

Leverage your architecture! Make decisions better, faster, and cheaper

Michael Idengren, Agile Architect
CIO Advisory
midengren@kpmg.com

Contents

PART 1: Opportunity, problem, solution

- Discussion about:
 - "EA as a service" based approach
 - People
 - Tools
- For:
 - CIO / CTO
 - Strategic Portfolio Managers
 - Project / Program managers
 - EA program manager
 - EA Architects
 - EA Analysts
 - IT DevOps manager

PART 2: Actually making it work: People & tools

- Discussion about:
 - "EA as a service" based approach
 - People
 - Tools
- For:
 - EA Program manager
 - EA Architects
 - EA Analysts
 - IT DevOps manager

PART 3: Case study: System rationalization in controls readiness context

- **Discussion about:**
 - "EA as a service" based approach
 - People
 - Tools
- **For:**
 - EA program manager
 - EA Architects
 - EA Analysts
 - IT DevOps manager

Executive summary

Why read this series?

89% of Forrester-surveyed organizations indicate immature or inconsistent use of operational and business data to support strategic decisions¹, yet CIOs indicate that Business Analysis and Enterprise Architecture are the most in-demand and fastest-growing skill shortages².

Your organization has invested in an Enterprise Architecture / Business Process capability, but it feels "stuck in the mud". You know that EA is supposed to help provide decision-makers information needed to make better decisions, but that doesn't seem to be happening. You want to get more value out of this "EA thing" – that is, you want decision-makers to feel confident getting answers from it. This can be small questions, such as "should we invest in technology XYZ?" or bigger questions, such as "what are our options for modernizing our service"?

This series assumes you already have an EA/BPA platform (i.e., with web portal, flexible method, and tools/advanced scripting APIs) and you have data in the repository. You've got some people who work in the EA area, but you're not sure if you have the right mix of backgrounds and skills needed to define the problem, turn data into information, and provide answers targeted to the right stakeholders. Typical "EA input/modeling", governance, and architecture design/frameworks/theory discussions are out of scope.

¹ Forrester's Q3 2016 Global State of Enterprise Architecture and Portfolio Management Online Survey

² Harvey/Nash KPMG CIO 2017 Survey <https://home.kpmg.com/xx/en/home/insights/2017/05/harvey-nash-kpmg-cio-survey-2017.html>

Keywords / concepts

Agile EA, EA quality control, quantitative risk assessment, controls readiness, application portfolio rationalization, TIME chart (Tolerate, Invest, Maintain, Eliminate), EA/BPA analysis, EA/BPA service organization, solution architecture options, project portfolio management, EA data vs. information, KPIs (Key Performance Indicators) vs. metrics

PART 1: Opportunity, problem, solution

You've taken the plunge, and invested in an Enterprise Architecture (EA) / Business Process Analysis (BPA) platform to centrally maintain lots of content (processes, apps, data, infrastructure, etc.). You've got a governed method and standards for maintenance of the repository content (i.e., TOGAF, Zachman, DoDAF). This situation presents an opportunity, a problem, and a solution.

The opportunity

Eventually, someone is going to ask questions ("business problems"), which can be answered using a slice of architecture data. The trick is to simplify ("flatten") the EA data into "information", to help make decisions. This can help answer typical (smaller-scope) process analysis questions, such as as-is vs. to-be process differences. For a "medium complexity" application portfolio rationalization effort, it can be used to decide which systems to keep or retire. It can also be aligned towards industry/functional specific questions, such as "Why do we continue to have poor audit findings? Which systems are the biggest factors?" (Our Part 3 case study is based on this scenario).

The most mature use of EA will accommodate answering complex end-to-end solution analysis. We might want to boil complex options to simple questions, such as "what are the costs, benefits, and risks of solution A vs. solution B?" For example, in an emerging technology example -Blockchain- a scenario comparison quickly compares various solution derivatives to each other, focusing on using technology to improve key areas balancing risk, cost, and benefits. This might result in a decision to do "a little bit" of Blockchain up front, paired with a longer-term API/integration on-premise/cloud investment strategy.

We work with business stakeholders to describe high-value "key questions", and assemble data (i.e., processes, capabilities, systems) into "architecture views" needed to provide the desired information. We will typically need to gather new data, or cleanse existing EA repository data. Depending on the scenario, it could be a "big question" or a "small question". Small questions might result in a small effort to produce a report, while larger questions could involve efforts like capability/process/system rationalization and alignment (the bulk of which is typically working with business stakeholders to agree on terminology and levels of detail!)

Those business stakeholders might want that information in the form of a simple report (e.g., in PDF or Excel formats), perhaps e-mailed daily or weekly. Or, it might be more appropriate to provide a real-time dashboard view, depending on the complexity of the data and how "summarized" it needs to be. This might be a longer-term monitoring question (e.g., "We projected a 20% reduction in time needed to maintain these systems, so what's the monthly status of our KPI readouts?").

The problem

Now, we turn to answering the question – the EA repository has a lot of "stuff" (raw content), in a multi-level, inter-connected hierarchy (architecture), that people don't really understand (except maybe a few architects), and even fewer people know how to mine (i.e., JavaScript or SQL developers who are familiar with relevant EA methods and tool APIs).

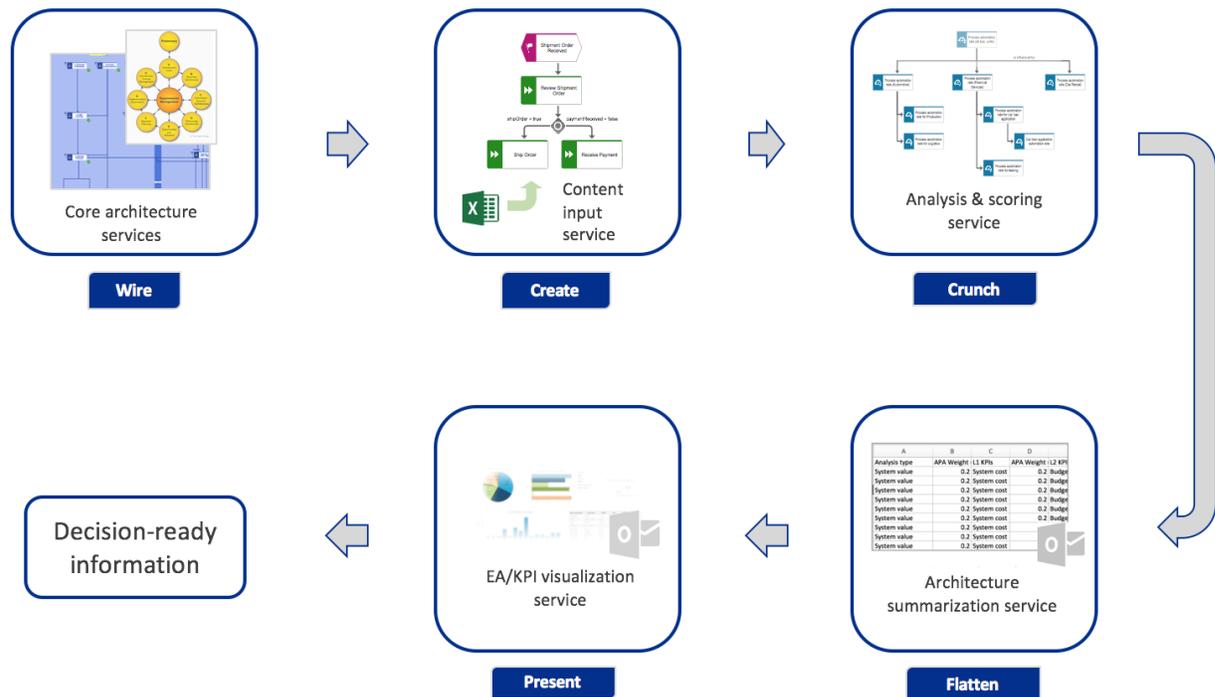
During the "opportunity" phase, we worked with key stakeholders who have business problems (questions) to be answered. After we helped to narrow down "the right" question, and collected/scrubbed data needed, we might have to scramble to find resources to provide an answer (in a way that is easy to understand). This turns into a mini-project, which takes time and specialized resources.

The solution

We will discuss three components to a holistic solution:

- (1) A services-based EA/BPA approach. This helps to clarify "what does EA do for me?"
- (2) A flexible EA/BPA/data visualization platform and toolchain. This lets our EA resources produce answers in a fast, repeatable manner.
- (3) A staffing/enabement model to define and answer questions at a low cost

Services approach. We divide the work done to "answer the question" into a continuum of services:



Each of the services has enabling assets, tools, governance, consumers, and producers. From the "end consumer" perspective (the stakeholder who wants to make a decision), most will only care about the end result – the "**present**" deliverable (i.e., dashboard or emailed report). However, intermediate analysts (i.e., IT, Line-of-Business, or project-focused analyst) might want a quick answer to their problem (the "flatten" service, asking "just give me the spreadsheet"). Some might want a report in their inbox weekly, a link to a dashboard, or just a single, one-time attached report. For the most part, none of them will want to hear about "enterprise architecture" or "BPM/BPA". They just want their questions answered.

For the most part, none of them will want to hear about "enterprise architecture" or "BPM/BPA". They just want their questions answered.

Taking a service-based approach allows the EA/BPA group to clearly show value (and, in mature organizations, commitment via service level agreements) to the services they provide to the enterprise. Of course, a successful service model requires more than just presentation slides or boxes on a screen; it needs a "marketing" campaign (to get "internal" customers who will ask the key questions), reliable technology, and people who can deliver. Using an incremental approach, such a model can "start small", show successes, gain credibility, and scale.

The simple "answer a question" context services listed here are intended for "non-EA" analysts and business stakeholders with specific questions. As an EA organization matures through visible contribution and credibility, Line-of-Business and IT leaders will continue to "ask for more". When an EA organization reaches a certain point of maturity, it makes sense to create an EA service catalog³ to leverage an "outside-in" approach; that is, primarily focus on what the business needs, and align EA services to deliver value. This "outside-in" concept can be

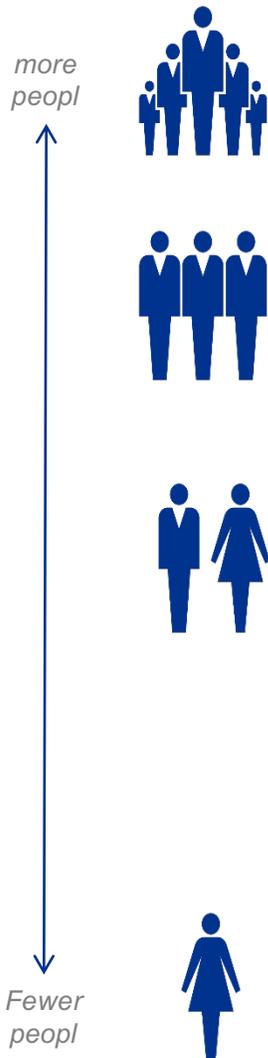
³ Forrester "Defining The EA Service Catalog", April 2017. Alex Cullen and Gordon Barnett.

expanded to more encompassing EA services, such as technology/target state solution service, or a business/technology portfolio review service.

Service enablement: Technology. For reliable service delivery, a base platform of EA technology/infrastructure needs to be in place. It needs to have sufficient capabilities in method customization and "app development" capabilities, including relevant APIs for interfacing and reporting. Most of the development is typically ETL-oriented (Extract, Transform, Load).

Service enablement: People. After defining the services and technology, let's have a look at the last, and most important ingredient needed to make things happen – the people. Not everyone has to be a programmer - some analysis questions are simple, and can be answered using typical Out-Of-Box (OOB) tools, such as reports. Fewer people are needed to do more complex development work; factors include EA integration environment complexity/maturity (higher complexity might mean more L3-L5 "scripters/developers", as described below), and EA tool "OOB capabilities (higher tool capability might mean fewer L3-L5 "scripters/developers" needed). Generally speaking, an organization will need more analysts, who are closer to the business problems, and fewer developers.

Here is a typical analysis roles profile, where "more analysts" and "fewer technical developers" are needed:



- **Level 1 (power user)** – Typically, an L1 power user creates on-the-fly, “easy” query creation for basic architecture / process analysis, resulting in spreadsheet/PDF document output. Requires typical tool-supported OOB query editor and built-in reports, with no coding.
- **Level 2 (analyst)** – L2 analysts do more detailed configuration of “template reports” to produce spreadsheets. This allows for complex filtering, downstream dashboard links to web portal models, and regularly emailed reports. Requires very little coding – example tasks are configuration of “template reports”.
- **Level 3 (basic scripter)** – An L3 scripter writes “real code” (i.e., JavaScript or SQL) using good SDLC practices, paired with some tool-relevant method/API training. A basic scripter produces custom reports (i.e., PDF, Word, XLS) by summarizing complex architecture relationships.
- **Level 4 (developer)** – An L4 developer does full custom solution development, such data sync between two systems (i.e., Archer – ARIS to exchange GRC and process information). This person might require Java/JavaScript experience and EA tool method/API proficiency, SDLC experience (i.e., code check-in/versioning, testing discipline)
- **Level 5 (developer)** – An L5 developer does custom infrastructure development, (i.e., web services/live API usage) to various endpoint UI components (i.e., web browser, iPad, custom app), Hadoop integration. More importantly, the L5 developer maintains toolchains for developers (especially L2) to be productive. Requires skills and experience such as JavaScript, EA tool method/API use, SDLC, distributed computing, custom java classloaders/Java, API development, and template/pattern development.

So, what kind of people can deliver? Look around! Who is the "Excel wizard"? That person might be a good L2/3 candidate, while someone who is more focused on business problems and simple analysis might be a good L1 candidate. Depending on your organization and staffing practices, it might make sense to outsource L5 developers, but maybe to have one or two in-house L3/4 developers.

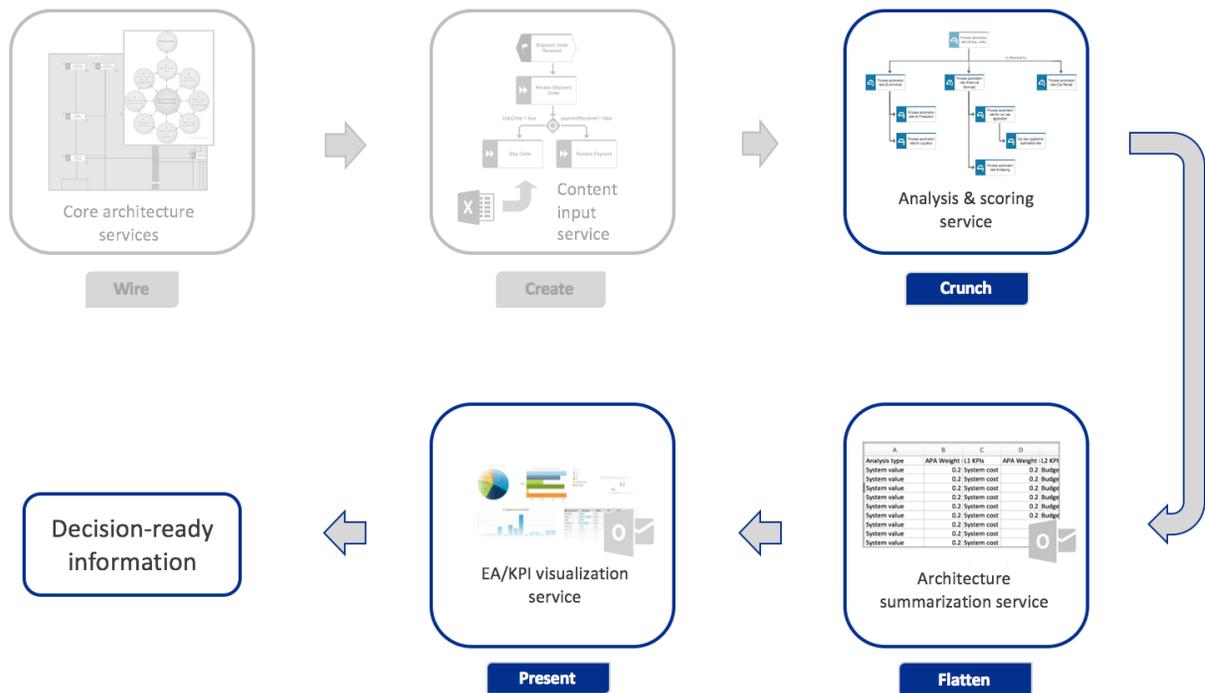
The key to effectively staffing an "EA Center of Excellence" includes identification and slotting candidates into the correct level, and supporting them with training and tools. However, unfortunately, it is all too easy to build an ivory tower of architects and analysts who can "do a lot of things", but somehow "don't show a lot of value" (at least, in the perception of the rest of the organization). Remember that EA groups need challenging and important questions to answer. Using the service-based approach outlined above, paired with executive leadership and interest (the "marketing" part) will define and assign the important "key questions" to answer. Then, the right mix of people and tool capabilities will meet the demand in an agile way.

PART 2: Making it work: People & toolchain

In part 1, we looked at techniques to define and answer questions using a service based approach and people's roles. Part 2 of our series will have a closer look at the people, and the setup of the EA toolchain to make a service-based approach happen. Part 3 will apply the concepts in part 1 and 2 in the context of an controls-readiness case study, where we attempt to determine systems that pose the highest risk to passing an audit.

Getting back to our original problem scenario: evaluation of a system portfolio using a set of customized criteria. To answer the important "business problem", we might need to:

- (1) **Wire** (*out of scope for this article*). Revise our architecture "method" – the standardized terminology and way things are connected, and how levels of detail are defined
- (2) **Create** (*out of scope for this article*). Collect, normalize, validate content
- (3) **Crunch, flatten**. Analyze the content using various methods, such as EA built-in query tools, or custom applications built in that platform
- (4) **Present**. Visualize the content, in a way that is appealing for the consumer (i.e., creating a dashboard)



While items (1) and (2) are outside the scope of this discussion, steps (3) and (4) are key to leveraging EA content and views to deliver on key services: **crunch, flatten, present**.

The toolchain. In the following section, we describe how to crunch, flatten, and present using a mix of built-in tools and custom apps, using Software AG tools. In our scenario, we will use icons to represent "out-of-box" (OOB), and custom-developed tools.

-  "Out-of-box" tools, such as Software AG's Mashzone (dashboard) and ARIS (EA/BPA modeling and analysis tool)
-  Custom ARIS JavaScript toolchain (re-usable apps that "plug in" to the ARIS API, using open-source components)

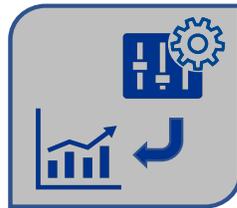
The steps and relevant toolchain components are:

- **Crunch.** Use a mix of custom-developed ARIS JavaScript apps and standard out-of-the-box tools.



ARIS Notifier app

*Event trigger-based rule engine
(for L2 analysts)*



Metric Scoring Engine app

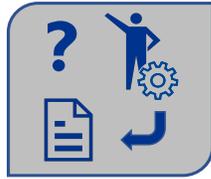
*Rule-based evaluation of metrics
and weighted KPIs*



Mashzone/ARIS Aware

*Software AG products - data
feed transformations &
dashboards*

- **Flatten.** Use a mix of custom apps and OOB tools. While the built-in standard ARIS query tool can help to easily create simple queries, they are not useful as scheduled reports or for running high-volume analysis. Furthermore, the OOB report editor cannot do advanced things such as auto-create web portal links, automatically send an email, or use complex filtering criteria. To "close the gap", a custom JavaScript utility (Query Generator) has been built by an L5 developer (especially for L2 analysts):



Query Generator app

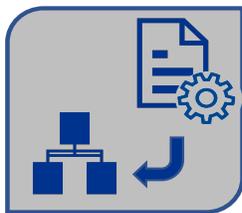
EA trace tool creates .CSV for runtime scheduled execution (especially for L2 analysts)



ARIS Query Builder

Built-in tool for visually designing queries

- **Present.** Use a mix of custom apps and OOB tools. For example, the custom app built by an L5 developer ("Model Layout Engine") uses layout algorithms to automatically arrange any number of "messy" modeled objects into profile-driven, architecture-layered layouts.



Model Layout Engine app

Auto-layout models according to template specifications; save time, increase quality (for all)



ARIS Connect Portal

View/maintain EA content (models, attributes, etc)

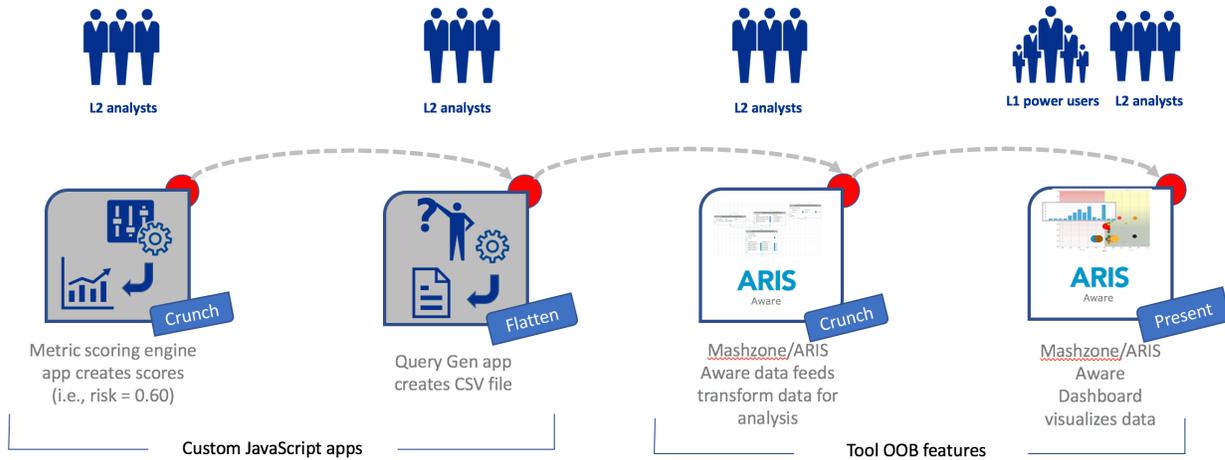


Mashzone/ARIS Aware

Software AG products - data feed transformations & dashboards

For our case study (in part 3), this combination of the toolchain will realize a decision-making (dashboard), end-to-end. To make this work -and reiterate what was said above- an important assumption is that the content already exists in the repository (for example, all applications have necessary attributes maintained, with a high degree of quality).

Then, we will crunch (summarize), flatten, crunch (transform), and present:



In the context of this example, the "tech" people-skills needed to operate (not develop) that toolchain are as follows:



Create scoring formulas in JSON format. A separate architect role might model KPIs and weights.



Create query trace specifications in JSON format. A separate analyst role might create requirements for data relationships.



Creates data feeds. These are necessary to support the charts & graphs that a UI/UX expert role might design.



UI / UX experienced analyst might design the "best way to tell the story" with dashboard visual. This might start with wireframes (L1 power user) and end with actual dashboard widgets (L2 analyst)

We do not need "heavy" L3-5 developers for this kind of low-to-medium complexity dashboard. The combination of this set of tools and corresponding training means that any analyst-type person can be trained to wire, create, crunch, flatten, and present decision-ready information.

Enabling L1/L2 analysts in this way to allows us to scale operations (i.e., "create 10 dashboards next month") faster, at low cost, and with stronger business experience.

PART 3: Case study: System rationalization in controls readiness context

In part 1, we looked at techniques to define and answer questions using a service based approach and people's roles. In part 2, we had a closer look at the people, and the setup of the EA toolchain to implement a service-based approach. Part 3 will apply the concepts in part 1 and 2 in the context of a controls-readiness case study, where we attempt to determine systems that pose the highest risk to passing an audit.

In our case study, we present an application portfolio rationalization in the context of a "controls readiness" engagement. The organization has the goal to reduce overall risk of system ownership and operation, which will result in fewer system-related audit findings. Many of those findings in the past were the result of expensive-to-operate systems that are out of compliance, for various reasons.

The "key question" (or "goal") here is: Which systems have the highest risk?

- WIRE/CREATE - First, we collect, normalize, and validate metrics, and align to KPIs in our architecture. This will form the rational basis for the analysis.

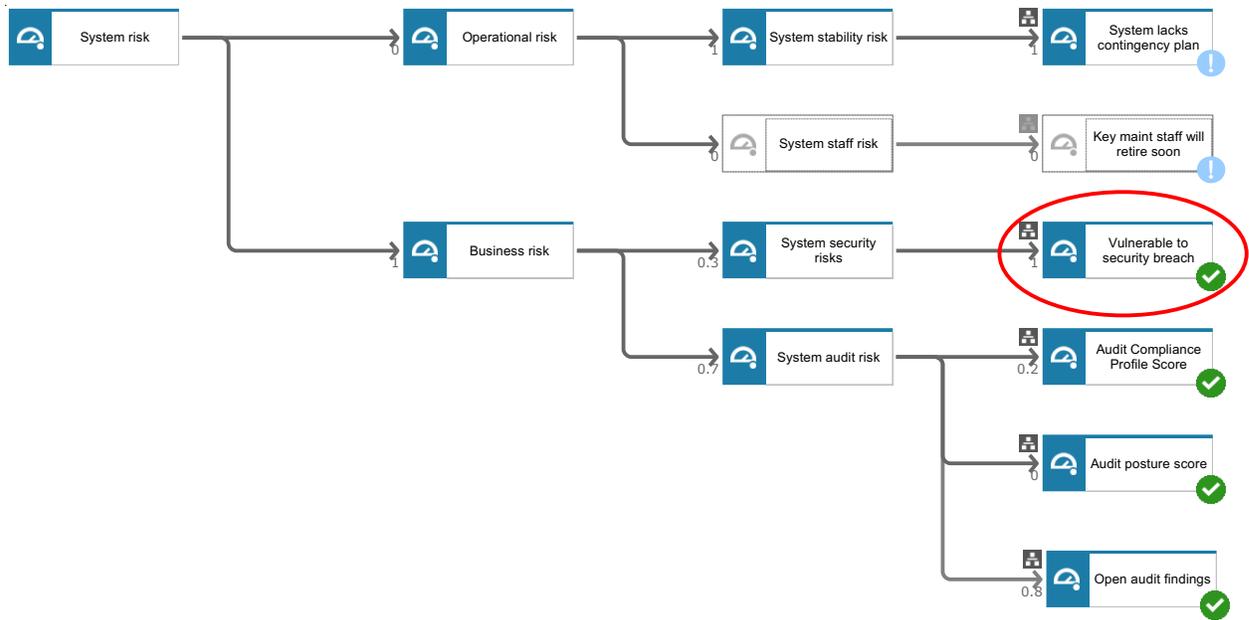
Then, to perform analysis needed to answer the key question, we followed these steps:

- CRUNCH - Score metrics.
- FLATTEN - Summarize complex metrics/KPIs into spreadsheet).
- CRUNCH - Transform data to be suitable for a dashboard)
- PRESENT - Create dashboard widgets to consume transformed data)

Analyze the problem, model the KPIs, and collect data.



While we have a focus on audit risks, there are also other risks to owning and operating systems from a business and operational perspective. To determine the system risk, we construct a visual KPI tree that allows us to decompose the goal into various levels of detail. We use the *Model Layout Engine* to visually assemble the tree automatically. Each level of detail compiles a score based on weighting criteria (which must add to the full value of 1.0), which are shown in the small gray numbers next to the arrow heads.



At the lowest level, (all the way on the right) there are additional details attached (metrics), which use the architecture in a flexible way to produce meaningful scores. The "checkbox" or "blue exclamation points" are data-quality trackers that visually indicate "how done we are" collecting data needed to create those scores.

Shown below is an "audible" way to read the KPI tree. Whereas Q1 is the main information we are looking for, notice that Q1.2.1.1 is the **leaf-node** question which collects metrics using the enterprise architecture repository data.

Q1: Which systems are the highest risk to operate?

Q1.1: Which systems have the highest operational risk?

Q1.1.1: Which systems have the highest staff risk?

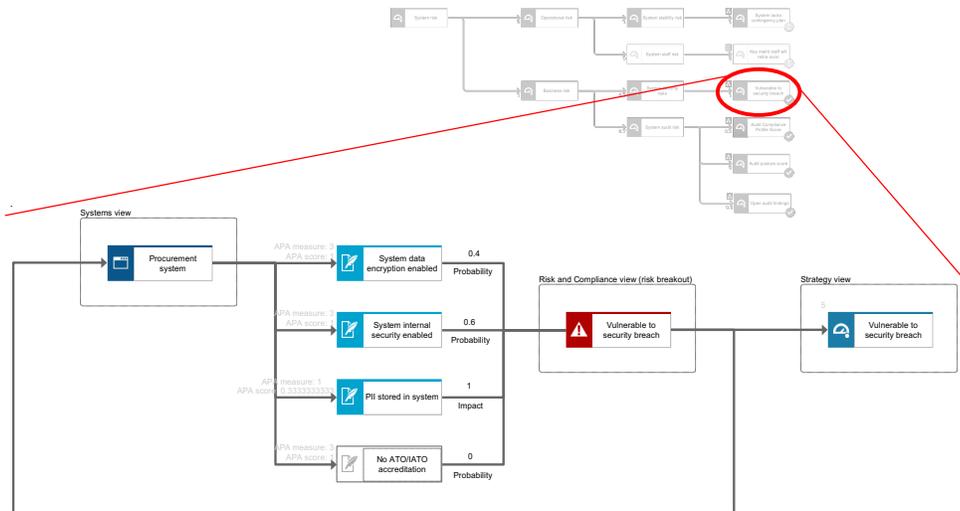
Q1.1.1.1: Which systems' key maintenance staff will retire soon?

Q1.2: Which systems have the highest business risk?

Q1.2.1: Which systems have the greatest security risk?

Q1.2.1.1: Which systems are the most vulnerable to security breach?

Continuing to the next level of detail, let's look at the metrics needed to assess the "Vulnerable to security breach" leaf-node KPI. It is determined using several metrics, as follows:



In this case, we have determined that we will collect three out of four proposed metrics (at this time). Those metrics are:

- System data encryption enabled? [true/false]
- System internal security enabled? [true/false]
- Personal Identifiable Information (PII) stored in system? [true/false]
- No Authority To Operate (ATO) accreditation? [true/false]. Notice that this metric is set to "0" probability, so it will be ignored (but is a placeholder for later). This helps us prioritize what detailed information to collect for our analysis – because collecting quality information usually takes a long time, and in the spirit of "agile EA", we may want to provide a preliminary answer faster.

Score the metric data and create normalized scores.



The metrics (i.e., "PII stored in system") are evaluated against the application system (i.e., Procurement system) and a score is assessed according to a formula:

```
"scoreType": "singleAttr", "evalType" : "boolean",
"NOT_MAINTAINED" : 3, "payload": { "false": 3, "true": 1 }
```

Using the *Metric Scoring Engine*, the score is assessed from 0-1, and this contributes to the "Impact" portion of a risk assessment ("Vulnerable to security breach"). This (red risk object) is from the Risk and Compliance view, and the same value is assigned to the "Vulnerable to security breach" leaf-node KPI.

Flatten the complex metric scores and KPIs into a spreadsheet.



The scores from all leaf-node KPIs are collected and summarized "up the tree", using a generic query defined in the *Query Generator* app. This results in a flattened CSV file, created on a daily report schedule automatically:

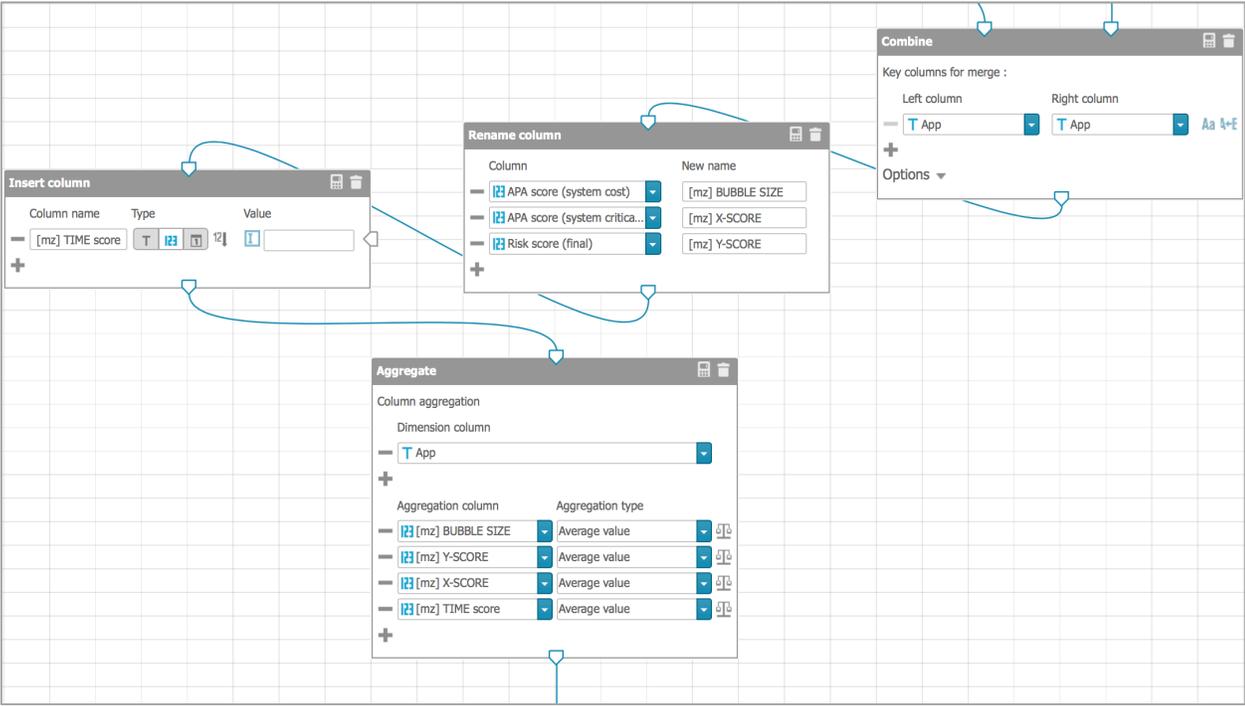
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	KPI(L0)	weight	KPI(L1)	weight	KPI(L2)	weight	KPI(L3)	Risk	APA - Risk as	weight	measure	max	score	App	
2	System risk	0.8	Business risk	0.3	System security risks	1	Vulnerable to security	Vulnerable to security	Probability	0.4	System data encryptk	3	1	Procurement system	
3	System risk	0.8	Business risk	0.3	System security risks	1	Vulnerable to security	Vulnerable to security	Probability	0.4	System internal secur	3	1	ERP system	
4	System risk	0.8	Business risk	0.3	System security risks	1	Vulnerable to security	Vulnerable to security	Probability	0.6	PII stored in system	3	1	Treasury system	

Crunch/transform the data (for dashboards)



Now that the data has been extracted from the architecture in a "flattened" format, it is ready to be transformed in a way suitable for dashboard consumption (i.e., pie charts, bar charts). In our scenario, this is done using the "Mashzone NG" product, but can also be done using ARIS Aware, or dashboard system that can receive CSV (Comma Separated Value) input.

Using Mashzone datafeeds can be constructed in a visual editor, without coding:



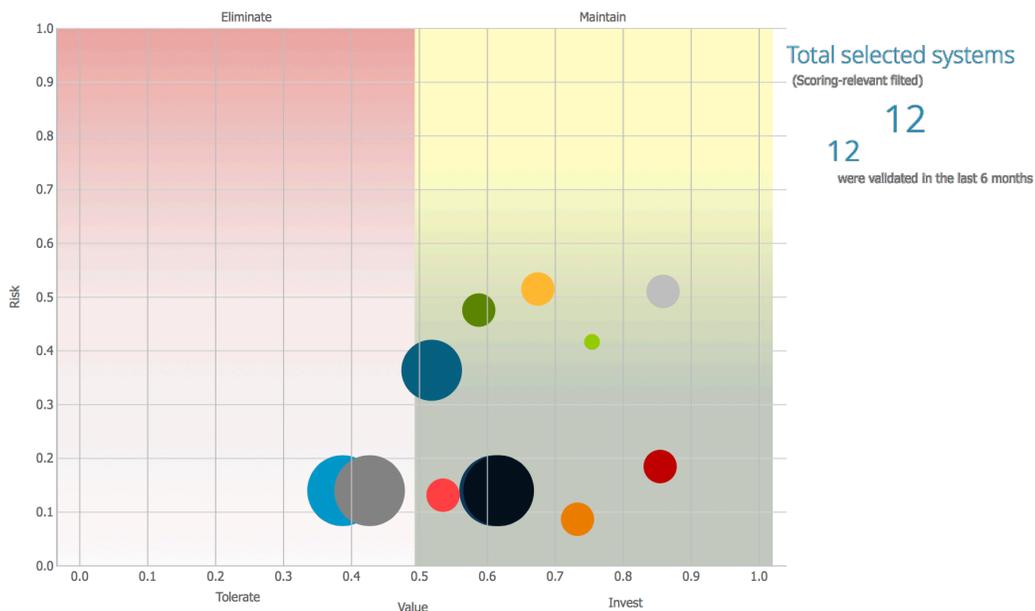
In this example, values from the summarized ("flattened") architecture – CSV files – are ingested into Mashzone, and the data is prepared to be presented in Bubble Chart representation.

In this step, it is important that analysts (who understand the business problem) are technical enough to work with the raw data input, and transform it in such a way that it can "tell a story" with a visual output.

Present the result in a dashboard.



In our case, the visual presentation is a Bubble Chart (using the TIME⁴ method). The four quadrants are a typical of a TIME chart, but the audit risk emphasis in this case aligns the X-axis to overall system risk. The Y-axis represents the value aspect of the system, which transforms a separate set of measures related to cost and benefits.



As part of the analysis, it is important for stakeholders to understand that they can trust the data. This is done by analyzing the quality attributes of the metrics that are the basis for the

⁴ Tolerate, Invest, Maintain, Eliminate. Source: "Gartner Application Portfolio Triage: TIME for APM" <https://www.gartner.com/doc/1115314/application-portfolio-triage-time-apm>

analysis. For example, "when was the data last updated" or "what is the source of the data"? In this case, it is sufficient to indicate how many of the objects in the Bubble Chart have been validated (by a knowledgeable person) in the last 6 months.

Value summary

When we opened the article, we promised several benefits of using an EA service-based approach to analysis: better, faster, and cheaper. Let's review and see how we did:

- Better. Using an EA service-based approach to answer "key business questions" can be justified as "better" than manual techniques (such as merging spreadsheets every time a new/updated answer is needed). The EA repository stores data from several sources (in our case study, from an audit findings team, combined with the input from the system owners). Information can be tagged with quality control attributes (i.e., the last time it was updated) and results can be based on "quality assured" information.
- Faster. Using the EA-service based approach, we don't re-collect information that has already been collected and previously validated. Using a combination of out-of-the-box analysis tools from professional EA products, pre-built toolchains and technical/non-technical people, we can produce answers faster, instead of having to launch a recurring mini projects to answer important business questions. Borrowing from the spirit of agile software development, we showed how to provide preliminary answers faster, and adding "placeholder" higher-quality analysis to metrics that might need more time to collect.
- Cheaper. Combined with an investment in professional EA tools, people, and role-specific training, day-to-day operations will scale and cost less in the long run. For example, the right mix of outsourcing "L4-developers" vs. training in-house "L2-analysts" will optimize value, bringing stronger analysis capabilities to existing full-time staff.

Contact us

Michael Idengren, Agile Architect CIO Advisory

T +1 352-870-8424

E midengren@kpmg.com

www.kpmg.com/us/ea

KPMG CIO Advisory helps your organization deliver EA value in the context of IT/operations efficiency, process improvements, technical solution assessments, and "strategy-to-implementation" digital transformation program governance.

© 2016 KPMG LLP, a Delaware limited liability partnership and the U.S. member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavor to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

The KPMG name and logo are registered trademarks or trademarks of KPMG International.

