Agile Business Architecture

Jeffrey Dreher
Aileen Johnson
Jeremy Moede
Troy Nelson
David Quimby

Table of Content

Executive Summary

Many of the traditional business architecture models do not take into the consideration of how fast the landscape can change for some businesses. In order to stay relevant in these industries, business architecture needs to adopt a new model that allows it to adapt faster in these markets. The Agile Business Architecture models takes several Agile principles into consideration when trying to respond in these rapidly changing industries.

Rather than having long lead times to deliver capabilities to customers, the model uses small experiments that last a couple of week to a few months to get fast learning back to determine if a strategy is still worth pursuing. The model uses capability assessments to determine what capabilities are currently available in the organization, identifies what is needed, and then works quickly to pull the right people, process, and technology together. It uses Agile thinking and small experiments to promote quick learning in the simplest way possible. Learning from these experiments allows an organization to quickly adapt through decentralized decision making, and determine if a strategy is accomplished, needs to be changed, or is worth abandoning.

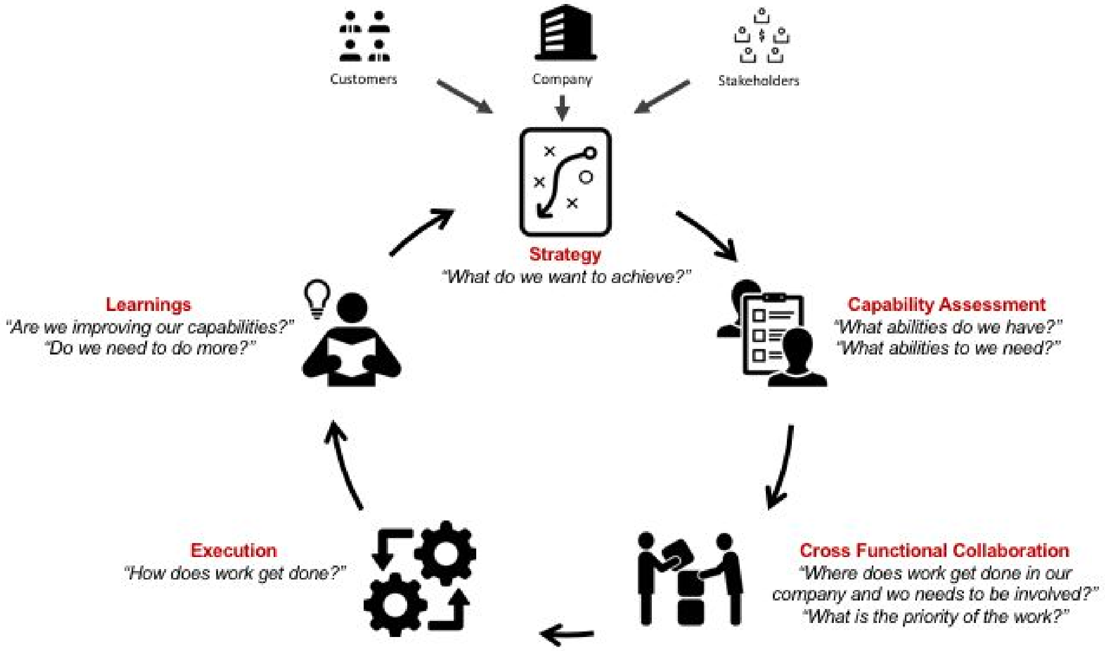

Customers
Company
Stakeholders
Strategy
"What do we want to achieve?"
Capability Assessment
"What abilities do we have?"
"What abilities to we need?"
Cross Functional Collaboration
"Where does work get done in our company and wo needs to be involved?"
"What is the priority of the work?"
Execution
"How does work get done?"
Learnings
"Are we improving our capabilities?"
"Do we need to do more?"

Figure 1: Agile Business Architecture Model

History of Business Architecture Frameworks and Tools

The history of business architecture has its origins in several disciplines: Enterprise Architecture (EA) and many techniques from the process improvement community. The concept of business architecture has alternately been proposed as a “blueprint of the enterprise,” as business strategy, and also as the representation of the overall business design. The concept of business architecture has evolved over the years.

It was introduced in the 1980s as one of the EA architectural domains in (e.g., Business, Application, Information/Data and Technology) and as activity of business design. In the 2000s, the study and concept development of business architecture accelerated. By the end of the 2000s, the first handbooks on business architecture were published, separate frameworks for business architecture were being developed, separate views and models for business architecture were further under construction, the business architect as a profession evolved, and an increasing number of business added business architecture to their agenda.

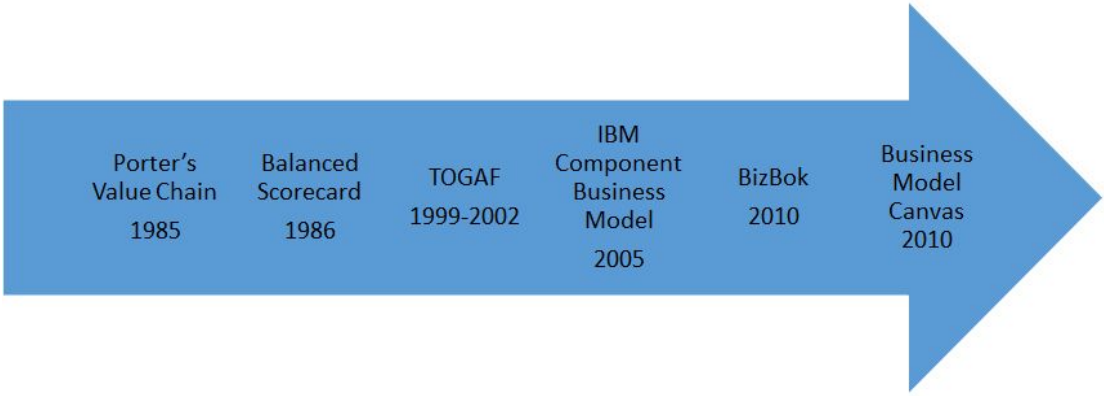

Porter’s Value Chain 1985
Balanced Scorecard 1986
TOGAF 1999-2002
IBM Component Business Model 2005
BizBok 2010
Business Model Canvas 2010

Figure 2: Business Architecture Timeline

Each of the business architecture frameworks and tools have their merits and all have helped business architects and other practitioners in the difficult work of building a bridge from strategy to execution. Business is constantly evolving and sometimes reacting to or driving disruptive change. In the current economy, with the increasing adoption of Agile software development methodologies, and digitally-driven, customer-obsessed business models, it is a good time to look at where business architecture has been and how it must change in order to remain relevant.

Amongst the first frameworks, Porter's Value Chain came to prominence at a time when being big and having scale was in itself a key aspect to competitive advantage and profitability.

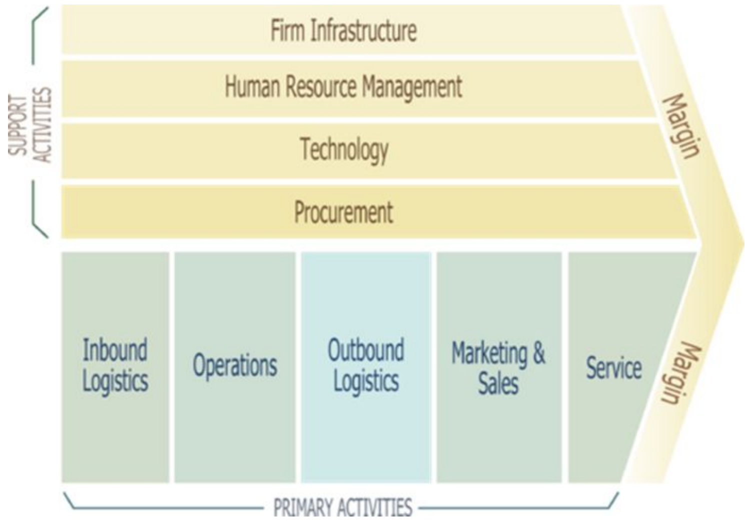


Figure 3: Porter's Value Chain

The framework is very well suited for any traditional mass-market, cost-driven approach to doing business. This model is optimized for efficient delivery of a known thing. Organizationally, it assumes that Z follows Y, which follows X. Porter's Value Chain Model takes an Inside/Out approach where customers remain at the end of the value chain. The model is not flexible nor is it responsive to social signals and so it seems incomplete in our age of social media as a demand driver.

Fast on the heels of Porter's Value Chain model came the Balanced Scorecard.

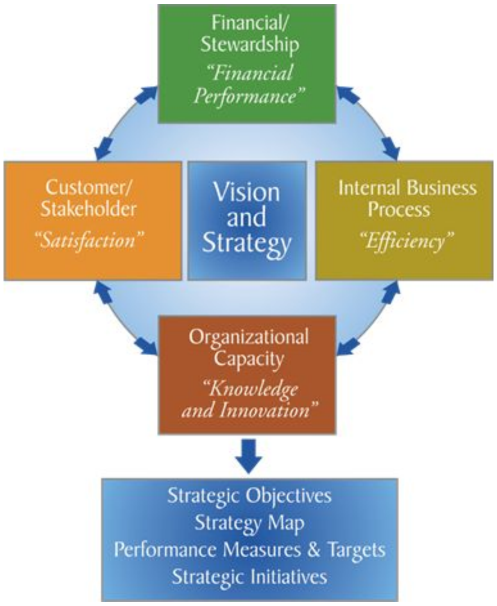

Financial/ Stewardship "Financial Performance"
Customer/ Stakeholder "Satisfaction"
Vision and Strategy
Internal Business Process "Efficiency"
Organizational Capacity "Knowledge and Innovation"
Strategic Objectives
Strategy Map
Performance Measures & Targets
Strategic Initiatives

Figure 4: Balanced Scorecard

While it lacks the robust feedback loop that many customer-obsessed businesses have established or are establishing today, it is well-regarded for promoting customer perspective as a key business consideration. It does not go so far as to ensure that it is the customer and not the company who is defining satisfaction which is something that many companies are pivoting to in the age of social media. Some people regard it as an incomplete model because performance measurement is subjective and can be quantified only through surveys or management opinion and because it doesn't provide recommendations.

The Open Group Architecture Framework (TOGAF) incorporated Business Architecture into its model in 2002. The primary concern about TOGAF as a structuring framework is that it is complex to learn and expensive to implement in its entirety were an organization to decide to do that. The complexity of TOGAF prompts some to label it as a framework that benefits architecture and its governance more that it does the business. It is most often implemented in organizations where business architecture remains a part of enterprise architecture and where enterprise architecture is keen to establish a baseline of technical assets. While there can be huge business benefit to be gained by knowing the state of the state of technical assets, many companies never get beyond the cataloging step to to a rich understanding business capabilities that can help unleash additional business potential.

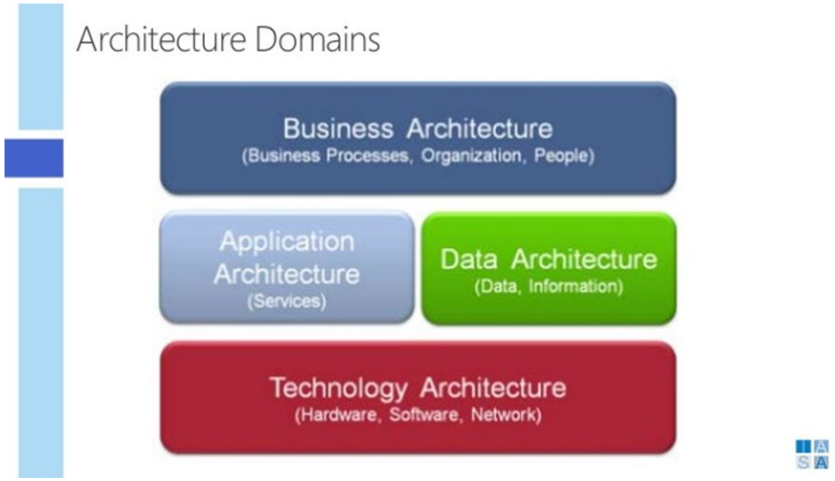
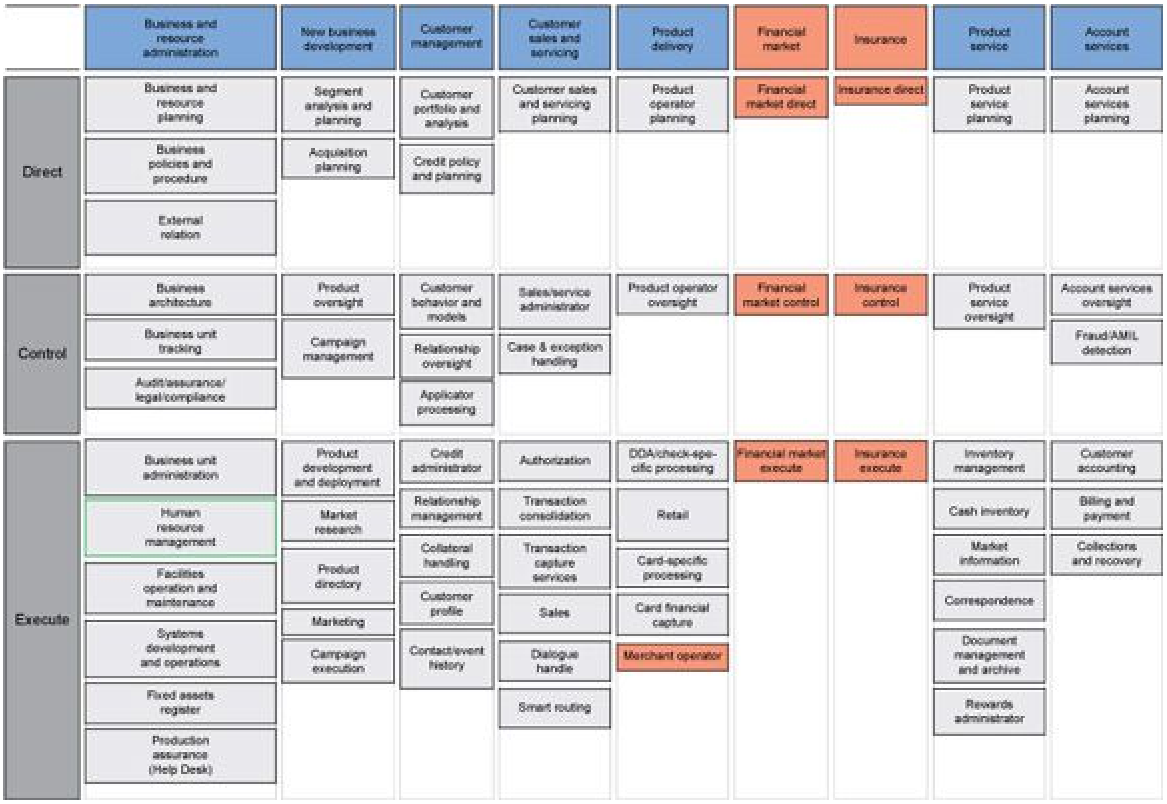


Figure 5: The Open Group Architecture Framework (TOGAF)

The IBM Business Component Model framework is particularly useful for organizations seeking to map the individual components of a stand-alone capability or product. It considers people, process and technology and, in doing so, gives a well-rounded picture of individual capabilities. In today's environment, where creation of end-to-end value streams and concepts of continuous improvement are central to business thinking, this model is constrained because it doesn't consider cause and effect linkages between the capabilities, desired outcomes and measurements.



	Business and resource administration	New business development	Customer management	Customer sales and servicing	Product delivery	Financial market	Insurance	Product service	Account services
Direct	Business and resource planning Business policies and procedure External relation	Segment analysis and planning Acquisition planning	Customer portfolio and analysis Credit policy and planning	Customer sales and servicing planning	Product operator planning	Financial market direct	Insurance direct	Product service planning	Account services planning
Control	Business architecture Business unit tracking Audit/assurance/legal/compliance	Product oversight Campaign management	Customer behavior and models Relationship oversight Applicator processing	Sales/service administrator Case & exception handling	Product operator oversight	Financial market control	Insurance control	Product service oversight	Account services oversight Fraud/AMIL detection
Execute	Business unit administration Human resource management Facilities operation and maintenance Systems development and operations Fixed assets register Production assurance (Help Desk)	Product development and deployment Market research Product directory Marketing Campaign execution	Credit administrator Relationship management Collateral handling Customer profile Contact/event history	Authorization Transaction consolidation Transaction capture services Sales Dialogue handle Smart routing	DDA/check-specific processing Retail Card-specific processing Card financial capture Merchant operator	Financial market execute	Insurance execute	Inventory management Cash inventory Market information Correspondence Document management and archive Rewards administrator	Customer accounting Billing and payment Collections and recovery

Figure 6: The IBM Component Business Model

The Business Model Canvas is a template used by many strategic management teams and lean start-ups to describe the attributes of capabilities that exist or that are proposed to be developed. It is easy to create and it's one page-format drives quick learning. The value of the Business Model Canvas for an Agile organization is in its iterative nature. It is easy to create and it can be constantly adapted to be in sync with additional learnings, demands and strategies. It is important to remember that the Business Model Canvas generally provides only a cursory view of any given capability and that, in many cases, additional analysis will be required to develop the first incremental piece of value for a new capability.

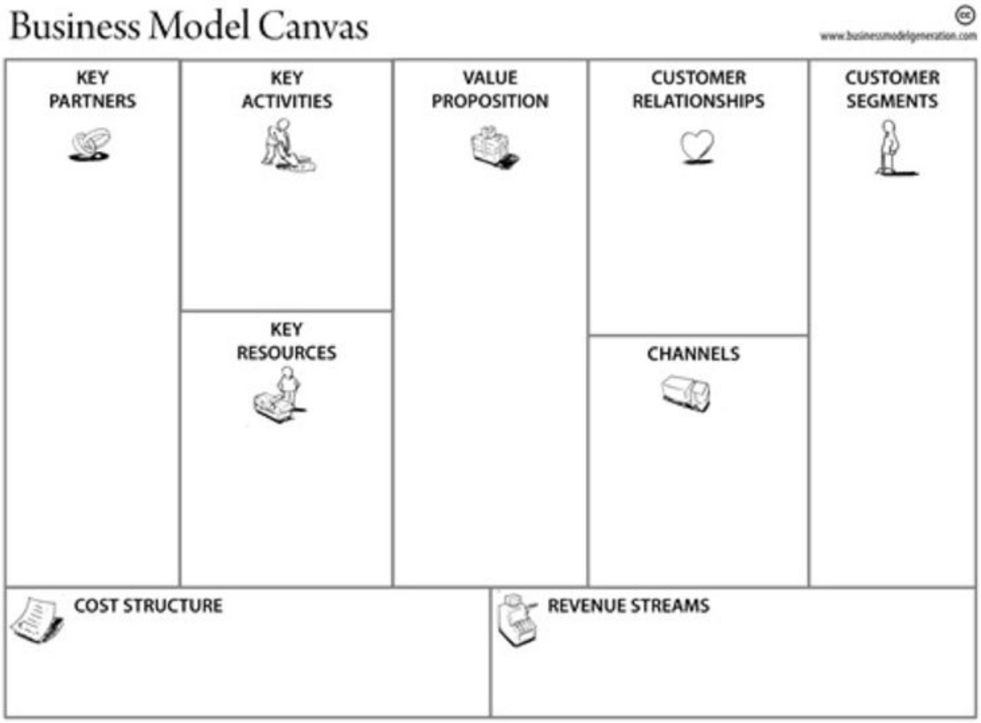


Figure 7: Business Model Canvas

The Business Architecture Body of Knowledge (BIZBOK) has been developed and is updated multiple times each year, stewarded by the Business Architecture Guild.

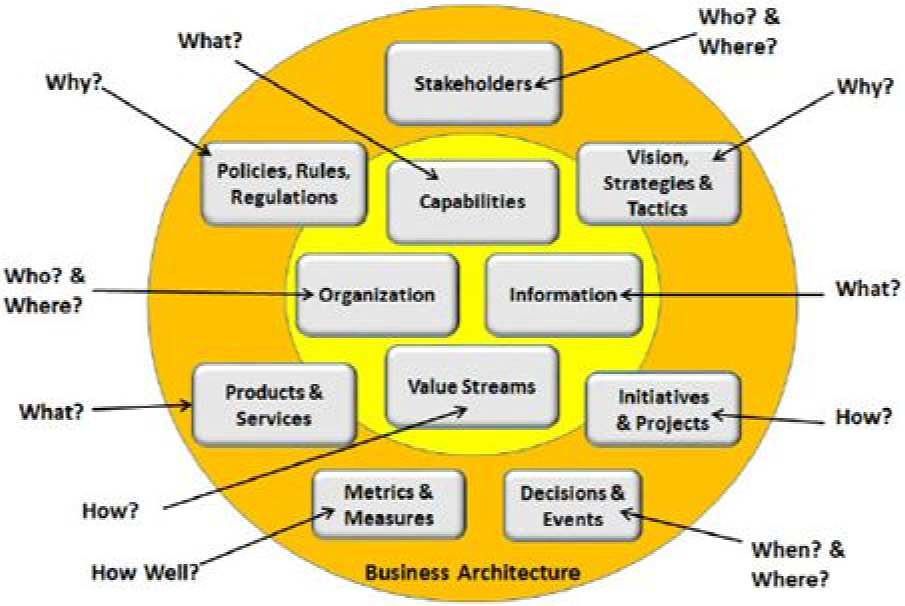


Figure 8: Business Architecture Body of Knowledge (BIZBOK)

The “DNA” of Enterprise Architecture is evident in the BIZBOK. It is a set of methods that relies heavily on creating a series of artifacts (see diagram). The authors of this whitepaper suggest that business architecture teams that are new, primarily start with the concepts in the inner circle: Capabilities, Value Streams (how capabilities are assembled to deliver value on behalf of a defined stakeholder), Information and Organization. In addition, the context of the organization’s overall vision and strategy are imperative to understand the WHY behind your business architecture efforts. In the next section, we will cover a series of Agile principles such as be quick to deliver value. In applying these principles, we’ll suggest that not all of the artifacts and points-of-view from the diagram be blow out. Rather, focus on the problems that can drive the most value, and focus there.

Business Architecture in an Agile World

Overview

When looking at some of the traditional planning methods for Business Architecture, they have some underlying thought patterns between them. At a high level, they first defining strategies, looking at the capabilities that are needed to deliver one or many of those strategies, figure out where are the gaps within the company to accomplish the strategies, and then implement a plan to get work going to fill those gaps. Once the plan is rolled out, the company would kick off multiple initiatives/programs/projects to accomplish that implementation plan.

Although this steps are all important, when taking a look at this from an Agile perspective, several principles of Agile are missing from these methods. There is a lack of focus on delivering value early and often, welcoming changing requirements, business and technology working together often, working deliverables as primary measure of progress, simplicity, and feedback at regular intervals. These are some of the reasons that business architecture need to look at changing the mindset behind these frameworks to be more Agile.

Delivering Value Early and Often

When defining work at a company level, these pieces of work are often large, take long periods of time, and are usually based on an annual budgeting process. At the beginning of the year, all the strategies are shared, then the company kicks off a project or several projects that end up taking several months, the full year, or multiple years to try to make those strategies are a reality.

The challenge with these long lead times is that our customers have to wait that period of time, several months or even years before they are actually able to use whatever new products or functionality was a part of those strategies. In faster paced industries, waiting that long may have caused the value of that strategy to have already passed and no longer be relevant, a competitor has already made it to market first, or realize after delivering the solution, it may not have actually even delivered the outcomes desired to accomplish that strategy and it is too late to adapt.

One of the Agile principles is around delivering value early and often to our customers. Business Architecture needs to focus on reducing the lead time that is needed to deliver these pieces of work to ensure the company is meeting our customer needs on time, get to market quickly, and get fast feedback from our customers to make decisions quickly on the next strategies that should be executed based on learnings from the previous cycles. To do this, companies need to start defining small experiments to start validating that the capabilities being built are going to accomplish the desired outcomes, and then incrementally deliver on top of those first experiments to meet the customer's expectations.

An example of a company doing this would be Apple when they came out with their first version of the iPhone. The first version of the product did not have every bell and whistle that that was expected, including the ability to copy and paste. This was not necessary to prove out the idea of people like the idea of a touch screen device for a phone. It wasn’t until the third release that Apple added copy and paste, probably based on the requests of the consumer, but beat the competition to the market with one of the first full featured touch screen phone, helping instill their position in the market they have today.

Welcome Changing Requirements

In traditionally define strategies,capabilities, and work, the mindset is that the requirements and outcomes must be fully understood prior to even starting the work. Once work is also under way, because of our execution practices, introducing change once work has begun requires extra process and paperwork to integrate those changes into the solution. This limits our ability to change, going against an Agile principle of responding to change. Responding to change is important because the reason change happens is that new information or learning has happened from the last time a decision was made, and not leveraging this learning will prevent us from making good decisions going forward.

Forcing the mindset of having a complete understanding of the solution up front causes substantial up front planning and analysis to try to define the requirement. Once the solution has been defined, the customer must then sign off that those are exactly what they want to be delivered. Trying to define exactly what the solution will be up front and how to build it is an almost impossible task. If the work is 100% predefined, this may be possible, but most of the time, the customer does not even understand what exactly they want until they actually use the solution. Only once a solution in the customer’s hands are they truly able to explain what they like or don’t like, but it’s too late, the solution is already fully built and being rolled out in traditional methods and change is impossible. A good indication that a company is unable to completely define their work up front is if there is a formal change request process defined.

Even if we are able to define requirements perfectly up front, force customers to put everything they want in these requirement gathering practices and documents causes them request functionality they may not even use. This process forces the customer to request everything they can dream of now, because they may not be able to ask for anything again for another year when the next cycle begins. This causes unnecessary functionality to be built that is not actually needed, wasting valuable resources and time that could have been used on other work that would be value add.

Welcoming changing requirements as work is in flight requires a slightly different mindset when it comes to defining work. Instead of defining solutions from the ground up in horizontal layers of functionality, we want to build the solution in vertical slices of functionality. In determining what that first slice of functionality is, we need to think of the smallest piece of the solution that will allow the customer to decide if the solution may work for them. There needs to be reassurance to the customer that this is not the complete solution and more functionality will be added based on their feedback, so the first slice does not include unnecessary and unneeded functionality. Once we build and give this small working piece of the solution to the customer, they can determine if it will work for them or not, and able to give real feedback to guide us in the right direction.

Responding to change does not mean that every change identified has to be implement, or should be implemented. Sometimes the change is so great that the cost of implementing that change may cause more harm than good. There is a constant balance between being flexible and changing as learning happens, and being in a constant swirl of change where nothing is being completed because everything is changing.

Going from a culture where change is not easy to a culture that embraces change for the customer's competitive advantage is not a simple switch. Compared to when we had our change request process where everyone was informed of a change in a process and there as a big approval process, the system now need to allow for change to happen at all levels, and a quick way. In order to allow for this change ,without slowing the whole system down, the decision to make a change need to be decentralized so that the decision can be made quickly by the people who have the best and latest information to make the decision. This change can then be rolled up to the right levels as to why the change was made while the change is being integrated, or may have already been complete.

Business and Technology Working Together Often

Customers (sometimes referred to as the business) were usually engaged at the beginning of an effort to help define what they think they needed, and then were brought in at the end to verify that what was built meet though needs defined earlier. In between these two contact points, the customer is not really engaged unless there is a problem or they figure out that they need to do a change request because they remembered something after the requirements were written.

If embracing change becomes part of the way of working, only engaging the customers at the begin and end will not work as decisions on the changes need to be made throughout the process. Ongoing collaboration between the people trying to build the solution and the customer of the solution is needed, so as learning happens, decisions can be quickly made to keep the work moving forward.

But how can the customer be expected to constantly be made available to the builders? What if there are multiple customer of the solution, how do decisions get made? Sometimes it's not always possible to have the end customer of a solution always available or need someone to make decisions for the best of all customers. For these scenarios, it may be necessary to pick a liaison for the customer to overcome these challenge. Picking the right person for this role is extremely important and requires a balance of domain knowledge, availability to the builders, and empowerment to make decisions on behalf of all customers. These liaisons can make decisions on the fly to keep the work moving forward, and can determine the best solution for all customers involved.

Working Deliverables as Primary Measure of Progress

Determine if progress is being made on the solution is a common question that everyone wants to determine to make sure that the end solution will be achieved at some point. The thinking of what progress means can be very different with an Agile mindset. If asked the question of, “Is it better to have a solution 80% started and nothing complete, or 20% working functionality?”, groups may answer differently depending on the mindset instilled in them.

Milestones based on steps in getting to the final solution, such as requirements sign-off, development complete, testing complete, User Acceptance Testing complete are traditionally how progress is measured. Although these milestones show progress being made against a process, this is not showing progress against delivering value. Getting a working piece of the solution in the hands of the customer to get feedback is a more Agile way of thinking of showing progress. Therefore, when setting milestones, don’t set it based on steps in the process, but rather what functionality is working as milestones to work towards, showing incremental progress of adding new functionality on top of what is already working.

Although working deliverables in the primary measure of progress, this does not mean that it's the only item that is measured. Only measure this will lead to sacrifices in other aspects of the delivery. Some examples of other ideas to look at are quality, outcomes, happiness, and other business specific measures.

Simplicity

In many aspects of the current way of working, there is a lot of process and tools that are used to complete the work. There are templates, meetings, and frameworks that companies have done things for years to accomplish pieces of work. Although there is still a lot of good that these processes and tools have helped, not all of them are still helping the way they were meant to.

When looking at every aspect of the current system to get work done, re-evaluate if the way it normally gets done still makes sense. Does the entire solution need to be planned out at the beginning? Do all the requirements need to gathered right now? Are all the documents for training, documentation, or status keeping really needed and adding value, or are they just distracting people for valuable work? Asking these question can promote a leaner way of working and eliminate non-value add work, or postpone work that can be done later, because learning will happen until then, therefore preventing rework.

A Model for Agile Business Architecture

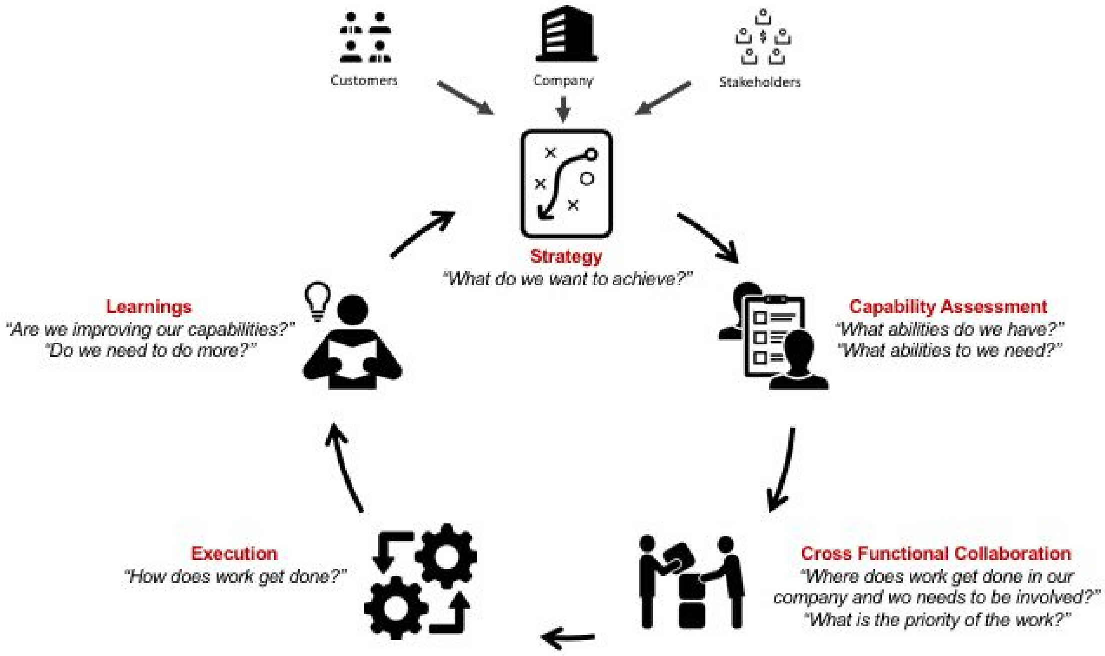

Customers
Company
Stakeholders
Strategy
"What do we want to achieve?"
Capability Assessment
"What abilities do we have?"
"What abilities to we need?"
Cross Functional Collaboration
"Where does work get done in our company and wo needs to be involved?"
"What is the priority of the work?"
Execution
"How does work get done?"
Learnings
"Are we improving our capabilities?"
"Do we need to do more?"

Figure 1: Agile Business Architecture Model

Strategy

Business architects often are not the ones to set strategy for an organization. More often this is done by senior leadership, sometimes in partnership with a corporate strategy function. Good strategies take into account inputs from customers, the marketplace, what competitors are doing and what other stakeholders (such as shareholders) want. However, business architects should be very familiar with an organization's strategy. The strategy sets the context, sets the WHY and PURPOSE, of the organization. Without these, it is hard to understand how to prioritize efforts – and how to focus on the activities that will drive the highest level of value.

Oftentimes, strategies will outline "where to play" (the geographies, the markets, the customer segments) as well as "how to win" (how will the organization create something that is uniquely of value that will attract customers, outfox competitors, etc.) Some of the common ways that business architects and other planning-related roles choice to codify strategies are: strategy maps, Objectives, Goals, Strategies and Measures (OGSM), and Objectives and Key Results (OKRs). Any of these are good methods to capture and codify an organization's strategies. Often, organizations do not have strategies that are written down. They often live in the minds of the leaders who run the organization. In these instances, it is still helpful to document what you perceive are the primary strategies of the organization and play them back to leadership to clarify your understanding. This dialog is beneficial in several ways. First, it helps the leadership team clarify and get on the same page with each other with regards to what they believe the fundamental strategies are. Second, once strategies are written down, they can be shared with the entire organization so that the whole team can get aligned with what needs to be done. Finally, it give focus to us as business architects of where to focus. It will help narrow down our focus on what capabilities to start with, as we adopt an agile mindset.

Capability Assessment

The next step in our approach is to focus on which capabilities to assess. As a preface, in an agile approach, you may wish to narrowly focus on a handful of capabilities (vs. the entire enterprise), and conduct the steps below in a rapid 2-3 week sprint. This way you can begin to collect a point of view, and test out the value proposition in a rapid fashion – and not end of spending months of effort that may yield little or no fruit. The primary steps are

- **Identify which business capabilities to assess**: create a matrix of your strategies to your capabilities in our capability model. Highlight those that are most essential to achieve the strategy – focus on these first.
- **Create a standard unit of measure for your assessment**: For the priority capabilities – understand if you are going to use a simple red/yellow/green or 1-5 scale to capture the current state of affairs. Understand and document the criteria for each level of assessment.
- **Identify who to include in the assessment**: stick to a short list of those leaders who understand the capabilities both operationally and strategically.
- **Perform the assessment**: Interview or survey the key stakeholders. Capture the key gaps or opportunities for each capability.
- **Use the results**: Create a hypothesis of what gaps or opportunities you want to focus on this sprint. Do not try to boil the ocean and stay focused.

Coming out of this assessment, gaps or opportunities should be identified. The change here from traditional methods would be to not go after the full gap or opportunity right away, but instead, go after a small experiment to provide learning that close that gap or opportunity will have the desired outcome. Take a small slice of that gap or opportunity that could be quickly build out and then, once confirmed that the increment of the capability is helping to accomplish the overall strategy, continue with the next increment of the capability to add additional functionality until the strategy has been accomplished, or is no longer relevant.

Cross-Functional Collaboration

Successful cross-functional collaboration begins with the business architecture assessment of capability impacts. Identifying the people, process and technologies that support the capabilities in question allows the business architect to identify the key stakeholders needed to complete the capability increment. Similar assessments can be made using the value stream maps and any existing organizational maps. The key is identifying the stakeholders needed to implement the changes to capabilities and any stakeholders impacted by the change. This helps ensure the right people are at the table from the start of the initiative.

Using a capability increment approach to deliver value allows the business architect to be more precise in identifying stakeholders. Capability increments impacting lower-level capabilities means that stakeholders who “own” the higher level capabilities may only need to be kept informed of the initiative vs. actively engaged. This can help manage the costs of the initiative from an internal billing perspective and keeps the right people focused on the right level of detail.

With the stakeholders identified across all aspects of the capability change (those linked to the process, technology or information) the team can be formed and work assignments and targets communicated. Cross-functional collaboration frameworks can support team interaction and information sharing. These tools and frameworks will vary by company, but should ideally keep the focus on measuring progress and keeping all members informed of changes and their impacts on deliverables.

Execution

The execution stage focuses on how the work, as defined in the earlier stages, gets done. Organizations have multiple options in their choice of delivery methodologies. Some organizations my deliver using projects, products, values streams, or a combination of methodologies. This paper will not espouse any single methodology, but will leave it to the organizations to select the best approach. The only recommendation made here is to ensure the methodology supports the concepts of feedback loops both upstream and downstream to support the overall Agile Business Architecture model.

Learnings

As an output of the execution stage, capability increments (or components of the increment), in the form of working deliverables, are deployed. While the delivery teams will focus their learning assessments on how they functioned as a delivery team or on the use of tools and methods, the business architect will focus on the the value delivered to the stakeholders. Did the expected value get delivered? Are there signs that any leading or lagging indicators that measure the capability change are moving in the expected direction? Updated Value Stream maps should reflect the changes (or remaining gaps). What about the overall capability? Did the increment drive any changes to the performance measures used to assess overall capability health? Why or why not? Are there other changes to the organization and information maps as a result of the newly completed capability increment?

While individual capability increment deliveries may miss the mark, if multiple releases fall short of stakeholder expectations, the business architect may need to engage the strategy team to assess changes.

Business architects will need to use these updated blueprints and maps to communicate back any adjustments to (or confirmation of the success of) the source strategy. After review with the owners and stakeholders of the strategy, executive leadership will either need to confirm a continuation of the strategy as planned, make adjustments to the strategy, or abandon the strategy. In their role of translators of strategy to the execution side, the business architect will use their architecture blueprints and maps to communicate the changes back to the enterprise.

The advantage of using capability increments as the method of delivering value, vs. a “big bang” approach to deliver larger changes, is that this cycle can be repeated quickly and iteratively in relatively short periods of time (monthly, quarterly or annually depending on the organization’s needs). This allows for faster course corrections and less investment dollars at risk of not delivering the required returns.

Welcome Changing Requirements

The Agile principle of welcoming changing requirements is embedded every stage of the model. Learning is possible at every stage, not only at the last one, and therefore change needs to be welcomed at every point in the system as it makes sense. Although the model may show a loop of with learning is happening at the end, learning is happening all the time throughout, with an emphasis of looking at the outcomes of our capability increments, as that is when customer focused behaviors can be observed.

Strategies should not be static goals that are set once a year and not revisited anymore. As learning is occurring throughout the model, the current strategies should be under constant observation, and an organization should not be afraid to adjust those strategies. Adjusting these strategies should not be treated as a failure for define them right the first time, but a success on being able to adjust to better direct our focus.

The Capability Assessment phase is tailored to communicate changes to requirements. By managing and publishing the underlying business architecture blueprints, the business architect is in position to communicate how the changes in strategy impact the organization. Blueprints like the Balanced Scorecard can be revised to show the new measures in place for the revised strategy. As requirements change, these blueprints can be quickly updated and published.

Capability maps can be re-heat mapped to reflect changes in priority and to highlight where a given capability may now require additional focus, shifting resources to that effort.

Cross-functional collaboration is one of the key success factors in supporting changes to requirements. As requirements change, either at the strategy level or at the deliverables level, teams must be in a position to integrate the changes into their work and to understand how the changes cascade to other teams. It is no longer sufficient for a team to ignore changes as “out of scope” or to not move into the execution phases of a project without formal sign-off on all requirements. The fact that requirements will change must be baked into the planning and execution phases and cross-functional connections must be in place to support and adjust to those changes. Critical to this level of thinking is looking at how changes in one area, can affect another area to think of the overall goals that the group is trying to hit.

Execution is where the rubber hits the road, and possibly the most amount of change many need to take place. Hopefully all these changes are relatively small and don’t cause big changes that would need to be raised to higher levels of coordination and communication, but if necessary, that might be the case. Because of the number of changes at this level, it’s important to empower the people at this level to make quick decisions to keep progress moving, and trust they will bring forth appropriate communication if necessary to ensure alignment between teams.

In the Learnings stage, time should be set aside to assess how requirement changes, especially those occurring during the Execution stage, impacted the stakeholder value delivered by the increment. Of particular interest to the business architect will be the learnings with can be used from this capability increment to further inform future the capability maps, value streams and related blueprints. Learnings from this deliver can also influence future prioritization discussions and determine if another capability increment should be added, if the strategy has been accomplished, or if the strategy needs to be abandoned. This learning is essential to the system and can not be skipped. If learning is not gathered and acted on, it’s easy to keep building the solution, but not to be adding true value.

Conclusion

This Agile Business Architecture Model, unlike most Business Architecture models, takes into consideration the fast moving landscape of some businesses. The model uses small experiments that last a couple of week to a few months to get fast learning back to determine if a strategy is still worth pursuing. The model uses capability assessments to determine what capabilities are currently available in the organization, identifies what is needed, and then works quickly to pull the right people, process, and technology together. It uses Agile thinking and small experiments to promote quick learning in the simplest way possible.