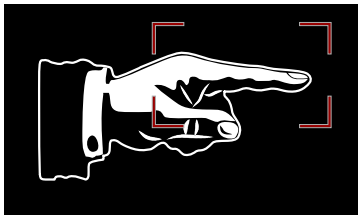
The

SEMANTIC EYE

Framework 1.0



[White Paper]

The Engineering Toolbox
for Effective Design, Evolution and Digitalisation
of your Organisation and its Processes

Oliver Reinhard

Version 0.7.6, Copyright 2026

TABLE OF CONTENTS

1. Abstract

Digital transformation (short: digitalisation) is the support or the partial or full automation of manual activities and processes with software. The Semantic Eye focuses on processes of organisations where humans, machines and software collaborate to achieve goals and create value in a controlled and repeatable fashion.

Judging by media reports—and possibly also in the reader's own experience—, something is going terribly wrong in most organisations' digitalisation projects. Too often, such projects become an existential threat to the organisation. At the same time, other industries complete projects of similar complexity with much higher success rates and without exhausting their staff.

For decades, the *Technology tier* of digitalisation has been the preferred «battleground» of organisations while they neglected the *Organisation tier*. They have been looking at the wrong map. This misguided focus of activities is the source of a large part of the problems these organisations experience.

The Semantic Eye changes the narrative. It takes successful industries as role models and proposes a disruptive, *strategic approach* for the digitalisation of organisations:

1. A shift of the organisation's focus of activities away from a passive involvement at the Technology tier and to an *active, leading role at the Organisation tier*.

   From there, the organisation take full ownership and control of their *value-generating processes*.
2. The introduction of an explicit, formal, software-supported *model of subject areas and processes*, named the organisation's *Cosmos*.

   It takes the pivotal role that *standardised blueprints or models* have held in mature industries for decades or even centuries (e.g. in building construction, music, automotive and aerospace, logistics, electronics).

This document first highlights typical digitalisation blunders; then reveals predominant causes for the current *digitalisation malaise*; and finally presents a practical solution: the *Semantic Eye Framework*. Which, apart from pragmatic concepts and procedures, introduces powerful digital tools and unlocks effective use of AI.

2. About

The Semantic Eye Initiative

The initiative is named the Semantic Eye for its defining *curious mindset* and its aim to look carefully; to discern the true meanings (*semantics*) of things; to *understand* actual *causes and effects*; and to closely examine strategic goals before jumping to conclusions or premature action.

The Semantic Eye has four main objectives:

- to distinguish between *goals*, *concepts* and *implementation* of organisational processes;
- to empower organisations as the source, the target and the driver of their own digital transformation;
- to reduce dependencies on IT suppliers by relying on new, impactful *methods*, *tools* and *processes*;
- to make organisational processes and digital transformation a successful and satisfying endeavour.

The Semantic Eye embraces an agile approach, however, first at the organisation tier until clarity and alignment is reached; then at the technology tier.

This Document

This document is directed to anyone looking for insight on the problems of and a brighter future for organisation and their digitalisation. The principal goals of this document are:

- to make readers realise and understand the immaturity, shortcomings and dire state of digital-transformations and the associated industry;
- to enable and encourage readers in exploring whether their own organisation and processes could benefit from the Semantic Eye Framework.

Make sure you are reading the latest version of this white paper. If this paper looks overwhelming, consider reading its «nutshell» brother. Both documents are available on www.semantic-eye.org.

The Author

Oliver has a master's degree in Software Engineering from ETH Zurich in Switzerland. During his professional career of 30+ years, he has actively led, worked in or contributed to the successful digital transformation of many different kinds of organisations. He got into model-based software development and requirements engineering over 20 years ago, mostly using UML [1].

Irritated by the same recurring problems in digital-transformation projects over the last 10 years, he started to analyse the causes in detail. This led him to develop a precursor of the Semantic Eye Framework in 2018, then test new ideas and tools in all his projects.

3. Orientation

First, let's introduce a few key terms and then clarify the scope of the Semantic Eye Framework.

- An *organisation* is an interdependent human-centred system in which social and technical agents constantly interact and co-create.
- An *agent* is an active resource of an organisation who can act independently and purposefully within a shared environment shaped by both social rules and technical infrastructure; an agent can be a human, a piece of smart software, a robot, etc.
- A *subject area* is a defined area of knowledge, control and activity in «the world» (i.e. in business, administration, transportation, daily life, etc.) within which organisations operate; also called a *domain*[1].
- An *organisational process* is the coordinative glue between the active elements of an organisation and the passive elements of one or more subject areas.
- The *digital transformation* of an organisation is a substantial campaign with the objective of hoisting the organisation and its processes to a significantly higher level of digital enablement and automation.

Transformation Tiers

Digital transformations break down into activity on three mutually-dependant tiers as shown in figure 3-1:

Figure 3-1 | Transformation tiers and mutual influences (green)

- The Strategy tier must identify and define the *goals* and the future *products and services* of the organisation; it must explain *why* a transformation is needed.
- The Technology tier defines what is *feasible* and *how*, and eventually provides the *resources* to do so.
- The Organisation tier ❶ masters the *knowledge*, rules and complexity of the relevant *subject areas*; ❷ coordinates the resources provided by the Technology tier, to realise the given strategy.

The Organisation tier is the scope and the focus of both this document and the Semantic Eye.

The stack of tiers, in its linear form, insinuates a top-down order of activities, i.e. first Strategy is tackled, then Organisation, then Technology. This is not the case. All tiers mutually depend on each other. What technologies are needed is determined by strategy and by the an organisation's subject areas and processes. However, what strategy and organisation can actually achieve is limited by the capabilities at the technology tier. A strategy cannot be successful if technology or organisation cannot deliver, e.g. for lack of capabilities, resources, legal foundations, etc..

Due to the mutual dependencies of the tiers, the best solution is found by iterating over all three tiers until a mature solution is reached.

In practise, responsibilities and activities tend to span organisation and technology tiers or to blend. A clear separation of those tiers is crucial for project success as we will discover over the next chapters. Complemented by solid methodology and diligent work on both tiers.

Here are the responsibilities of the two tiers during the analysis, planning and realisation phases of a transformation in more detail:

- Organisation describes the subject areas from a conceptual point of view[2], including their work objects, relations, business rules, know-how, etc. It identifies and defines the active and passive, material and virtual resources it requires, including their roles, responsibilities, qualities, etc. It defines organisation-level goals, allocates and coordinates all the resources through organisational processes. It trains and organises the human agents according to their roles and responsibilities.
- Technology describes the capabilities and quality of the resources it can provide, typically in the form of specifications of hardware, software, tools, interfaces, etc., including human skills. It hires, procures, builds or develops all the required human resources (individuals, groups), machines, tools, software, offices, trucks, licences, power, etc. and provides them to the organisation.

The responsibilities of the two tiers during the operational (i.e. execution) phase are outside the scope of the Semantic Eye.

[1] The terms *domain of interest*, *business domain* and *problem domain* are also in use; for brevity, this document uses *domain* expert over *subject-matter* expert or an acronym.

[2] A *conceptual view or model* conveys an abstracted, reflected picture of an underlying, more complex world for the sake of focus, ease of understanding and communication.

4. What is the Problem?

Sooner or later, most organisations on their mission to higher levels of digital enablement and automation pass through the «Valley of Tears». Despite great efforts, much frustration and ever-growing budgets, transformation projects fail to meet essential objectives or are canceled altogether. A shocking performance !

But what is the main cause? — While other industries have developed powerful concepts, engineering practises and sophisticated software tools at great cost, digital transformations still apply amateurish methodologies [2] and use no digital tools worth mentioning.

However, any conversation about this subject quickly becomes very abstract. So, instead, let's walk through a fictional *case study* of a non-digital transformation. The purpose of the study is to demonstrate how the digitalisation industry for organisations would tackle this kind of transformation. A certain magnification is intended to highlight the weak points.

For centuries, Venice had been one of the leading trading powers in Europe, its merchant families had accumulated immeasurable wealth. But after the fall of Constantinople in 1453, a long and slow decline set in. During the 17th century, European trade with the Americas started to flourish to the detriment of Venice. In 1688, two wealthy Venice families emigrated to the Southern tip of South America to found an even more lavish and beautiful twin which they called New Venice. The city flourished quickly due to its favourable position at the Magellan straits: every ship en route to or from the America's west coast was calling on to resupply. But when the Panama Canal was inaugurated in 1914, New Venice lost most of its trade. In the 1980s, tourists started to discover the forgotten gem with its hundreds of canals, gondolas, bridges, squares, churches and magnificent buildings. They flocked to the «Southern Peal» in ever-greater numbers. However, the only way to get there was by bus (26 hours from Buenos Aires) or by ship (16 days from Europe). For this reason, visitor numbers stagnated. The city fathers decided that visitors should travel *by air*.

So, they called in the local building company Agilicca to build a suitable *airport* that the city would then operate. Agilicca had previously built a large hydro dam, shopping centres and many residential buildings. And they had successfully used a new construction material, called Cloudite. Made from wood fibres, it is stronger, lighter and cheaper than concrete.

On day one, the Agilicca engineers started to interview the head of the New Venice tourist office. They used a technique known as «user stories» to capture the *requirements* for the new airport and scribbled them down on little yellow sticky notes. For example: «As a new arrival, I want to get my bags quickly so that I can spend more time enjoying the sights».

To the surprise of many, excavators started to tear up the ground after only a few weeks. Soon, the first building was erected. Meanwhile, the engineers continued to produce more yellow notes and grouped them into clusters on a large wall. Every morning, all the crafts got together to discuss the work of the day based on the sticky notes in a cluster named «This week». After a few more weeks, a first small plane landed on the still-very-short runway.

It was unfortunate that the need for elevators came up late and required significant rework of the buildings. Also, the first building soon had to be moved by 200 metres because it blocked a critical view from the air-traffic control tower; which was put up to fulfil *aviation-safety regulations* the project team had only become aware of recently. A new, unanticipated concern popped up every day and forced the hand of the project team.

Eventually, flights with real passengers and larger planes came in. But arrivals and departures were severely hampered by the *chaotic baggage situation* for the lack of centralised baggage check-in facilities— there had not been a yellow sticky note asking for that. Hence, passengers carried their baggage to and from the airplanes themselves. In a hurry, *baggage trolleys* were introduced. But after every flight, dozens of abandoned trolleys were left by the runways and created a real danger and nuisance.

Most flights were delayed in landing and departure. The airlines were annoyed. The airport should have served six million passengers per year[3], yet half a million was its practical limit. It had acquired a bad reputation. International interest in New Venice dwindled quickly. The transformation to having most passengers arrive by air instead of by bus and ship had failed.

[3] Venice (Italy) sees ±12 million overnight visitors per year

What Happened?

So, where did things go wrong in connecting New Venice to the international airways? Let's use the transformation tiers from figure 3-1 as a tool to locate what happened at which tier and when (figure 4-1).

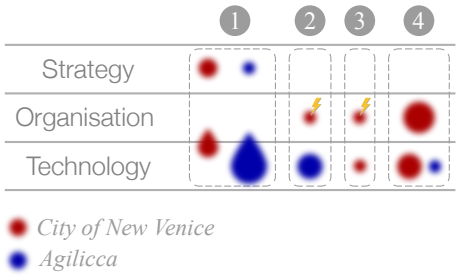


Figure 4-1 | Major activities: tier and magnitude (flash symbols indicate unexpected insights)

❶ The city fathers acted on the Strategy tier when they decided the city should open itself to commercial aviation. The first move to implement their strategy was contracting Agilicca to build an airport. Airport buildings and runways are technology, so it is this tier where Agilicca was active and where the city participated with requirements.

❷ The *overlooked* aviation-safety regulations are subject matter and thus belong on the Organisation tier. This required moving a building, which is Technology tier.

❸ The chaotic baggage situation was owed to a *lack of deeper understanding* of passenger and baggage flows, and to the ensuing *missed innovation*, i.e. not recognising the opportunity and benefits of centralised baggage handling. Both are on the Organisation tier. The abatement with baggage trolleys is Technology tier.

❹ Finally, the city implemented all the *airport processes* required to handle arriving and departing airplanes and passengers. The definition of the required roles, responsibilities, resources and process is Organisation tier. The provisioning of the people organisation and of the virtual and material resources is Technology tier.

What went Wrong?

For aviation passengers, flying is a *service* that starts at booking time and ends at the moment they leave the destination airport. This service consists of many individual sub-services provided by a great number of organisational processes. The airline is responsible for some of those processes; the airport, the border police, air-traffic control, etc. for others. Most processes are collaborations that involve multiple agents.

The processes rely on hundreds of *resources* like airplanes, runways, gates, buildings, offices, conveyor belts, computers, cables, monitors, etc. But also on people, checklists, computer programs, permits, licenses, etc.

So, what went wrong? — Important parts of the resources were designed and put in place before a clear picture of the processes existed and had been approved by all stakeholders. A lack of alignment among stakeholders creates a high risk of procuring and/or building the wrong resources. This either leads to inefficient or even ineffective processes, or else it requires unplanned work or expenses to end up with suitable resources.

Figure 4-1 also tells a bigger story: First (❶), the Organisation tier is skipped entirely. Later, it shows some *reaction* when prompted by the encounter of overlooked regulations (❷) and operational issues (❸).

Generally, the strategy to bring commercial aviation to New Venice was as farsighted as the decision to immediately start building an airport was short-sighted.

- The airport is a crucial resource and is absolutely needed at *some* point to realise the strategy.
- But the biggest mistake was the city's *failure to first acquire a deep understanding* of the core processes. That is, of what it means to work with commercial airlines and to operate an airport with respect to aviation safety, established processes and efficiency.

Instead, the city should have first developed a «big picture» at the Organisation tier. The «big picture» must provide an end-to-end view of at least the core processes, the relevant subject areas and resources. It must use a standardised language (text and/or graphics) that all stakeholders can learn, understand and challenge.

Requirements-Driven Feature Negotiation

Let's analyse the collaboration between the city and Agilicca during activity ❶ in more detail because—on purpose—it represents the typical collaboration of an organisation and its technology providers throughout a *digital*-transformation project, figure 4-2:

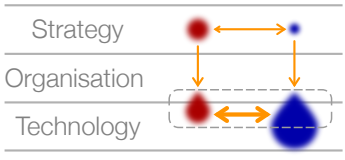


Figure 4-2 | Feature negotiation (in dashed box)

The main characteristic of the collaboration is the feature negotiation taking place mostly at the Technology tier. The organisation provides the *requirements*; the technology partner, in its own dark corner, designs and implements *solutions* that satisfies the requirements. Or so, the organisation should *hope*, because it will only

find out later—when deviations and errors cost money and effort to amend.

The collaboration is problematic in various ways:

- The requirements are often described by members of the organisation who have no formal training to do so; often they were hired or appointed specifically for this role but lack deep know-how of the subject matter. Thus the completeness, coherence and quality of the requirements is dubious at best.
- The members of the technology partner are not trained in requirements engineering either. Thus they are not in a position to assist or compensate.
- The members of the technology partner typically have no previous, deep knowledge of the relevant subject areas and processes. Thus, the two groups don't share a common «domain language». This makes the collaboration slow and prone to misunderstandings.
- The *estimates of effort and cost* are solely based on requirements rather than on an explicit, aligned solution description—and are thus bound to be *unreliable*.

If this situation sounds like «the blind leading the blind», then because that is exactly what it is. This *naive approach* of discovering requirements and designing features sequentially and tackling problems as they come up bears not only unmanaged risk but even *unmanageable risk*. The team never knows what's up next and can never anticipate and act with certainty. Every day, an unforeseen issue could break the project's neck.

No Explicit Solution Description

At least on the part of Agilicca, who used yellow sticky notes to capture the requirements that were later lost to the winds, there was no explicit documentation of the solution before or after it was realised. The knowledge of *how* things were built and *why* they are the way they are, rests solely with the people who built them. When workers leave the project, airport staff often needs days to figure out how cables, ducts and pipes were routed; maintenance becomes a nightmare, and it is very difficult to train new staff.

This is also a frequent flaw of *digital*-transformation projects: the only reliable «documentation» of the solution in place is its program code. Not only is it so rich in different functional aspects and detail that the forest is hard to see for the trees. But it is also totally unaccessible to domain experts. Firstly because domain experts typically don't have access to program code, secondly because very few are skilled software developers and are thus unable to read and make sense of it—this is even hard for developers who have not produced that code themselves.

This situation makes the organisation highly dependent on their IT-solution provider or even to individual software developers for the lifespan of the software. Artificial Intelligence could certainly help here by creating documentation of the software solution in a form that is readable by domain experts. However, since only a part of the activities of most processes is supported or automated by software, the other part of those processes remains either undocumented or is likely to be documented in a different system and style.

So, why is an entire industry struggling here?

5. Why is it so Hard?

Let's get back to digital transformations and understand what makes them such a hard problem to solve.

Automation is Hard

Firstly, *automation* is the holy grail of any discipline. Automating even a small part of a subject area requires awareness and domination of the involved complexity over a much larger part of that subject area, in both detail and precision. Moreover, the choice and exploitation of suitable technology for the purpose of automation is yet another—and very different—subject area to master.

Innovation is Hard

Just like automation, innovation is hard and—for digital transformations—goes hand in hand with automation. Designing a new or simply a «better» solution requires several «ingredients»:

- a deep understanding of the involved subject areas;
- a thorough understanding of the current processes;
- a good understanding of the strategy;
- a body of precise requirements for the new solution;
- a solid understanding of the technology currently in use and of candidate technologies for the new solution, such as mobile technologies, cryptography, cloud-based software, AI.;
- creativity, courage and persistence.

Obviously, understanding is key. But what actually *is* understanding? A few individuals reading lots of documents, studying software code and conducting interviews is not enough. In complex environments, understanding is a group exercise in communication and knowledge management. Knowledge management, as a topic, is beyond the scope of this document. However, at the very least, essential knowledge must be captured in a language that all group members can understand and that let's them challenge and align the results. Furthermore, it must cover these key aspects:

- what?
- when?
- why?

The *why* aspect is the hardest of the three and the essence of understanding: only if we know why, can we make changes with reason and confidence.

If more resources and time can be dedicated to analysis, covering the following aspects is also very helpful:

- how?
- who?
- how many? (quality)

This enables *systematic innovation* by challenging each of these aspects individually and looking for automation and improvement [3].

Tackling Complexity

Designing and building an airport and integrating with commercial aviation is an extremely complex exercise. This is why it was chosen for the case study over the construction of just a family home. Of course, in reality, the project would be handled very differently due to the expertise gained from building airports at hundreds of cities around the globe. But remember that the purpose of the study was to expose typical methodology applied by the *digital*-transformation industry.

Pundits of that industry often claim that their software is pushing the complexity boundaries of what is humanly possible. This is a typical excuse when projects are delayed or when they don't deliver on requirements or promises. The argument is hard to believe in the face of what mind-bending complexity other industries are able to master. E.g. computer-chip design and manufacturing at nanometre resolution [4]; aerospace; self-driving cars; computer games in 3-dimensional worlds based on physics and behavioural simulations, for example with the Unreal Engine® [5].

However, those pundits are probably even right! The reason lies in *humanly* possible: building software solutions for organisations is largely a manual affair and is thus limited by what those humans can handle. Note, however, that the weak point is not the actual development of the software programs and components. Both software development and software quality have made enormous progress in terms of methods and tools over the last 20 years thanks to automation and, recently, AI. Smartphones and smartphone apps are incredibly complex machines. But when did an app last crash on you? When did you last have to restart your phone because it misbehaved? How often did you restart Windows computers 20 years ago?—At least daily.

So, where *is* the problem?

Organisations do not master the combined complexity of their strategy, requirements, subject areas and processes at the level required for automation and innovation.

Organisational Complexity

Organisations like an airport or any enterprise with more than a handful employees manage incredible *organisational complexity*. Only: when humans are in charge of handling it, the level of complexity is greatly underestimated because they are so good at it and find effective solutions to new situations and requirements on the fly. But when we try to get to the bottom of a subject area, suddenly the perceived complexity grows exponentially with the exploration of each new topic. Because within it, lay ever-more fractal-like topics. Talk to any time-tested expert of any subject area and be amazed by the richness of their knowledge and by the complexity and the subtleties of even a seemingly «simple» process. Exceptions wherever you look.

For an organisation, obtaining the full picture and understanding all the tiny-but-vital rules, dependencies and idiosyncrasies of an existing solution is a science in itself. It is a science that goes far beyond what has been the requirements-engineering discipline [6] for decades. And for which there seems to be no accepted, software-tool-assisted approach yet. It is thus one of the main elements of the Semantic Eye Framework.

Agile for Digital Transformations

A popular approach to mastering complexity is *agile* methodology. In the case study, instead of proactive analysis resulting in a «big picture» and careful planning, the city of New Venice and Agilicca went with a lightweight, rapidly-iterating approach to overcoming complexity. They started on one end of an extremely complex problem, gathered a bunch of requirements, and implemented a solution for them. Then, they evaluated the result, gathered necessary improvements and the next bunch of requirements, addressed those, and so on.

Now, advocates of agile methodology like Scrum [7] will argue that eating an elephant in small bytes is the only way to go; and that errors need to be made in order to learn and improve. Their assumption for this methodology is that all organisations and subject areas are *chaotic*. And their rationale is that mastering the complexity of chaotic environments up front is impossible.

Let's have a closer look:

– Through its trial-and-error approach, agile is not efficient. There is a reason that, in real life, airports are not built with Scrum. Also, errors are expensive (see below).
– If need arises, agile teams adapt the solution already in place to achieve a next, better version — and call it *learning*. Errors and learning are important, but does every team need to make the same mistakes all over again? Or can we learn collectively as an industry, like other industries have before us? Can we find faster and more reliable methods of solving the same kinds of problems the same, proven way?
– Agile teams do perform in highly unclear or even chaotic environments where other methodologies try to plan ahead with little success.
– In most cases, *unclear or chaotic* situations can be converted into *understood and predictable* ones. One discipline that does this is engineering. It typically delivers with the help of mathematics (including graph theory or statistics), other formalisms (e.g. chemical formulae), or formal models (e.g. 2D or 3D blueprints, musical scores).
– Organisations and organisational processes are precisely such a case that can be formalised, standardised and thus addressed once and for all. Hold that thought for chapter 6.

The Cost of Errors

It is an accepted truth that the cost of fixing errors grows exponentially with each phase. Figure 5-1 visualises the cost of fixing errors (defects) through the height of the bars depending on when errors are made and when they are detected [8].

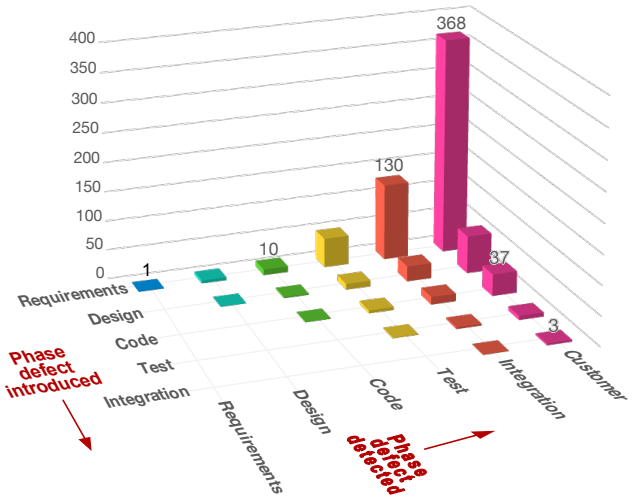


Figure 5-1 | The cost of errors

The diagram strikingly conveys that errors made early and detected late cause far greater costs than those not made at all or detected within a short time. This is intuitive: if you order a new table at your local furniture-maker, it is far, far cheaper to change the table design from seating three people to seating four on paper than

it is on the finished table.

As a consequence:

- we must avoid errors, especially early ones (they are the most expensive to fix);
- we must detect errors as soon as possible;
- we can invest significant effort and money in methods and tools and still save overall cost and time;
- if we (as an industry) invest in methods and tools, we help many projects, not just one.

These insights are not new. As mentioned, other industries had these insights decades or even centuries ago and have developed effective processes, roles and tools to overcome the crippling issues.

So, what *did* they do?.

6. Learning from Other Industries

Digital solutions for organisations are a relatively young discipline when compared to other industries. As discussed, digital transformations of organisations have an extremely high risk of unplanned cost, missed goals and failure. So, what is the generalised recipe for success of comparable industries? It is surprisingly simple:

When products or services require a lot of money or effort to produce, it pays to create the right thing and quality at first attempt.

Success at first attempt does not imply design and production as a monolith. Delivering and validating intermediate versions is a very effective risk-mitigation strategy when the production process is complex and/or when new requirements or technology shifts are likely and would lead to a different product.

The most popular method of ensuring first-attempt success of the future product or service is *investing* in a *standardised model of the solution* that:

- captures the *essence of the future solution* with sufficient detail, precision and accuracy;
- enables *conclusive feedback* on the functionality, quality and acceptance of the future product;
- is much cheaper—and possibly much faster—to *build, validate and adapt* than the actual solution.

However, to provide significant value, the model needs to be complemented by two more facets:

- *digital tools* to build, share and evolve such models;
- *a process* to use tools and models effectively and efficiently in a project.

In order to avoid confusion, we will use the term *journey* instead of process when referring to making proper use of model and tools.

One industry that has perfected this method for over 500 years is *building construction*. More recent examples are *electronic circuit boards* or the *aerodynamics* of cars, boats, airplanes, etc. Let's look a these examples.

1—Building Blueprints, Standards and Codes

Over the centuries, many bridges had to crumble under weight and water pressure; many people had to die before the required knowledge, building statics, and tested design patterns for the popular technologies had emerged. Together, they ensure safe, effective, affordable and aesthetic buildings that last for centuries.

Building construction introduced precise, *true-to-scale blueprints* in the 15th century [9] in order to address precisely the issues discussed in chapter 4. True scaling is essential because it allows accurate measurement between any two points of the blueprint and comparison with its physical realisation.

Construction blueprints changed everything. They are able to convey both the big picture and accurate details through a standardised visual language that everybody in the industry understands. The disciplines responsible for producing accurate construction blueprints are *architecture* and *drafting*.

Before the introduction of blueprints, future building owners provided their general idea and their wishes («requirements») verbally, often altering them again and again. From these, an architect created informal sketches which were not to scale. During construction, the architect huddled with the craftsmen every morning to discuss the works of the day. New castles and cathedrals took decades to complete, also because they were riddled with technical and quality problems and thus with exploding costs.

Later, building codes and standards created a dedicated vocabulary for effective and efficient communication among stakeholders. New blueprinting symbols and practises were introduced; more and more rules to ensure safe and accessible solutions were defined.

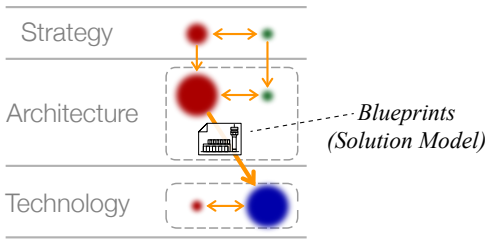


Figure 6-1 | Explicit «big picture» in building construction (green dots: technology experts)

Today, construction projects use sophisticated digital tools like CAD and BIM[4] to create *3-dimensional models of buildings*. These models are carefully validated and improved by many experts in order to eliminate errors, unplanned costs and delays once construction starts. People work on such models sometimes for years before, eventually, excavators set to work. Blueprints allow for a clear, geographic separation of planning, costing, offering, tendering and execution.

2—Electronic-Circuit Models

Electric circuits of the early 19th century were manual trial-and-error affairs. By the advent of semi-conductors in the 1950s, solid engineering practises and *modelling of electrical and electronic components and circuits*

[4] Computer-Aided Design, Building Integration Model

had developed. Soon, software-simulation applications allowed testing and improving circuits without any physical components and soldering. Today, multi-layer circuit-board layouts are generated and optimised for rapid production, high performance and low cost. It no longer takes an electronics engineer to design and test a circuit and a corresponding circuit board. The model of a printed circuit board (PCB) can be uploaded to manufacturers via internet. These use fully automated production processes to create just a few units, or hundreds or millions of units. Small PCBs cost a few cents. The involved tiers are very similar to figure 6-1. Modelling and simulations happen at the middle tier, which, in turn, are based on models of electrical properties of actual components and circuits at the bottom tier.

3—CFD with 3D Solid Models

Good aerodynamics have always been crucial for rockets, airplanes and racing vehicles. Wind tunnels soon became indispensable. However, they were expensive to operate, and optimising the aerodynamic performance of a vehicle was a matter of experience, trial and error. Smoke and woollen threads were rather primitive tools visualise wind flows and turbulences. *CAD models* for solid 3-dimensional shapes, again, changed everything. Together with computational fluid dynamics (CFD), those shapes can be optimised with respect to their aero- or hydrodynamic performance with software alone. Hundreds of versions of body designs are thus tested and optimised to improve the operation range of electrical vehicles (EVs). Actual wind tunnels are only used to validate promising computed results. Again, the involved tiers are very similar to figure 6-1. Modelling and CFD happen at the middle tier.

Lessons from Building Construction

The construction industry offers a close-enough analogy for digitalisation to lean on and learn from:

- a building is always built for the *purpose* of facilitating *processes*, e.g. of living, education, manufacturing, storage, transportation, etc.;
- the actual building is made from *base resources* like rocks, concrete, wood, metal, tiles, pipes, etc.;
- the *semantic elements* of a building (e.g. floors, walls, roof, elevators, stairs, windows, power and piping networks) and their relationships and interactions, make up *higher-level resources*. Each one has specific meaning, functions, qualities and individual characteristics; together, they follow established rules and practises (i.e. standards and building codes).

Architects are highly-trained and experienced professionals who fulfil a pivotal role in the construction (or renovation or conversion) of buildings. They:

- act as *mediators* between the future building owners with their needs, the craftspeople and experts;
- know and understand the capabilities of the base resources;
- study and come to terms with the processes that the future building must support;
- master a *highly standardised, visual language* for accurate, detailed, true-to-scale blueprints of the future building and its components;
- use the blueprints to ensure together with all stakeholders that the future building fulfils all requirements and is feasible (technically, financially).

So, what can we learn for the digital transformation of organisations?

- Requirements must be complemented by *explicit models of the solution* at the Organisation tier.
- A shared «blueprinting» or modelling language must facilitate the collaboration of the organisation and its technology partners.
- The organisation must be responsible for: ❶ content, completeness and quality of the solution models at the Organisation tier; ❷ minimise and manage business and organisational risk.
- The technology partners must be responsible for: ❶ the completeness and quality of the solution implementation; ❷ the choice and use of suitable technologies at the Technology tier; ❸ minimise and manage technical risk.
- A new, professional mediator role (e.g. *process architect*) must: ❶ support the organisation in capturing and understanding subject areas and processes in their current shape (*as is)* and future shape *(to be*); ❷ ensure with the *technology experts* that the future solution fulfils all requirements and is feasible.

In summary, there are three main take-aways for digital-transformation projects:

- shift the focus of the organisation's activities from the Technology tier to the Organisation tier;
- create a sound «big picture» of the future solution before starting to even look for solution partners;
- introduce a new role dedicated to supporting organisations in creating a high-quality «big picture».

Let's look at the first two recommendations in more detail.

Shift Focus to the Organisation Tier

Comparing the three examples presented above, the lesson to learn is that all of the problems exposed by the case study could have been avoided *proactively* at the Organisation tier. And since the Organisation tier is the organisation's own domain, here's the **disruptive change to the narrative** of digital transformations:

Organisations must take full responsibility and leadership of their own digital transformation—instead of trying to offload it to technology partners and hope for the best.

Nobody in the world can know better what an organisation is, how it works and what it needs to evolve to than the organisation itself. Technology partners cannot help here—instead they must focus on the best use of technology. The corollary is that technology partners cannot be held responsible or even accountable for the needs, change and adaptation of the organisation at the Organisation tier. Only the organisation is in a position to be successful here in the first place.

If this message sounds somehow familiar, then this is because the same kind of responsibility shift was proposed to humanity 250 years ago:

«Immaturity is the incapacity to use one's intelligence without the guidance of another»—Kant [10]

Now, let's examine what this means to the organisation. It certainly looks like a heavy burden. But despair not, the Semantic Eye Framework was specifically created to fill the abyss that just opened. Taking the three industries presented above as successful role models, we can sharpen the picture of the future.

As identified on pages 6 and 7, the two weakest points of digital transformations are ❶ the lack of a thorough, documented, shared understanding of subject areas and processes; and ❷ the organisations' focus of activity on the Technology tier. Figure 6-2 (right) shows what the distribution of activities across the transformation tiers and the collaborations between organisation and technology partners should look like.

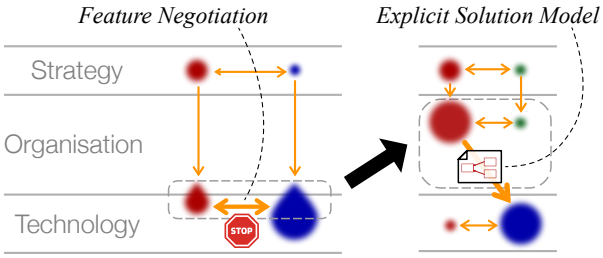


Figure 6-2 | Shifting the focus of activity to the organisation tier

Now, let's see what the «*Explicit Solution Model*» in figure 6-2 should be in order to solve problem ❶.

First, Create a Sound «Big Picture»

Starting a transformation at the technology tier («we need to take advantage of AI»)—or doing only that—covers only the *how* and is a recipe for trouble. First, projects need to clarify the *why* and the *what.* The «homework» of acquiring a deep understanding is an absolute prerequisite before even starting to look for potential realisation partners. The *early inclusion of technology experts* is required to reduce the risk of conceiving a solution that is not feasible or too expensive. Technology experts should be independent of future solution partners.

Solution blueprints should be the basis for all cost and effort calculations, tenders, negotiations, and for actual realisation and quality control. This may sound obvious, however, for the lack of blueprints, the digitalisation industry often relies on the requirements instead.

Let this sink in for a second: software projects in the highly-paid enterprise market often use descriptions of the clients' needs for planning and estimate purposes rather than an explicit blueprint for the solution they intend to deliver. In other words, each of the involved experts creates his or her own *ad-hoc solution blueprint* in their head and uses that as a base for the task at hand. Given the levels of complexity discussed above, nobody should be surprised for a minute if estimates are factors off reality; if projects take 2, 3 or 10 times longer to complete than planned; if solutions overlooked essential elements or dependencies.

So, creating solutions models that make scope and complexity of digital transformation projects explicit must be of great value.

The question that remains is: what should be the principal subject or focus of such solution blueprints?

7. Organisational Processes

Organisational processes are arguably the most frequent and relevant subjects of digital transformations. They are part of the Organisation tier. The very nature of organisational processes means that:

Every organisation has organisational processes.

Since organisational processes are so important, here's a more formal definition, based on [11], than on page 4:

An organisational process consists of a set of activities that are performed in coordination in an organisational and technical environment. These activities jointly realise organisation-level goals.

The definition is illustrated in figure 7-1. In the following, let's use simply «process» for organisational process unless this were to create ambiguity.

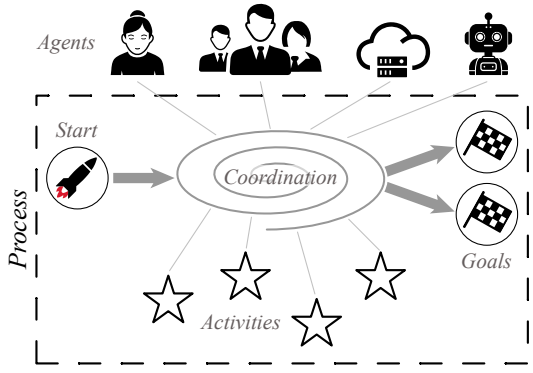


Figure 7-1 | What is a process?

Organisations maintain viability by providing products and services to consumers and partners inside and/or outside the organisation. Consumers and partners derive sufficient value from the products and services to reciprocate in ways that sustain the organisation.

If your organisation provides products and services through coordinated interactions of humans, then it relies on organisational processes. Most goals in planning, management or control are met through processes: sales, orders, billing, logistics, complaints, maintenance, human resources, projects, training, events, etc. ISO 9000 and many other commercial standards have processes at their very foundation.

Organisational processes are a great instrument to convey a coherent and complete picture of what an organisation is and does.

Organisational processes are an end-to-end description of where and how an organisation creates value.

8. Why another Framework?

Before we take a look at the actual Semantic Eye Framework, let's turn to the question of why a new framework is needed in the first place.

Even inside the transformation industry catering to organisations and their processes, most people seem unaware of just how *immature* the process methodology and tools are in comparison to other disciplines. Meanwhile, the tools of the latter have filtered down to small businesses: Order a new dining table from your local furniture maker today and expect it to be designed—and possibly manufactured—using some digital tools and automation. But start a project to digitally enable and automate the Word-, Excel- and email-based value chain of your company, and be sure to see highly-paid transformation experts use whiteboards, sticky notes, Miro®, Excel®, Powerpoint®, Jira®, etc. to capture and come to terms with your processes. They simply do not have, know or use an equivalent to construction blueprints, CAD, BIM, CFD, Unreal Engine®, etc.

From the air-travel case study, a first reaction could be to blame Agilicca as incompetent or unprofessional. However, they are a construction company with no prior experience in airports—let alone in airport processes. So, was it the fault of the city of New Venice? Knowing nothing about air traffic, they clearly should have studied existing airports and air transport first. But how, then, should they have captured the thousands and thousands of pieces of the commercial-aviation puzzle to get a clear and coherent picture? For buildings and infrastructure, the real world has the tools. But how to capture and optimise processes in equal detail and in a form that is understandable by all stakeholders?

Literally hundreds of books have been written on the subject of processes in general and of process management in enterprise contexts in particular (e.g. [11], [12], [13]). But try to apply their best practices to your organisation and its ways of doing «business», the truly relevant chapters will be either strangely vague or missing altogether. This leaves organisations stranded or in the hands of «process consultants» and IT suppliers offering to design and build a software solution—but who neither master the subject areas nor provide solid process methodology and tools. So, there you have it: the malaise that sparked the Semantic Eye initiative in the first place.

9. The Semantic Eye Framework

This chapter introduces the main objectives of the framework and then provides a «nutshell» overview of its elements.

High-Level Objectives

The framework strives to:

- make the digital-transformation of organisations, subject areas and processes a methodical, efficient, satisfying and, above all, *successful* affair;
- *empower organisations*—and more specifically, domain experts and process owners—in gaining ownership of their processes and in emancipating their organisations from nerdy technology partners;
- overcome the building of immature, *ineffective* software in projects as a *slow* method of domain analysis, conceptual alignment and *knowledge transfer*.
- establish a *shared language* for unambiguous description of processes and for effective collaboration among all process stakeholders, particularly between domain experts and technology experts;
- encourage the creation of *aligned end-to-end process blueprints* (models) at the organisation tier covering all activities irrespective of whether they will be *manual*, *assisted* or *automated* in the future.
- provide a high-quality *digital modelling workbench* to organisations, which is on par with state-of-the-art, AI-assisted software-development tools.

Key Areas

The Semantic Eye Framework addresses five key areas (figure 9-1) that build on each other and offer increasing value to projects.

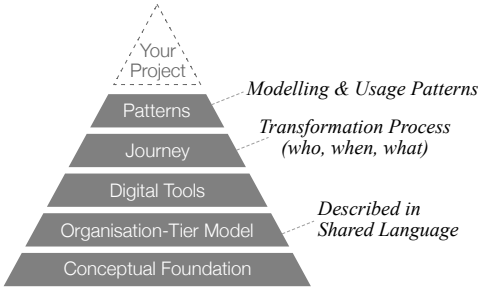


Figure 9-1 | Key framework areas of concern

The remainder of this chapter climbs the pyramid from *bottom to top*.

Conceptual Foundation

The conceptual foundation of the framework consists of multiple elements of which the *Digital-Transformation Tiers* from page 4 and the *Semantic Compass* are the most important. Both are frameworks within the Framework.

The semantic compass (inspired by [14]) provides orientation and guidance (hence «compass») on how to get from the vague idea for a product («Air-transport for 6 million annual visitors») to a working solution (buildings, infrastructure, a people organisation and processes). It spans a two-dimensional area where the vertical axis ranges from *abstract* at the top to *detailed* at the bottom, and the horizontal axis ranges from *vague* on the left to *precise* on the right. Natural-language is very vague and requires a lot of care and redundancy by both speakers and writers in order to avoid ambiguities and thus errors and misunderstandings.

In typical digital transformations today, most of the results are either documents in natural language, informal diagrams or program code. This results in a massive *precision gap* that software-development teams need to jump (figure 9-2, illustrating a classic, non-agile setup). And since domain experts are unfamiliar with the resulting program code, they cannot assist in the correct translation from natural language to software artefacts. All they are left to do is finding deviations («this is not what I meant») manually during software testing, which is both tedious and incomplete.

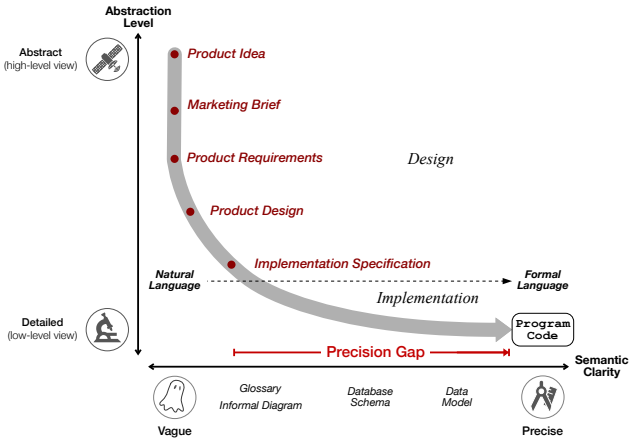


Figure 9-2 | Semantic Compass: details first (reality)

Construction blueprints, on the other hand, especially when they are created using CAD software, are precise and unambiguous; they prevent misinterpretation and can contain arbitrary levels of detail. To reap the same benefits, the Semantic Eye crosses the precision gap as early as possible by making the main deliverables more

precise and more formal (figure 9-3). This enables early digital-tool support and automated quality checking (highlighted in green in the figure).

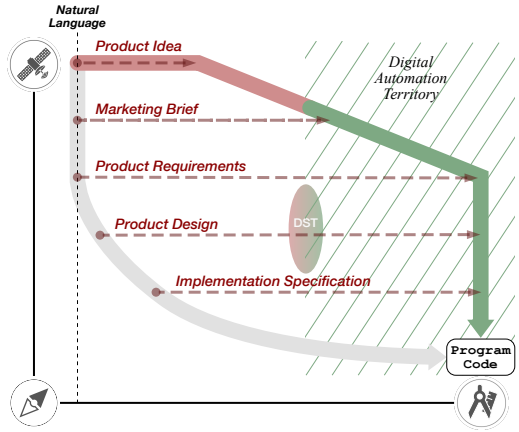


Figure 9-3 | Semantic Compass: precision first
(DST will be introduced on page 17)

The Organisation's «Cosmos»

Chapter 6 presented three types of highly successful conceptual models and their tools and recommended a strategy for sustaining success.

Following this, at the heart of the Semantic Eye lies the «Cosmos», a *conceptual organisation-tier model*, which consists of three fundamental parts that make up, describe and later govern the organisation and its activities:

- the *organisation* itself;
- the relevant *subject areas*;
- the organisational *processes*.

But before we get into the details of the organisational Cosmos, we need to understand the role of the subject ares and their relationship to the real world. The part of the Cosmos that covers subject areas is called *World Model* (figure 9-4) because it captures the relevant elements of «daily life»; it unites virtual and material objects; it defines structural and behavioural rules; and it explains cause and effect. Eventually, that part holds most of the expert knowledge of a subject area. All processes share the same world model to ensure conceptual and logical consistency across all processes.

The world model of e.g. the New Venice airport consists of elements like passenger, baggage, flight, airplane, gate, departure, delay, check-in, last call, etc. (the last two represent concepts, not activities; the corresponding activities are in the process part). These elements are in specific relationships with one another.

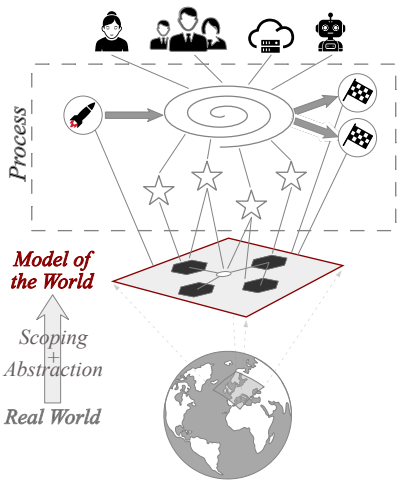


Figure 9-4 | The Semantic Eye Model of the World

An organisation's model-based Cosmos (inspired by [14], [15]) is a composite of sub-models that captures the *organisational solution space* (*cosmos*) as shown in figure 9-5. The process part of a Cosmos links the organisation model to a world model, which describes the relevant and/or affected subject areas.

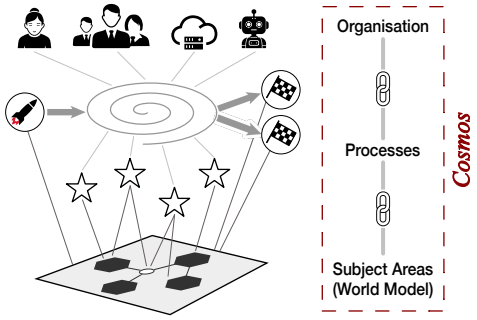


Figure 9-5 | The organisation's cosmos

As with the transformation tiers of figure 3-1, the three parts of the Cosmos mutually depend on one other and are not in a linear or hierarchical order like figure 9-5 might suggest. That is, organisations can shape *some* subject areas and processes, but, as is the case for commercial aviation, some subject areas come with predefined, immutable vocabulary, protocols or even processes that an organisation must either adapt to or remain excluded from.

The Cosmos belongs to the Organisation tier (identified in figure 6-2 as «explicit solution model»). The Technology tier should build its own models that reference the elements of the Cosmos models for traceability.

Shared Language

In order to let domain experts, process owners, partners, technical people, etc. talk *about* processes, the Semantic Eye provides a fixed set of named *meta elements* and rules for how they can be combined [5] to describe the Cosmos. Meta elements represent familiar things (*concepts*) and have clear meanings (*semantics*), like Role (e.g. a passenger), Form (boarding pass), Wor Object (baggage, plane), Task (Verify passenger ID), etc. The meta elements are the vocabulary of the formal Clarity language, a *language shared by everyone*. It enables unambiguous description and collaboration—and the implementation of tools.

Technically, a model is a structure of elements related to each other according to the rules of its metamodel. But unless this structure is materialised in some form, it is *invisible* to us. So, in order to study, build or change a model, at least one *view* is required. Views can be text, diagrams, forms, etc. Models like Cosmos rely on different types of views, each focusing on a specific aspect of the model (e.g. building floor plan, electrical schema, energy flows, etc.) or catering to specific roles or needs. Each type of view (e.g. type of diagram) has its own set of elements and layout rules and thus constitutes a language of its own. A Cosmos is accessible through about a dozen types of views, two of which are presented in figure 9-6 and figure 9-8. It also uses object views, state-based views, transaction views, snapshot views, etc.

The Cosmos is not limited to fully precise and detailed models and views (as per the semantic compass of figure 9-3). It is also able to integrate natural language, structured text and semi-formal diagrams like the Domain Storytelling (DST) notation [16]. Figure 9-3 locates the DST notation near its centre, which means that DST diagrams cover a medium level of detail and precision. The main strength of DST diagrams is their ability to act as a *bridge* between informal and formal descriptions by combining semi-structured language with precise shapes. DST diagrams are easily understood by all stakeholders («① Passenger steps up to Boarding Desk») and make for an invaluable base towards the creation of more formal models and diagrams.

Figure 9-6 shows a a semi-formal example (*domain story*) of a process for the boarding of a single passenger. Wherever possible, the elements of such diagrams are later referenced by elements of formal models for traceability.

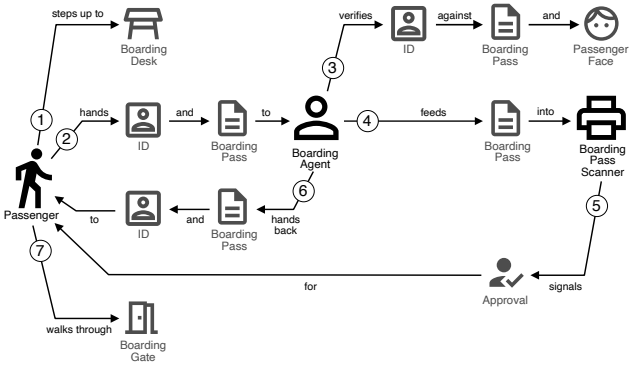


Figure 9-6 | Boarding process told as a domain story (created with the free Egon app [17]; agents are shown in black)

Declarative Processes

A formal Semantic Eye process is a *goal-driven, declarative* [15] definition of a coherent change to the world model. It is activated when a set of precise entry criteria is met, and ends when the goal criteria are met (depicted by figure 7-1).

A process breaks down into subprocesses, activities and tasks, each with the same characteristics as the process itself. Note that some activities and tasks of a process can or will never be automated with software alone, like lifting a bag onto the conveyor belt in figure 9-7 , task *Drop Bag*:

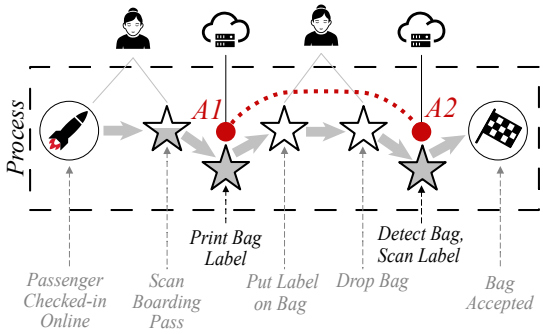


Figure 9-7 | Arrival at airport with check-in bag: the process creates a semantic bracket e.g. between the *fully automated* activities A1 and A2. *Scan Boarding Pass* is an *assisted* activity.

Semantic Eye processes are not functional input-output mechanisms like the processes of the popular BPMN ([18], [19]), thus they use *neither control flows nor information flows* to define their outcomes. Instead, they rely on

- formal entry and goal criteria;
- multiple exits (with non-overlapping goal criteria);
- decomposition (subprocesses, activities, tasks);
- dependencies among components (on goal criteria).

(if this list did not immediately make sense to you, you can safely ignore it for this document).

[5] Together these elements and rules make up the *metamodel*.

Explicit process models, reviewed by all relevant stakeholders (including technical experts), will tell an aligned and coherent process story. They let the technical folks make meaningful and traceable decisions and designs later.

The exact anatomy and characteristics of Semantic Eye processes are a framework within the Framework and are outside the scope of this document.

Tools

Formal Cosmos models are a prerequisite for the design and implementation of digital tools able to create *stringent* «blueprints» of organisations, process and subject areas.

Figure 9-8 depicts a basic structural world model of airport operations as (reusable and reused) components. There is a lot more detail to any of those components; they can be «zoomed into» in the Semantic Eye digital workbench to reveal more details.

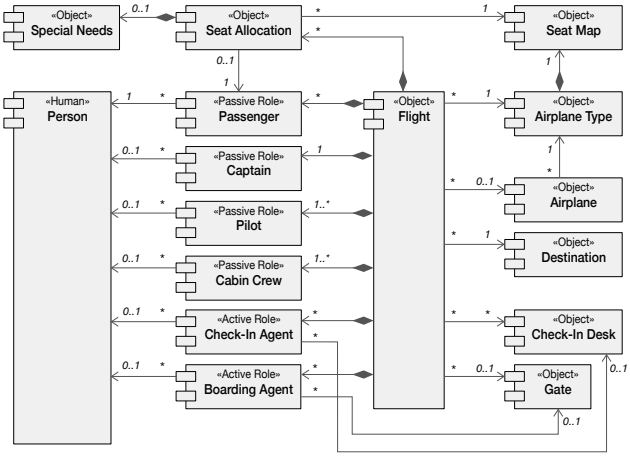


Figure 9-8 | Subject-area blueprint (shown as UML components; arrows show direction of dependency, not navigability)

A first integrated digital workbench is under development. It is based on the state-of-the art, open-source *Eclipse Theia* platform [20], which is able to run in an internet browser; it also fully supports *Visual Studio Code* extensions [21] and offers collaborative development like Google Docs® or Miro® boards (if you don't know what any of this means, just ignore it).

Theia comes with first-class LLM support that will be harnessed to assist the building of Cosmoses.

Transformation Journey

The framework promotes the collaborative and iterative building of a comprehensive, explicit Cosmos for organisations, with strong domain semantics, both before their digital transformation («*as is*») and after («*to be*»).

In addition to the *what* and *how* aspects discussed earlier, there is another major aspect to transformations: the timeline (*when*). More specifically, what transformational activities are carried out at which point in time. This defines the transformation journey of an organisation.

The Semantic Eye divides the journey into a *before* and an *after* choosing and onboarding software and technical partners. The journey consists of four phases, the last of which repeats forever, figure 9-9:

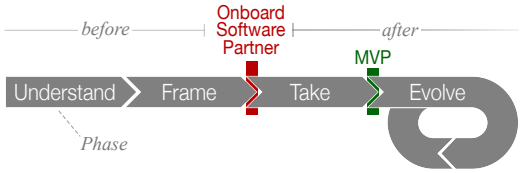


Figure 9-9 | Phases of the transformation journey

- *Understand*: Create a Cosmos of the solution «as is», carefully analyse it and the new strategy;
- *Frame*: define the organisational solution «to be» in a new or updated Cosmos;
- *Take*: Scope, design, implement and deliver an *MVP*, a *minimum viable product* (here: a *process*);
- *Evolve*: frame and take ever-more highly digitalised Cosmos and solution versions.

Each phase is best carried out in iterations towards its phase milestones. The phase names are inspired by photography, from where the «eye» was borrowed (also see page 3): analyse and understand your scene, frame it[6], take the first shot, then evolve it to a compelling visual story over time.

The underrated *Understand* and *Frame* phases are where organisations can make or break their digital transformation endeavour. The city of New Venice broke it because they did not use those phases as an opportunity to learn and to deeply understand how air travel works—and perhaps could work. In fact, they skipped both phases entirely. If nothing else, they could have studied airlines and airports and should have captured all the processes they could find with Domain Storytelling like in figure 9-6. Not only would they have been overwhelmed by the sheer number of different activities, but it would have provided them with a huge base of knowledge about the «*as is*» situation.

Together, the *Understand* and the *Frame* phases can take a third to half of the time spent on the «first 80%» (Pareto principle) of a digital transformation. Organi-

[6] Framing is the activity of getting the composition of a photography right, just before pressing the shutter button.

sations must take this time to get into the driver seat of their own transformation, and to avoid early, costly mistakes. The Semantic Eye Framework provides the tools for this.

Figure 9-10 gives an overview of the first three phases of the transformation journey including where intentional innovation should take place. Innovation is another framework within the Semantic Eye Framework; its whereabouts are outside the scope of this document.

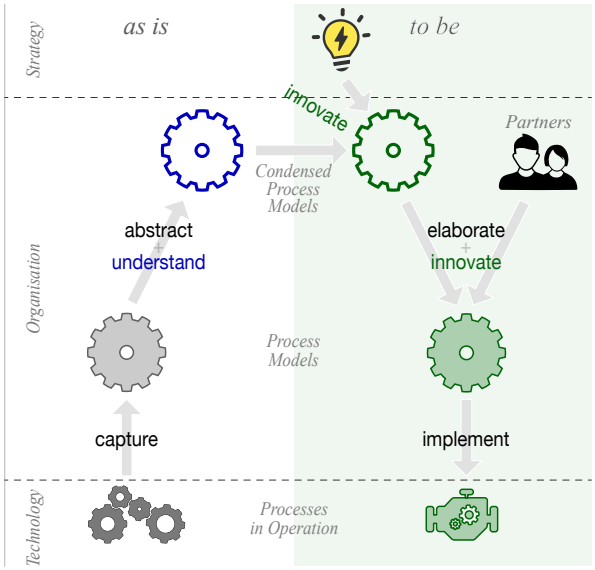


Figure 9-10 | Understand – Frame – Take (the vertical dimension of the Organisation tier is the abstraction level)

Patterns

Whenever languages and tools are being applied, best practises for their application emerge over time and can be documented as solution *patterns* [22]. Especially for formal languages like Clarity, the underlying *tool platform* can provide explicit support for patterns. LLMs[7] (see below) favour repeating patterns by their statistical nature, and can play a big role in suggesting suitable patterns and in applying them automagically.

The Role LLMs and AI

By making ChatGPT available to the public in 2022, OpenAI launched a raced that is possibly likened only by the race to the moon in the 1960s. Due to the strict, formal nature of programming languages, LLMs are particularly good at creating program code based on the billions of lines of code and on established coding patterns «out there». At the time of this writing, some AI agents—and the underlying LLMs—are capable of generating the entire code base for highly complex software services and applications.

Are software developers soon becoming an endangered species?—No, only those not taking advantage of AI. The others will produce much more functionality—and hopefully of better quality than manually—in a the same amount time. However, it is a reality that generative AI has the potential to create applications that no human can understand. Which is a problem given the non-deterministic nature of generative AI and its capacity for hallucination. Unfortunately, hallucination is a feature not a bug as it is an intrinsic (i.e. inevitable) byproduct of the statistical nature of LLMs. As a consequence, and even though AI agents can also generate test code that validates the results of generated production code, organisations must take extra care that their software partners create sustainable, high-quality software. Software that adheres to defined software-architecture principles, that can be understood and evolved by future developers with ease and confidence—probably with support from AI. In any case:

> *Organisations must maintain control of—and manage the risks associated with—their value chains, in which software is a first-class asset.*

How do developers actually create software with the help of AI? Assuming they interact with their code-generating infrastructure through some form of prompt (either text or diagrams or both), what is the basis for their prompts? It is in the hands of the organisation whether it continues hand-to-mouth feature negotiation as discussed on page 6, or whether development is based on a precise, aligned and coherent Cosmos developed with the Semantic Eye Framework.

The formal sub-models of the Semantic Eye Cosmos, as well as the semi-formal sub-models or even natural text lend themselves very well as input for LLMs and AI agents. Moreover, once there is a number of published Cosmos instances out there, LLMs can assist organisations in defining meaningful organisation models, processes models and world models, and in quality-checking them. However, to ensure correct and dependable models, organisations must be able to validate what AI agents are generating., i.e. they must be able to understand and validate every element of their Cosmos regardless of its source.

Let's not forget that AI is just a tool. Until further notice, an organisation must assume that damage or legal infringement or violation created by AI in its name will see the very organisation in charge. This is specially true with regards to regulations, data protection and system security.

[7] Large Language Models

10. How Organisations benefit from the Semantic Eye Framework

Chapter 7 established that, by definition, all organisations have organisational processes. Chapter 9 puts processes and process models at the centre of an organisation's Cosmos, which, in turn, is at the heart of the Semantic Eye Framework. It follows that:

Every organisation can benefit from the Semantic Eye.

Key Benefits of the Semantic Eye Framework

Here are some of the key benefits the Semantic Eye Framework offers to organisations:

- The framework gives organisations a playbook that can be studied, taught, discussed, challenged, adapted and improved over time.
- The expertise embedded in the framework, its goal-driven and methodical approach, and its tool-based application significantly reduce the risks of transformation projects from their start by aligning stakeholders and avoiding unforced errors.
- An organisation's Cosmos is a first-quality knowledge-capturing and -management instrument. It enables the organisation to describe, study, learn, plan, align and optimise subject areas and processes long before software is being built.
- The tools of the framework make capturing requirements and modelling current and future solutions a productive and enjoyable activity; they help detect logical and consistency errors early.
- The first comprehensive version of a Cosmos can be built without implementation partners pressing for new requirements, specifications and testing.
- The creation of an organisation's Cosmos eliminates the need for trial-and-error software as an inefficient tool to learn and formalise a subject area and its processes. The Cosmos can be (iteratively) improved much quicker and cheaper than solutions already cast in software or hardware.
- A standardised framework lets organisations share models and lessons learnt with other organisations, and lets them benefit from other models and lessons.
- The formal Cosmos models and their semi-formal and natural-language elements lend themselves very well to LLM learning and generation.

Where to Start?

Organisations that see potential benefits in using the Semantic Eye Framework (or parts of it) for their own digital transformation should consider these steps:

- Focus on the Organisation tier (page 13).
- Review the transformation journey on page 18 and start at phase *Understand*.
- Find two curious and motivated organisation members with good communication and analytical skills. They should study Domain Storytelling and then use it to methodically document the top three digitalisation-candidate processes of your organisation (strictly «*as is*»). First from a high flight altitude, then from a medium one.
- Review the resulting domain stories with all relevant stakeholders and align them on *one* version.
- Ensure your organisation's strategy is clear on goals, future products and services, and on reasons why digitalisation is needed.
- Get professional guidance and support from Semantic Labs (www.semantic-labs.ch).
- If your organisation's digital transformation is already under way or is even showing the usual problems (exploding complexity and cost, missed milestones, insufficient quality, unhealthy dependencies solution providers or individuals, exhaustion of team members, etc.): suspend the project immediately. Take time to explore, understand and evaluate the Semantic Eye Framework. Then decide to resume the project—or go back to square one and restart with the Semantic Eye.

Caveat

Make no mistake:

- Starting to apply the Semantic Eye Framework is a serious hill to climb. The involved members of the organisation will face a steep learning curve. But they will be rewarded with guidance, insight and clarity; and with the absence of chaos, uncertainty and unrealistic expectations. Other industries have climbed the same kind of hill with great success.
- Building a Cosmos is not a walk in the park: it often uncovers unclear or even chaotic responsibilities and work practises. These are best untangled by creating the Cosmos using agile practises.

11. Closing Notes

Formal Process Models are the Missing Link

Now, the Semantic Eye obviously did not invent processes. But why has all the advice on process management by myriad experts, books and tools mentioned in chapter 8 been ineffective? — Because either ❶ they do not propose process models at all, or ❷ they do not offer and/or rely on powerful and productive digital tools; or ❸ their models are not formal or good enough to create significant value. That last point is crucial: in construction, blueprints have been in use for centuries, but were hand-drafted until a few decades ago. Why were people willing to go to such lengths?—Because there was real value in the blueprints. The same holds for building a formal Cosmos for an organisation.

However, the digital age no longer accepts three years of training for draftspeople in meticulous hand-drawing of highly standardised blueprints when they can do it better and faster with CAD tools. From which the results can be presented to customers in 3D, with realistic colour, light and texture. For this reason, it is compulsory that the Semantic Eye offer powerful, productive digital tools. Otherwise, building the Cosmos is skipped; organisations continue to hope their IT suppliers will do the job they should be doing themselves; and the only dependable legacy of digital-transformation teams is program code.

Isn't the Cosmos just a UML Retake?

Most process-management books propose UML as a process-modelling language. But UML completely lacks specific support for organisation, process or subject-area semantics. Practitioners first need to figure out how they can create real value [23] with UML and get acceptance by teams and stakeholders. The Semantic Eye borrows many concepts from UML and makes organisation, process and subject areas first-class citizens with dedicated method and tool support.

Additional Reading

If you find this introductory white paper helpful and are looking for more information:

- The document «*Introduction to the Semantic Eye Framework*» is in the making; it mostly elaborates on the topics presented in chapter 9.
- Check out www.semantic-eye.org
- Contact the author at the-eye@semantic-eye.org

Contributing to the Semantic Eye

If you would like to become a reviewer or contributor to the Semantic Eye methodology or documents, to its tools, or otherwise, then please get in touch with the author at www.semantic-labs.ch.

Acknowledgements

This work has only been possible with the continued understanding, support and encouragement of my loving wife Jeannine; with the relentless requests for that first version of the Semantic Eye document by my friends and employers Marc-André and Kay; with the faith and generosity of the Swiss Blood Stem Cell organisation who allowed me to both draw inspiration and try out new ideas and concepts in their digital-transformation project over more than two years.

Also, this work would not have seen the daylight without other people's outstanding work and publications from which I was able to extract that basis of my own work. Notably Richard Mitchell who sparked «the whole thing» for me and from which I was able to borrow many of the main concepts around precision-modelling and transactional changes [14]. Kudos also to John Krogstie whose publications like [15] had huge influence on the current framework, and also for pointing me to the «Seven Rs» [3], which became a pillar of the Semantic Eye. Thanks also to John Cheesman and John Daniels for their early and influential work [23] in this field.

And last but not least to all my friends and colleagues who have patiently endured my ongoing enthusiasm and endless praise of model-based requirements engineering; and who, sometimes, without ever getting a feedback or thanks, have sparked another idea or feature of the Semantic Eye with an innocent question.

Why the Semantic Eye was called into Existence

«The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore, all progress depends on the unreasonable man.»
— George Bernard Shaw, Man and Superman

BIBLIOGRAPHY

1. **OMG: UML 2** — *Unified Modeling Language (UML) 2.5.1, 2017* (Internet Resource)
   Object-Management Group, https://www.omg.org/spec/UML/ (accessed Jan 15, 2025).
2. **Meyer, B. 2022** — *Handbook of Requirements and Business Analysis.*
   Springer, 2022, https://bertrandmeyer.com/2022/10/28/new-book-requirements-handbook/.
3. **Andersen Consulting 1997** — *Process Excellence Handbook* (Internet Resource)
   Andersen Consulting, https://www.scribd.com/doc/31730974/Process-Excellence-Handbook (accessed Jan 11, 2024).
4. **ASML** — *EUV Lithography Systems* (Internet Resource)
   https://www.asml.com/en/products/euv-lithography-systems (accessed March 25, 2026).
5. **Epic Games 2024** — *Unreal Engine 5.4* (Internet Resource)
   Epic Games, Inc., https://www.unrealengine.com/ (accessed May 9, 2024).
6. **Pohl, K. 2025** — *Requirements Engineering: Fundamentals, Principles, and Techniques, Second Edition.*
   Springer, Second Edition edition, 2025, https://link.springer.com/book/9783662518885.
7. **Schwaber, K. & Sutherland, J. 2020** — *The Definitive Guide to Scrum: The Rules of the Game* (Internet Resource)
   https://scrumguides.org/ (accessed Jul 12, 2023).
8. **Bennett, T. L. & Wennberg, P. W. 2005** — *Eliminating embedded software defects prior to integration test.*
   Crosstalk, https://citeseerx.ist.psu.edu/
   document?repid=rep1&type=pdf&doi=f54b4ab5eb4ab06e5ced6ae99d5e9a3d4ad6a23a.
9. **VDC Netzwerk FHNW 2024** — *Von der Bauhütte zu BIM: Was das Mittelalter mit digitalem Bauen zu tun hat* (Internet Resource)
   Fachhochschule Nordwestschweiz, https://vdc-netzwerk.ch/von-der-bauhuette-zu-bim/ (accessed May 9, 2024).
10. **Kant, I. 1784** — *Was ist Aufklärung?*
    Berlinische Monatsschrift,
11. **Weske, M. 2019** — *Business Process Management: Concepts, Languages, Architectures.*
    Springer-Verlag GmbH, Third edition, 2019,
12. **Dumas, M. et al. 2018** — *Fundamentals of Business Process Management.*
    Springer, 2018,
13. **Schmelzer, H. J. & Sesselmann, W. 2020** — *Geschäftsprozessmanagement in der Praxis.*
    Carl Hanser Verlag GmbH & Co. KG, (German only), 9th edition, 2020, https://www.hanser-elibrary.com/doi/10.3139/9783446467095.
14. **Mitchell, R., Graff, P. & Vladimir, B. 2004** — *High-quality modeling in UML* (Internet Resource)
    inferdata, https://www.inferdata.com/resources/whitepapers/HQmodeling.4.pdf (accessed Oct 30, 2025).
15. **Krogstie, J. 2012** — *Model-Based Development and Evolution of Information Systems: A Quality Approach.*
    Springer Science & Business Media, 2012, https://www.researchgate.net/publication/230636097_Model-Based_Development_and_Evolution_of_Information_Systems_A_Quality_Approach.
16. **Hofer, S. & Schwentner, H. 2021** — *Domain Storytelling.*
    Addison-Wesley Professional, 2021, https://domainstorytelling.org/.
17. **Modeler, T. D. S.** — *Egon.io* (Internet Resource)
    https://egon.io/ (accessed Nov 19, 2025).
18. **OMG: BMPN 2.0** — *Business Process Model and Notation (BPMN) 2.0.2, 2012* (Internet Resource)
    Object-Management Group, https://www.omg.org/spec/BPMN/2.0.2/ (accessed Jan 21, 2025).
19. **Aagesen, G. & Krogstie, J. 2015** — *BPMN 2.0 for Modeling Business Processes* (Internet Resource)
    Research Gate, https://www.researchgate.net/publication/283772827_BPMN_20_for_Modeling_Business_Processes (accessed Dec 14, 2023).
20. **Eclipse Foundation** — *Eclipse Theia Platform* (Internet Resource)
    https://theia-ide.org/ (accessed Oct 30, 2025).
21. **Microsoft** — *Extensions for Visual Studio Code* (Internet Resource)
    https://marketplace.visualstudio.com/vscode (accessed Oct 30, 2025).
22. **Gamma, E. et al. 1995** — *Design Patterns: Elements of Reusable Object-Oriented Software.*
    Pearson Deutschland GmbH, 1995,
23. **Cheesman, J. & Daniels, J. 2001** — *UML Components: A Simple Process for Specifying Component-based Software.*
    Addison-Wesley Professional, 2001,