Enterprise Architecture
Vol. 8, No. 9

CUTTER CONSORTIUM

The Hitchhiker's Guide to Enterprise Architecture Roadmapping

by Sebastian Konkol; with contributions from Wojciech Ozimek, Bartosz Kiepuszewski, and Borys Stokalski, Senior Consultants, Cutter Consortium

This *Executive Report* presents a technique that supports the development of the enterprise architecture roadmap. Our goal was to design an agile approach that can be seen as a core process for enterprise architecture management that can be used as is or modified as needed. The report concludes with a review of case studies in which the principles of enterprise architecture roadmapping (EARM) have been successfully adopted.

Executive Report

Cutter Business Technology Council

Rob Austin | Tom DeMarco | Christine Davis | Lynne Ellyn | Jim Highsmith | Tim Lister | Ken Orr | Lou Mazzucchelli | Ed Yourdon

About Cutter Consortium

Cutter Consortium is a truly unique IT advisory firm, comprising a group of more than 100 internationally recognized experts who have come together to offer content, consulting, and training to our clients. These experts are committed to delivering top-level, critical, and objective advice. They have done, and are doing, groundbreaking work in organizations worldwide, helping companies deal with issues in the core areas of software development and agile project management, enterprise architecture, business technology trends and strategies, enterprise risk management, metrics, and sourcing.

Cutter offers a different value proposition than other IT research firms: We give you Access to the Experts. You get practitioners' points of view, derived from hands-on experience with the same critical issues you are facing, not the perspective of a desk-bound analyst who can only make predictions and observations on what's happening in the marketplace. With Cutter Consortium, you get the best practices and lessons learned from the world's leading experts; experts who are implementing these techniques at companies like yours right now.

Cutter's clients are able to tap into its expertise in a variety of formats including content via online advisory services and journals, mentoring, workshops, training, and consulting. And by customizing our information products and training/consulting services, you get the solutions you need, while staying within your budget.

Cutter Consortium's philosophy is that there is no single right solution for all enterprises, or all departments within one enterprise, or even all projects within a department. Cutter believes that the complexity of the business technology issues confronting corporations today demands multiple detailed perspectives from which a company can view its opportunities and risks in order to make the right strategic and tactical decisions. The simplistic pronouncements other analyst firms make do not take into account the unique situation of each organization. This is another reason to present the several sides to each issue: to enable clients to determine the course of action that best fits their unique situation.

For more information, contact Cutter Consortium at +1 781 648 8700 or sales@cutter.com.

The Hitchhiker's Guide to Enterprise Architecture Roadmapping

ENTERPRISE ARCHITECTURE ADVISORY SERVICE

Executive Report, Vol. 8, No. 9

by Sebastian Konkol; with contributions from Wojciech Ozimek, Bartosz Kiepuszewski, and Borys Stokalski, Senior Consultants, Cutter Consortium

At a recent IT conference, one of the speakers — an experienced industry analyst — defined the term "enterprise architecture" as a concept, which makes business decision makers immediately lose interest in further conversations with CIOs. This remark is a good illustration of the communications deficiencies that are plaguing the territory of business-IT alignment. It is as if technology management is protected by some kind of Douglas Adams–type SEP field, a protective shield that can cover a "bizarre and unbelievable scene so that the unconscious minds of the observers instantly abdicate responsibility for its existence."[1]

Similarly, the growing tendency to outsource any activities related to information technology indicates that decision makers are more comfortable managing business relationships with an outsourcer than managing technology as a business asset. However, this fact seems to be a paradox: with decades of exponential growth of IT capabilities still ahead, the idea of turning technology investments into Somebody Else's Problem is not a viable proposition, as the following three reasons explain:

1. The development of technology (materials, processors, networks, power cells, wireless devices, etc.) systematically delivers disruptive business capabilities and creates new market niches. We cannot foresee exactly what business changes will be triggered by technology advances, but some will.
2. The technology lifecycle is to a large extent independent from the evolution of markets that

[1] A definition of the Somebody Else's Problem (SEP) field, from *The Hitchhiker's Guide to the Galaxy* by Douglas Adams, can be found on Wikipedia (http://en.wikipedia.org/wiki/Somebody_Else%27s_Problem_Field): "An SEP field can be erected on, or projected around a bizarre and unbelievable scene so that the unconscious minds of the observers instantly abdicate responsibility for its existence, assert that it's 'somebody else's problem,' and therefore don't perceive it at all. The primary example of this was given in the third book *Life, the Universe and Everything*, when a UFO … landed in the middle of a cricket ground during a match, and the assembled crowd failed to notice it. Another prime example is when [a] ship's field is extended so that the characters fail to notice the fact that they cannot breathe or the fact that the asteroid that they are standing on does not have enough gravitational force to hold them down. The SEP field requires much less energy than a normal invisibility field (a single flashlight battery can run it for over a hundred years) due to the natural propensity of humans to see things as Somebody Else's Problem."

implement solutions based on technology and exploit technology advancements (such as modern financial services and telecommunications). Therefore, any strategy linking markets with technology has to analyze and exploit the relationships between what is feasible and what is possible. It is important to understand events in both market and technology evolution as they are equally important sources of business value or business risk.

3. The key strategic motives for competing in the 21st century — time-based competition, knowledge work productivity, business agility — rely on technology innovations and efficient management of enterprise architecture (EA).

These arguments may appear insufficient to many decision makers who seem to abdicate their responsibility for technology management as if the subject indeed was something bizarre. Nevertheless we believe this attitude toward IT signals an important transformation of corporate computing rather than its end (recently announced by the famous IT-buster Nicholas Carr in the Spring 2005 *MIT Sloan Management Review* article "The End of Corporate Computing").

As the rate of change in business continues to increase, technology management is clearly being divided into two domains. The first domain is where technology becomes part of the business services, products, and management processes. In this domain, technology management must become integrated — not aligned — with business management, becoming one of the assets that is consciously used as a strategic resource. The second domain is that of solutions and technologies, which (using the rhetoric borrowed from Carr) are essential to business continuity but bear little or no strategic consequences. In this area, technology becomes an infrastructure, and the technology management is exchanged with sourcing. Additional dimension to this picture is added by the fact that the components of enterprise architecture constantly migrate from strategic to commodity domains. The relatively short lifecycle of IT solutions and products is further enabled by the current growth of service-oriented architectures (SOAs) in both custom system development and enterprise application vendor strategies.

The transformation of corporate computing can be visualized with the help of a grid defined by Warren F. McFarlan — Cutter Consortium *Summit 2005* keynoter and Baker Foundation Professor and Albert H. Gordon Professor of Business Administration Emeritus at Harvard Business School — plotting components of enterprise information architecture across two dimensions: strategic dependence (how much are business operations and decision cycles dependent on the availability and functionality of IT components) and strategic impact (how big is the current competitive advantage provided by the capabilities delivered by IT components). (See Figure 1.)

The important question is which management practices are appropriate for organizations that experience the effects of transformation. The main coordination mechanisms implemented in recent years to manage corporate IT have been focusing on aligning IT capabilities to business requirements by dictating cost structures, service levels, and deadlines. The transformation of corporate computing requires a more diversified approach to be developed, resulting in sourcing-oriented practices in the commodity domain and integrated business-IT planning practices in the strategic domain. The strategic domain is where enterprise architecture

The ***Enterprise Architecture Advisory Service Executive Report*** is published by Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA. Tel: +1 781 641 9876 or, within North America, +1 800 492 1650; Fax: +1 781 648 1950 or, within North America, +1 800 888 1816; E-mail: service@cutter.com; Web site: www.cutter.com. Group Publisher: Kara Letourneau, E-mail: kletourneau@cutter.com. Managing Editor: Cindy Swain, E-mail: cswain@cutter.com. Production Editor: Linda M. Dias, E-mail: ldias@cutter.com. ISSN: 1530-3462.

roadmapping (EARM) takes its role. This approach was detailed in the *Executive Report* "Applying EA Roadmapping: An SOA Roadmap" [5].

In this *Executive Report,* we present a technique — a set of steps to be followed and artifacts to be used — supporting the development of the enterprise architecture roadmap. The approach presented is not intended to be a substitute for formal enterprise architecture management (EAM) methodologies. Our goal was to design an agile approach that can be seen as a "core" or "governing" process for EAM. This core process can be extended and formalized if needed or can be used as is to achieve quick, but valuable, results. To give the reader some idea of how the approach can be used in various contexts, at the end of the report we present case studies of projects where the principles of EARM have been successfully adopted.

THE QUESTION

The "correctness" of the idea of enterprise architecture management very often is not a good enough argument for investing in a management practice when you are confronting the fear of bureaucracy, uncertainty about goals, and doubt about benefits. While in some cases such concerns could be valid, this does not necessarily have to be true in general. Let us borrow another metaphor from the world created by Douglas Adams: the Ultimate Computer.[2] The Ultimate Computer is a powerful IT artifact designed to deliver the "Ultimate Answer" — explaining Life, the Universe, and Everything. It did its job well, delivering the answer, which happened to be "forty two."[3] It was then that its makers realized that the answer makes no sense unless you know precisely what it is you are asking. And coming up with the right question is often much more difficult than coming up with the right answer. Similarly, admirable concepts such as enterprise architecture frameworks created by thought leaders such as John Zachman or Bernard Boar are only successful to the extent that the purpose and goals for their implementation are well defined and understood. So before explaining details about the EARM process, it is worthwhile to understand what kinds of questions this "answer" is addressing.

Enterprise architecture roadmapping is the discipline of planning the evolution of enterprise architecture in a way that anticipates and enables business changes and maximizes the opportunities provided by the technology innovations. The process is

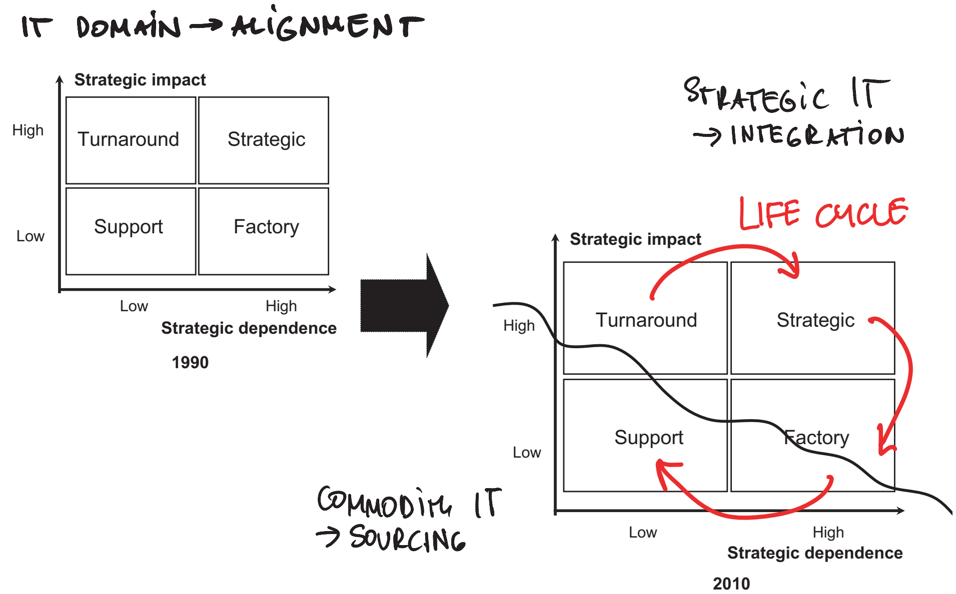

Figure 1 — Transformation of corporate computing.

[2] For those unaware, according to Adams's *Hitchhiker's Guide to the Galaxy*, researchers from a pan-dimensional, hyper-intelligent race of beings construct Deep Thought, the second-greatest computer of all time and space, to calculate the answer to the Ultimate Question. After seven and a half million years of pondering the question, Deep Thought provides the answer: "Forty-two."

[3] A curious reader might try the query "what is the answer to life the universe and everything" (enter exactly as written here, but without the quotes) with Google or MSN Search. Apparently the Web is as good as the Ultimate Computer.

intended to facilitate the communication necessary to integrate business and technology management. It is inspired by research and a growing practice of technology roadmapping, which encompasses methods, techniques, and tools for strategy development in environments with high rates of change. It is not intended to replace architecture frameworks or EAM methodologies, but rather serves as a technique that enables their efficient implementation. EARM consists of the following three components:

1. A planning framework, which lays out the key "dimensions of knowledge."
2. A roadmapping technique (or process), which implements the planning framework.
3. Roadmap templates, which are focused on the key themes in enterprise architecture evolution. Templates support the quick creation of company-specific architecture roadmaps.

While our previous Cutter report on EARM focused on the first and third aspects of the approach above [5], this report gives a more detailed overview of the EARM technique (process) and presents case studies illustrating the actual implementation. The process is focused on extensive communication involving business and technology experts providing the following critical ingredients:

- **Shared area of knowledge —** basis for communication; common understanding of fundamental concepts
- **Common means of communication —** agreed artifacts allowing the use of shared concepts in order to build new ones
- **Method for dealing with problems —** accepted way of proceeding to solve a defined problem

The EARM technique forms a framework for running communication, enabling integrated business and technology management. It structures communication as well as forms a language necessary for communication. But most importantly, its relative simplicity and iterative characteristics allow it to receive valuable results without falling into excessive and time-consuming activities.

SHARED VISION

Achieving such results requires some shifts in the basic concepts of describing a company's information environment and its dynamics. Traditionally, the information environment has been described by modeling its information systems, applications, and business processes. Moreover, a business user has been forced to understand and use such a description method in communicating with IT; otherwise, IT could not fulfill the business user's requirements.

As long as such effort was to be done only once, it was regarded as a necessary evil. Unfortunately, business processes have become so volatile and information systems so complex that the method becomes unproductive: its use requires huge overhead just for the maintenance of models. This slows down, or even inhibits, the communication necessary for an integrated planning of business *and* technology. In the following sections (and in the accompanying sidebars), we present practices used in EARM to speed up and simplify the communication. Cutter Consortium Senior Consultant Scott Ambler's *Executive Report* "An Agile Approach to Enterprise Architecture" provides useful advice concerning the way modeling techniques can be used in an agile, adaptive manner [1].

Business Cycle

The process defined as a sequence of interrelated tasks presents only one possibility that the company should be geared up for. What builds the company's competitive advantage is its ability to reorganize tasks, or groups of tasks, based on an emerging business requirements context and according to its time frame. It means the ability to find and define business tasks' modules, which could be executed in the same manner regardless, to some extent, of the context. Moreover, within each module, the company should be able to make decisions based on gained experience (see sidebar "Capturing Business Processes"). Based on such an approach, the company's business would be perceived as running many interrelated business cycles, each having a purpose and regular set of steps. Additionally, some form of

CAPTURING BUSINESS PROCESSES

PROCESS FRAMEWORKS

Thomas Davenport, in a recent *Harvard Business Review* article [3], argues that effective outsourcing requires the development of a "broad set of process standards" that will enable "plug-and-play" outsourcing. There are areas where such standards are indeed emerging, a good example being the Supply-Chain Operations Reference (SCOR) model developed by the Supply-Chain Council. Such process frameworks can be used in various contexts as the inventory of activities that can be either quickly evolved into specific models or used as is for various cross-reference activities (such as identification of EA components supporting a given business area).

VALUE CHAIN TEMPLATES

Many approaches to strategic management define a simplified yet comprehensive model of business activities in the form of value chain templates. Such templates are included in methods promoted by such management gurus as Robert Kaplan and David Norton [4] and Michael Porter (the value chain framework [8]).

An important class of value chain templates are the templates describing various kinds of end-to-end business cycles (such as order-to-cash, requirement-to-resource, or new product development). These may be very useful tools for analyzing how information architecture affects the capabilities related to time-based competition.

SERVICES: FOCUSING ON BUILDING BLOCKS

In an organization where processes and organizational charts often change, the focus on "building blocks" may sometimes enable us to capture a portfolio of activities that serve as "bricks" from which processes are constructed. These often come in the form of services. This kind of approach to process modeling may be useful for front-office organizations (such as customer service and sales) and internal units that are defined as "shared services centers." Services can be captured quickly and relatively easily with a small subset of standard UML modeling techniques (e.g., use cases and component models). It may also be useful for immature organizations that have not developed internal rules and procedures but have a clearly defined interface to their environment.

BUSINESS RULES MODELING

Modeling processes as workflows or document flows is time-consuming in part because a process model often comprises a large amount of heavily interrelated diagrams and definitions, especially when there are many exceptions and potential ways of doing things. These can make even a relatively simple process look complicated and require a lot of effort to build. An alternative approach comes from the business rules community. Rules are a powerful yet simple method for defining the way business activities are performed. Arguably, business rules are also more powerful — considering the underlying semantic and formal aspects — than most workflow modeling languages. At the same time, business rules can be directly mapped on the technical architecture with the use of specialized middleware — rule engines. The price tag for this approach is related to the fact that the simplicity of defining individual building blocks for processes (the rules) needs to be balanced by the complexity of managing large sets of interrelated rules.

ORGANIZATIONS AS VALUE NETWORKS

Research in the area of social networks reveals the hidden structure dwelling under the cover of an official org chart. Many knowledge-based organizations are de facto value networks, comprising groups, teams, and units that exchange services and information in order to create value for the customers. The approaches listed above are well suited for modeling such environments, by defining service portfolios and associating them with rules that define their internal and external behavior.

feedback should be included in each cycle to allow decisions to be made based on gained experience.

The detailed analysis and definition of business processes is in most cases a very tedious exercise, which, from the viewpoint of results expected from enterprise architecture management, often proves to be unproductive. This is not to say that having a detailed business process map doesn't deliver any advantages; it is only that, given the complexity and the rate of change in modern business, the effort required to define models that truly reflect business practices and intentions and maintain their currency is in most cases too large to justify the potential benefits.

It is therefore necessary to use a tool that gives a simplified yet good enough view of how business activities are organized and supported by IT services.

BUSINESS CYCLE CLASSIFICATION

Business cycles are the equivalent of end-to-end processes: sequences of activities, triggered by a business event, that deliver some business value. Often-cited examples of business cycles include:

- Sales cycle (order to cash)
- Procurement cycle (procure to pay)
- Service cycle (demand to service)
- New product/service development cycle (concept to market)
- Innovation cycle (idea to implementation)

Our experience shows that a simplified view of end-to-end processes (aka business cycles) proves to be good enough even in fairly complex environments such as a telecommunications company or a large government agency.

Let's now take a closer look at business cycle anatomy. The most important characteristic is its timing — the time frame in which the individual cycle iteration is to be completed, as required by business constraints. Each business cycle represents a sequence of tasks being done for a particular purpose (see sidebar "Business Cycle Classification") and within a particular time frame.

The adaptive process model we use in the business cycle analysis — the so-called observe, orient, decide, and act (OODA) loop — comes from US Air Force Colonel John Boyd, who used it to present concepts of maneuver warfare. Boyd disseminated his ideas through numerous briefings to members of the defense community. The content of these briefings has changed over time, echoing the evolution in Boyd's thinking about competitive strategy. OODA has been introduced in briefings titled "Patterns of Conflict" and "The Essence of Winning and Losing," later becoming one of the focal points for a series of briefings called "A Discourse on Winning and Losing" [2].

OODA and maneuver warfare have been adopted in modern military thinking as well as in business and information technology. One of the most interesting applications of OODA to strategic IT planning comes from Boar, who has for many years been among the pioneers of enterprise architecture management (see Figure 2).

The components of OODA — observe, orient, decide, and act — are the four critical areas of any adaptive process based on a "sense and respond" model. The flow of information among these areas, along with actions specific for each of the areas, creates a series of loops shaping the behavior of an entity in a competitive environment, be it a fighter pilot engaged in a dogfight or an organization involved in a highly competitive market.

Several of Boyd's observations are worth considering (these come from Boyd's writings and from the briefings as they are reported by his biographers and acolytes):

- From the perspective of the OODA model, the key to winning a competitive game is to be able to execute the OODA loop faster than the opponent or competitor.
- The heart of the OODA loop is the orientation phase; it is here that data is transformed to a meaningful picture of reality enabling meaningful actions to be taken in order to exploit opportunities or take evasive

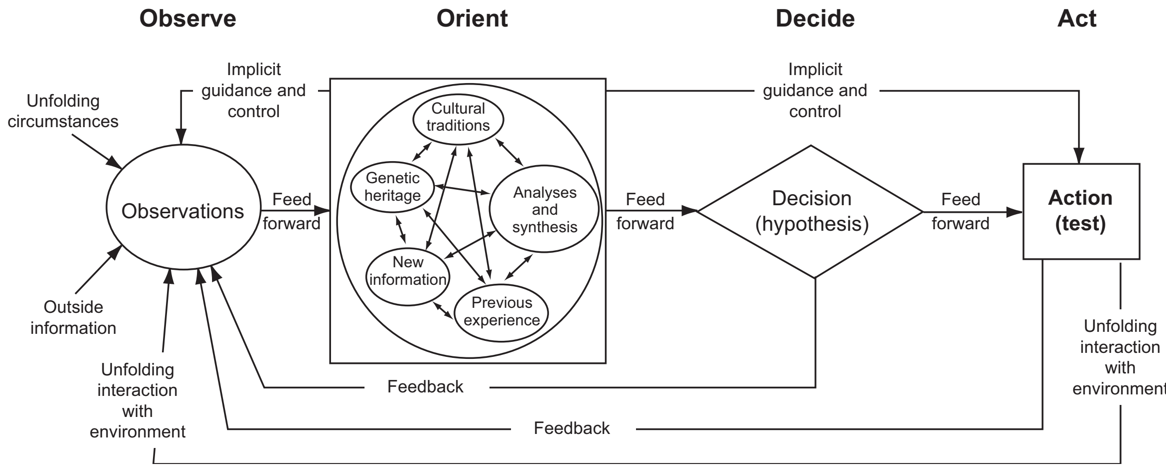

Figure 2 — OODA: the adaptive loop [2].

actions to mitigate risks. The orientation phase exploits various mental and formal models to evaluate information and filter events on which some action should be taken.

- The fastest loops are those that do not require explicit decision making. Implicit guidance and control or reflex-like behavior that can be achieved through automation, training, and distribution of the decision-making process lead to the best competitive performance, as long as the "reflexes" are relevant.

- The reasons why even the best models become obsolete stem from changes in the environment and from the changing rules of competition. An opponent may fly a better aircraft next time or might be better trained. A competitor may copy and improve our ideas and products, change, or successfully use emerging technology to disrupt the entire market, changing the basis of competition.

If orientation is the heart of the adaptive loop, then thorough understanding of the evolving requirements and motivations concerning various aspects of enterprise architecture (such as flexibility, cost of ownership, and information quality) is needed to maintain the business value of the technology investment portfolio. The roadmapping process can prevent a situation in which the IT legacy becomes unmanaged — a situation where the constraints created by technology overshadow any benefits and opportunities. Such a situation — a "bad legacy" — represents a friction in the architecture, impairing the agility of an enterprise. The friction manifests itself in issues such as the need for reconciliation of inconsistent or absent data, growing integration spaghetti, unwanted latency, and so on.

VALUE CHAIN

In reference to Figure 2, one cycle's "act" task results could be a trigger for another cycle's "observe" task. This defines a way business cycles can cooperate — a way of describing volatile business rules. For example, let's consider two cycles: demand to service and order to cash. The first observes the market to create new services; the second observes service requests to ensure they are paid. Each time a new service is created in the first cycle, it should be included in the set of services observed by the second cycle. Such relations among business cycles create a value chain in which each cycle in the chain adds value to the organization (see Figure 3).

The value chain presents a good view of business rules dynamics in the enterprise and thus could be regarded as an applicable means of describing dynamics

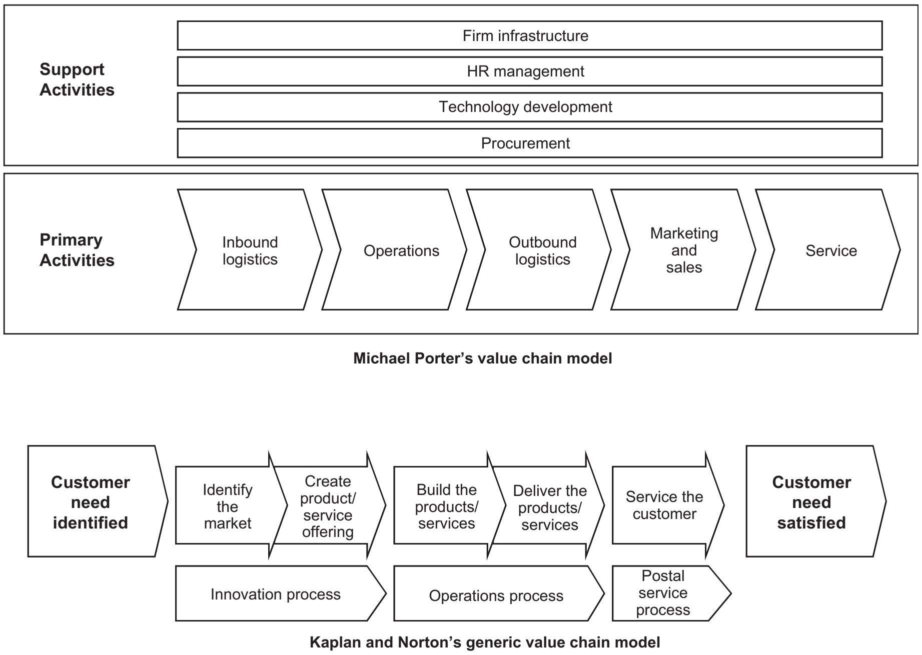

Figure 3 — Value chain examples.

instead of business processes. This is the first ingredient of efficient communication between business and IT.

Logical Architecture

The business is not concerned with systems technology, database structures, or data models. Instead, business users quite easily understand system functions, simplified to some extent, and use them quite appropriately. Describing each information system by means of its main functions — or rather services — is sufficient to give business users a view of the information environment that allows them to communicate unambiguously. The set of services offered by each information system forms a base for the logical architecture.

From the EA point of view, a logical architecture represents the most precise level of detail in the architecture description on which its elements' categorization (grouping and layering) is still possible. This categorization should be based on elements' features — for example, presentation features should be grouped into the presentation layer, while business logic features should be grouped in the business logic layer. This is the level on which most architecture patterns are applicable. Logical architecture elements (architecture blocks) should describe precisely the purpose of a particular service and its features creation, refraining from talking about a particular technology or software package. Since logical architecture is the EA language, this level of abstraction should be of the highest importance to enterprise architects.

Since understanding logical architecture components and the overall logical architecture concept forms a good basis for communication, it could be regarded as the second ingredient of efficient communication between business and IT. In combination with the business cycles concept, the overall information environment could be understood by business users and thus create communication language. In this language, services defined within the logical architecture are being used within business cycles, and all architecture features important for business become visible, especially capabilities and constraints.

As an example for the language use, consider real-time rating requirements for a telecommunications operator. The rating process assigns a price to each telecom service usage data. It requires the collection of call data in the form of call data records (CDRs) from network elements. These CDRs are processed in order to be presented to the telecom customer. As the collection cycle is the only one that cannot be run in a real-time manner, this is the place where development effort should be put. Such statements are understood to business users without lengthy discussions on rating systems capabilities, and they are understood to technical people as well.

COMMON ARTIFACTS

Business cycles and logical architecture should be used as means of communication with the business, but the result is not yet business-IT alignment. Staying within the set of concepts, the goal for architecture management should be closing and shortening business cycles according to business requirements. So the questions to be answered could be stated as follows:

- How should we build cycles? Which components of logical architecture should be used to achieve the desired result?
- How should we close cycles? How do we connect components of logical architecture to achieve feedback?
- How should we shorten cycles? What cycle elements should be modified to achieve the required timing?

The answer to each of these questions could be as straightforward as the real-time rating example presented above. What's most important is that both parties — business and IT — understand the change required in the same manner and do not force the other party to get deeply into its area of expertise. So the goal for communication between business users and IT would be answering questions like the ones above.

To sum up, with reference to the generic EA planning framework, the goal of IT communication with the business is to understand *why* and to define *what*. To be precise, the means of communication described above touches on two aspects of enterprise architecture management:

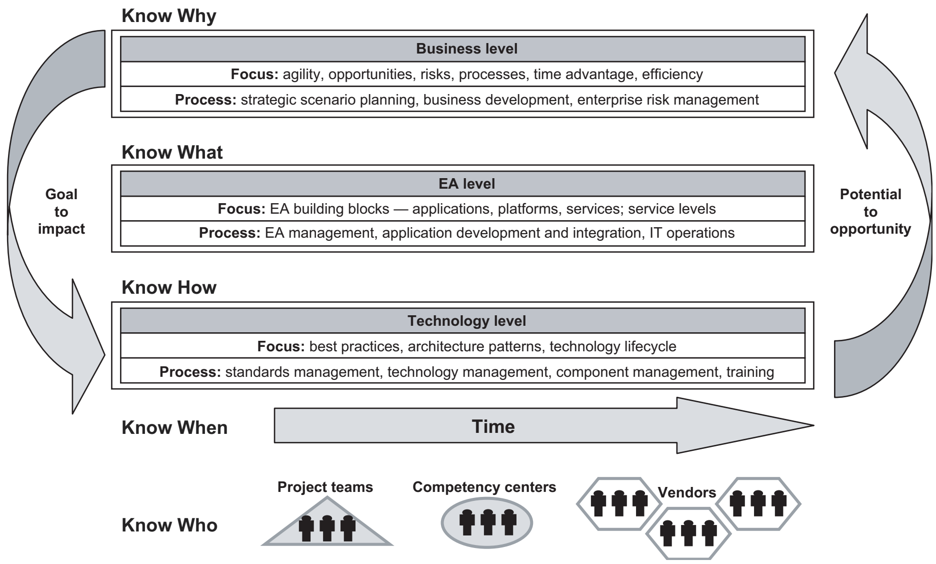

Figure 4 — EA planning framework.

1. **Know why** (business intentions, objectives, and imperatives) — understanding the cycles, their relations, and the value chains these relations create; we call it business architecture.
2. **Know what** (EA, its components, and capabilities) — describing architecture components, services offered to the information environment; we call it logical architecture.

Furthermore, in order to define EA, there are three remaining aspects that should be determined:

3. **Know how** (patterns, standards, and tools) — determining the way architecture capabilities should be implemented; we call it technical architecture.
4. **Know when** (timing) — assessing the time relations time frames for rollout of abovementioned aspects.
5. **Know who** (organization, people, and partners) — assigning roles and responsibilities to applicable activities.

The EA planning framework is derived from the technology planning framework presented in Robert Phaal, Clare Farrukh, and David Probert's "Fast-Start Technology Roadmapping" [6] and shown in Figure 4.

What any business would certainly request as the result of interrogation is a high-level plan detailing when its requirements would be fulfilled — a "schedule" showing IT initiatives being taken to achieve business objectives marked with delivery time and the "relations" describing how the initiatives support business objectives. This is an almost complete definition of the technology roadmap.

According to Phaal, Farrukh, and Probert:

> The generic roadmap is a time-based chart, comprising a number of layers that typically include both commercial and technological perspectives. The roadmap enables the evolution of markets, products and technologies to be explored, together with the linkages between the various perspectives. [7]

Roadmaps can take various forms (see sidebar "Examples of Various Roadmap Visualization Forms"), from which the service/capability planning roadmap is closest to the EA planning framework concepts.

Hopefully, at this stage, the applicability of roadmapping in communication between business and IT is not being questioned. Now, it is high time to get to the EARM technique and present the steps by which IT understands why, describes what, determines how, and assesses when.

The EARM technique description is based on some basic concepts and is described using a step-by-step approach — from determining the list of modifications in EA, through sequencing them in alignment with business objectives, to creation of the roadmap itself.

Before going deeper into the method, it should be stressed that the method is iterative, and the main result — the roadmap — should be regarded as a living document. Each shift in business objectives or architecture capabilities could result in running the next EARM technique, resulting in a roadmap update.

ROADMAP DEVELOPMENT METHOD

As presented in Figure 5, the EARM method consists of the

EXAMPLES OF VARIOUS ROADMAP VISUALIZATION FORMS

As presented in [5], depending on its purpose, the roadmap can take some visual forms. For instance, consider the following examples:

- **Product planning roadmap.** This is by far the most common type of technology roadmap, relating to the insertion of technology into manufactured products, often including more than one generation of the product. Consists of the product layer and technology layer. Relations between the two layers show the link between planned technology and product developments.
- **Service/capability planning.** Similar to the product planning roadmap but more suited to service-based enterprises, this focuses on how technology supports organizational capabilities. Consists of triggers, business and market drivers, capabilities to meet drivers, and technology developments layers. Shows organizational capabilities as the bridge between technology and the business, rather than products.
- **Strategic planning.** This includes a strategic dimension, in terms of supporting the evaluation of different opportunities or threats, typically at the business level. Consists of market, business, product, technology, skills, and organization layers. Focuses on the development of a vision of the future business in terms of markets, business, products, technologies, skills, culture, and so on. Gaps are identified, and strategic options are explored to bridge the gaps.
- **Knowledge asset planning.** This is used to align knowledge assets and knowledge management initiatives with business objectives. Consists of knowledge-related processes, knowledge management enablers, leading projects and actions, business objectives, and knowledge asset layers. Enables organizations to visualize their critical knowledge assets and the linkages to the skills, technologies, and competences required to meet future market demands.
- **Program planning.** This is the implementation of strategy and more directly relates to project planning (e.g., R&D programs). Consists of technology developments, key decision points, project milestones, and project flow layers. Shows the relationships between technology development and program phases and milestones.
- **Integration planning.** This centers on integration and/or evolution of technology in terms of how different technologies combine within products and systems, or to form new technologies (often without showing the time dimension explicitly). Focuses on technology flow, showing how technology feeds into systems, to support goals.

following generic steps driving the roadmap development:

1. Capturing current architecture definition
2. Creating architecture vision
3. Determining list of required architecture changes
4. Prioritizing changes
5. Adding time scale to the prioritized list

As a result, a target roadmap is being created. Each step in the method is discussed in more detail in the following sections with respect to its goals, its description, tools and techniques used, and the result or expected outcomes.

1. Capturing Current Architecture Definition

In this step, the current architecture should be documented to be available for comparison with the architecture vision.

Goal

The goal of this step is to understand the current architecture in terms of logical and technical architecture components. For the EARM process, it is typically not necessary to obtain a complete IT infrastructure assets inventory. The documentation created in this step follows the principles of agile modeling; that is, it should be barely sufficient to allow future matching of existing IT capabilities with the desired business vision.

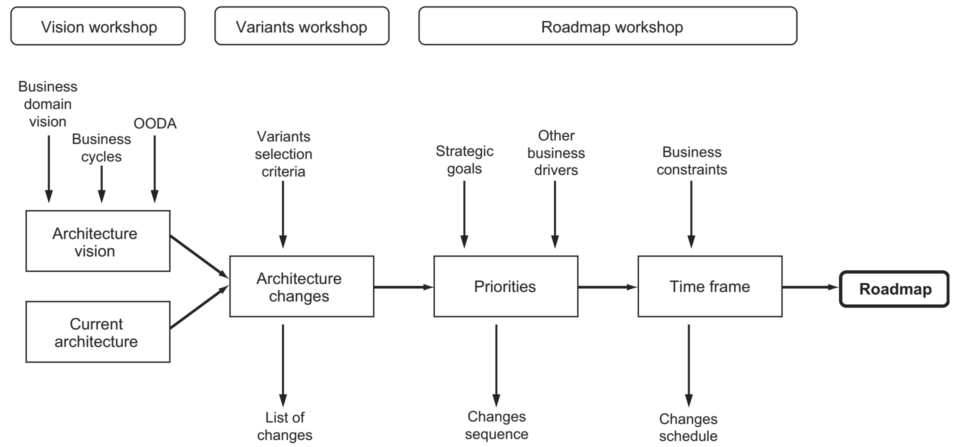

Figure 5 — EARM method.

Description

For the EARM method, the description of the current architecture includes both a logical and technical view. The logical architecture of the current IT infrastructure comprises existing components grouped according to business features they perform. It focuses on logical components and abstracts from concrete implementations; it presents a "relational database" instead of a "database provided by supplier X." It is important at this stage to acquire a thorough understanding of the dependencies between different architecture components. If it is possible, the logical architecture could present services being offered by current IT systems.

The technical architecture presents technical components of the IT infrastructure currently in operation (e.g., particular [named] application servers or databases).

While assessing current logical and technical architecture, care should be taken not to lose relations between logical architecture components and their implementation — technical architecture components.

The EARM process does not require that particular models be used to describe either of the architectural views. The types of models and the amount of information captured for the purpose of getting an understanding of the current IT infrastructure depend on the type of roadmap being developed, its scope, and its future purpose. In practice, even though capturing the existing IT architecture is shown as the first step of the EARM process, in subsequent steps it is often necessary to augment the collected information with the missing data that might be required to do, for example, a gap analysis.

Tools and Techniques

In processing this step, the following tools and techniques could be helpful:

- **Logical and technical architecture models**
- **Architecture patterns —** presenting a generic view for solving some common architecture problems

Result

As a result of this step, current architecture blueprints should be created on logical and technical levels of description. If this is not the first time the EARM cycle is being run, such a description should already exist.

2. Creating Architecture Vision

In this step, the architecture vision is created to be available for comparison with the current architecture.

Goal

The goal of this step is to create a vision of the required architecture within the scope of the architecture roadmap to be created.

Description

In the EARM method, the architecture vision is described in terms of its business and logical view. The business view of the architecture presents a set of activities that are performed within the business area defined as the scope of the architecture roadmap creation. These activities are the result of business value chain analysis — each business area is represented as a chain of interconnected business cycles that is further decomposed using the OODA model. Such decomposition maps business activities onto OODA tasks (see Figure 6). For each OODA cycle step, the business cycle task is described, time constraints required for the cycle to run appropriately are defined, and there is an explanation of the issues concerning task timing requirements and systems to be involved.

Since time frames are defined for each OODA cycle, the time regime required for business

OODA	Business cycle: demand to service (operational cycle)				
	Description	Timing	Time-based issues	Systems	Remarks
Observe	Catch event describing service order arrival	Online	Order management takes care not to put delay in service delivery process	Order management, configurator	Online integration between order management and configurator necessary
Orient	Preparation of network configuration and checking for resources availability	Online	No delays between network inventory and configurator; automatic request processing	Configurator, network inventory	Online integration between configurator and network inventory necessary
Decide	Configurator decides on configuration implementation based on available resources	Online		Configurator	
Act	Decision is passed to activator taking care of service delivery	Online	No delays in communication between configurator and activator	Configurator, activator	Online integration between configurator and activator necessary

Figure 6 — OODA cycle decomposition matrix.

activities implementation could be easily determined within the business architecture. The decomposition of the business cycle into OODA steps can be seen as a kind of "validity" test that determines whether the actual selection of particular business cycles for the given value chain does indeed build required business value. If the business cycle cannot be decomposed, it might need reconsideration.

The logical architecture view for the architecture vision describes the same concepts as those used in the logical architecture view of the current IT infrastructure — components and their features — but with respect to the architecture vision being created. It is based on business domain knowledge (regarding, for example, state-of-the-art architecture patterns and future trends) and developed further according to business architecture requirements. Components of the logical architecture vision should present architecture blocks (classes of technology solutions). These components should be functionally connected according to the sequence of business activities mapped onto OODA tasks. Each business cycle mapped onto OODA tasks presents time constraints for the running of that cycle. This, in turn, presents a time regime, in which architecture components must operate to fulfill business requirements. Care should be taken to properly classify components' features and organize them accordingly (e.g., into layers). In such an approach, each logical architecture component should offer services implementing business activities mapped onto OODA tasks, able to run within the time regime required by the business cycle to be executed.

The creation of a complete architecture vision requires extensive communication with business users; in practice, several facilitated workshops are needed to achieve the goals of this step. In some cases, during such workshops the current architecture could be discussed and clarified as well.

Tools and Techniques

In processing this step, the following tools and techniques could be helpful:

- **Business cycle classification** — defining parts of the value chain in the organization
- **Architecture patterns** — presenting a generic view for solving some of the common architecture problems
- **OODA cycle** — defining tasks being performed in each business cycle

Result

As a result, architecture vision blueprints are created on business and logical levels of description.

3. Determining List of Required Architecture Changes

In this step, the current architecture is compared to the architecture vision to create a list of required architecture changes.

Goal

The goal of this step is to determine the list of changes required in the information environment based on a comparison between the current architecture and the desired architecture vision. During analysis, implementation variants are evaluated and the optimal variant is chosen.

Description

A comparison of the current architecture and the architecture vision is done using a gap analysis technique. A set of services, together with time regimes, implemented in the current architecture is compared to the services required to achieve the architecture vision. Typical questions asked during the comparison of the current architecture service with the architecture vision service focus on:

- Existence of applicable logical architecture service for each task in each OODA cycle
- Applicability of time regime for a particular service
- Existence of required logical architecture service in the current architecture

- Need for modification of current architecture service in order to fulfill requirements derived from the architecture vision
- Ability to cut out some current architecture services without negative impact

While proceeding with the gap analysis, quite often additional features are discovered: optional services, components, or their features that could extend the value of the architecture. Such additional features could be incorporated into the architecture vision as a logical architecture variant. These variants are evaluated according to business constraints, and an optimal variant should be chosen. In case there are no straightforward criteria for variant selection, such decisions should be made together with business users during a variants selection workshop.

Results of the comparison between the current logical architecture and the optimal variant of the logical architecture vision — decisions regarding adding, modifying, or deleting particular logical architecture services — are documented in the form of a list. If applicable, architecture patterns are assigned for each change to be implemented. Usage of architecture patterns results in better EA organization and control. In addition, it simplifies EA description.

Changes in the logical architecture are subsequently mapped into changes in the technical architecture. For each change (addition or modification) in the logical architecture, the applicable component of the technical architecture is determined. This is where “adding database service” is changed into “adding instance of database provided by supplier X”; this presents real changes to be done to the information infrastructure. Each component of the logical architecture could be implemented by many technical components (e.g., the application server could be implemented as J2EE or .NET application servers supplied by one of the available IT vendors). All variants of the technical architecture and all technical variants for each logical component implementation are documented. Finally, an optimal variant is selected; conflicting variants are eliminated, and unification should be enforced (e.g., applicable technical standards).

Tools and Techniques

In processing this step, the following tools and techniques could be helpful:

- **Gap analysis matrix** — comparing current architecture and architecture vision
- **Architecture variants presentation**

Result

As a result of this step, the list of changes to be implemented is determined. Additionally, optimal technical architecture implementing required changes to EA is being selected, and usable architecture patterns are determined.

4. Prioritizing Changes

In this step, the list of architecture changes should be sequenced to reflect business priorities and technical dependencies.

Goal

The goal of this step is to prioritize the list of changes prepared in the previous step; the changes must be prioritized according to business drivers and technical dependencies.

Description

The priority of changes to the IT architecture typically depends on business issues, but some technical dependencies or constraints could also have an impact. Prioritization is based on an assessment of to what extent the changes support strategic goals as well as their level of support for other business drivers applicable to the business area (product strategy, legal obligation, operational necessity, etc.). Such ordering can then be modified by technical dependencies between required changes.

Therefore, each change is assessed as to whether it supports each of the strategic goals. This assessment is done using a weighted scores matrix (see Figure 7). Each strategic goal is assigned a weight: for example, 3 = must have; 2 = should be; 1 = nice to have; and 0 = not important. Next, the level of support is quantified to create a support scale: for example, 0 = no support; 1 = partial support; and 2 = strong support. The support level of each change for each strategic goal is then assessed and placed on the support scale. Weighted support is calculated as support scale value multiplied by strategic goal weight. Finally, the overall change score is calculated as a sum of the weighted support levels. The changes are sorted in descending order by the overall change score with the most important changes at the top of the list; this forms the base prioritization scheme.

In most cases, some additional business drivers influence the requirements for architecture changes. Each set of the business drivers (e.g., product strategy or legal obligations) could be assessed using the same weighted scores matrix. The result would then be presented in the form of shifts in the required ordering of the architecture changes together with its explanation.

During the analysis of additional business drivers and changes in ordering, some additional business values could be discovered; particularly the prioritization of changes could represent new competitive advantage that was not visible before. This represents variants of the architecture changes ordering. Each variant should be assessed in order to select the most promising one; for example, using SWOT (strengths, weaknesses, opportunities, and threats) analysis technique. Based on the assessment, the optimal variant is then selected. Business users should decide if the reasons for the shifts in ordering are important enough to accept them, as they change the business value of the strategic goals. Such decisions could be made during a roadmap workshop.

		Strategic goals									
Architecture changes		Sales online	1	One ID for customer	2	Real-time credit check	1	Customer satisfaction	2	Cost reduction	1
Single and separate order and problem management platform (EAI-based)	18	Order management starts the whole cycle	3	Order management manages singular customer ID	3	Order management simplifies order processing and credit check	1	Effective order tracking	3	Order management reduces manual work costs	2
Coherent xDSL configuration mechanism (EAI-based)	13	Configurator makes the activation process totally automated	2	Order management manages singular customer ID	2		0	Effective order tracking	2	Network operations costs decrease	3
Network inventory consolidation (EAI-based)	2		0		0		0		0		2
Unified billing method inside company	11		0	Rating and billing support unified customer identification	3	Rating and billing systems integration supports single credit limit	3	Single and coherent billing information presented to customer	1	Network management simplification — cost reduction	0

Figure 7 — Weighted scores matrix.

Once architecture changes are ordered according to business goals, it is then examined as to whether such ordering is technically feasible — whether there exist technical dependencies or constraints, which would modify the optimal sequence of architecture changes. It is determined whether a particular change depends technically on another or whether some changes have common areas of implementation. Besides obvious technical dependencies, some additional dependencies could be discovered when looking at the architecture changes from a particular architectural viewpoint (see sidebar "Architecture Views"). Every dependency discovered is identified and documented. The optimal sequence of architecture changes resulting from the business goals assessment is then analyzed with respect to technical dependencies. Each shift in the optimal sequence resulting from technical dependencies is documented and communicated to business users; if accepted, it modifies the final ordering of changes in the architecture.

ARCHITECTURE VIEWS

The following represent the different architecture viewpoints:

- **Business architecture view** — focuses on the functional aspects of the system from the perspective of the users of the system. It addresses the concerns of the users and includes consideration of people, process, function, business information, usability, and performance.
- **Enterprise security view** — focuses on how the system is implemented from the perspective of security and how security affects the system properties. It examines the system to establish what information is stored and processed, how valuable it is, what threats exist, and how they can be addressed.
- **Software engineering view** — focuses on the fact that building a software-intensive system is both expensive and time consuming. Because of this, it is necessary to establish guidelines to help minimize the effort required and the risks involved.
- **System engineering view** — focuses on how the system is implemented from the perspective of hardware/software and networking. It typically is concerned with location, modifiability, reusability, and availability of all components of the system.
- **Communications engineering view** — focuses on how the system is implemented from the perspective of the communications engineer. It typically is concerned with location, modifiability, reusability, and availability of communications and networking services.
- **Data flow view** — focuses on understanding how to provide data to the right people and applications with the right interfaces at the right time. This view deals with the architecture of the storage, retrieval, processing, archiving, and security of data.
- **Enterprise manageability view** — focuses on understanding how the system is managed as a whole and how all components of the system are managed. The key concern is managing change in the system and predicting necessary preventative maintenance.
- **Acquirer's view** — focuses on understanding what building blocks of the architecture can be bought and what constraints (or rules) exist that are relevant to the purchase.

Tools and Techniques

In processing this step, the following tools and techniques could be helpful:

- **Weighted scores matrix —** providing synthetic assessment of each change's impact on business goals rollout
- **SWOT matrix —** giving analytical business variant assessment

Result

As a result of this step, the list of changes is ordered according to strategic goals and other business drivers. This list incorporates technical feasibility dependencies as well.

5. Adding Time Scale to the Prioritized List

In this step the sequence of changes should be assigned time frames based on business goals and known constraints.

Goal

The goal of this step is to prepare a roadmap — to add time scale to the sequence of changes in the architecture prepared previously.

Description

Assigning time frames to the sequence of architecture changes is initially based on business milestones. This timing could be modified by results of rough time estimations prepared for each architecture change implementation.

Aligning architecture changes with time is like scheduling interrelated tasks of a project. All the dependencies among the architecture changes represent interrelations, while time estimation for a change implementation represents duration. Strategic goals are usually long-term issues, and as such they have no strict deadlines defined. In the case of building a roadmap without any additional business constraints, adding time frames to the sequence of architecture changes is quite straightforward. Other business drivers could underpin some time constraints (milestones), which should be evaluated, whether architecture is able to fulfill them or not.

In cases where the business constraint could not be fulfilled, it should be communicated to business users along with a description of the reasons. If accepted, it modifies the final time frames of the architecture changes implementation.

The final sequence of architecture changes together with expected time scheduling is documented in the form of a roadmap (see Figure 8). Additionally, each required architecture change is documented based on information collected: its goal, description, dependencies, business justification, and estimations.

Tools and Techniques

In processing this step, the following tools and techniques could be helpful:

- **Roadmap presentation —** giving a template of roadmap visualisation
- **Architecture change description —** giving a template for documenting required architecture change

Result

As a result of this step, a complete and final roadmap is created. It presents a time-framed sequence of changes in the architecture aligned with business goals and incorporating technical dependencies.

EA MANAGEMENT AND EA ROADMAPPING

As stated above, EARM proposes a technique for EA planning, according to the EA planning framework, and the resulting roadmap itself is applicable as a conceptual tool for communication between business and IT to better align their activities. Since EA management is a more general concept, it is worthwhile to understand how EARM could support a broader EA initiative.

Most approaches to EA management are designed as an iterative effort, with incremental advancement from “as is” toward a “to be” state. Typical EA management iteration starts with defining (updating) an architecture vision, proceeds through a brief comparison with the current architecture, planning for changes migration, and ends with some kind of implementation governance. In complex organizational and IT environments, the planning effort

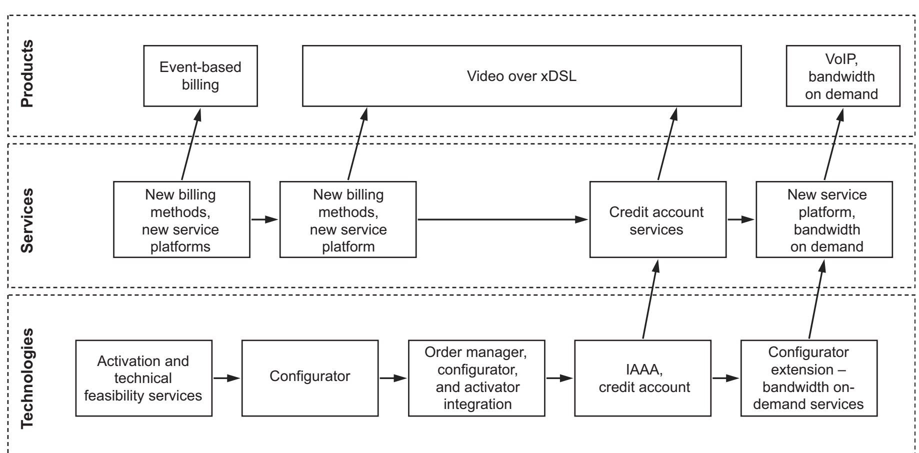

Figure 8 — Roadmap presentation.

may need to be executed on multiple levels. An example shown in Figure 9 illustrates a process that consists of two levels. High-level (strategic) planning transforms the strategic intentions into a set of major IT initiatives (such as a customer relationship management program, a distribution channel integration program, or a consolidation of IT resources). This may be complemented by series of operational plans representing how business objectives are delivered by transforming the architecture toward the "nearest" set of high-level milestones.

In such situations, the major challenge is to capture and manage changes resulting from the complex relationships between the changes in business environment, technologies, and components of the IT puzzle. A well-designed set of roadmaps, developed in a collaborative process, may help conceptualize, capture, and communicate these relationships in a way that improves the capability to absorb changes without creating chaos in the enterprise architecture.

This strategic plan could be presented in the form of a strategic EA roadmap. EA management cycles are executed, each containing its own planning activities regarding the scope of changes to be implemented in that cycle. As a result, each EA management cycle delivers an operational roadmap. Each time an operational roadmap is created, the strategic roadmap may be adjusted — changing timing or some other aspects of architecture vision. The strategic roadmap, however, should be a stabilizing factor as long as the business motives on which the long-term architecture vision has been built are stable.

What should be stressed at this stage are the iterative characteristics of both EAM concepts and EARM. Iterations in roadmap creation and adjustment allow you to correct plans as new issues emerge. These issues could regard both technical and business aspects impacting the roadmap. The iterative nature of EARM results in the ability to introduce refactoring cycles, in which changes made to the architecture in previous cycles could be lined up and made common or shared.

This leads to a generalized "procedure" of EARM cycle scope selection, which plays an important role in the method rollout. The main assumption for the method is that it is hardly possible to create a complete architecture vision for the entire enterprise. It is possible, however, to identify some business areas, in which architectural work could be done and in which "interfaces" with other parts of the company business are quite able to be defined. Such a business area with its interface could be regarded, with some degree of simplification, as the scope of an EARM cycle. Such an approach makes it possible to conduct EAM on an area-by-area basis.

Another concept permitting the narrowing of the scope of an individual EARM cycle is the architecture view. It could be regarded as a representation of the overall architecture that is meaningful to one or more stakeholders. It enables the architecture to be communicated to, and understood by, all the stakeholders and enables them to verify that it will address their concerns. The architecture could be seen through, for

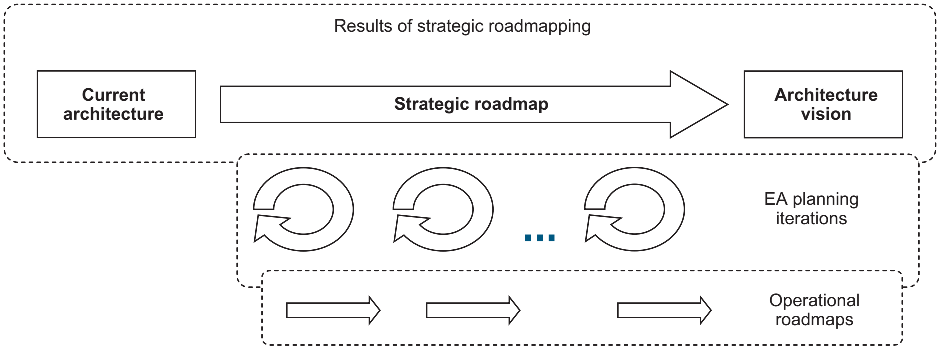

Figure 9 — Strategic and operational roadmaps.

example, enterprise security, system engineering, manageability, procurement, or particular system owner views. Each of these views, or a set of them, could create a scope for the EARM cycle run.

It should be noted that each time the scope of an EARM cycle is narrowed, some aspects of the architecture vision could be missed. This is the tradeoff necessary for the work to be effective and efficient. As a contingency to this drawback, refactoring EAM cycles can be introduced after a number of regular EAM cycles.

The EARM cycle scope selection procedure mentioned above takes these two dimensions and selects the scope of the individual EARM cycle, adding concepts of the EA planning framework to them. To select the scope, the business area should be identified and applicable architecture views and a level of description (know why, know what, know how) should be decided (see Figure 10). It could be a quite common case, where the strategic roadmap presents overall EA migration, while each cycle operational roadmap EARM cycle should take care of the individual business area to be covered. In other cases, the result could be perceived as a broader framework for thinking about EA. Its main purpose is to prepare a roadmap, but particular steps, tools, or techniques could be used even without the roadmap goal in mind. Depending on the scope of the EARM cycle, its steps could be processed or not. For example, (see Figure 11), if the scope of the EARM cycle is defined as discovering "know why" and "know what" for many business areas, no technical solutions or time frames could be determined. In that case, the roadmap is of a less precise nature, but it could be made more precise in subsequent EARM cycle iterations being run for each business area to determine "know how" and "know when."

(Text continues on page 21.)

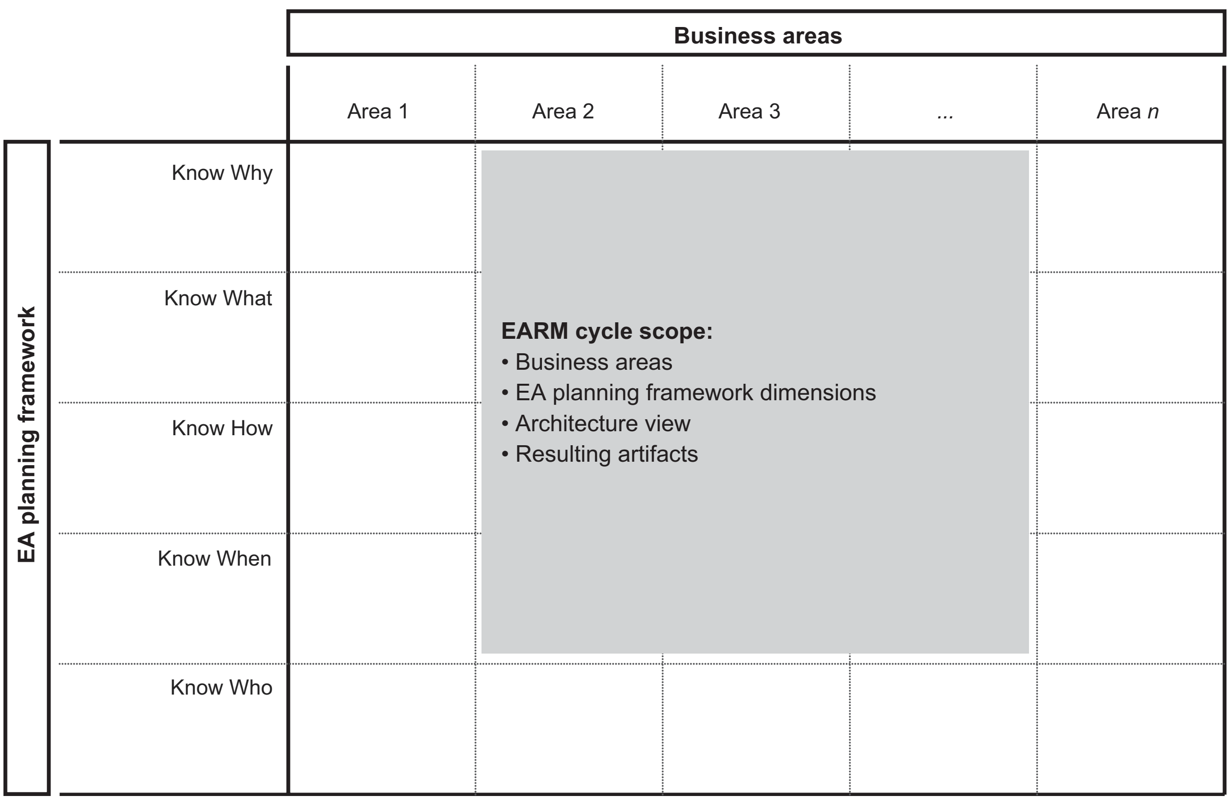

Figure 10 — EARM cycle scope selection.

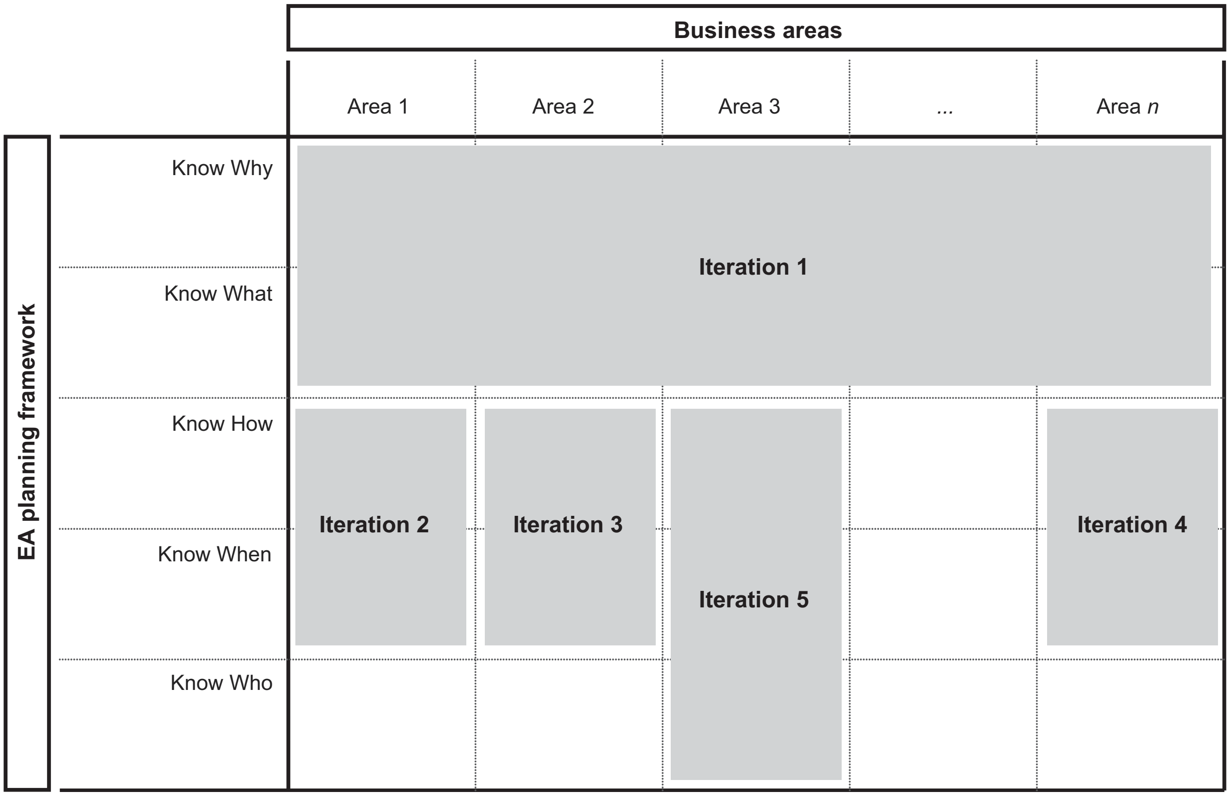

Figure 11 — EARM cycle scope: broad business, precise technical.

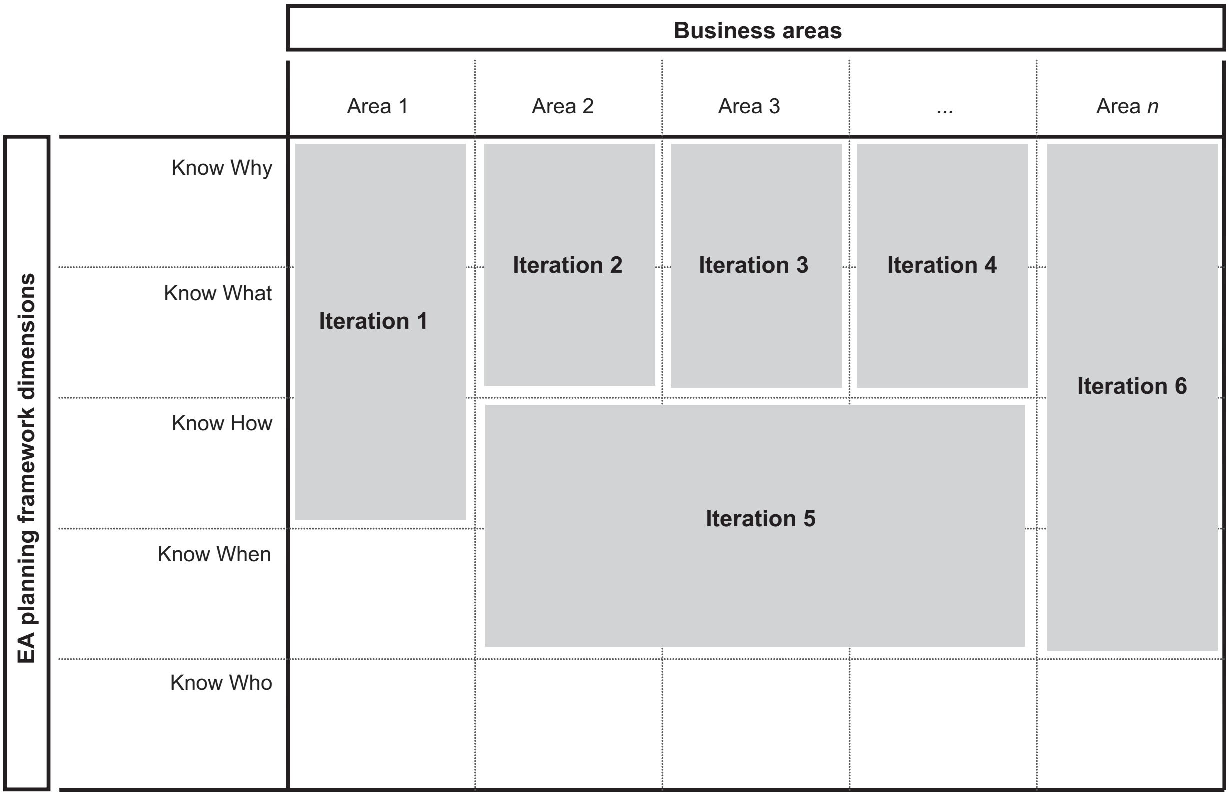

Figure 12 — EARM cycle scope: technical for many business areas.

(Text continued from page 19.)

In another example (see Figure 12), if the variants of the technical architecture are too complicated or could affect other business areas' architecture, the current EARM cycle could be finished without roadmap creation and the next cycle could be started in order to assess the impact of changes on other business areas in order to select the optimal variant of the technical architecture and create a roadmap based on that optimal variant.

EXAMPLES

This section follows the above discussion with some real-world examples.

Integration Architecture

The first project has been executed for a telecommunications company that has been in the process of business and technology consolidation with its mobile subsidiary. The process began with offering common products to their customers, which required some integration of the two companies' IT environments. The goal of the roadmapping project was to propose an integration architecture in four business areas (see Figure 13). Four business areas were chosen for the purpose of determining the optimal integration architecture. The project was successfully completed in six months.

The first business area (customer self-care) regarded common care for a customer having business with either of the two companies. This area had been deeply defined in terms of technical solution; we started with broad customer-care business vision, identified applicable logical

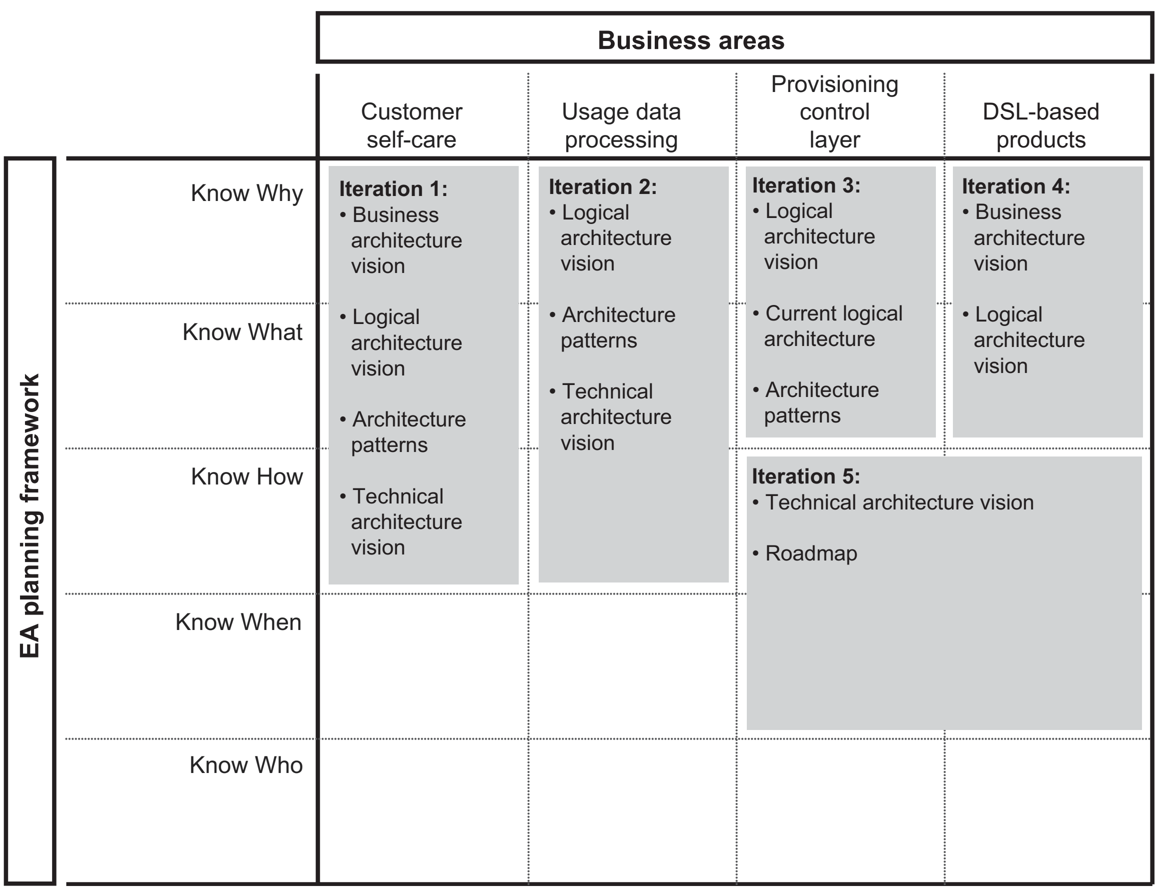

Figure 13 — Integration architecture: scope of EARM iterations.

architecture and some architecture patterns, and finally defined the technical architecture for that area.

The second business area (usage data processing) concerned unified mediation, rating, and billing for both voice and IP services billing data. Since business architecture was quite well-defined here, we focused on logical architecture and technical architecture patterns.

The third business area (provisioning control layer) described the way services provision should be controlled in order to achieve a smoothly running and transactional provisioning process. Since we quickly discovered that it strongly depends on the services planned to be provisioned in the near future, we decided to split this business area analysis into two iterations. Within the first one, we defined architecture on the business and logical level with some applicable architecture patterns. The second one was decided to be postponed until the next business area was processed.

The fourth business area (DSL-based products) concerned the roadmap definition responding to business plans regarding DSL technology-based products and services. We focused here on business architecture and defined logical architecture as well. As we discovered previously, this business area roadmap had been strongly correlated with provisioning control features implementation, so we decided to build a common roadmap for the third and fourth business areas.

The roadmap created was based on the fourth business area business architecture and concerned consolidated logical architectures resulting from the third and fourth business areas analyses. For such scope, we defined the technical architecture vision and a roadmap presenting the way in which both provisioning and services platform technology should support business plans regarding the rollout of new products.

eGovernment Gateway to Poland

In 2004, the Polish government decided to run an initiative for building electronic access to public administration services in Poland. One of the challenges was to define the overall architecture for such an initiative, aligned with the grassroots initiatives accomplished by some local governments. The architecture needed to find a balance for the role of provider of integrated interagency services with the role of supporting infrastructure and aggregator of services developed by independent local governments and nongovernmental organizations. Adopting the EARM principles enabled the team to define the architecture on business and logical levels (see Figure 14) and build a strategic roadmap on the business architecture level in just two months.

One of the key architectural principles employed in the gateway design was the assumption that the gateway will serve different roles depending on the level of integration and the amount of bespoke business logic required by a given service. These roles formed an evolution path starting from a simple directory providing a single entry point for information browsing, through a gateway role extending the directory with identification services and finishing with notifier and coordinator roles, in which the eGovernment Gateway mediates messages between government agencies and citizen (G2C) and among government agencies (G2G), respectively. Based on that business architecture vision, we developed the logical architecture vision, optimized to support the business architecture vision evolution.

The result of the project was an architecture vision defined on business and logical levels and an architecture roadmap presenting the increments of eGovernment Gateway. Additionally, a catalog of architecture patterns was created to assist the design of e-government implemented on the Gateway platform.

CONCLUSIONS

The EARM method affords an opportunity for rapid creation of valuable results in the field of

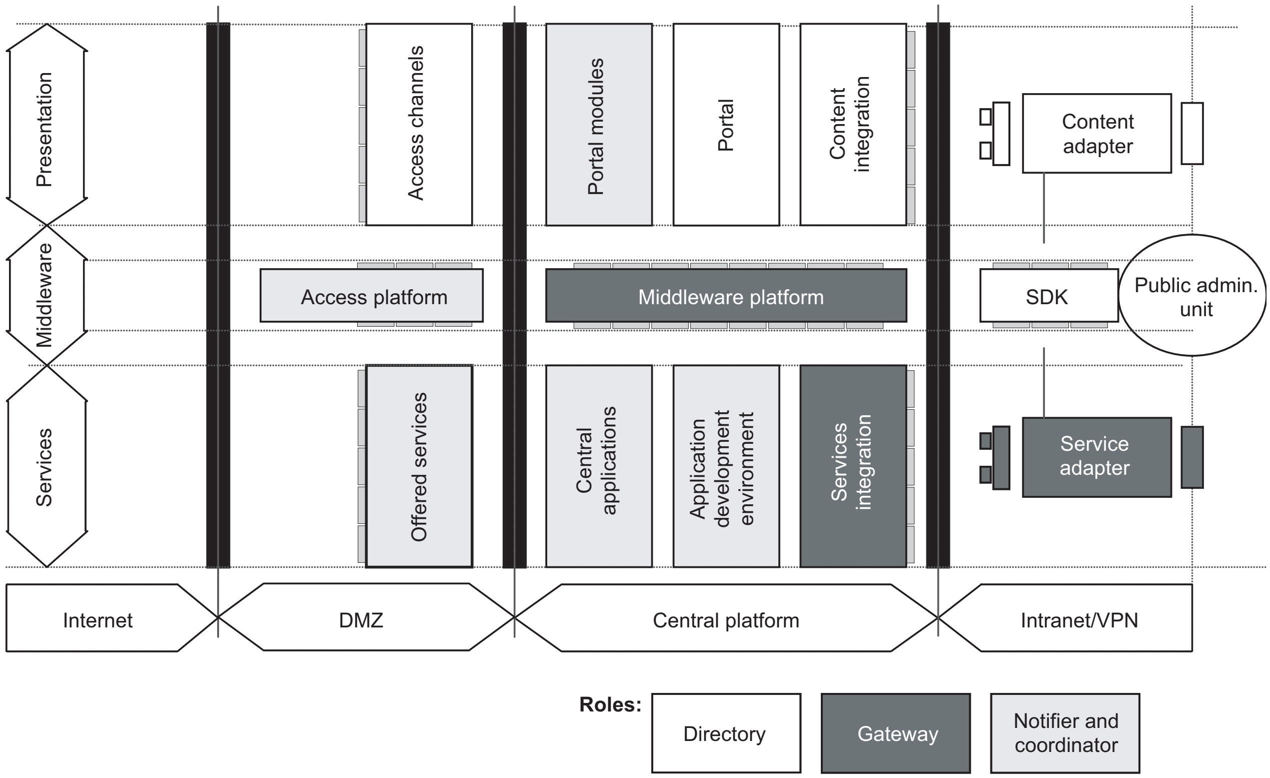

Figure 14 — Polish eGovernment Gateway evolution.

integrated business-IT planning. Decision-making time could be of essence, so the fact that this method focuses on the most important issues and reduces the number and volume of artifacts to a necessary minimum makes it highly usable, effective, and efficient in the field of EA changes planning.

The method presented in this report proposes a framework for consciously simplified thinking about changes to EA. Strategically, this method allows you to plan for medium- and long-term modifications in the information environment on a level that gives you the ability to justify and communicate with the business, without getting into deep and broad business analyses. More importantly, the method gives answers on the strategic level — whether technology is able to support company strategy or not. From an IT operational point of view, the results produced by the EARM cycle allow you to place IT endeavors in context of the architecture vision more broadly than the individual technical area. As a result, investment decisions can be made faster and can more accurately reflect the changing business and technology environment, supporting the time-based competitive strategies so typical for modern business and for improving capital efficiency.

EARM is in many respects an agile approach to integrated business-IT planning and management. It allows you to focus on the most important issues, adapt to changes rapidly, and does not require extensive effort for producing valuable results. The basic concept of iterations affords the possibility to adapt and react, while selecting the scope of the EARM cycle gives the opportunity to focus on important pieces of business. These iterative characteristics form a basis for the self-healing features of the EARM cycle — refactoring cycles could be introduced in case of a risk of no compatibility among previously finished EARM cycle results.

This method, however, should not be considered (and was never intended to be) an "Ultimate Answer" (in Adams's terms) to all architecture problems. Its end result depends strongly on the context of EARM cycle processing, the correctness of its scope selection, and the company culture in which the method operates. It gives means of communication, but communication itself requires people to cooperate.

Last, but not least, this method is a tool to be used by an aware enterprise architect. It does not require formal organizational structure like an architecture office. Nevertheless, in case the organization wants to build an architecture office, EARM can be used successfully in such organizational frameworks as well.

ABOUT THE AUTHORS

Bartosz Kiepuszewski is a Senior Consultant with Cutter Consortium's Agile Project Management and Enterprise Architecture Practices. He is also a Senior Consultant at Infovide, the exclusive representative of Cutter Consortium in Poland that specializes in EA consulting and implementation. Dr. Kiepuszewski, who specializes in J2EE architectures, workflow systems, and agile software development, holds a Ph.D. from Queensland University of Technology, where he worked on theoretical foundations for workflow modeling languages. Before joining Infovide in 2001, he worked for five years in Australia, first at the Distributed Systems Technology Centre as a research scientist, and later at Mincom Ltd. as a Senior J2EE Architect. Dr. Kiepuszewski has authored numerous research papers, primarily in the workflow area, and he is a frequent speaker at national and international conferences, seminars, and trade shows. He can be at bkiepuszewski@cutter.com.

Sebastian Konkol is a Senior Consultant at Infovide. He is coauthor of the EAM approach at Infovide. He splits his time among running strategic IT consulting endeavors, performing project manager duties, and being an EAM expert. Before joining Infovide in the beginning of 2003, he worked in a mobile telecom operator company as program manager responsible for running projects aimed at the creation of GSM VAS (value-added services) offered to the mass market. Mr. Konkol has authored numerous publications dealing with IT and telecom information environment management and IT organization management.

Wojciech Ozimek is a Senior Consultant with Cutter Consortium's Business-IT Strategies and Enterprise Architecture Practices. He is also the R&D Director at Infovide and principal and cofounder of one2tribe. Mr. Ozimek began his career at Polish Telecom, where he was involved in the design and implementation of high-availability distributed systems, OO methodologies, and human-computer interaction design. In 1998, Mr. Ozimek joined Infovide, where he has participated in numerous projects as consultant, architect, and project manager. As R&D Director, he has focused on strategy and business development. Today Mr. Ozimek's responsibilities include exploring new business areas for the company as well as performing high-level architecture and strategy consulting for select Infovide clients. In 2003, Mr. Ozimek founded one2tribe, a small company that specializes in creating large mobile communities. He is the author of many articles for Polish business and IT publications and is a frequent speaker at conferences, seminars, and trade shows. Mr. Ozimek can be reached at wozimek@cutter.com.

Borys Stokalski is a Senior Consultant with Cutter Consortium's Business-IT Strategies and Enterprise Architecture Practices and is a cofounder and principal of Infovide. Mr. Stokalski began his career in 1985 as a research worker and then as a lecturer in the area of human-computer interaction at Warsaw University. Today his primary responsibilities include business development,

innovation, and strategy. He takes an active part in Infovide initiatives targeted at new service areas and consulting products. Mr. Stokalski occasionally takes the opportunity to perform high-level consulting and coaching for select Infovide clients. He frequently publishes articles on enterprise architectures, business-IT alignment, IT strategy, and IT-based innovations, and is a frequent speaker at conferences and seminars. He can be reached at bstokalski@cutter.com.

REFERENCES

1. Ambler, Scott. "An Agile Approach to Enterprise Architecture." Cutter Consortium Enterprise Architecture *Executive Report*, Vol. 7, No. 5, 2004.

2. Boyd, John. "A Discourse on Winning and Losing," 1996 (www.d-n-i.net/second_level/boyd_military.htm#discourse).

3. Davenport, Thomas. "The Coming Commoditization of Processes." *Harvard Business Review*, 1 June 2005.

4. Kaplan, Robert S., and David P. Norton. *The Balanced Scorecard*. Harvard Business School Press, 1996.

5. Kiepuszewski, Bartosz, Michal Paluskiewicz, Borys Stokalski, Sebastian Konkol, and Maciej Ruszynski. "Applying EA Roadmapping: An SOA Roadmap." Cutter Consortium Enterprise Architecture *Executive Report*, Vol. 7, No. 9, 2004.

6. Phaal, Robert, Clare Farrukh, and David Probert. "Fast-Start Technology Roadmapping." Department of Engineering, University of Cambridge, 2000.

7. Phaal, Robert, Clare Farrukh, and David Probert. "Technology Roadmapping: Linking Technology Resources to Business Objectives." Centre for Technology Management, University of Cambridge, 2001.

8. Porter, Michael E. *Competitive Strategy: Techniques for Analyzing Industries and Competitors*. Free Press, 1980.

RESOURCES

Orr, Ken, in collaboration with Bill Roth and Ben Nelson. "Business Enterprise Architecture Modeling," Cutter Consortium Enterprise Architecture *Executive Report*, Vol. 8, No. 3, 2005.

Enterprise Architecture Practice

Today the demands on corporate IT have never been greater. Cutting costs and accelerating time to market for individual line-of-business projects are still priorities, but even that's not nearly enough anymore. Companies are now looking for strategies to better leverage their entire IT infrastructure. They want IT to deliver sophisticated enterprise applications that can provide value across many lines of business and provide marked differentiation from their competitors. The Enterprise Architecture Practice provides the information, analysis, and strategic advice to help organizations commit to and develop an overarching plan that ensures their whole system fits together and performs seamlessly.

The Enterprise Architecture Practice offer continuous research into the latest developments in this area, including Web services, enterprise application integration, XML, security, emerging and established methodologies, Model Driven Architecture, how to build an enterprise architecture, plus unbiased reports on the vendors and products in this market. Consulting and training offerings, which are customized, can range from mapping an infrastructure architecture to transitioning to a distributed computing environment.

Products and Services Available from the Enterprise Architecture Practice

- The Enterprise Architecture Advisory Service
- Consulting
- Inhouse Workshops
- Mentoring
- Research Reports

Other Cutter Consortium Practices

Cutter Consortium aligns its products and services into the nine practice areas below. Each of these practices includes a subscription-based periodical service, plus consulting and training services.

- Agile Software Development and Project Management
- Business Intelligence
- Business-IT Strategies
- Business Technology Trends and Impacts
- Enterprise Architecture
- IT Management
- Measurement and Benchmarking Strategies
- Enterprise Risk Management and Governance
- Sourcing and Vendor Relationships

Senior Consultant Team

Our team of internationally recognized specialists offers expertise in security issues, e-business implementation, XML, e-business methodologies, agents, Web services, J2EE, .NET, high-level architecture and systems integration planning, managing distributed systems, performing architecture assessments, providing mentoring and training, overseeing or executing pilot projects, and more. The team includes:

- Michael Rosen, Practice Director
- Scott W. Ambler
- Douglas Barry
- Don Estes
- Michael Guttman
- Paul Harmon
- David Hay
- Ian S. Hayes
- Tushar Hazra
- Peter Herzum
- J. Bradford Kain
- Bartosz Kiepuszewski
- André LeClerc
- Jean Pierre LeJacq
- Arun Majumdar
- Jason Matthews
- Tom Marzolf
- Terry Merriman
- James Odell
- Ken Orr
- Wojciech Ozimek
- Rob Shelton
- Oliver Sims
- Borys Stokalski
- William Ulrich
- Jim Watson
- Tom Welsh