0	0	0	0	3	3	0	0
PPM	0	0	0	0	0	0	0	0
Scope								
Business Support	3	0	0	0	1	2	2	3
Cost, Risk, and Value	3	0	0	1	1	1	2	3
Technical Health	1	2	0	3	0	0	3	0
Operational Performance	2	0	0	1	0	0	2	0
Lifecycle	3	0	1	0	1	2	0	2
Functional Wealth	0	0	0	0	0	0	0	2
Compliance	3	0	0	0	0	0	0	2
Dependency	3	1	1	1	2	2	2	2
Vendor	1	0	0	0	0	0	0	0
Scope Flexibility	3	1	1	1	1	1	2	3

ABOUT THE AUTHORS

Daniel Simon is a Consultant with act! consulting and specializes in Enterprise Architecture Management and IT Portfolio Management. He is TOGAF 9-certified and holds a diploma degree in Information Systems from the University of Cologne, Germany. He has gained practical experience in several industries and within different company departments in Germany and the United Kingdom.

Kai Fischbach is an Assistant Professor in the Department of Information Systems and Information Management at the University of Cologne. He is also Associate Director of the Center for Applied Social Network Analysis (CASNA) in Cologne. Previously, he worked at the WHU-Otto Beisheim School of Management (Germany) and has been a visiting scholar at the Massachusetts Institute of Technology (USA) and the University of Illinois at Urbana-Champaign (USA). His research focuses on the formation of social and complex networks, open innovation, and the design of efficient information exchanges.

Detlef Schoder is a Professor of Information Systems and Information Management at the University of Cologne. Previously, he served as Chair of Electronic Business at WHU-Otto Beisheim School of Management. He is the author of a large number of reviewed publications, including journal articles in leading international outlets such as the Journal of Product Innovation Management, Journal of Marketing, Journal of Electronic Commerce Research, Journal of the Association of Information Systems, Harvard Business Review, International Journal on Media Management, Electronic Markets, and Communications of the ACM. Professor Schoder has worked not only in Germany, but also in the United States, Republic of Kazakhstan, and Japan. He has been an invited Visiting Scholar at Stanford University, the University of California, Berkeley, and the Massachusetts Institute of Technology (MIT).

Communications of the Association for Information Systems

ISSN: 1529-3181