To develop a useful enterprise architecture (EA) it is important to first understand the questions you want to answer with your architecture. Then based on these questions, you can develop an approach and identify the models that you need. Finally, you can do both quantitative and qualitative analysis on your architecture to either see where the business can be improved or identify necessary changes or enhancements to the architecture. This article offers a summary of an enterprise architecture program and its processes.

Enterprise architecture is a logical organization of a business and its supporting data, applications, and IT infrastructure, with clearly defined goals and objectives for the future success of the business. A typical architecture consists of diagrams, or models, that show how aspects of your business relate. For example an organizational chart is a model of how business units relate to each other.

Businesses should have an "as-is" architecture that represents its current state, and a planned architecture to show the direction of the business over the next one to five years.

Enterprise architecture aligns the following key areas. Note the examples in each area:

- **Business**: Processes, strategies, organization charts, and functions
- **Information**: Conceptual, logical and physical data models to show what information is needed and how it relates to other information For example, a customer and an order
- **Application**: Portfolios, interfaces, and services
- **Infrastructure**: Network concept diagrams, technology reference models

To achieve alignment, you model each key area from its own perspective, and then link the models from each perspective. For example, model business processes from a business perspective. Do not include things like applications. Then, link the business processes to the applications that support them, which helps you achieve alignment. We do this to ensure that every decision is based on a business need; therefore, an application is not dictating the way a business process is designed.

Throughout this article, we assume that you have a modeling tool to create your architecture. The implementation specific information in this article is based on Rational System Architect.
"We are doing architecture... to do architecture!"

Figure 1. If you do not have a purpose, your project will fail

The simplest way to ensure that your architecture fails is to not have a purpose for doing it. I've worked on architecture projects with hundreds of clients. When projects are not successful, I ask why they are creating an enterprise architecture. They respond, "Because we want an architecture!"

Step 1. Identify the purpose of your architecture

You can define the purpose of your architecture by asking the following questions:

• What information is important for the architecture?
• How much detail is needed to support analysis and decision making?
• Who will produce or use the architecture?
• What is the expected ROI of the architecture?
• What are the maintenance considerations?

If you cannot answer these questions, your architecture project will likely fail. Without a purpose you can waste months drawing business process diagrams that no one cares about. Or you may draw complex diagrams of application interfaces that cannot be presented to senior management because it will make their heads explode.

As an example, for a hotel chain, hotel managers were identified as the audience for EA.

By knowing the purpose of the architecture, you can scope the necessary models and data that are needed to ensure people use your architecture for analysis and business decisions.

Do not go overboard on your first foray into architecture. Even if you have a very large and experienced team, you will not be able to capture all of the information about your organization.

It is also important to remember that comprehensive architecture can obfuscate the important things. For example, there's no point in capturing 5,000 business processes if only 50 of them are critical to your business. Identify your critical business questions, and use those as the focus of your first architecture project.
Figure 2. Architecture provides a route for answering questions

The first thing I do with a client is discuss the questions that are critical to their business; and then I help them identify the ones that are hard for them to answer. The following questions are ones that many clients need answered:

- What is the impact of retiring an application?
- What is the impact of moving a location?
- What applications are needed to support a business process?
- What is the impact of replacing servers?
- What processes need to be developed to support a new strategy?
- Where are the gaps or redundancies in our application portfolio?

Your questions drive the content of your architecture. If most questions concern your application portfolio, then focus on defining the application area. If you need to understand how your processes support a new strategy, then focus on the business area. Then you can begin to expand the scope of your architecture with new business questions.

Step 2. Identify your business questions

Step 3. Identify assumptions and business rules

Now that you have identified the audience, purpose, and questions, you should identify the business rules that constrain or explain the area of interest.

Every business has rules. For example, if you are capturing information about critical business processes, you must also capture any regulations or corporate standards for the process. An
example of a regulation is the Health Insurance Portability and Accountability Act (HIPAA), which protects health insurance coverage for people who change jobs. A corporate regulation would then be created to show that the company is meeting the requirements of HIPAA.

You should capture assumptions about your architecture, such as "New application information will be uploaded on Friday" or "Every business unit is responsible for documenting business processes."

**Step 4. Identify your framework**

The following industry standard frameworks can help you create an enterprise architecture:

- ToGAF
- Zachman
- EA3
- DoDAF

Using a standard framework gives your architecture a "skeleton" that you can then build out with your models.

A framework also provides guidance on what information you need to capture based on the stakeholders who will use the architecture. It provides guidance on organizing information but does not suggest a specific implementation for your architecture.

There is plenty of information on the Internet about each of these frameworks. The framework you choose depends on the goal of your architecture, the experience of your team, and whether you want to follow a defined process like ToGAF, or just need help identifying which model to use for what purpose as in Zachman.

You can also combine frameworks. ToGAF and Zachman are often used together.
Figure 3. Your choice should be based on your purpose; do not make a random selection

How does a framework fit in to your architecture?

A framework provides guidance on what to model. Methodologies are then used to create models.

A methodology is a rule set that explains how to model something. For example, the Business Process Modeling Notation (BPMN) methodology gives precise rules and symbols to model a business process.

Figure 4. A framework helps with methodology selection

<table>
<thead>
<tr>
<th>Why</th>
<th>How</th>
<th>What</th>
<th>Who</th>
<th>Where</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual</td>
<td>Goal list</td>
<td>Process list</td>
<td>Material list</td>
<td>Organizational unit and role list</td>
<td>Geographical location list</td>
</tr>
<tr>
<td>Conceptual</td>
<td>Relationship</td>
<td>Entity relationship model</td>
<td>Project model</td>
<td>Organizational unit and role relationship model</td>
<td>Locations model</td>
</tr>
<tr>
<td>Logical</td>
<td>Process model</td>
<td>Using a framework you can identify that if you want to model how your business works from a conceptual perspective, you use a process model. You then select a process modelling methodology, in this example, BPMN.</td>
<td>Location diagram</td>
<td>Event diagram</td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td>Location specification</td>
<td>Event specification</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detailed</td>
<td>Details</td>
<td>Details</td>
<td>Details</td>
<td>Location details</td>
<td>Event details</td>
</tr>
</tbody>
</table>

A framework helps organize the key areas of the architecture and identifies the views you need to model, such as the perspective and the data needed to answer business questions.
The hotel chain decided to use the Zachman framework.

Whenever possible, use an industry standard methodology rather than something "home grown". Industry standard methodologies have rule sets and standard ways of modeling. Most homegrown methodologies fail to capture information in a useful way because the rule set is not clearly defined, which allows people to model the same information in multiple ways. This also affects analysis because the information is not captured according to a standard.

Multiple models are produced to support the framework based on the type of information you need.

**Figure 5. Framework with supported methodologies**

**Step 5. Create a metamodel**

A metamodel is an abstract view of your architecture. It shows the data you are trying to capture, and the relationships among the data. This is where you realize alignment, which is based on answers to your business questions.

For example, if you need to know the application that supports a certain business process, there must be a relationship between those two things in your metamodel. Otherwise, there is no connection between the data, you cannot answer your business question, and the architecture is not functional.

Note that you do not want a direct relationship between everything in your metamodel, and you should only link things together that have logical relationships. For example, linking an organizational department to a technology does not make sense, but linking a technology to an application does. A good modeling tool such as Rational System Architect supports traversing the metamodel to create complex reports. So, in this metamodel example you can report on the hardware that supports a business function even though there is not a direct relationship in the metamodel. In a metamodel you can potentially traverse from a business function, to a business process owned by that function, to a location of the business process, to a supporting application the process needs, and finally to the technologies that support that application.
Your metamodel should include the following features:

- Relationships between the architecture elements. For example, a business process to an application.
- Definitions of the elements. For example, the meaning of the term "application" and what properties you will capture.
- Traceability to business questions. For example, if your question is "What applications support what business processes?" You know you need a business process and an application in your metamodel, with a direct or indirect relationship between them.

**Step 6. Identify the models needed in the architecture**

Now that you have identified your business questions, your framework, and the metamodel you need to answer your questions, you need to figure out what models to draw.

Using a business process as an example, there are many industry standards that support modeling business processes, such as BPMN and flow charts. Choose your modeling methodology based on the following criteria:

- The audience for the information. Managers understand simple diagrams like BPMN; software developers normally prefer UML sequence diagrams or use cases.
- The elements of the metamodel. If in your metamodel you need to understand data as it relates to business processes, consider using BPMN to model that. If instead you are just worried about the sequence of process steps consider creating a flow chart.

After knowing the audience and the content you want to model you can then identify the diagrams you need to create. In the above example, since you needed information about business processes and system interfaces, you could select the following models:

- BPMN (captures business processes)
- System architecture (captures applications)
It is important to remember that you cannot use a single diagram to model everything in your EA. Further, separation of the architectural views, such as the application view from business view, is a best practice. If you try to model two views in the same diagram it often creates confusion and does not capture information in a meaningful way.

**Figure 7. Use models to answer business questions so the architecture is useful**
As Will Gadd said, "There's just something about getting out and doing nothing useful that I find wonderfully engaging."

**Use the right tools**

A single modeling tool or methodology does not provide a full solution. Besides a tool for developing models, you should also have tools for publishing, requirements management, and displaying on a dashboard. A dashboard presents your enterprise architecture information in easy to understand graphs like pie charts and bar charts.

If your architecture tool is customizable, question customizations that change the way the tool is meant to be used. Extensive customization is usually a sign that the wrong tool or approach is being used. Also remember that customization creates administrative overhead on the architecture.

Some customers customize architecture tools to create their own model. This is not a best practice approach, especially if the "model" is a single diagram taking up an entire wall that contains all the information about your architecture. Instead of creating wallpaper, create reports. Not everything needs to be shown on a diagram.

**Figure 8. ...instead of modeling for the sake of modeling**
People are just as important as tools when creating an architecture. A single person cannot be an expert in every aspect of architecture. Develop a well-rounded team to support the architecture. For a list of ideal characteristics of an architecture team, see the first article in this series Get the maximum value from your enterprise architecture consultant.

Step 7. Integrate the architecture

Link the data that you captured together based on the relationships you identified earlier. No, a tool does not magically do this, regardless of what sales people may tell you. And yes, relationship linkages are really hard to do without a repository. If someone suggests that the project can do this using a spreadsheet, you are wise to find another project to work on.

If you have existing architectures for projects or lines of business, and you want to create an enterprise architecture, the easiest approach is to populate your EA from the bottom up. Take existing architectures and pull common elements into a repository. Moving forward, try to standardize the models and terminology that is used across the organization, because everyone uses the same name for an organization, such as standardizing on "Finance" rather than having variations such as "Finance dept.", Accounting, or Accounting & Finance.

If this is your first enterprise architecture, use a common blueprint across your organization so that an architecture that was created for a line of business uses the same framework, terminology, and models as the enterprise architecture. That way you can report across the entire business.

Analyze the architecture

Figure 9. Save some energy for analysis!

If you do not allow time to analyze the architecture, there won't be time to analyze it. If you don't analyze it, why did you build it? This key step is often overlooked in scheduling. Allow for at least
50% of the time allotted to developing a model to be for analysis; this includes reviewing the model to verify and validate it.

Do quantitative as well as qualitative analysis. Math is important, especially for showing ROI. Quantitative analysis can be used to show bottlenecks in a process, time savings, cost savings, and elimination of redundancies if you use an industry standard method, such as BPMN. BPMN is "structured" because it has a rule set you cannot violate. The rules ensure you can do analysis such as simulating a change to a business process to see if the change saves time or money, or has a negative impact such as causing a bottle neck.

Qualitative analysis is done by looking at a model to see where potential problems are. For example, if you have a business process that has feedback to an early part of the process it is usually a sign that rework has to be done. Eliminating feedback loops in a process is one way to improve it.

After your analysis is complete, share the results. People will see value in architecture if they learn how to use it. Reporting is the key so when selecting an enterprise architecture tool, make sure it has a powerful reporting capability.

**Don't forget – you need a game plan**

**Figure 10. Develop your EA game plan**

People often forget there are many administrative issues that need to be addressed to start and support an EA project. Some of the administrative issues that need to be addressed include:

- How to deploy the enterprise architecture?
- Where to deploy it (web site, etc.)?
- Who's on the team?
  - Review boards
• Project management
• Administration
  • Who will use the information?
  • Who will have access to the information?
  • What standards will be followed?
    • Naming conventions
    • Color coding

There's an "EA" in team

Figure 11. A good EA team ensures success

You cannot create architecture in a vacuum. You have to be prepared to work with people outside the EA team, otherwise your architecture cannot be adopted and used. Also make sure stakeholders (e.g. people who are paying for you to build an architecture or people who are helping you build the architecture) are involved in your decision making.

Governance is required for decision making. Governance helps define the rules and strategies you will use for architecture. An example is governance on the naming of lines of business in your organization so one person doesn’t call the department "Accounting" and another "Finance". Governance also determines what models are ready to be released as "approved". Typical boards that are needed to have a successful EA include:

• Architecture review board
• Configuration and control board
• Administration guidelines (For example, who can create models, what is the approval process, what is the process for a change request)

The more people who are involved in supporting the architecture, the better the chances it will be used
A few final pointers

If it seems hard, that's because it is hard. But, it could also indicate that you are doing something wrong with either one of the following aspects:

- Models. For example, using a BPMN model to capture application interfaces,
- Stakeholders. For example, people who are unfamiliar with a business process that provide input and feedback instead of the people who actually do the process

Learn to separate political reactions from architecture

- "Don't document that! Then people will know we're doing things wrong!" – This is a common reaction to enterprise architecture. But there is no point in documenting an idealized view of the world because you won't be able to plan how to fix things in the future.
- "My organization is using a different method for capturing our architecture." – There is always one line of business in every organization I have worked with that does not want to develop their architecture in the standard way. No one is special. No one has a legitimate reason for not doing things in a standard way. The best approach for dealing with this situation is training for the knuckle heads that want to be different. If they feel comfortable and understand what is expected of them they should be willing to follow standards. If that does not work, your only option is to break their legs.

Use architecture to break down the stove pipes

- "We can't talk to those guys. They're, you know, software guys." – At most companies, there are people who find it challenging to talk to the software developers. They are nice people once you get to know them. And their attitude towards you will change once they discover they have an easy way to communicate with you that involves pictures instead of meaningless 500 page requirement documents.
- "I would like to give you that data, but I need to check with my management first. How about I get back to you next never?" – Some people consider the withholding of information job security. You cannot fire Barney if he is the only person who understands how your applications are wired together. Other lines of business may not want to share information out of fear that you will use it to fire people. Handle these situations by explaining the information will be used and how it will benefit the person who provides the information. If you are using the information for nefarious purposes, such as firing people, expect to see a sharp decline in the number of people willing to give you information.

Special thanks to Joe Josephson, First Ascent Press, for providing pictures and Will Gadd, www.gravsports.com, for providing pictures and quotes.
Related topics

- Improve your skills. Check the Rational training and certification catalog, which includes many types of courses on a wide range of topics. You can take some of them anywhere, any time, and many of the "Getting Started" ones are free.
- Article: Get the maximum value from your enterprise architecture consultant
- Download a free, fully enabled trial version of Rational System Architect.
- Evaluate IBM software in the way that suits you best: Download it for a trial, try it online.

© Copyright IBM Corporation 2011
Trademark
(www.ibm.com/developerworks/ibm/trademarks/)