White Paper

Modeling Tool Evaluation 101

WP0184 | April 2015

Peter Harrad

Peter has worked with modeling standards and techniques throughout his 20 years in IT, in a career that has covered software development, solutions architecture and international consulting.

Peter's particular areas of interest are opportunities arising from interdisciplinary touchpoints, how to balance practicality and rigor when modeling, and the importance of viewpoints in addressing different stakeholder perspectives.

Modeling Tool Evaluation 101

There are two things that I find striking about the modeling tools arena. First is the number of organizations that spend a 5- or 6-figure sum on a tool and then don't seriously use it (rarely is this because of any deficiencies of the tool). The second is how many organizations approach the purchasing of a tool in a disorganized fashion. There are enough recommendations out there about what a tool should or should not offer, that it seems pointless to add to it.

Instead, in this paper I'm going to outline a recommended approach on how to go about evaluating a modeling tool, based on what I've seen work, and what I've seen fail. It may seem to be stating the obvious that evaluating a tool is a project in its own right – but too many organizations don't act as if this is the case. The evaluation approach I will outline consists of the following 5 stages:

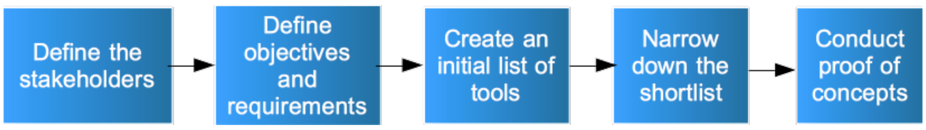
Define the stakeholders
Define objectives and requirements
Create an initial list of tools
Narrow down the shortlist
Conduct proof of concepts

Figure 1 – The evaluation process

Define the Stakeholders

Before you even start to talk about what your tool should do, you need to ask yourself who needs to be involved.

This stage might seem obvious; the stakeholders are the budget holder, and the people who will be using the tool. But in practice, once you start looking at a modeling tool, questions arise. First of all; what other initiatives might be interested? Most modeling tools don't just support one area of modeling – they'd be fools to. So the natural question that arises again and again is 'what other teams might be able to use this, now or in the future?'

The next area where you may run into unexpected stakeholders come from the touchpoints to other modeling environments. A tool that you plan to use for process mapping or business analysis has natural touchpoints. You're going to want to align your architecture models to your application portfolio, and so on. Thus it becomes worth engaging with the owners of other such tools and repositories to see if they have an opinion. At this stage, if some form of integration is under consideration, it's worth checking what interfaces the other tool supports (e.g., XML, CSV import) and what formats they have.

Architecture modeling tools	Process modeling tools
Content Management Systems	Content Management Systems
CMDBs	Architecture tools
Requirements management tools	Requirements management tools
Application Portfolios	Business activity monitoring
Project portfolio management systems	Workflow automation tools

Table 1 – Common touchpoints for modeling tools

The final area where unexpected stakeholders may appear is in the owners of processes that the tool might affect. For example, architectures may need to go through a review stage involving the project management office. Likewise, if the organization has a dedicated knowledge manager, they may have something to say about a process repository for common operating models.

This seems complex – and it is. There's a balancing act between pre-handling objections from other quarters and having so many signoffs that nothing can be done... and that's why I state that the first step should be a stakeholder analysis, so that you identify who needs to have a say, and who needs to be kept informed only. TOGAF has a decent section on stakeholder management in Chapter 24; ActiMate has a good set of objects for modeling stakeholder motivations (you can use Orbus' ArchiMate templates for this). It is effort, but it can save a lot of time further on.

Recommendation: Look at other initiatives, tool touchpoints and affected processes to identify stakeholders – and then work out who needs to be involved, and at what level.

Defining Objectives and Requirements

The starting point for designing your evaluation should be the business case for purchasing the tool in the first place. There are some standard reasons why an organization might be looking at a tool. What prompted your organization to start looking? Are you rolling out a new EA practice, or are you finding that the existing approach (usually what Gartner calls EVP – some mix of Excel, Visio and PowerPoint) isn't enabling sufficient reporting?

However, a 500-word statement that lists every possible benefit is as useless as no statement at all – it should be something that you would feel comfortable presenting to the C-suite. The goal at this point is to provide a focus. The following can be used as examples:

> *"We had a warning on our last compliance audit that our architectural controls were weak. One of the things that we will do to address this is implement a shared model managed in a modeling tool, so that each architect has visibility of what regulations affect what systems."*

> *"Each business unit currently operates in a different manner, so that processes and IT systems work differently in each unit, increasing costs. A centralized modeling tool will enable us to define shared processes, enabling standardization and cost reduction."*

Recommendation: Start with a clear, explicit statement of what the main benefits of the tool are intended to be.

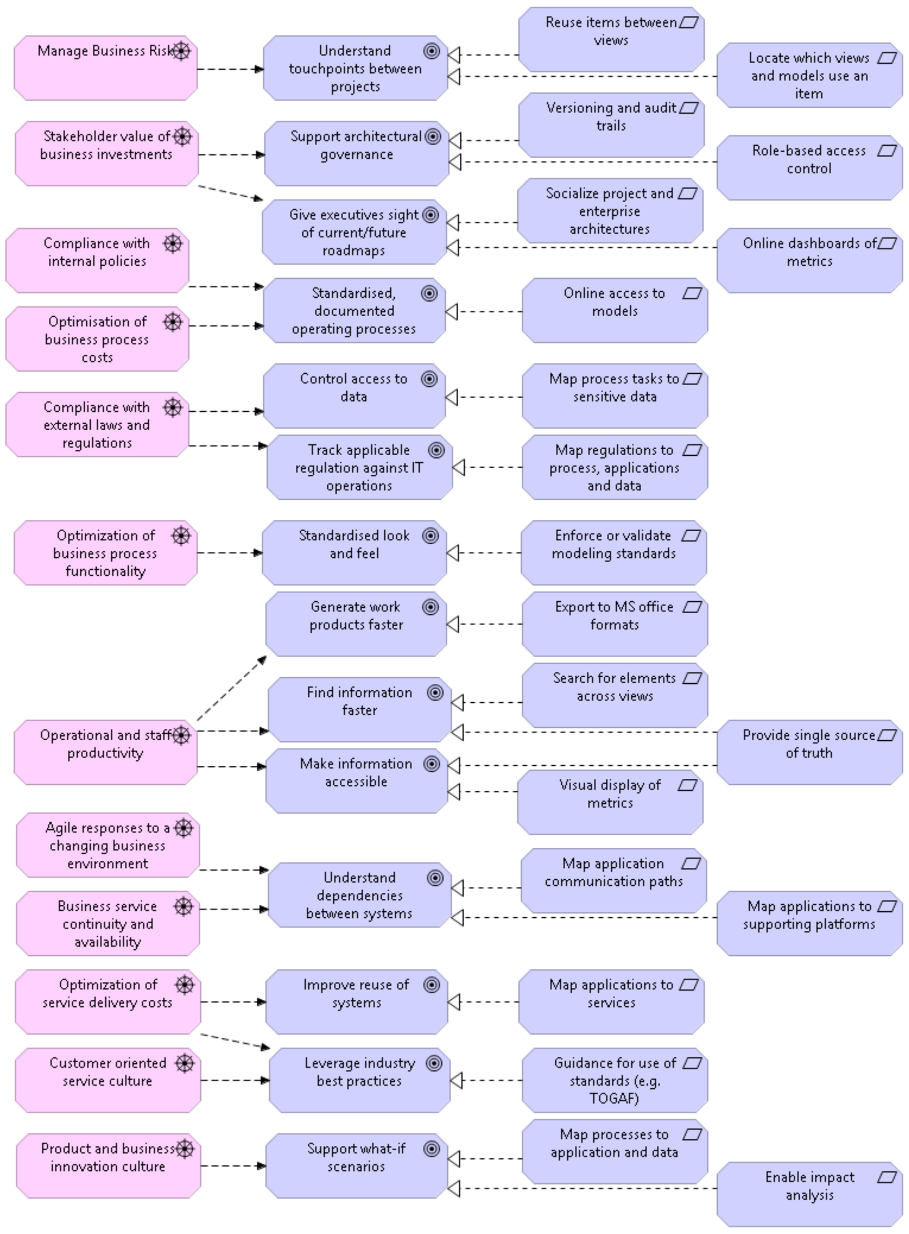
Manage Business Risk
Understand touchpoints between projects
Reuse items between views
Locate which views and models use an item
Stakeholder value of business investments
Support architectural governance
Versioning and audit trails
Role-based access control
Give executives sight of current/future roadmaps
Socialize project and enterprise architectures
Online dashboards of metrics
Compliance with internal policies
Optimisation of business process costs
Standardised, documented operating processes
Online access to models
Compliance with external laws and regulations
Control access to data
Map process tasks to sensitive data
Track applicable regulation against IT operations
Map regulations to process, applications and data
Optimization of business process functionality
Standardised look and feel
Enforce or validate modeling standards
Generate work products faster
Export to MS office formats
Search for elements across views
Find information faster
Operational and staff productivity
Provide single source of truth
Make information accessible
Visual display of metrics
Agile responses to a changing business environment
Map application communication paths
Understand dependencies between systems
Business service continuity and availability
Map applications to supporting platforms
Optimization of service delivery costs
Improve reuse of systems
Map applications to services
Customer oriented service culture
Leverage industry best practices
Guidance for use of standards (e.g. TOGAF)
Product and business innovation culture
Support what-if scenarios
Map processes to application and data
Enable impact analysis

Figure 2 - Common requirement areas for a tool (drivers taken from COBIT 5)

Once you have a clear statement of purpose, you can start defining requirements. It's the nature of tools that they all offer something – the point is to define which offers the right tradeoff for your organization, to achieve the goals listed. There are various documents available on the web that suggest what to look for in a modeling tool (some examples are listed at the end of this paper). It's also worth assigning a weighting function to your requirements, to let you prioritize which ones are most important. There are a few ways to approach this; High/Medium/Low, or the MoSCoW method. At the risk of sounding cynical and in the absence of any team preference, choose the approach that will speak to the budget holder.

Recommendation: Use the statement of benefits to prioritize your requirements in an agreed fashion.

Creating the Vendor List for Demonstrations

The next step is to identify a list of vendors to receive demonstrations from. There are several sources that you can use here.

The first place many organizations will refer to for a list of vendors are industry surveys, such as the Gartner Magic Quadrant and the Forrester Wave. This is a reasonable approach, but with one caveat – appearance in these reports requires that an organization make the investment to get a place. It's not true, as has sometimes been alleged, that placement in the Magic Quadrant is dependent on buying services from Gartner... but placement does depend on spending the time engaging with the analyst teams from these research firms. There are a minority of organizations that choose not to do so for various reasons, therefore the Magic Quadrant and the Wave should act as recommendations, not exclusion criteria.

A supplementary source of vendors may also come from looking at tool certification registers. There are lists of TOGAF and ArchiMate certified tools on the Open Group website. However, as with research organizations, there is a caution – this should be treated as a source of names only. The certification questionnaires are self-certified and many of the certification questions are open to interpretation, so the responses given in the certification questionnaires are usually not a reliable apples-to-apples comparison.

The third resource option could be personal recommendations from the professional network of the evaluation team. Do former colleagues have experience with any tools?

Once you have your wide-ranging list of vendors, it will be time to slim it down to those vendors that you will take a preliminary product demonstration from. You should budget 1-2 hours for each demonstration, including time for follow up questions and post-demonstration discussions. Most organizations that I've dealt with will have demonstrations from around 5-6 vendors.

Some questions to check when creating the list of demonstrators are:

- What are the vendors support hours (and in what time zone)?
- What customers can they reference in your region and industry?
- What standards bodies do they participate in?

Creating the Shortlist – Demonstration Time

With a set of vendors established, the next step is to slim the list down to 2-3 that you perform some hands-on evaluation time with. To reach this step, it's time to invite the vendors to demonstrate their products to you.

Now, while most vendors will have some kind of demonstration script, it is highly recommended you produce a list of questions that you want answered and some functionality points that you want to see demonstrated. This is actually not something to feel uncomfortable about doing – the sales prospect who can't decide what they want is a vendor nightmare.

Some sample questions to ask during a demonstration are:

- What are the ballpark investments? (You should know how many seats of each type of role you will need as well as have an idea of 'must-have' functionality here).
- How long does it take to get up to speed with the tool?
- What is the administrative overhead?
- How often do you release?
- Is there a user group?
- How do you support
    - reference models
    - governance
    - transformation road mapping

The functionality points you should ask to see should be driven by the use cases that you've identified. Some examples:

- Reusing an item in an architecture
- Publishing a diagram to HTML
- Collecting a review comment
- Generating a document
- Importing from a spreadsheet (supplied ahead of time)
- Running a custom report

Recommendation: Have a list of questions ready to provide to vendors and 3-5 things that you'd like to see demonstrated.

Each demonstration attendee should give a score on the functionality area in question – this enables more effective pooling of opinions to refer back to.

Recommendation: Collate a set of independent scores from the demonstrations.

Depending on the respective locations of the vendor and your team members, the demonstration might be conducted remotely via some kind of conferencing software such as Webex or GoToMeeting. In such a case, it's worth checking connectivity prior. Most software for remote meetings has some form of test meeting. On the plus side, most such software also allows you to record the session.

Recommendation: Ask if the demonstration can be recorded.

Running a Proof of Concept

At this point you should have a list of two, or maximum three vendors to perform a Proof of Concept with. Just as the demonstrations will be much more productive if there are checklists of questions and functionality points to see defined ahead of time, the proof of concept will be much more effective if you state a set of use cases that you will test.

Use Case:	Import from CMDB
Description:	This use case tests the ability of the tool to batch import from the CMDB overnight
Preconditions:	- An import job has been configured in the tool
	- Tool contains one item that is in the CMDB with an attribute that has a different value to that in the CMDB
	- Tool contains one item that is in the CMDB with identical attributes
	- Tool does not contain one item that is in the CMDB
Actions:	- Take an export from the CMDB and place it in the import folder
Success Criteria:	- Import completes successfully
	- The item that existed with a different attribute is updated
	- The item that did not exist in the tool is imported with the correct attribute values
	- The item that existed in the tool is unchanged

Table 3 – Sample Use Case

Recommendation: Have a set of use cases ahead of time with success criteria for each.

The second thing to get in place before the Proof of Concept officially kicks off is to ask for access to the vendor's support resources. The purpose of the Proof of Concept is to try and test the product - a software product is more than just a load of code, it also includes the support organization behind the software, and the resources that are available to customers. This very fact means that a thorough evaluation of the product should also include a thorough evaluation of the support for the software package.

Recommendation: Ask for access to the vendor's user groups and support forum as part of the Proof of Concept.

Now, the biggest problem that I see with organizations that are conducting a Proof of Concept is time pressure. I have even seen organizations that get a Proof of Concept established and no-one logs in to the system for a week (as an aside, this is one reason why vendors often charge for Proof of Concepts, to concentrate minds). So it's vital to reserve time for the activities of the PoC. It's true that in reality, you won't be able to spend the entire time on the evaluation, but unless you reserve the necessary time on the calendar, and try very hard to protect it, you more or less guarantee that you won't get through your use cases.

Below is a sample timetable; in practice, each day should only require an hour or two of effort

	Mon	Tue	Wed	Thurs	Fri
Week 1	Creation of models	Creation of models	Reuse between models	Finding information	Publication to stakeholders
Week 2	Support for internal review	Governance	Impact Analysis	Custom Reporting	Conclusions

Table 4 – Sample two-week evaluation timetable

Recommendation: Establish a time boxed schedule for your evaluations and reserve time for it.

Now, the above recommendation does demand a time investment from busy people. One way to address this is to assign specific areas of responsibility. That is, different members of the team each have specific use cases to evaluate. Others can do so and give a score if they want, but there is only one person who is specifically responsible for a given area.

Recommendation: Consider a divide-and-conquer approach; different members of the team own the evaluation of a given area.

Conclusion

- Look at other initiatives, tool touchpoints and affected processes to identify stakeholders, and then work out who needs to be involved at what level
- Start with a clear, explicit statement of what the main benefits of the tool are intended to be
- Use this statement to establish and prioritize your requirements in an agreed fashion
- Research the tools market to create a shortlist
- Use product demonstrations to identify 2-3 tools for Proof of Concepts
- Collate a set of independent scores from the demonstrations
- Have a set of use cases ahead of time with success criteria for each
- Establish a time boxed schedule for your Proof of Concepts and reserve time for them
- Divide-and-conquer; different members of the team own the evaluation of a given area

References

TOGAF, chapter 24: “Stakeholder Management”
pubs.opengroup.org/architecture/togaf9-doc/arch/chap24.html

The Open Group, “Why do I need an enterprise architecture?”
pubs.opengroup.org/architecture/togaf9-doc/arch/chap01.html#tag_01_02

The Enterprise Architecture Benefits Framework
www.cs.uu.nl/research/techreps/repo/CS-2010/2010-014.pdf

Gartner Assessment of Enterprise Architecture Tool Capabilities, 2014
www.gartner.com/doc/2707223/gartner-assessment-enterprise-architecture-tool

Mastering ArchiMate, chapter 30: “Tool Requirements”
en.wikipedia.org/wiki/MoSCoW_method

Orbus Software
3rd Floor
111 Buckingham Palace Road
London
SW1W 0SR
United Kingdom

+44 (0) 870 991 1851
enquiries@orbussoftware.com
www.orbussoftware.com