

Foundations of the Architectural Thinking Framework®



Dr. Wolfgang Goebel, V1.0 August 9th, 2018
wolfgang@architectural-thinking.com

architectural-thinking.com

Definition

The Architectural Thinking Framework® is a truly open, lightweight architectural framework that is based on the experience of many practitioners. It has the *goal* to ensure that all the solutions of a company fit together in order to balance the dimensions

- customer value
- finance
- sustainability

in a way that maximises the overall value from an enterprise-wide viewpoint in the short and the long term.

Architectural Thinking is not a process or discipline. The Architectural Thinking Framework® is a content oriented framework that defines artefacts and their relations that need to be created by various processes (such as governance or solution development). The Architectural Thinking Framework® enables consistent, connected, company-wide structures that ensure traceability from business vision to technology implementation. It is lean enough for Agile but works as well with classical project management methods.

Values

All deliverables to be produced under the brand of the Architectural Thinking Association must comply with the following values:

Lean:

Each and every architectural model, map, principle and integration artefact and its significance must be self explanatory and instantly understood by relevant stakeholders.

The Architectural Thinking Guide only includes models, maps, principles and integration points that have proven to be valuable and work in any mid to large-sized company. It defines an invariant minimal core that companies can easily extend.

Collaborative:

In Architectural Thinking, 80 percent of architectural work is carried out by many people, for example, by autonomous, cross-functional teams. Thus, everybody becomes an architect on a micro level contributing to the overall big picture of the company. Dedicated architect roles are used in order to ensure conceptual integrity on capability and enterprise-wide levels only.

Collaboration is fostered by easy to understand, lean models and maps, and by using Web 2.0 features (such as wikis) as central architecture repositories, where everybody can contribute and comment.

Business-Orientated:

Business Architecture drives Technology Architecture, not vice-versa. Business people are encouraged to start thinking in architectural structures that are connected to each other. Thus, business is treated as part of the architected system, not only as its user. Four out of five artefacts of the architectural model (Value Streams, Capabilities, Business Objects, Application) are purely business related. We value the work of the Business Architecture Guild¹ and their BIZBOK

¹ businessarchitectureguild.org

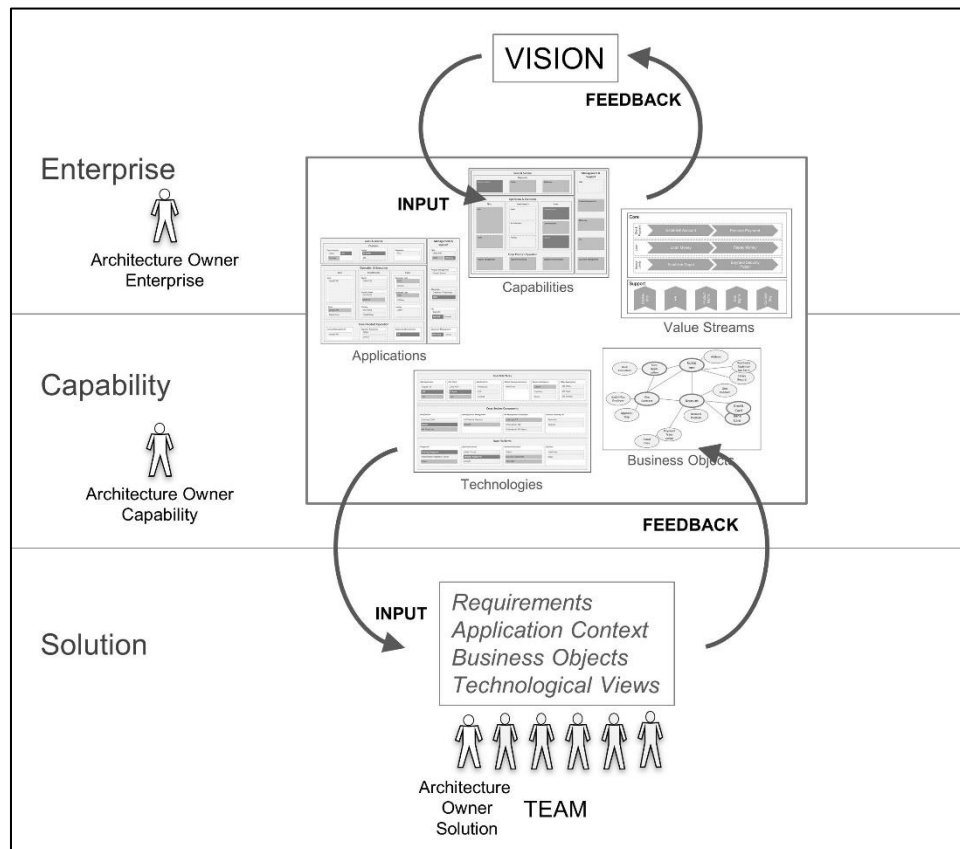
Architectural Scopes

The Architectural Thinking Framework® defines three scopes of architecture: (i) enterprise, (ii) top-level capability and (iii) team.

Approximately 80% of architectural work is carried out during implementation with a *solution scope*. A team creates architectural maps that are specific for the solution collaboratively, but connected well with the maps on the higher (capability/enterprise) levels. An 'Architecture Owner – Solution' is responsible for the conceptual integrity of the solution, which means that the micro-architectures of each team member fit together.

In order to ensure that solutions that support the same top-level business capability fit together, single minded architectural work with a *capability scope* is required. An 'Architectural Owner/Capability' creates the capability-wide architectural maps by aggregating and consolidating the work carried out at team level.

The remaining maybe 1% of architectural work is carried out at *enterprise level*. An 'Architectural Owner/Enterprise' creates the enterprise-wide architectural maps by aggregating and consolidating the work carried out at capability and solution level.



The architectural maps at solution level are connected to the enterprise-wide maps via unique references (e.g. solution requirements must refer to exactly one business capability). This allows permanent feedback cycles between solution architecture and capability/enterprise architecture.

The business vision is an important input for the design of the enterprise-wide architectural maps. On the other hand, architectural work at enterprise-level typically leads to questions that must be addressed in the vision statement and is the perfect feedback mechanism to challenge the vision.

The enterprise-wide architectural maps are an important input for the scoping of solutions. These permanent feedback-cycles ensure that findings at solution levels that are relevant at enterprise-level are taken to the next level. Thus, the architecture owner at capability level uses the findings of the solution level as an input for the target architecture of his/her capability.

Architectural work is permanently carried out at the detailed solution level and at the capability and enterprise level side by side. Many solution teams create architecture within their scope simultaneously. Whenever they find issues relevant at the capability level, they take them to this higher level, where the architectural owner has to integrate them into the target architectural maps for his capability.

State of Practice

Nowadays, business vision statements are created by top executives almost only top-down. A feedback loop that challenges the vision statement against the enterprise wide architecture is uncommon.

Enterprise-wide architecture models are designed by a few enterprise architects and typically not widely accepted by solution architects. Strong links between solution architecture and enterprise architecture seldom exist.

Parts of the Architectural Thinking Framework®

The Architectural Thinking Framework® consists of four parts:

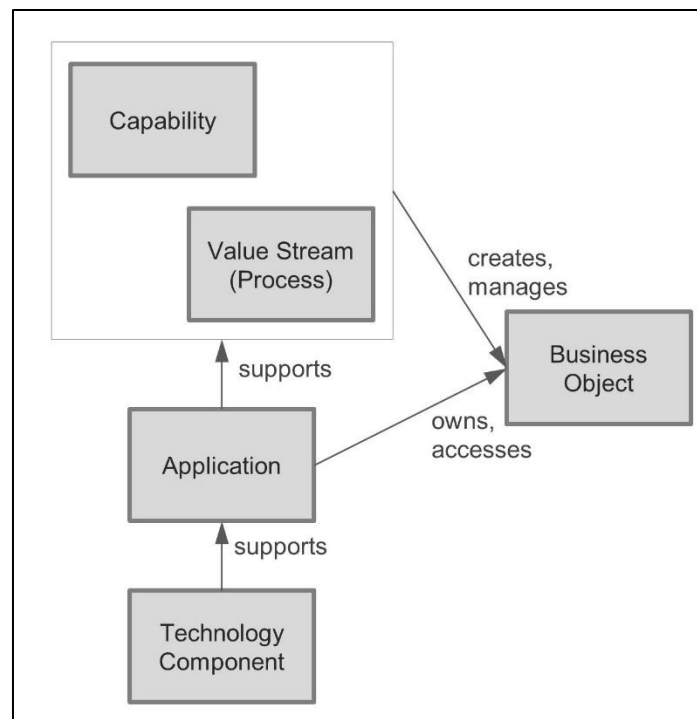
- Lightweight [Architectural Model](#)
- Small set of [Architectural Maps](#)
- [Architecture Principles](#)
- *Integration Points*, i.e. *interfaces* to relevant processes of the enterprise

For each part, the Architectural Thinking Framework® provides [how-tos](#) that have been proven in practice and gives detailed instructions on how to model the corresponding artefacts.

Lightweight Architectural Model

The core architectural model of Architectural Thinking consists of just five artefacts:

- *Business Capabilities* describe what a business needs to do in order to generate customer value.
- *Value Streams* define how the processes of the company create this customer value.
- *Business Objects* define which information is needed in the capabilities and value streams.
- *Applications* are computer programs that support value streams and business capabilities and store business objects in the form of data.
- *Technology Components* support applications.



See the appendix for more details.

If linked consistently and used widely throughout the entire company, these five artefacts are sufficient in order to achieve the goals of Architectural Thinking as mentioned above. They are the core skeleton model to which more specialised architecture models (such as business models, product models, customer journey models, process models, IT software and system architecture models) can be linked. The architectural model of Architectural Thinking is not in contradiction to the models used in existing tools and frameworks; it simply heavily downsizes what is in use today and focuses on the essence.

The Architectural Thinking Framework® comes with detailed, practicable cookbooks that explain how to build your architectural model.

State of Practice

The two most common architectural meta-models are provided by The Open Group:

- ArchiMate®
- TOGAF® Content Metamodel

Both meta-models are voluminous and consist of 30+ artefacts and 100+ potential relations between artefacts. They provide a maximum that attempts to cover each and every potential usage scenario. Companies typically need to heavily downsize them before implementation, which means that they end up with a proprietary rather than a standard model.

Vendors of enterprise architecture tools typically define their own meta-model, more or less compliant to the standards defined by The Open Group®, which can be customized to a company-proprietary model.

So yes, there are standards, but they are so universal and require intense customisation, so that it questions their role as a standard.

What is different in Architectural Thinking?

The Architectural Thinking Framework® consists of a minimum, core set of artefacts that are beyond dispute and exist in any company. This is important, because in Architectural Thinking many business roles implement architecture. Thus, the model must be simple and self-explanatory otherwise it will not be accepted by a wide range of stakeholders.

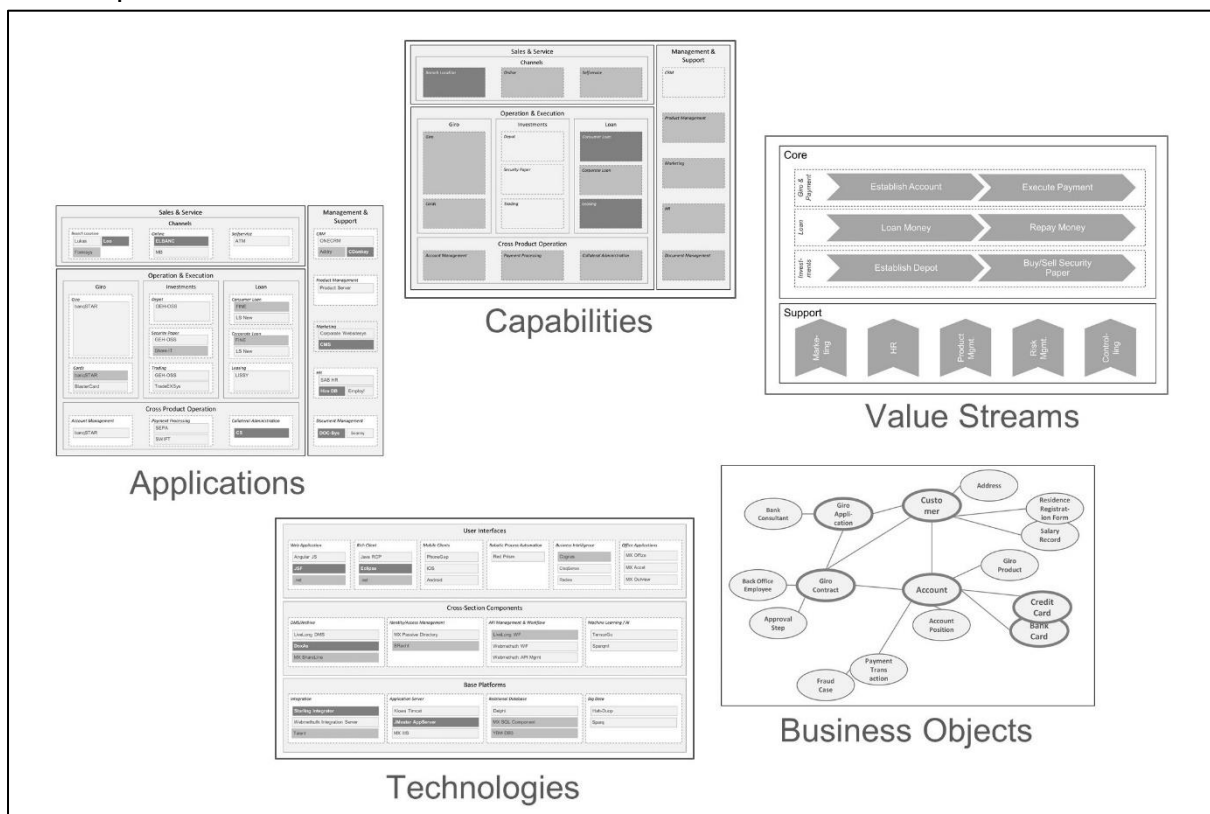
Companies are encouraged to extend the core model according to their usage scenarios. Compared to the standards mentioned above, the approach is vice-versa: while existing standards provide a maximum-fits-all approach that must be downsized, Architectural Thinking defines a powerful core that can be extended by companies.

The Architectural Thinking Framework® defines a lightweight architectural model that wants to be a true standard.

Architectural Maps - Enterprise Level

Architectural Thinking defines a small set of maps that are valid for the entire enterprise. The maps enable informed decisions at any level of the organisation. They can be used for strategic decisions by top executives as well as for capturing more detailed solution architectures at team level. The maps are linked in a way that allows traceability from the enterprise-wide to the solution level.

Architectural Thinking only provides maps that have proven to be valuable in many companies and have been accepted by business people and executives. The maps are based on the artefacts of the architectural model.



Here are some examples²:

- Strategic Theme Map
- Value Stream Map Levels 1 & 2
- Capability Map with various heat mapping viewpoints

² Work under discussion.

- Capability Cost Map
- Application->Capability Map
- Technology Map
- Business Object Model
- Application Interface Map
- Solution Context Map

Architectural Thinking comes with detailed, practicable cookbooks that explain how to model these maps.

State of Practice

The number of maps of potential interest for stakeholders is almost infinite. A comprehensive [report by TU Munich](#) shows the vast number of maps that can be generated by enterprise architecture tools. The most expensive tools come with more than two hundred maps. Downsizing the number and complexity of the maps is usually carried out by the tool vendors. The result is that there are no real standards for architectural maps.

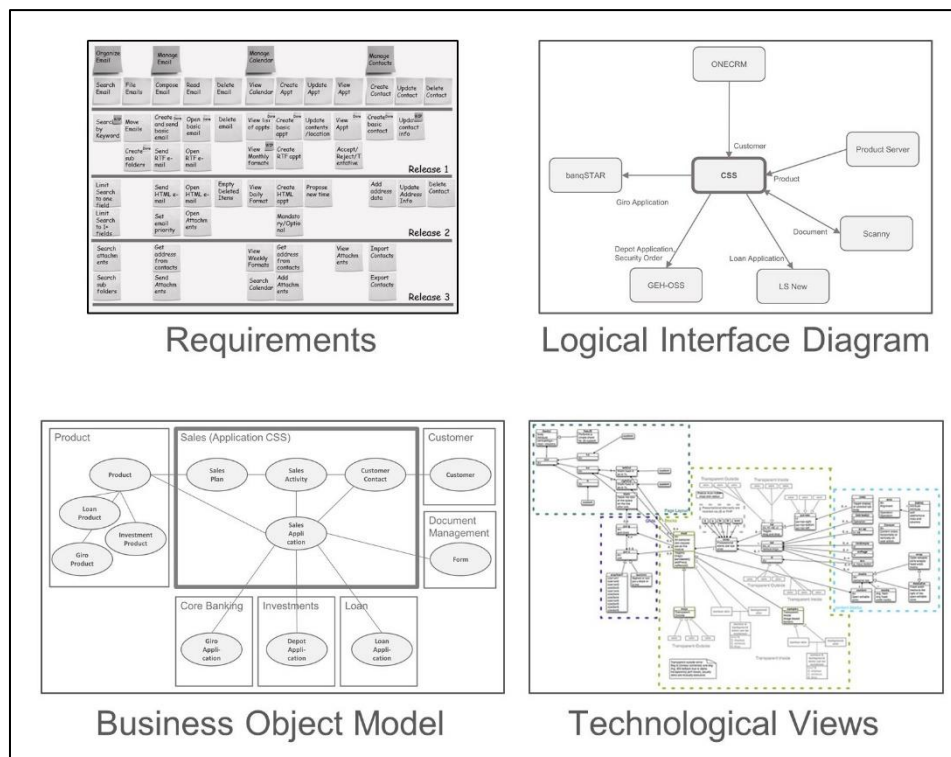
What is different in Architectural Thinking?

Architectural Thinking consists of a minimum, core set of maps that are independent of tool vendors, self explanatory and have been proven in practice. All of the maps have been accepted by various business roles. This is important to fostering the buy-in of roles that are typically not architecture oriented.

Companies are encouraged to extend the small set of valuable maps according to their usage scenarios. Compared to what enterprise architecture tools provide, the approach is vice-versa: while existing tools provide a maximum-fits-all approach that must be downsized, Architectural Thinking defines a powerful core that can be extended by companies.

Architectural Maps - Solution Level

Architectural Thinking defines a small set of maps that are used at solution level and connected to the enterprise-wide architectural model. This connection makes it possible to trace each requirement and design decisions to the enterprise-wide level.



Here are some examples:

- Solution Requirement Map
- Solution Business Object Model
- Solution Application Interface Map
- Software Architecture - Technical Interface Map
- Software Architecture - Whitebox Component Map
- Software Architecture - Relational Database Model
- System Architecture - Deployment View

State of Practice

In practice, solution teams produce two kinds of architectural maps:

Technological Maps (software and system architecture)

Software and system architecture are usually modelled with the widely used Unified Modelling Language (UML). Both disciplines are mature. arc42.org is an excellent source of information for technical architects.

However, solution architecture with UML usually jumbles technological with business related architectural elements. This suits the needs of the software development team, but makes the business architectural elements hard to grasp for their stakeholders (business people).

Solution Requirement Map

There are many approaches to structure requirements. From more strictly formalised (user cases) to less structured (user stories/epics/themes). There are only a few rules for structuring requirements in agile teams. Every team structures them according to other dimensions. Concepts such as epics/themes/stories allow a lot of freedom for interpretation.

What is different in Architectural Thinking?

Technological Maps

Software and system architecture are both mature disciplines. For this reason, Architectural Thinking does not make any further suggestions as to how to model the technical aspects of this kind of architectural maps, but refers to arc42.org as an excellent source of information for technical architects only.

However, a strict separation of technical and business-related architectural elements is mandatory in Architectural Thinking. For this reason, Architectural Thinking defines two mandatory, purely business-oriented maps: the 'Solution Business Object Model' and the 'Solution Application Interface Map' that define the business objects of the solution and how they are passed between surrounding applications. Technological maps must be derived from these maps.

In Architectural Thinking, business people are the owners of the business object model, the applications and the logical, business orientated interfaces between the applications. Thus, architectural maps must be designed in a way that is understandable for them, which means that they must be kept simple and free from technological details.

Solution Requirement Map

A strong connection between architecture at enterprise and at solution level is mandatory in order to make Architectural Thinking work. For this reason, it defines a simple yet powerful rule: each and every requirement must be easily traceable to the business architecture models of the enterprise level. This means, for example, that naming is used consistently at enterprise and solution level, and that requirements should be linked to the business capabilities they support.

Architecture Principles

Architectural Thinking comes with a list of ten Architecture Principles that guides the implementation of solutions.

Here are some examples³:

We act in a sustainable manner and as ONE enterprise

- Solutions are built with a company-wide focus rather than in business unit silos.
- Inter-divisional functionalities are implemented in company wide platforms, not in several isolated applications.

Business goals and value streams drive our application landscape

- We implement the 'right' projects - those with business values we believe in.
- We use structured business analysis, business architecture and innovation methods early in the ideation process.

We do not re-invent the wheel

- We check whether existing applications can be re-used before implementing new ones.
- We look for standard solutions before we implement individual solutions.

³ Work under discussion.

State of Practice

TOGAF 9.2® provides guidelines for creating architecture principles and a list of 21 example architecture principles. However, it is unclear whether these examples are based on practical experiences and used widely. In practice, most companies create their own principles, which questions the use of a standard.

What is different in Architectural Thinking?

Architectural Thinking provides a simple set of ten principles that have emerged by open participation of many companies. You can benefit from the input of these companies and customise them according to your specific needs.

Integration Points

Integration Points are connectors between processes of the company and the contents of the Architectural Thinking Framework®. They are there to keep Architectural Thinking out of discussions around (Agile) development- and other processes. It does not matter how the surrounding processes create defined artefacts, it just matters that they are created according to the templates defined in the Architectural Thinking Framework®.

Integration artefacts can be passed in two directions: (i) they can be produced by an external process (e.g. vision statement) and needed as a prerequisite in Architectural Thinking, or (ii) can be held in the model of the Architectural Thinking Framework® (e.g. capability model) and be a mandatory input for a process (e.g. solution development).

In the current draft, the Architectural Thinking Framework® defines Integration Points for the following processes:

- [Vision Building](#)
- [Strategic Planning and Governance](#)
- [Solution Development](#)

The Architectural Thinking Association®

The Architectural Thinking Framework is owned by the 'Architectural Thinking Association®', a non-profit, registered association according to Austrian law. The articles of the association have been defined by the founding members and can be downloaded from our website (German only, English coming soon). According to these articles, the association is purely non-profit.

The only goal of the Architectural Thinking Association is to spread the values and ideas of Architectural Thinking. It does this by:

- permanent improvement of the Architectural Thinking Framework driven by open participation;
- publications in journals, speeches at conferences of various communities;
- creation of training material for public use.

The Architectural Thinking Association is organized as follows:

The *Executive Board* consists of the President, the Vice-President and two members responsible for finance and administration. It is elected by the members. The executive board has duties that include the following:

- admission and dismissal of members;
- delegating members to the content committee;
- financial management;
- calling a general assembly of all members;
- communicating changes of the articles of association to public authorities.

The *Content Committee* is the core of the organisation. This committee makes decisions about changes/extensions of the Architectural Thinking Framework and approves the association's publications. It consists of highly experienced, renowned professionals from various fields (such as enterprise/business architecture, business analysis, business process management, Agile, software architecture) that agree to the values of the Architectural Thinking Association®. The Executive Board is responsible for delegating Members to the Content Committee.

Every individual or legal person is encouraged to apply for membership on the association's website. Once the application is approved, persons become *regular Members*.

Regular Members have the right to publish content under the logo of the association. This content can be used for commercial purposes and can remain the intellectual property of the member, but must be approved by the content committee for its compliance to the values of the association and to the ideas of the Architectural Thinking Framework.

Regular Members have the right to vote at the General Meeting held via teleconferencing at least once a year.

All regular *Members* are called upon to create content to become the intellectual property of the association.

PARTICIPATE!

We are currently in the start-up phase. We plan to release version 1.0 of the Architectural Thinking Framework (ATF) by the end of 2018.

What has happened so far:

- the Association was founded according to Austrian Law;
- the logo and the name have been registered as a trademark;
- the core concepts have been published on the association's wiki for open participation.

The content available in the wiki will be steadily extended.

We plan to release version 1.0 of the Architectural Thinking Framework by the end of 2018.

If you like the idea of our lean, collaborative, open framework, you can participate in many ways:

Comment or make suggestions for changes at architectural-thinking.com/ATF

Everybody is invited to comment on the published content or make changes in our wiki. It is possible to edit the content of pages. In this case, a member of the association will check the suggestion, and if it applies to our ideas, he/she will publish it.

If you are interested in a specific topic, you can register for change-notifications of the corresponding pages.

Register at our blog

Every Friday, we write a blog post that discusses specific ideas of architectural thinking. So, if you want to stay informed, just register on the blog and join in our discussions.

Every month we send out a newsletter that summarises the progress of the previous month and includes the current news of the association.

Donate money or skills

The initial setup of the content, the graphics, the marketing, the website, the administration was funded privately. In order to reach the next level of professionalism, skills (or money to buy these skills) in these areas are required.

We would be grateful for any donations of money or skills that can help us with the topics mentioned above.

If you are interested, please send an email to office@architectural-thinking.com

Apply to become a member of the content committee

The *Content Committee* makes decisions about changes/extensions of the Architectural Thinking Framework and approves publications of the association.

We are looking for highly experienced, renowned professionals in various fields (such as enterprise/business architecture, business analysis, business process management, Agile, software engineering) that agree to the values of the Architectural Thinking Association®.

If you are interested in becoming a member of the content committee, please send an email to: office@architectural-thinking.com including your CV, a letter of motivation and reasons why you are the right person for the content committee.

Apply to become a member of the board

At the present time, there are two members of the executive board. We are still searching for two more board members.

If you are interested in becoming part of the executive board, please send an email to office@architectural-thinking.com including your CV, letter of motivation and reasons why you are the right person for the board.

Q&A

Q: What is the difference between Architectural Thinking and the discipline of Enterprise Architecture Management?

A: Architectural Thinking is not a discipline carried out by a specific role. Instead, it is a connector between disciplines, targeted at a broad range of (mainly business-related) roles.

Q: What is the difference between the Architectural Thinking Framework® and existing Enterprise Architecture (EA) Frameworks?

A: The Architectural Thinking Framework® defines a minimalist framework that can be easily integrated into Agile methodologies. It is based on the proven experience of many practitioners in various fields (such as Agile, EA, business architecture, business analysis, Agile, software architecture). Architectural Thinking provides lean but detailed, practical how-tos that are permanently improved by mechanisms of open participation. Architectural Thinking is free, and owned by a non-profit registered association that has no financial interests.

Q: Can I use Architectural Thinking with my existing (Enterprise) Architecture Tools?

A: Yes, if (i) the meta-model of your architecture tool can be customised (which is true for most of them), and (ii) the tool provides strong Web 2.0, wiki-like functionalities for collaborative architectural work.

Q: How are Architectural Thinking and Agile related?

A: Approaches for company-wide agility (such as SAFe, LeSS, Disciplined Agile Framework) and agile methodologies such as Kanban or SCRUM, define how people work together to create solutions. Architectural Thinking defines the deliverables that are required to ensure that these solutions fit together and into the big architectural picture of the enterprise. Thus, Architectural Thinking can be seen as an extension of Agile that enables structured cooperation between teams based on clear target architectural models.

Architectural Thinking is process agnostic, i.e. it works with any development methodology like other waterfall-like approaches.

Q: How are Architectural Thinking and Design Thinking related?

A: Design Thinking focuses on creating innovative solutions. Architectural Thinking can be easily added to make sure that these solutions are sustainable and fit together.

Q: How are Architectural Thinking and Business Vision related?

A: Business vision is a mandatory input. Architectural Thinking defines how the vision must be formulated in order to derive the target architecture model, but does not deal with the process of vision creation.

Q: How are Architectural Thinking and Business Strategy related?

A: The connection between vision, strategy and the architectural model is mandatory in Architectural Thinking. For this reason, it defines the structure of the integration artefact 'Strategic Theme' that connects the business vision with the architecture model.

Q: How does Architectural Thinking fit into other enterprise processes?

A: The Architectural Thinking Framework® is process agnostic, which means that it does not include processes. It simply defines deliverables (such as architectural models and maps) and cookbooks that show how to create them. These deliverables can be integrated with existing enterprise processes via 'Integration Points'.

Q: Which roles are defined in the Architectural Thinking Framework®?

A: To ensure conceptual integrity between solutions, dedicated 'Architecture Owner' roles are defined for each level of architecture (Solution/Capability/Enterprise). However, due to the fact that most of the work is carried out by many people, Architectural Thinking invites existing or emerging roles such as business analysts, business process managers, business architects, enterprise architects, software architects, system architects or software developers to participate in the creation of consistent, linked architecture models.

Q: How are Architectural Thinking and Solution Development related?

A: The Architectural Thinking Framework® provides simple architectural maps that connect artefacts that are produced during solution development (e.g. User Stories, software and solution architecture models) with the enterprise-wide big picture.

Q: How are Architectural Thinking and Business Architecture/Business Analysis related?

A: The Architectural Thinking Framework® is business orientated and heavily influenced by the work of the 'Business Architecture Body of Knowledge (BIZBOK)' of the Business Architecture Guild. Business Analysis practices as promoted in the Business Analysis Body of Knowledge (BABOK) of the International Institute for Business Architecture (IIBA) are of great help for creating the business-related artefacts (Value Streams, Business Capabilities, Business Objects, Applications)

Why Architectural Thinking?

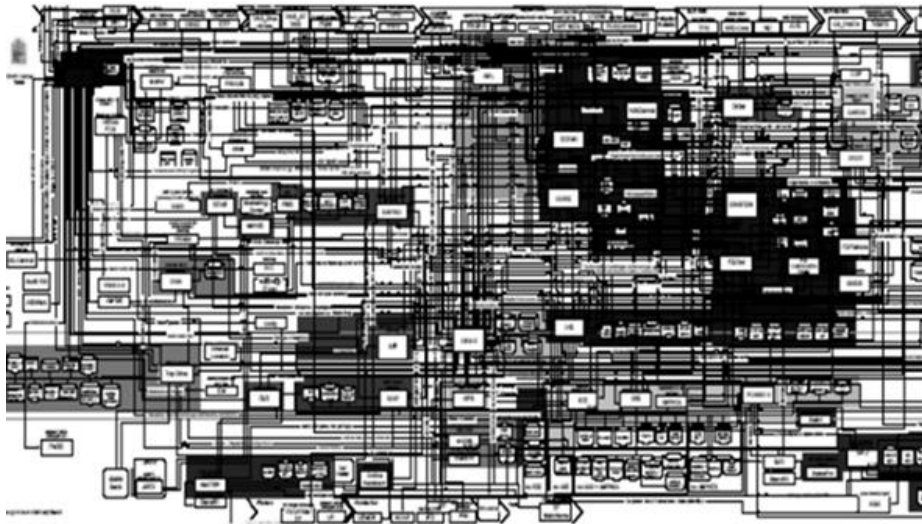
Nowadays, Enterprises are in an uncomfortable position. On the one hand, their industries are being disrupted by new competitors, and the pace of innovation has increased dramatically. On the other hand, they need to run a vast legacy IT landscape which thwarts the implementation of innovative digital solutions that integrate the old with the new business & IT.

Companies put a tremendous effort into innovation initiatives. These efforts are fuelled by the necessity to have more modern digital business capabilities in the organisation, in order to stay competitive. They must be capable of quickly deploying technology that produces business value.

Driven by this demand, communities surrounding Agile and Design Thinking are growing rapidly. Both ideas focus on speed and innovation, but do not answer the question of how to ensure that created solutions fit together and into the big picture of the enterprise. They do not answer the question of how to deal with the enormous 'technical debt' that is created by grubby integration of new digital solutions with legacy or with each other. They are speedy in the short term, but do not answer the question of how to maintain this speed in the long term.

Focusing solely on new ways to create customer value, without seriously considering sustainability and architectural integrity, creates point-solutions and enormous costs in the long-term. This can be taken as a proven fact, if you only think of what has happened to application landscapes⁴ over the past decades with a much slower pace of innovation:

⁴ This example is from a real bank and typical for the status-quo of most medium to large organisations.



In our experience of carrying out enterprise architecture, I would estimate that the IT of a typical organisation of the 'old economy' has ten times more applications, servers, databases, than would be necessary if business and IT had been properly architected. This is the reason why current application landscapes are not agile for adding new functionalities. 90% of the IT costs could have been saved, if architectural thinking had been applied properly.

And this is not an IT problem. The severe situation has been caused by the absence of the notion of architecture in the business world.

Issues such as:

- overlapping responsibilities between business units;
- unclear data and process ownerships;
- business unit silos;
- weak connections between departments that structure the business (such as strategy creation, business model generation, product management, business process management)

are common for any large organisation I have ever worked in.

If we project this existing situation into the future of faster and faster innovation cycles, it becomes clear that without a structured approach to managing the dependencies between business & IT structures, complexity will increase exponentially. Companies simply must solve this problem as

long as they can afford to do so. They must invest in legacy business & IT renovation in order to reduce complexity, and must learn how to integrate new digital solutions into their legacy IT and business processes in a way that keeps technical and organisational debt within reasonable bounds. They must learn how to make informed decisions that are based on holistic viewpoints that integrate business & IT architecture. They must start to build structured connections between any level of the business and IT organisation.

They need to start Architectural Thinking.

Appendix:

Artefacts of the Architectural Thinking Model

Business Capability

A [business capability](#) is defined as 'a particular ability or capacity that a business may possess or exchange to achieve a specific purpose or outcome. Capabilities represent the basic building blocks, or DNA, of a business'. Capabilities define what, not how, a business does something' [BIZBOK].

Examples of Level 1 business capabilities are: sales, risk & compliance, payments

Why is it important to model business capabilities?

- *Capabilities clarify terms and concepts across organisational borders.*
- *Capabilities provide a robust skeleton, a framework for assigning all the other elements of the enterprise architecture.*
- *Capabilities can be used as the central structure for heat mapping in order to answer questions such as: 'Which strategic fields of actions do we see in which capability'; 'In which capabilities are we planning to invest how much?'; 'Which capabilities are not supported enough by IT?'*
- *Assigning IT-applications to capabilities is a powerful way to support business & IT alignment.*

Value Stream (Business Process)

A [*value stream*](#) is a visual depiction of how an organisation achieves value for a given stakeholder or stakeholders within the context of a given set of business activities [BIZBOK®]. It consists of *value stages* that are performed within the organisation in order to create *customer value*.

If you are experienced in business process management, this definition sounds like the definition of a business process. And this is correct. A value stream is a business process that focuses on maximising value for the customer. Value Stream Mapping (VSM) is a technique used in process methodologies such as 'Six Sigma' in order to understand how customer value flows through a process and how to identify waste in processes. When modelling value streams, organisations view themselves from an 'outside-in' perspective rather than the 'inside-out' perspective that were common in the 'business process reengineering' projects of the last two decades.

Value streams are cross-mapped to *capabilities*, to show how a company orchestrates capabilities in order to create customer value.

Some examples of value streams are: 'open account', 'buy stocks', 'order bank card'.

Why is it important to manage value streams?

- *While capabilities define what a company needs to do, value streams provide a common understanding of how a company delivers customer value.*
- *Modelling value streams challenges business capabilities. Capabilities that do not participate in creating customer value can be dismissed.*
- *Value streams indicate which business capabilities are required to create customer value.*
- *Assigning IT-applications to stages of the value stream is a powerful way to discover process-cracks of insufficient IT-application coverage.*

Business Object

A [*business object*](#) is a thing that business people handle and use. Such business objects can be tangible, with a physical presence (such as cars or computers), or intangible (such as accounts or payments). Business objects are stored in IT-applications. They hold the information required or produced in the corresponding business capabilities, in the form of attributes. Banks, for example, create intangible business objects such as accounts or payment transactions. An account may hold attributes such as 'balance' and 'transaction limit' which are stored in the IT-application 'core banking system'.

Some examples of business objects are: account, stocks, depot, bank card.

As we will see later, it is important to model *relations* between business objects. The business object 'payment transaction', for example, has a relation 'is processed for' to the object 'account'.

Why is it important to manage business objects?

- Business objects represent the business vocabulary. They are a basis for the consistent naming of capabilities and value streams.
- *Business Objects are the foundation for data governance, as it can be used as the structural basis for defining who is liable for which business object.*
- *When mapped to IT-applications, data redundancies can be discovered and reduced.*
- *A Business Object is a bridging artefact that defines the information the business people handle in a way that can easily be transferred into the world of IT-applications, while still retaining a purely business viewpoint. This makes them perfect for business/IT alignment.*

Application

An [*application*](#) is a computer program that provides useful business functionality to its users. It supports a bounded set of business capabilities that are provided as user interfaces. An application stores owned business objects and makes them available to other applications by interfaces.

An application owns business objects (e.g. a CRM application owns business object 'customer') and may process business objects owned by other applications if connected by an interface.

Historically, applications have been in the hands of IT as they are implemented in the form of custom or standard software (such as SAP). In the Architectural Thinking Framework, however, business people are accountable for their applications and the business objects that exist in the applications. Thus, architectural responsibility is shifted to where it belongs - the business.

Applications are implemented on top of software technology components (such as application servers or web browsers) that run on hardware such as mobile devices, servers in a computing centre (on-site) or in the cloud.

Some examples of applications are: CRM, SAP HR, e-banking risk management system.

Why is it important to manage applications?

- *As applications map capabilities/value streams to technology, they enable more effective planning of IT projects.*
- *By assigning applications to value streams, their cost can be weighed against their business value.*
- *By grouping applications by the top level capability, they support functional redundancies.*

Technology Component

A [*technology component*](#) represents a software or hardware resource that hosts or interacts with other technology components. It executes computational behaviour or stores and processes data. Technology components are used to support applications.

A technology component can compose other technology components, for example, an 'operating system' consists of a 'file system' and a 'user interface'.

Technology Components can be related to other technology components. A 'database', for example, runs on top of an 'operating system' and a 'hardware server'.

Why is it important to manage technology components?

- *Categorising technology components by technology domains helps to identify redundant technologies.*
- *To know which technologies you already have is mandatory to assess new technologies that fit with legacy.*
- *Assigning operation and maintenance costs to technology components is mandatory for IT cost controlling.*
- *The cost of IT-application can be calculated by summarising the cost of supporting technology components. Thus, the cost of IT can be assessed in comparison to its business value.*