White Paper

The Top 10 Factors to Consider When Selecting a TOGAF® 9 Repository

WP0005 | July 2011

Louw Labuschagne CBPA®

Louw is a Managing Partner at CS Interactive Training, a specialist IT consultancy focused on providing methodology consulting, training and systems to organizations who need to build internal capacity within their Analysis, Architecture, Design, and Requirements Management environments. Louw is passionate about all aspects of information management and has had the opportunity to act as strategist, architect, speaker, trainer, analyst, modeler and developer within this field over the past 20 years.

To establish a sustainable Enterprise Architecture Practice based on TOGAF 9 you need a solid architecture repository. When doing an internet search for Enterprise Architecture (EA) repository tools I found everything, from open source ontology projects to master data management software being packaged as EA repositories, so finding a proper EA tool that will support your initiatives is not a trivial task. I have worked on a variety of EA projects using different tools to capture and manage the concepts, and the relationships between those concepts, that I require to produce useful information that I can package for different stakeholders.

Based on my own experience, and those of the architecture professionals that I have worked with and trained over the past 8 years, I believe the following 10 key factors must be considered before jumping into a licensing agreement. Furthermore, I don't base my decision on which tool to buy just because the vendor tool is placed in a certain quadrant on a report by a research analyst.

Number 1: EA Repository

(It must be a real repository)

During my earlier years as a consultant I was exposed to a whole range of products that could be very loosely defined as "architecture repositories". I grudgingly used these tools out of desperation, just to save my models in a re-usable format whilst trying to convince the client or boss to invest in a sustainable architecture repository. The standard answer I almost always got was; "You have Visio installed on your laptop, what more do you need?".

Well what I needed then and now, as a bare minimum, was to have my models shared with the rest of the team and the ability to re-use the models on multiple views and projects. After a while, as I started working on new projects, I only had one rule when asking the IT Department for a modeling tool; the tool must use a repository.

Soon after I made this rule I worked at an organization that provided IT outsourcing services to large clients, and after requesting a modeling environment with a single repository, I received a UML modeling tool linked to a Subversion Version Control Repository; not what I wanted, but that was what I asked for! I must admit, it did work and we did complete the project, but at most I can say we did not strictly follow the UML rules, leading to an unsustainable repository in the long run.

After the project I updated my rule: I now want to be able to access the data (or building blocks in TOGAF terminology) without being forced to use the modeling front-end. The benefit I find with this is that you have a self-contained, real repository with referential integrity (I prefer relation databases) and with the ability to extract data in different formats as the projects require without using predefined reports or viewpoints.

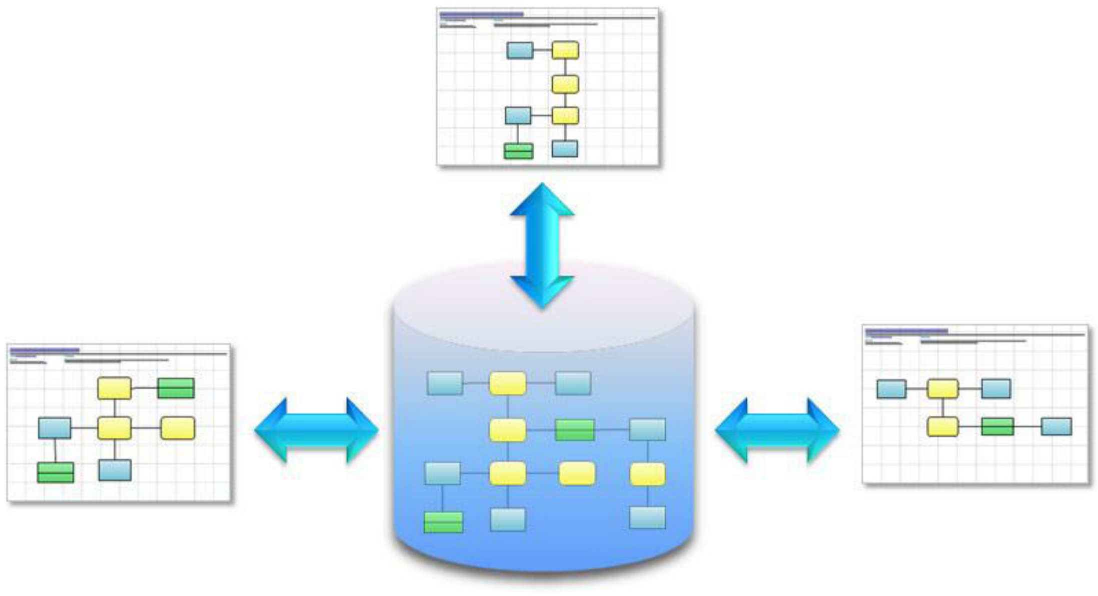


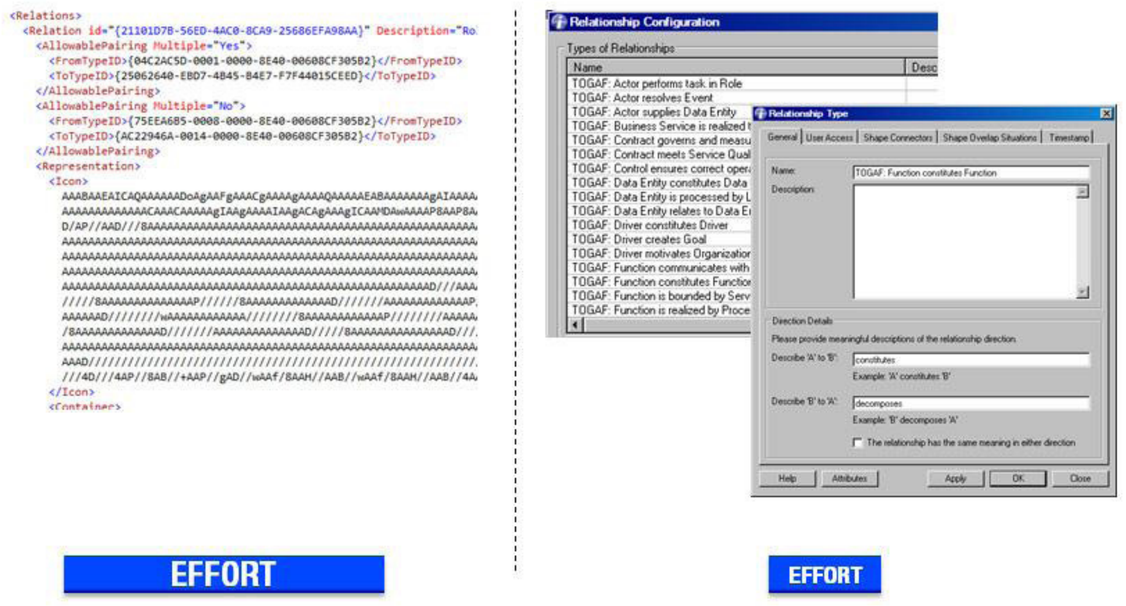
Figure 1: A Central Enterprise Architecture Repository with the ability to extract different views of the information

Number 2: Extending the meta-model

(it must be simple to extend the meta-model)

Starting my modeling career using a UML modeling tool and an inflexible repository, I soon became agitated with a tool that will not allow me to extend the basic model types to fit the requirement of the client (although the UML language is great for creating extensions). I had no choice, but to create a little instruction manual on how our project team interpreted the UML notation and how we used different colours to represent different building block types in the organization. If someone forgot to change the colour of a shape, we had data inconsistency!

The result was my second rule, the repository meta-model must be updatable by someone who does not necessarily hold an advanced degree in microprocessor design or quantum physics.



```
<Relations>
  <Relation id="{21101D7B-56ED-4AC0-8CA9-25686EFA98AA}" Description="Ro
    <AllowablePairing Multiple="Yes">
      <FromTypeID>{04C2AC5D-0001-0000-8E40-00608CF305B2}</FromTypeID>
      <ToTypeID>{25062640-EBD7-4845-B4E7-F7F44015CEED}</ToTypeID>
    </AllowablePairing>
    <AllowablePairing Multiple="No">
      <FromTypeID>{75EEA685-0008-0000-8E40-00608CF305B2}</FromTypeID>
      <ToTypeID>{AC22946A-0014-0000-8E40-00608CF305B2}</ToTypeID>
    </AllowablePairing>
    <Representation>
      <Icon>
        AAABAAEAICAQAAAAAADoAgAAFgAAACgAAAAgAAAAQAAAAAEABAAAAAAAgAIAAAA
        AAAAAAAAAAAAACAAACAAAAAgIAAgAAAAIAAgACAgAAAgICAAMDAwAAAAP8AAP8A
        D/AP//AAD///8AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD///AAA
        /////8AAAAAAAAAAAAAP//////8AAAAAAAAAAAAAD///////AAAAAAAAAAAAAP
        AAAAAAD////////wAAAAAAAAAAAA////////8AAAAAAAAAAAAP////////AAAAA
        /8AAAAAAAAAAAAAD///////AAAAAAAAAAAAAD/////8AAAAAAAAAAAAAAAD///
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAD//////////////////////////////////////////////////////////
        ///4D///4AP//8AB//+AAP//gAD//wAAf/8AAH//AAB//wAAf/8AAH//AAB//4A
      </Icon>
      <Container>
```

Relationship Configuration

Types of Relationships

Name	Desc
TOGAF: Actor performs task in Role	
TOGAF: Actor resolves Event	
TOGAF: Actor supplies Data Entity	
TOGAF: Business Service is realized t	
TOGAF: Contract governs and measu	
TOGAF: Contract meets Service Qual	
TOGAF: Control ensures correct oper	
TOGAF: Data Entity constitutes Data	
TOGAF: Data Entity is processed by L	
TOGAF: Data Entity relates to Data E	
TOGAF: Driver constitutes Driver	
TOGAF: Driver creates Goal	
TOGAF: Driver motivates Organizatior	
TOGAF: Function communicates with	
TOGAF: Function constitutes Functio	
TOGAF: Function is bounded by Serv	
TOGAF: Function is realized by Proce	

Relationship Type

Name: TOGAF: Function constitutes Function

Description:

Direction Details

Please provide meaningful descriptions of the relationship direction.

Describe 'A' to 'B': constitutes

Example: 'A' constitutes 'B'

Describe 'B' to 'A': decomposes

Example: 'B' decomposes 'A'

☐ The relationship has the same meaning in either direction

Figure 2: Ability to extend the metamodel with minimal effort is essential

Number 3: User interface

(It must have an intuitive User interface)

I can honestly say that when I see the user interface designs, or Human Computer Interface (to use the correct term), of some of the larger Enterprise Systems, the average Architecture Repository vendor is not doing too badly. On average I find that an experienced modeller will find them intuitive and of benefit, but the occasional modeller, Business Analyst or Senior Enterprise Architect will prefer not to break the rhythm of the experienced modellers by asking silly questions, so they continue to model in Visio and then print the models for the experienced (and more junior) modellers to recapture into their sophisticated modeling tool.

This way, everyone is productive and the timesheets are always filled with extra hours of modeling. However, this bliss only lasts until the project timelines start slipping, and the project managers start asking the difficult question of why he need to pay for two resources who are duplicating the same work (and of course the client also wants to know why the effort is duplicated; he prefers the Visio model because it looks pretty, so just stop re-doing the work!).

Rule number 3 states that any tool that I use on a project must be able to use Visio, import Visio diagrams and map it to my custom meta-model or visually look like Visio!

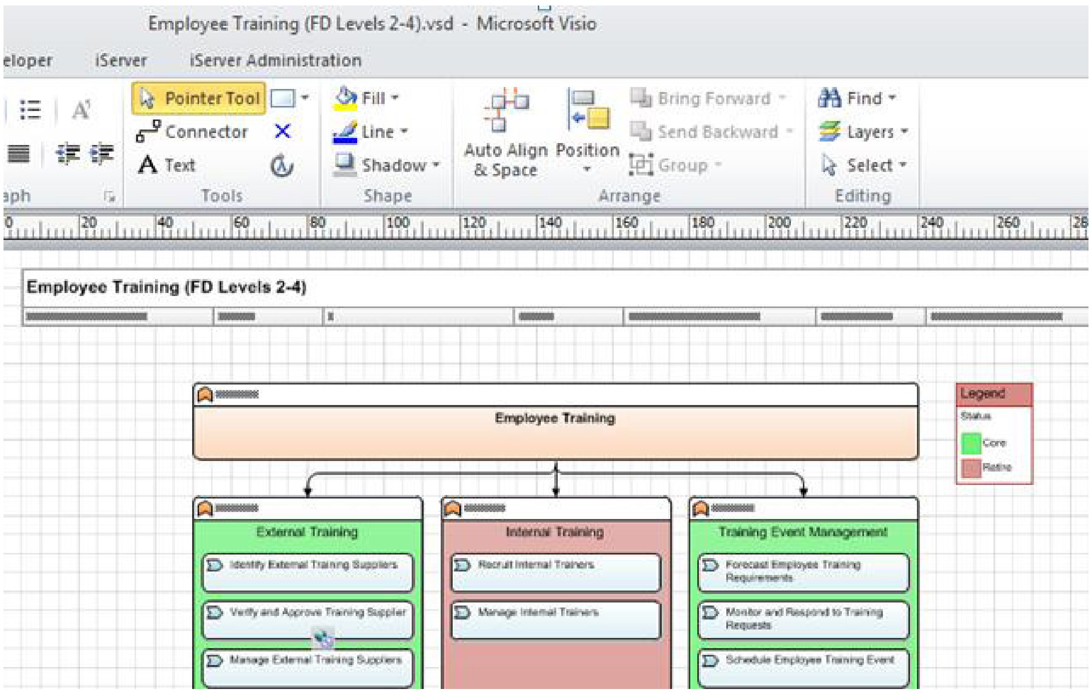

Employee Training (FD Levels 2-4).vsd - Microsoft Visio
eloper
iServer
iServer Administration
Pointer Tool
Connector
Text
Tools
Fill
Line
Shadow
Shape
Auto Align & Space
Position
Bring Forward
Send Backward
Group
Arrange
Find
Layers
Select
Editing
Employee Training (FD Levels 2-4)
Employee Training
External Training
Identify External Training Suppliers
Verify and Approve Training Supplier
Manage External Training Suppliers
Internal Training
Recruit Internal Trainers
Manage Internal Trainers
Training Event Management
Forecast Employee Training Requirements
Monitor and Respond to Training Requests
Schedule Employee Training Event
Legend
Status
Core
Retire

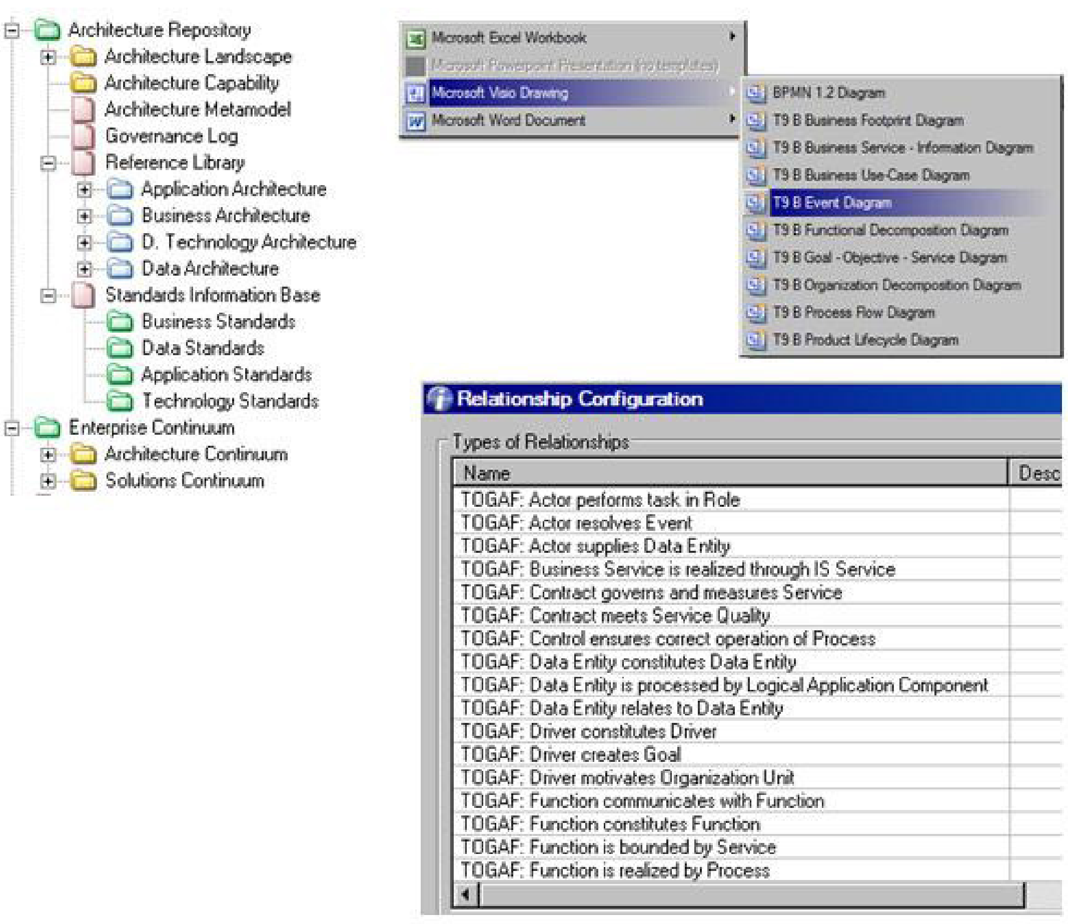
Figure 3: The Visio interface: clear and easy to use

Number 4: TOGAF 9 support

(Out-of-the-box TOGAF 9 Support is required)

You cannot just create random models. Well, if you have Visio the sky is the limit, so maybe you can just model, but the sustainability and re-use of the models will sometimes not even reach the end of the project. The biggest concern I have with Open Source software (and Visio) is that you only own a tool that will draw the shapes; there is no intelligence behind it or a methodology that gives it a purpose in life! The major tool vendors all ship with integrated modeling environments and a standard methodology that helps the modeller understand how and when to use the modeling shapes (my first exposure to this was with a German developed modeling tool, using a house as a starting point).

Enterprise Architecture projects are more difficult to tackle than Business Process Improvement or Solution design projects, where standards like BPMN, UML and Entity Relationship Models are standard notations with solid methodologies to support the modeling effort. The Open Group's TOGAF 9 framework together with the new ArchiMate notation is an excellent starting point for any Architecture project. Using TOGAF to define enough governance controls, even Visio modellers can be allowed to roam free on a project. The TOGAF 9 Architecture Development Method allows me to define a set of Viewpoints at the start of the project, which I can confirm with stakeholders and distribute to the architects and modellers to create, with a predictable, consistent result at the end. If I now have a choice in selecting a tool, I always prefer (rule 4) to have TOGAF 9 support out of the box.



Name	Desc
TOGAF: Actor performs task in Role	
TOGAF: Actor resolves Event	
TOGAF: Actor supplies Data Entity	
TOGAF: Business Service is realized through IS Service	
TOGAF: Contract governs and measures Service	
TOGAF: Contract meets Service Quality	
TOGAF: Control ensures correct operation of Process	
TOGAF: Data Entity constitutes Data Entity	
TOGAF: Data Entity is processed by Logical Application Component	
TOGAF: Data Entity relates to Data Entity	
TOGAF: Driver constitutes Driver	
TOGAF: Driver creates Goal	
TOGAF: Driver motivates Organization Unit	
TOGAF: Function communicates with Function	
TOGAF: Function constitutes Function	
TOGAF: Function is bounded by Service	
TOGAF: Function is realized by Process	

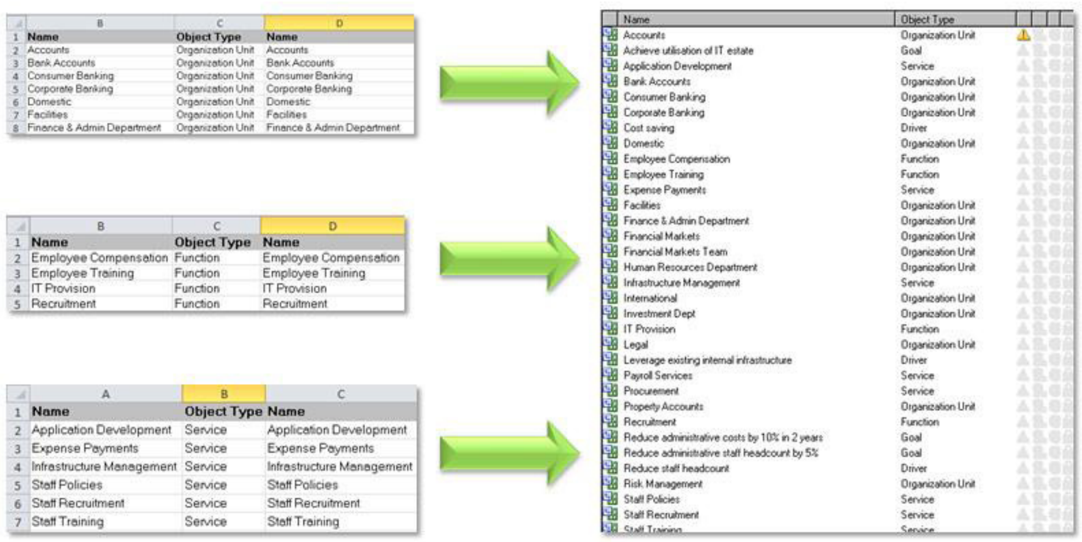
Figure 4: 'Out-of-the-Box' Support: A Pre-structured repository with pre-pre-defined templates and relationships will both guide and accelerate your EA initative

Number 5: Bulk Uploads of Data

(It must be easy to load data in bulk)

My love for modeling lasted until I had to create over 700 applications objects by hand and then just over 300 process objects (not counting the manual linking I had to do to get any value out of the process). Sticking to rule 1, I had access to a repository (with a relational database management system), which allowed me to peek into the back-end of the tool. Within a few hours I had the key tables and attributes identified. By running scripts on the database I was able to import the rest of the building blocks in a fraction of the time it took to create the original set.

I tried a different strategy with my next assignment and e-mailed excel spread sheets with fixed columns and rows to stakeholders and, by using the data extraction script, uploaded and linked a large amount of building blocks in a very short time. When I now evaluate modeling tools, I also include rule number 5 – the Modeling tool must be able to import excel spread sheet data.



	B	C	D
1	**Name**	**Object Type**	**Name**
2	Accounts	Organization Unit	Accounts
3	Bank Accounts	Organization Unit	Bank Accounts
4	Consumer Banking	Organization Unit	Consumer Banking
5	Corporate Banking	Organization Unit	Corporate Banking
6	Domestic	Organization Unit	Domestic
7	Facilities	Organization Unit	Facilities
8	Finance & Admin Department	Organization Unit	Finance & Admin Department

	B	C	D
1	**Name**	**Object Type**	**Name**
2	Employee Compensation	Function	Employee Compensation
3	Employee Training	Function	Employee Training
4	IT Provision	Function	IT Provision
5	Recruitment	Function	Recruitment

	A	B	C
1	**Name**	**Object Type**	**Name**
2	Application Development	Service	Application Development
3	Expense Payments	Service	Expense Payments
4	Infrastructure Management	Service	Infrastructure Management
5	Staff Policies	Service	Staff Policies
6	Staff Recruitment	Service	Staff Recruitment
7	Staff Training	Service	Staff Training

Name	Object Type
Accounts	Organization Unit
Achieve utilisation of IT estate	Goal
Application Development	Service
Bank Accounts	Organization Unit
Consumer Banking	Organization Unit
Corporate Banking	Organization Unit
Cost saving	Driver
Domestic	Organization Unit
Employee Compensation	Function
Employee Training	Function
Expense Payments	Service
Facilities	Organization Unit
Finance & Admin Department	Organization Unit
Financial Markets	Organization Unit
Financial Markets Team	Organization Unit
Human Resources Department	Organization Unit
Infrastructure Management	Service
International	Organization Unit
Investment Dept	Organization Unit
IT Provision	Function
Legal	Organization Unit
Leverage existing internal infrastructure	Driver
Payroll Services	Service
Procurement	Service
Property Accounts	Organization Unit
Recruitment	Function
Reduce administrative costs by 10% in 2 years	Goal
Reduce administrative staff headcount by 5%	Goal
Reduce staff headcount	Driver
Risk Management	Organization Unit
Staff Policies	Service
Staff Recruitment	Service
Staff Training	Service

Figure 5: Bulk importing artefacts (from MS Excel or other) to the repository saves hours of work!

Number 6: Visualisation of Views

(It must be varied, accessible and stunning)

I have spent a large amount of my time trying to get information into a repository, but with some tools it took me even longer just to generate a report or view that was presentable to non-technical stakeholders. Some tool vendors are building the modeling tools and repositories for architects and modellers but do not realize that a major beneficiary of the repository is the senior manager, who needs to make key decisions using information from the repository. Important information is often hidden to them by layers of inaccessible, obscure modeling notation and complicated report generation engines. I encountered this problem very early in my career, but am only now starting to see commercial vendors develop tools that are designed with the real end-user in mind. So, non-negotiable rule number 6 is: the views created from the repository must be visually impressive and easy to access.

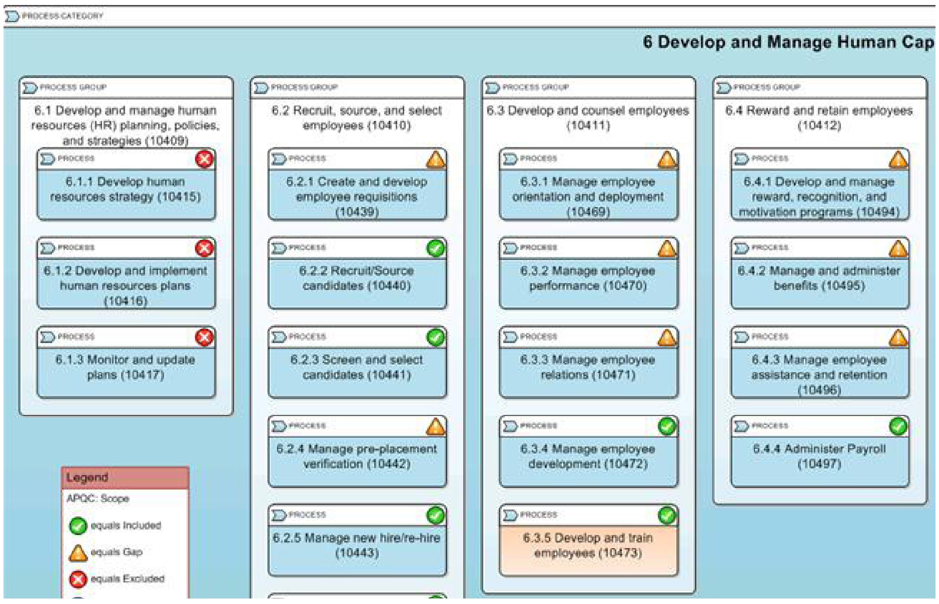

PROCESS CATEGORY
6 Develop and Manage Human Cap
PROCESS GROUP
6.1 Develop and manage human resources (HR) planning, policies, and strategies (10409)
PROCESS
6.1.1 Develop human resources strategy (10415)
6.1.2 Develop and implement human resources plans (10416)
6.1.3 Monitor and update plans (10417)
6.2 Recruit, source, and select employees (10410)
6.2.1 Create and develop employee requisitions (10439)
6.2.2 Recruit/Source candidates (10440)
6.2.3 Screen and select candidates (10441)
6.2.4 Manage pre-placement verification (10442)
6.2.5 Manage new hire/re-hire (10443)
6.3 Develop and counsel employees (10411)
6.3.1 Manage employee orientation and deployment (10469)
6.3.2 Manage employee performance (10470)
6.3.3 Manage employee relations (10471)
6.3.4 Manage employee development (10472)
6.3.5 Develop and train employees (10473)
6.4 Reward and retain employees (10412)
6.4.1 Develop and manage reward, recognition, and motivation programs (10494)
6.4.2 Manage and administer benefits (10495)
6.4.3 Manage employee assistance and retention (10496)
6.4.4 Administer Payroll (10497)
Legend
APQC: Scope
equals Included
equals Gap
equals Excluded

Figure 6: Clear, easy to understand and visually impressive views are a must for stakeholders

Number 7: Data analysis

(It must have native support for data analysis)

I have spent some time as a business intelligence and data warehouse consultant, so I am really passionate about empowering the business (and IT) decisions makers. If they can explore their domains and apply their knowledge and expertise to the information that they see, the business insight that flows from it places the organization in a very competitive and market leading position. Organizations, business stakeholders and architecture tools must still go through a maturing process to achieve this kind of insight. I am starting to apply rule number 7 now and I will not buy or implement a tool in the future that does not give me the capability to dynamically interact with the repository and perform data analysis on objects.

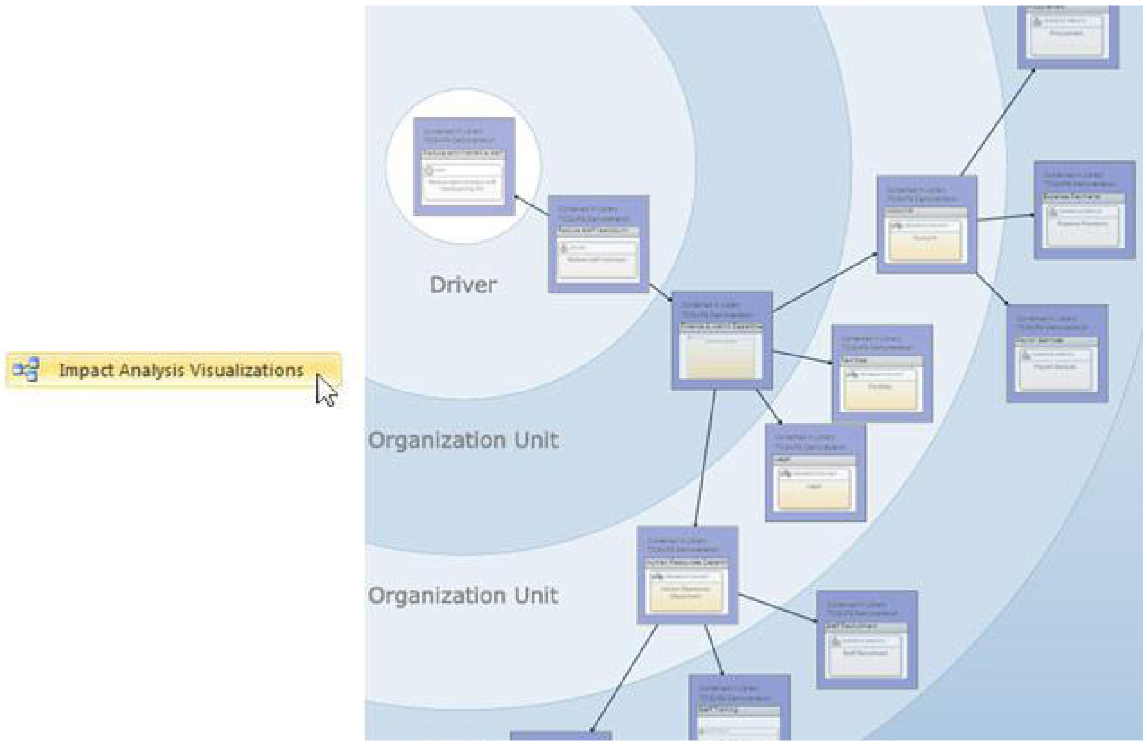


Figure 7: The value of any repository is limited without a capability to perform analysis on its content

Number 8: Information Assurance

(Users must believe the data is accurate and consistent and it must be readily available)

Nothing drains the energy out of a project or initiative as quickly as when stakeholders lose faith in the quality of the information being presented or by the process in which information is collected. Despite the fact that advanced architecture repositories are designed and implemented using cutting-edge technology, I have found that the information quality and the consistency of the repository must be reviewed on a regular basis by performing technical quality assurance (running internal consistency scripts in the repository).

Technical integrity tools and reporting must come standard with any good repository, so that is my rule number 8: ensure that the vendor and tool understands what is meant by technical integrity checking. Otherwise, run!

Unfortunately, the logical quality assurance is a human function and although I would have loved to embed an Artificial Intelligence module into the repository to do the checking, it is still a role that has to be performed by someone on the Enterprise Architecture team.

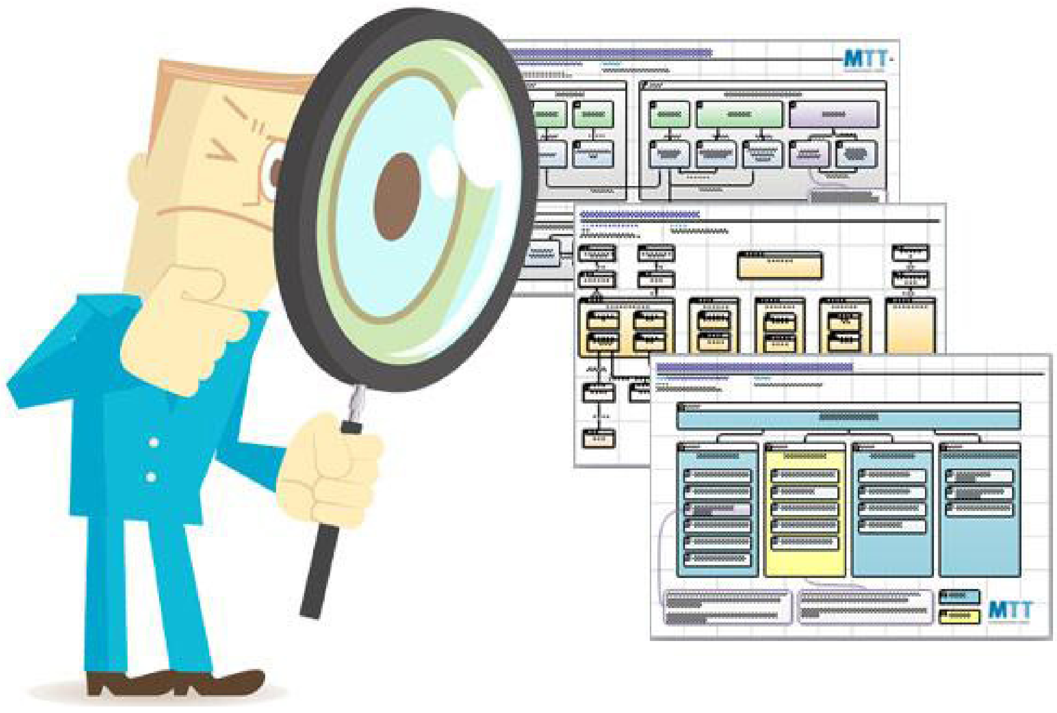


Figure 8: Human scrutiny is always required to ensure the integrity of repository data

Number 9: Technical Support

(It must use familiar infrastructure)

Technical support for your repository infrastructure is critical. My simple rule 9 is therefore: pick technology that your internal support staffs are comfortable with, i.e. if you are using SQL Server and Windows Server in your data centre then pick a tool with those technology components. It just makes life much simpler during routine support and maintenance tasks.

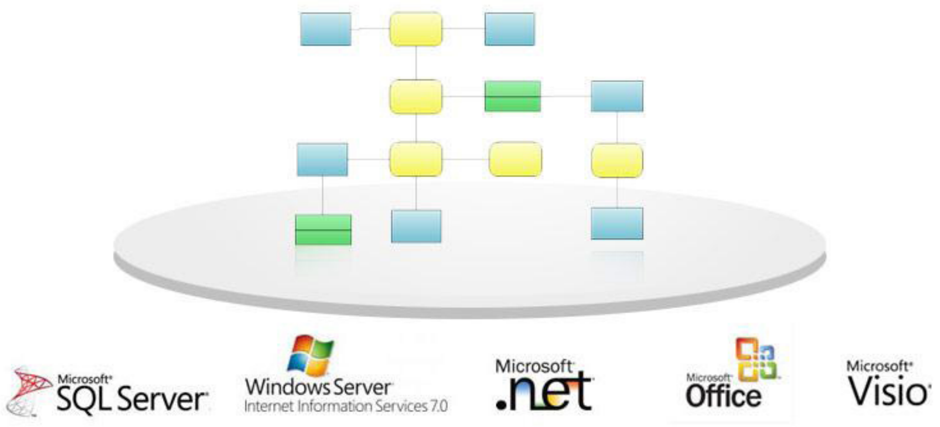


Figure 9: The technologies supporting the repository should be proven and familiar to your support staff.

Number 10: Vendor Support

(Vendor support must be credible)

The last point seems obvious, but buying software from a website or downloading “community edition” software will give you exactly the kind of support that you paid for… zero. Building your own repository is always an option; I had a successful SQL Server-based repository running for about a year before the novelty wore off and the D.I.Y. repository ended up in the store room with all the other novel ideas.

Maintaining and supporting an Enterprise Architecture Repository is a serious business and that is why I want to end with my rule number 10: get the professionals in.

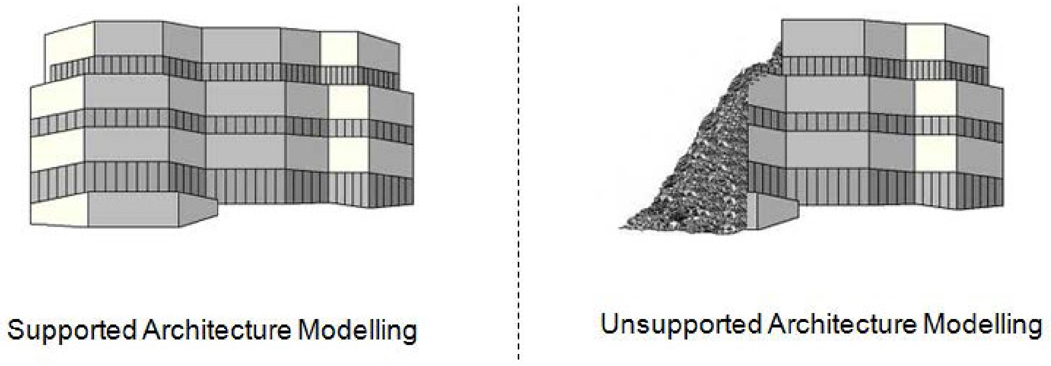

Supported Architecture Modelling
Unsupported Architecture Modelling

Figure 10: Solid support from your vender is the key to avoiding nasty surprises in the future.

Conclusions

In conclusion, I am not attempting to create an architecture tool comparison table, just because it is so subjective and because there are so many of those tables already available (and also if I excluded the list of business process management tools that charade as EA tools, then I might just get a backlash from the vendor community).

I do however think that when evaluating EA tools, practitioners should play close attention to the rules I have outlined in this document. If you have a tick in the box for all of the above you will have the best chance of creating a repository that is capable of delivering the business value you require from your EA efforts.

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

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