Informed Governance of Enterprise Transformations

F. Harmsen[1,2], H.A. Proper[1,3], and N. Kok[1]

[1] Capgemini, Utrecht, The Netherlands
[2] Maastricht University, Maastricht, The Netherlands
[3] Radboud University Nijmegen Nijmegen, The Netherlands

Abstract. Modern day enterprises are confronted with a variety of challenges, forcing them to continuously transform themselves to better meet these challenges. The diversity of the challenges and the resulting desires to transform (parts of) the enterprise, make it desirable to align all required and desired transformations in such a way that they complement each other rather than nullifying, or even undermining, each other's effects. Therefore, mature governance of these enterprise transformations is absolutely crucial. We will argue that this requires a *transformation authority*, being the organisational function which is responsible for the governance of enterprise transformations. In this chapter, our driving interest is the implementation of mature *transformation authorities*.

Judging whether a portfolio of enterprise transformations is well aligned requires insight into the desired overall result, as well as the planned and *achieved* effects of the individual transformations. This is what we refer to as *informed governance*. In this chapter we will position the discipline of *enterprise architecture* (referring to the architecture of the enterprise, and not just enterprise-wide IT architecture) as the core means to achieve informed governance. We will argue that mature governance of enterprises transformations presupposes the use of enterprise architecture to direct the portfolio of transformations.

Our discussions we will be based on theories from management science, as well as experiences from our own industrial practices. We will also discuss two cases of enterprises involved in the implementation of *transformation authorities* and use these to further refine our theoretical model.

1 Introduction

As a result of developments such as globalisation, the fusion of business and IT, the introduction of new technologies, novel business models, et cetera, enterprises are confronted with an increasing variety of options to deal with an ever faster changing environment. This results in a need for enterprises to be able to innovate, and to adapt themselves quickly to these changes in the environment, as well as a desire to proactively exploit these developments in an attempt to create new business opportunities. As a result, modern day enterprises are confronted with several challenges driving them to continuously transform themselves to put them in a position where they are better equipped to meet these challenges.

The diversity of the challenges and the resulting desires to transform (parts of) the enterprise, make it necessary to align all required and desired transformations in such a way that they complement each other rather than nullifying, or even undermining, each other's effects. This also puts a major challenge on an enterprise's management to make the right decisions at the right time and ensure that these decisions are translated into the right actions. At the same time, since enterprise transformations are executed in terms of projects, one needs to ensure that these projects comply to the decisions made. In practice, this proves to be a difficult task indeed. Even more, different stakeholders and/or problem-owners will have a different perception of the necessary changes and their priority. Unless properly governed, chaos will result. Mature governance of enterprise transformations is therefore absolutely crucial, requiring a dedicated *transformation authority* as the organisational function responsible for the governance of these enterprise transformations. In this chapter, our driving interest is the implementation of mature *transformation authorities*.

Judging whether a portfolio of enterprise transformations is well aligned, requires insight into the desired overall result as well as the planned and *achieved* effects of the individual transformations. This is what we refer to as *informed governance*. In line with [1], we position enterprise architecture[1] as the core means to achieve *informed governance*. This will be elaborated upon in Section 4, where we will argue that mature governance of enterprise transformations presupposes the use of enterprise architecture to direct the portfolio of transformations.

In our discussions we will take both a theoretical perspective, basing ourselves on theories from management science [2, 3] and cybernetics [4, 5] as well as a practical perspective based on experiences from industrial practice. More specifically, we also discuss two cases of enterprises aiming to implement *transformation authorities*. We will use these cases to further refine our theoretical model.

The remainder of this chapter is structured as follows. In Section 2, we start with a brief exploration of drivers for organisations to transform themselves. We will see how these drivers may actually pull the enterprise into different directions. Stakeholders with a stake in the outcome of the transformation, may want to direct the enterprise's transformation in different directions based on the discussed drivers. This begs for the implementation of an *transformation authority*. Before we can properly discuss the concept of a *transformation authority*, we need to define more specifically what we mean by *enterprise transformations* and their *governance*. Therefore, Section 3 provides a theoretical exploration of these concepts. In Section 4 we then position enterprise architecture as a necessary means for informed governance of portfolios of enterprise transformations. Using these definitions, Section 5 then identifies the requirements to be put on a *transformation authority*, as well as the processes involved in its maturation.

With our, initial, theoretical framework in place, we then proceed in Section 6 and 7 by discussing two (anonymised) cases drawn from industrial practice involving the implementation/maturation of a *transformation authority* in a pre-existing large organisation. In Section 8 we provide an analysis of the two cases in relation to the initial theoretical framework. An important conclusion from this analysis will be that the implementation of a transformation authority requires a broad maturity framework taking several important aspects into consideration that may lead to blockages during its imple-

[1] We understand enterprise architecture as the architecture of the enterprise, and not as a synonym for enterprise-wide IT architecture

mentation, and maybe even lead to erosion of already achieved results. This leads to the introduction of a refined theoretical framework in terms of a *transformation maturity framework* (TMF) in Section 9.

2 Drivers for enterprise transformation

This section is concerned with a brief exploration of the drivers which may trigger organisations to transform themselves. These drivers are likely to pull the enterprise into different directions. Stakeholders with a clear stake in the outcome of the transformation, may also want to direct the enterprise's transformation in different directions based on the drivers discussed. Without an effective governance mechanism making a clear univocal choice for future direction, and ensuring that the transformation stays on course. The discussed below is based on a more elaborate discussion provided in [1].

2.1 Keep up or perish

Enterprises face many changes, such as mergers, acquisitions, innovations, novel technologies, new business models, reduced protectionism, de-monopolisation of markets, deregulation of international trade, privatisation of state owned companies, increased global competition, etcetera. These changes are fuelled even more by the advances of eCommerce, Networked Business, Virtual Enterprises, Mashup Corporations, the availability of resourcing on a global scale, et cetera [6, 7, 8, 9]. These factors all contribute towards an increasingly dynamic environment in which enterprises want to thrive.

2.2 Shifting powers in the value chain

Clients of enterprises have become more demanding. A shift of power in the value chain is occurring. Clients have grown more powerful and demand customised, integrated and full life-cycle products and services. For example, rather than asking for a "printer", they require a guaranteed "printing service". Even more, customers have a tendency to ask for integrated service offerings. Rather than treating booking of a flight, a hotel, and a sight-seeing trip as separate services provided via separate outlets, customers opt for one-stop shopping. A shift from basic products to full services.

The creation and delivery of such complex products and services requires additional competencies which may not be readily available within a single (pre-existing) enterprise. In this pursuit they increasingly engage in complex product-offerings involving other parties, leading to cross selling and co-branding. To ensure the quality of such products and services, a high level of integration and orchestration between the processes involved in delivering them is required.

2.3 Comply or bust

In the networked economy, governance of enterprises becomes increasingly complex. One sees a shift in governance from individual departments within an organisation, to the entire organisation, and lately to the organisation's value web. Management does not only have to worry about the reputation of their own organisation, but also about the other organisations in their value web.

How daunting the latter might be can be illustrated by real life examples, such as a large shoe manufacturer who outsourced the production of shoes to another company, to only discover at a later stage that the latter made use of child labour. Although the latter company was not part of the shoe maker's own organisation, their reputation was still damaged, threatening their survival on the market-place.

Corporate governance is not only an issue to an organisation on its own, but also a major concern to society as a whole. As a result of undesired and uncontrollable effects of the increased socio-economical complexity and interdependency of organisations, services, products and financial instruments. Recent examples of such side-effects are the well-known Enron scandal, as well as the sub-prime mortgage crises. To control and/or prevent such effects, new legislation has been put in place to better regulate enterprise practices. An example being the Sarbanes-Oxley Act [10] forcing enterprises to increase the quality of their governance and appropriateness of audits.

2.4 Achieving competitive advantage

Enterprises try to achieve and maintain a competitive advantage. In order to do so, they need to choose an optimal strategic position. Porter [11] distinguishes four basic units of competitive advantage: product development, purchasing, operation, and distribution of products or services. Performing these four activities better than rivals do is called operational excel lence. Enterprises can, however, also opt for other ways of distinguishing themselves from their rivals. In [12], Treacy and Wiersema argue that enterprises should try and focus on one of the three disciplines of added value: product leadership, operational excel lence and customer intimacy.

In the recent past, enterprises needed to excel only in one of the above areas to be successful, and meet industry standards on the other areas [12]. Due to the network economy and globalisation, there is a growing need to excel in a minimum of two areas (or at least in one and significantly increasing in the other areas).

2.5 Making technology the business differentiator

The evolution of information technology brings an abundance of new opportunities to enterprises. Technology becomes part of almost everything and most processes have become IT reliant, if not fully automated. The technological evolutions confront enterprises with the question of which technologies are relevant to the enterprise? Which technology should be replaced and which technology could be of use for developing new products (or services) of to enter new markets?

2.6 Excel or outsource

Increasingly enterprises outsource business processes. Outsourcing of business processes requires organisations to precisely understand and describe what needs to be outsourced, as well as the implementation of measures to ensure the quality of the outsourced processes [13, 14, 15, 16].

In deciding on what to outsource and how to safeguard its quality, management needs insight into the extent to which processes can be outsourced, the risks that may need to be managed when doing so, as well as the interdependencies within the outsourced processes and between the outsourced processes and the retained organisation.

Conversely, organisations with a strong tradition in a certain business process may decide to become industry leader for such processes. For example, processing of payments, management of IT infrastructure and logistics.

3 Governed enterprise transformation

In this section we provide a theoretical perspective on enterprise transformations and in particular the governance thereof. In doing so, we will base ourselves on theories from management science [2, 3] and cybernetics [4, 5], as well as our own experiences from industrial practice.

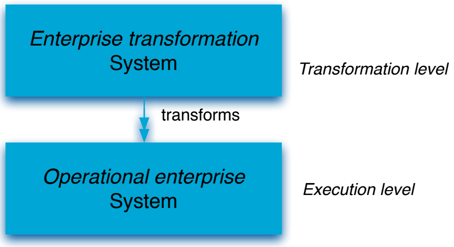

Fig. 1. Enterprise transformation

We regard an enterprise as an (open and active) system comprising a collective of actors, processes and technology which jointly engage in some *purposeful* activity. Being a system in the *general systems theory* sense of the word [4, 5], an enterprise can be divided into component systems (such as business units) as well as aspect systems (such as IT, business processes, et cetera).

An enterprise may evolve over the course of time. This evolution may be the result of a gradual change of the behaviour of individual elements in the enterprise, or it may be the result of a deliberate and conscious action. We define an enterprise transformation as the latter type of change, in other words, a deliberate and conscious action aiming to make changes to an enterprise. This is illustrated in Figure 1. At the execution level we find the operational enterprise concerned with "normal" operations (a *first order* system), while at the transformation level we find the enterprise transformation (a *second order* system).

Enterprise transformations may be triggered by several events. Management of an enterprise needs to make conscious decisions about the initiation and direction of enterprise transformations, balancing the desired benefits of the transformation in relation to its costs. Note: a special type of enterprise transformation would be the ‘undoing’ or ‘prevention’ of unwanted gradual change of the enterprise. More importantly, this also requires a conscious decision about the desiredness of the direction the ‘natural evolution’ takes and the need to counter or stimulate this. Furthermore, as argued before, when a series of transformations is executed, these transformations need to be aligned. The enterprise transformation and the operational enterprise may be sub-systems of the same enterprise, but they could equally well be part of different enterprises. In the first case, the enterprise is able to execute its own transformations, while in the second case an external party (e.g. a consultancy firm) is used to execute the transformation.

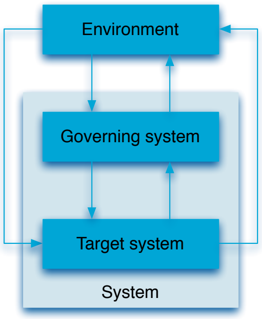

Fig. 2. The basic governance paradigm

According to [17], governance is “*the activity of [] controlling a company or an organisation*”. In management science this is embodied by the so-called governance paradigm [2, 3]. Figure 2, which is based on [2], depicts the basic governance paradigm. The governance paradigm involves three important assumptions:

1. there is some system, and not as a synonym to application system as is the case in software development. In the context of enterprise architecture, we are specifically interested in active systems [18], the target system, which interacts with its environment;
2. this target system needs to be governed;
3. there is another system, the governing system which does the actual governing.

The essence of the governance paradigm is that during the execution of a process (the *target system*) there is some kind of interaction with the environment (input and output), and that this process is controlled by some (internal) authority (the *governing system*) which monitors, and if necessary adjusts, the process to make sure the intended objectives are reached. Since an organisation is part of a larger system, the governing system also interacts with the environment to determine which services of products to deliver, to determine new opportunities and to determine changes in the environment.

When combining the governance paradigm with the view on enterprise transformations as depicted in Figure 1, we arrive at the situation as depicted in Figure 3. Both at the execution level and the transformation level, a distinction is made between the processes which are the *target system* of the governing processes, and the governing

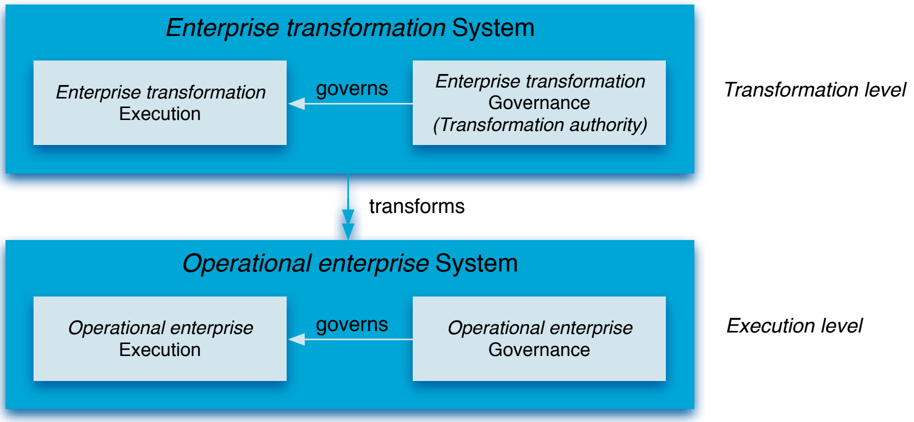
Enterprise transformation System
Enterprise transformation Execution
governs
Enterprise transformation Governance *(Transformation authority)*
Transformation level
transforms
Operational enterprise System
Operational enterprise Execution
governs
Operational enterprise Governance
Execution level

Fig. 3. Enterprise transformation governance

system itself. The *enterprise transformation governance system* constitute the (operational part of) the *transformation authority*. A transformation authority involves two important (sub)systems:

Process management – The management of the processes involved in the execution of the actual transformation.

Quality management – Management of the quality of the results produced by the transformation processes.

while the actual execution of the transformation can be thought of as comprising three core processes:

Assess – The *assess*ment (diagnose) of the problem/challenge the transformation seeks to solve/meet.

Aim – The identification of how the transformation *aim*s to solve/meet the problem/challenge (formulation/selection of the treatment).

Act – The *act*ing out of the actual transformation (performing the treatment).

This leads to the situation as depicted in Figure 4. Note: it is no accident that there are no arrows present. In general, the execution of the assess/aim/act processes will be highly iterative and cyclic in nature.

A comparrison can be made to Deming's [19] *Plan*, *Do*, *Check*, *Act* cycle for quality improvement. In terms of this cycle we would have the following mapping:

Assess – Involves Deming's notions of *Check*:

> *Measure the new processes and compare the results against the expected results to ascertain any differences*

and *Act*

> *Analyze the differences to determine their cause, and determine where to apply changes that will include improvement.*

Aim – Corresponds to *Plan*:

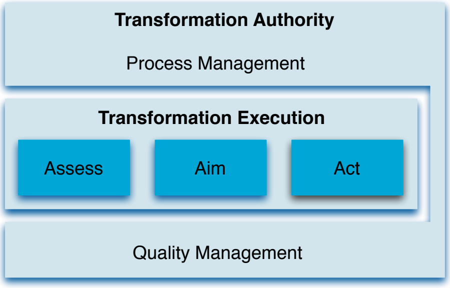

Fig. 4. Three perspectives on an enterprise transformation

Establish the objectives and processes necessary to deliver results in accordance with the expected output.

Act – Corresponds to *Do*:
Implement the new processes.

At an enterprise scale, transformations might be regarded at three key levels of granularity:

Project level – The level of specific projects having a clearly defined goal and time-frame in which to achieve this goal.

Program level – The level at which we consider several transformation projects contributing towards a larger overarching goal, still bound to a specific time-frame.

Portfolio level – The level at which enterprise transformation is regarded as a continuous collection of programs working towards the execution of the enterprise's strategy.

When taking these three levels into account, we end up with the situation as depicted in Figure 5. Note again: it is no accident that there are no arrows present. In general, the execution of the assess/aim/act processes will be highly iterative and cyclic in nature, and will even iterate and cycle between the levels.

4 Architecture as a means for informed governance

Several socio-economical and technological trends drive enterprises to transform themselves. As discussed before, the diversity of these challenges and the resulting desires to transform (parts of) the enterprise, make it desirable to align all required and desired transformations. Judging whether a portfolio of enterprise transformations is well aligned, requires insight into the desired overall result as well as the planned and *achieved* effects of the individual transformations. It also puts a major challenge on an enterprise's management to make the right decisions at the right time and ensure that these decisions are translated into the right actions.

As discussed in [1], architecture offers a means for management to obtain insight, as well as to make decisions about, the direction of enterprise transformations. This is what we refer to as *informed governance*. During the *assess* and *aim* processes (see Figure 5),

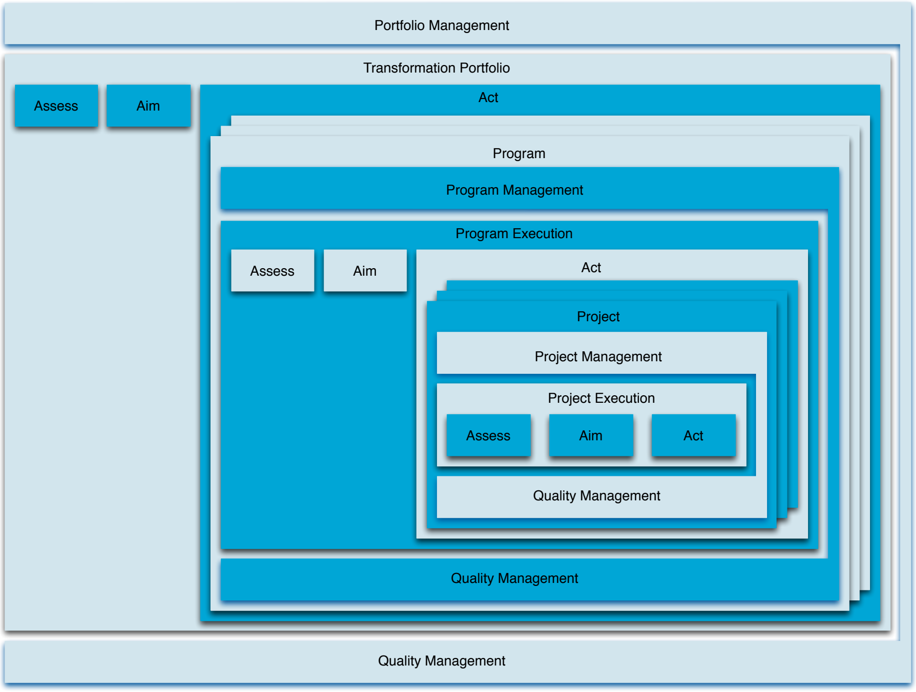

Fig. 5. Three levels of enterprise transformations

enterprise management needs insight into the way a transformation may improve the enterprise's ability to deal with these trends. Some concerns of enterprise management that need answers/insight:

- What is the rationale for this transformation? Will the transformation enable us to better deal/exploit with the socio-economic and technological trends?
- What are alternative transformation paths and their relative costs/benefits? What is the impact on current enterprise, its processes, structures, alliances, IT, et cetera? What are the risks during/after the transformation?
- What part of the enterprise will be impacted by the transformation?
- What are the relations and dependencies with other transformations/projects?
- When will the results of the transformation be effective?
- How sound is the business case for the transformation? What will it cost? How big are the benefits?
- What are consequences/opportunities of alliances with external parties or innovation networks?

More specific requirements on architecture as a means that follow from this (see [1]) are:

- Express/depict a coherent, comprehensive and concrete image of the desired future state(s) of the enterprise

- Provide a common language to a portfolio of changes/transformations of an enterprise.
- Identify a roadmap for the transformations needed.
- Distinguish between short-term solutions and long-term (structural) solutions
- Give a clear context and direction - limiting design freedom - to individual programs and projects that contribute to the desired overall transformation.
- Select available solutions and/or packages that are to remain or to become a part of the solution, whether in-house or sourced by a business partner.
- Enable traceability of design decisions from the strategic level via programs to specific projects.

The use of an enterprise architecture can also be likened to the use of a "dashboard" which allows the architect and stakeholders to steer the enterprise's transformation processes. When using this metaphor, the "*dashboard*" displays the *architecture* in terms of relevant aspects of the current state of the enterprise, its future direction and the desired states of the enterprise.

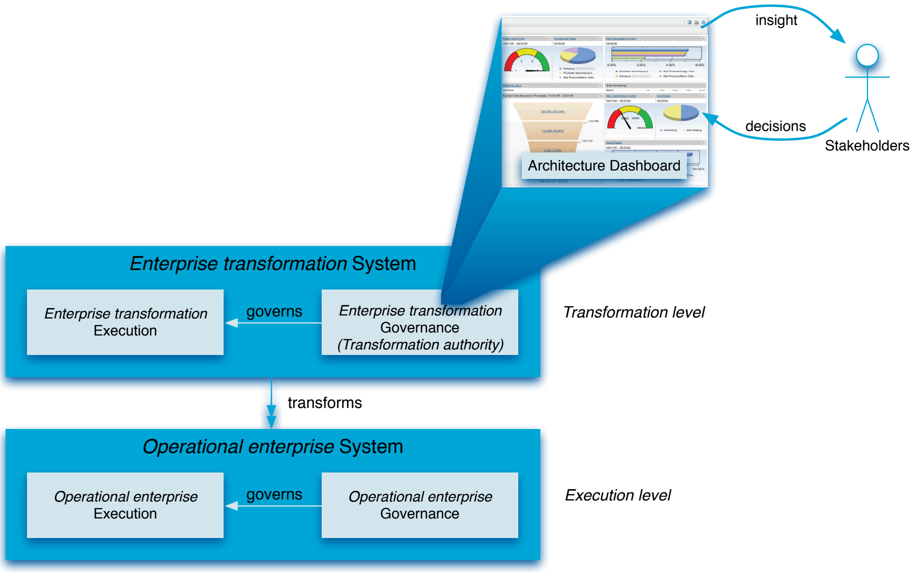
insight
decisions
Stakeholders
Architecture Dashboard
Enterprise transformation System
Enterprise transformation Execution
governs
Enterprise transformation Governance
(Transformation authority)
Transformation level
transforms
Operational enterprise System
Operational enterprise Execution
governs
Operational enterprise Governance
Execution level

Fig. 6. Enterprise architecture on a dashboard

Just as the selected/displayed speed, altitude and direction of an airplane is not *the dashboard*, but rather *displayed on* the dashboard, the dashboard is not the enterprise architecture. Analogously, it is the enterprise architecture, or rather a part thereof, what will be displayed on the dashboard. In addition, the dashboard may contain a report on the gaps between the current state and desired states, as well as its *operational performance* in terms of its current state. In an airplane, a "*dashboard*" may comprise of indicators (meters, lights, et cetera) and controls (levers, handles, pedals, and knobs). In the case of enterprise architecture as a means to govern transformations, the dashboard needs at least:

- *indicators* giving insight into:
  - the enterprise's current state,
  - the enterprise's current performance,
  - the enterprise's future (expected) performance,
  - the selected direction and progress of its transformation processes,
- *controls* allowing the transformation processes to be influenced:
  - the desired state of the enterprise,
  - plateaux of intermediary stages,
  - overall regulations.

The indicators may take the form of models, views, performance measurements, et cetera. The controls may take the form of (enforced) reference models, design principles, standards, et cetera. This is illustrated in Figure 6. More specifically, architecture is used during the *assess* and *aim* processes to analyse problems in the current situation and formulate the desired target situation. In line with [20] the former would involve the use of a *base-line architecture*, while the latter leads to the *target architecture*. The *target architecture* serves as input to the quality management (sub-)process of the realisation as it will conducted by the *act* process.

In [1], seven key applications for architecture as a means have been identified. In combination, these applications provide an instrument to make informed decisions as well as to ensure compliance of the transformation to these decisions, at several levels of specificity:

Situation description – Use architecture as a means for goal/cause analysis to investigate problems/shortcomings in an existing situation. This also involves the creation of a shared understanding (among stakeholders) of the existing situation.

Strategic direction – Use architecture to express (and motivate) the future direction of an enterprise, as well as investigate (and evaluate) different alternatives. This also involves the creation of a shared (among stakeholders) conceptualisation of the (possible) future directions, and shared agreement for the selected alternative.

Gap analysis – Use architecture to identify key problems, challenges, issues, impediments, chances, threats, etcetera, as well as make well motivated design decisions that enable a move from the existing situation into the desired strategic direction.

Tactical planning – Use architecture to provide boundaries and identify plateaux (intermediary steps) for the transformation of the enterprise towards the articulated strategic direction. In this context, enterprise architecture is used as a planning tool, making the realisation of a strategy more tangible.

Operational planning – Use architecture to give a clear context and direction for a portfolio of projects working towards the realisation of the first plateau as defined at the tactical planning level.

Selection of partial solutions – Use architecture as a means to select one or more standard solutions and/or packages that are to become part of the solution and/or decide to outsource an entire business process/service to another enterprise.

Solution crafting – Use architecture to create the high level design of an actual step in the enterprise transformation as it will be realised (and implemented) in the context of a specific project.

When adding the architecture to Figure 5 we end up with the situation as depicted in Figure 7, where a distinction has been made between an *enterprise architecture* and a *solution architecture*. The enterprise architecture is concerned with a longer term regulative perspective giving directions to a number of transformation programs, while a solution architecture is more concerned with specific choices pertaining to transformation projects within a single transformation program. Needless to say that the solution architecture should comply to the over-arching enterprise architecture.

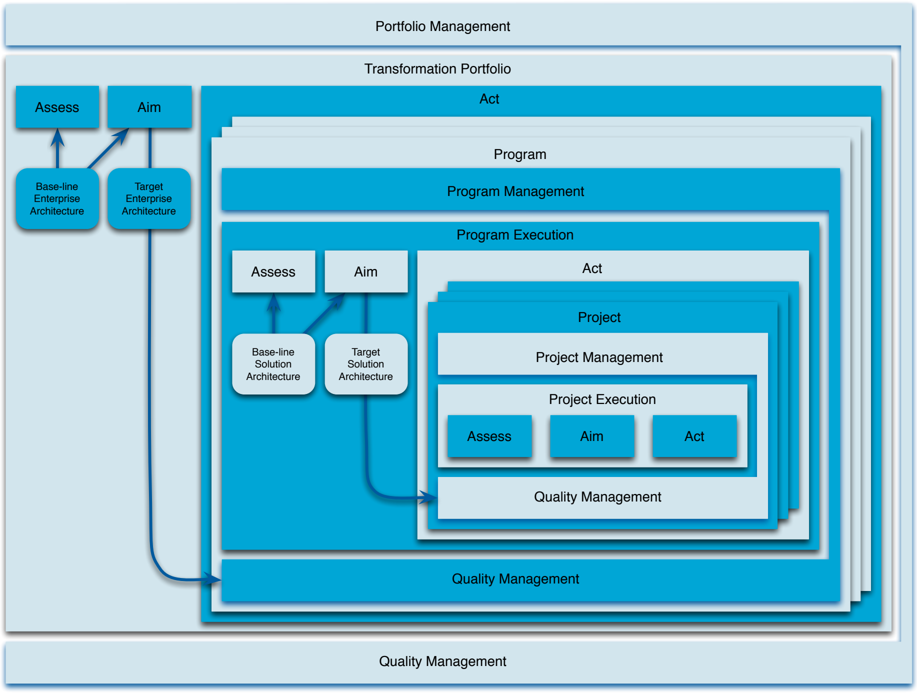

Fig. 7. The role of architecture

5 Mature transformation authorities

In this section we focus on the transformation authority as the function within an enterprise which is responsible for the activities involved in the governance of transformations. We start by identifying the requirements put on a *governing system* according to the governance paradigm (see Figure 2 and [2]):

- A governing system should have a goal with regards to the target system, providing guidance for steering.
- It should have information about the target system:
  - Its state (in the case of a moving object, this would include location, vector, speed and acceleration).
  - Environment variables influencing its state.
- It should have a (predictive) model of the behaviour of the target system, including its responses to steering signals.

  For example, to be effective, it needs an understanding of the current state (and evolution) of the enterprise, It also needs the ability to predict or gauge the effects of steering actions. Without this ability, the steering can only be reactive rather than pro-active.
- It should have enough requisite variety [4] to control the target system.

The latter requirement is one of the most challenging ones when implementing a transformation authority. First of all, it involves the challenge of involving senior management of the enterprise in the decision making process leading up to an enterprise/solution architecture. This typically requires a collaborative approach aiming to create as much understanding and commitment as possible. This commitment is more than needed during the actual communication/enforcement of these decisions to the transformation execution. During the execution of a transformation one typically has the inclination to favour short-term interests over longer-term interests especially when time pressures mount. In such cases, the *process management* and *quality management* streams (see Figure 4) of transformation governance are likely to clash as well. The decisions made in relation to the architecture tend to focus on the longer term interests/qualities of the portfolio of current/future enterprise transformations, while short-term pressures will favour faster and/or cheaper realisation from the perspective of a single project or program. These shorter-term versus longer-term clashes require commitment from senior management towards the original decisions. More details on strategies to achieve committed to decisions by senior management can be found in e.g. [21, 22, 23].

In addition to the trade-offs between longer-term and short-term interests, the directions formulated at the portfolio and/or program level (see Figure 7) may not always provide enough guidance to specific projects and/or may not be workable in a given practical situation. This also implies that a transformation authority should be willing to engage in the actual execution of the transformation, and be willing to learn how to make their directions (e.g. the architecture) more useful from a project/program's perspective.

More insight into the role of a transformation authority can be achieved by considering an enterprise transformation system from the perspective of the viable systems model [5]. The viable systems model takes the perspective that an organisation, such as an enterprise transformation system, comprises five key sub-systems:

System one (operations) – The primary activities of the organisation.

System two (co-ordination) – Communication & co-ordination channels allowing the activities of system one to communicate with each other.

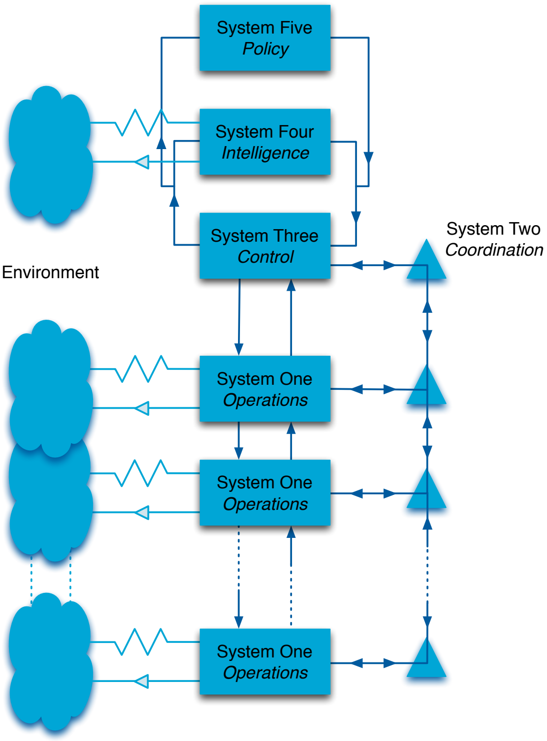

Fig. 8. Viable Systems Model

System three (control) – Structures and controls to establish rules, resources, rights and responsibilities of system one and two.

System four (intelligence) – Looks outward to the environment to monitor how the whole system needs to adopt to remain viable.

System five (policy) – Responsible for policy decisions within the organisation as a whole to balance demands from different parts of the organisation, and steer the organisation as a whole.

This is illustrated in Figure 8. When applied to an enterprise transformation system, we would have the following five systems:

System one (operations) – The projects and programs executing the enterprise transformation.

System two (co-ordination) – Communication & co-ordination between transformation programs and projects.

System three (control) – Structures and controls to establish governance of enterprise transformations, including the formulate and deployment of enterprise architectures.

System four (intelligence) – Monitor the environment for developments which require changes in the enterprise transformation system and/or the architectures used.

System five (policy) – Strategic management of the transformation authority. Monitors the effective functioning of the authority, making changes where/when needed in terms of structures and policies.

where, in terms of Figure 2 systems one and two are the *target system* and systems three, four and five are part of the *governing system*, i.e., the *transformation authority*.

Systems four and five are actually "new" in our discussions. As suggested by Figure 3, the core of the transformation authority's activities are formed by *enterprise transformation governance*, i.e., system three. System four and five essentially require a transformation authority to be able to change and improve itself, i.e., mature itself. This brings us to the introduction of the concept *maturation* of a transformation authority, the key driver of this chapter. In terms of Figure 3, this leads to the introduction of a *maturation level* on top of the *transformation* and *execution* levels. The result is illustrated in Figure 9.

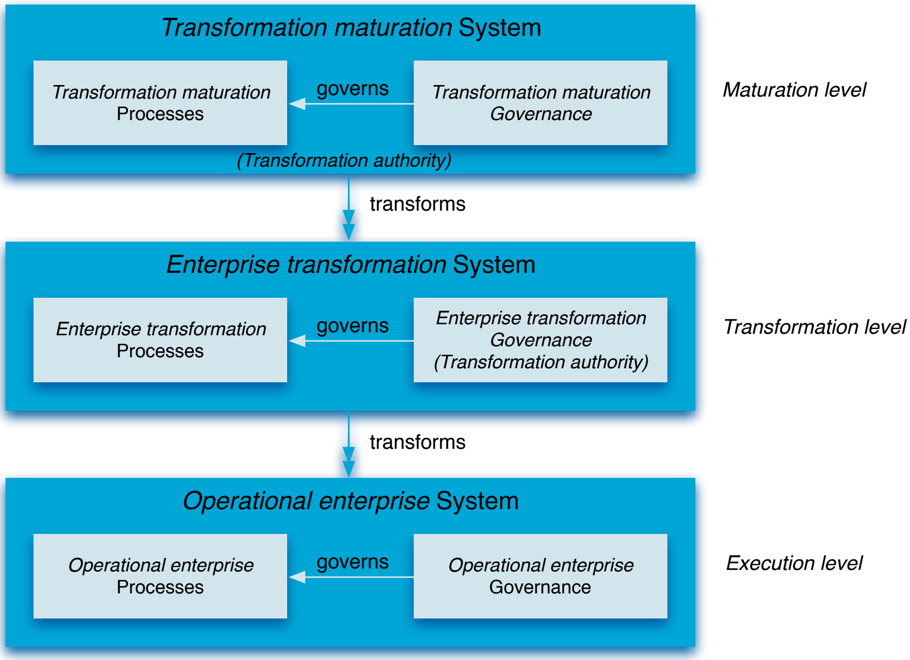
Transformation maturation System
Transformation maturation Processes
governs
Transformation maturation Governance
(Transformation authority)
Maturation level
transforms
Enterprise transformation System
Enterprise transformation Processes
governs
Enterprise transformation Governance (Transformation authority)
Transformation level
transforms
Operational enterprise System
Operational enterprise Processes
governs
Operational enterprise Governance
Execution level

Fig. 9. Adding a maturation level

An important aspect in the maturation of a transformation authority is the evaluation of its effectiveness. This effectiveness may differ from one organisation to another. In [24] a model is described on how to measure the effectiveness of an organisation's *architecture function*. An architecture function as defined in [24] is close to our concept of a transformation authority. However, we prefer to stress the primary role of the transformation authority, being the governance of transformations, rather than to refer to one of the *means* (albeit an important one) it uses to do so: *architecture*. An important indicator for the effectiveness of a transformation authority is the maturity level at which it operates. Analogously to the capability and maturity model for software development processes, maturity models have been developed to express an organisation's maturity in using architecture. A prime example is the USA Department of Commerce's maturity model [25].

Note that any enterprise executing enterprise transformations does have a transformation authority at some level of maturity. This might be at a purely "ad hoc" and totally immature level. The "implementation" of a transformation authority therefore can be seen as taking the enterprise's transformation authority to at least the first maturity level.

6 Case 1 – Post-merger improvement of a transformation authority

6.1 Situation

A large Netherlands-based multinational enterprise highly depends on IT for its core business processes. Over the past fifteen years, this organisation has grown to its current size as a result of several mergers and acquisitions. About five years ago, the organisation found itself hampered by a lack of synergy between the different business units originating from past acquisitions and mergers. To a larger extent this was traced back to the lack of integration between the IT departments of the different business units, as these departments where focussed on the needs of the respective business units (formally independent enterprises) rather than the needs at the enterprise level. To remedy this, the IT departments of the different business units were put together in one large IT business unit responsible for all IT operations for the entire enterprise. The original IT departments were, however, turned into sub-units within the newly created IT department.

Two years ago, much to the disappointment of top management, it was concluded that the situation had not really improved. The sub-units within the new IT department still maintained strong ties to the original business units they belonged to. As a result, during system development/transformation projects, the needs of the business units still prevailed over the needs of the enterprise as a whole.

Even more, due to missing insight into the relationship between decisions favouring a fast time-to-market of new business initiatives and the negative impact these decisions may have on the maintainability of the IT infrastructures, most key decisions had been made based on the shorter term interests. Meanwhile as a result of ill-guided pragmatism in achieving fast a time-to-market, the IT infrastructure had become a patchwork of platforms and interfaces, leading to a choice high operational risks for the enterprise. Recently, this hidden danger surfaced, leading to disruptions of the enterprise's primary processes.

6.2 The improvement programme

To improve on the existing situation, an improvement programme was put in place by senior management of enterprise, aiming to improve transformation governance of the IT department. The focus of this transformation programme was on: increasing the maturity of transformation processes, training of people in architectural standards, the standardisation of the development/transformation processes, as well as the creation of an explicit enterprise architecture.

Management of the enterprise, and senior management of the IT department, adhered to a rather Anglo-Saxon management style. As a result, the improvement programme used a top-down approach to initiate the changes needed. However, most of the lower management, as well as the people involved in IT operations and development projects where brought up in a more consensus based culture. This obviously produced a lot of tension within the organisation.

However, the strong top-down approach did enforce commitment from middle-level management during the transformation. When using a bottom-up approach based on the consensus culture, while creating intrinsic motivation by increasing awareness/insight into the advantages of a more mature way of working, the improvement programme might still have failed due to a lack of commitment from middle-level management. In the end, the combination of using a strict top-down approach forcing management within the IT department to move into the right direction, and increasing the knowledge level of people working in development/transformation projects by having them attend courses, proved to be a balanced approach. The top-down approach also brought about a kind of a "shock-therapy" in the sense that a lot of the scepticism about the IT department's own ability to create standards and workable architectures was swept away when the initial results were produced. The forced adoption of these standards consequently led to focussed discussions about the correctness of the standard. However, rather than paralysing progress in creating consensus based standards, the discussions were now more easy to focus on the improvement and fine-tuning of the top-down created standards.

While the improvement process took hold, it became clear that it was necessary to also increase the maturity of senior management in the IT department as well as the awareness of the impact of design decisions with business management. Thus far it was highly difficult to clearly trace back increased maintenance costs to undisciplined business desires. The next logical step was therefore to create reference architectures for standard solutions with low maintenance costs, offering a clear choice to business management: standardise or be prepared to pay extra. As a result of increased training of the people in the IT department, their personal esteem also increased, enabling them to indeed more clearly and *confidently* communicate these trade-offs back to business management.

During the improvement process, a complicating factor was that some people involved played roles in both the improvement programme and the daily workings of the IT department. The sponsors of the maturation programme needed maturing themselves as well in their role in the IT department, in particular in their senior management role.

6.3 Conclusion

In the first attempt by the enterprise to re-organise the existing IT departments, not enough attention was paid to integrating the former departments into a new whole, including the setting up of new relations (and ways of "doing business") between the new IT department and the business units, in particular the establishment of an able and effective transformation authority.

In the second attempt, a very interesting balance was struck between a top-down approach, creating a "shock-therapy" effect, and the enabling of people by training, also balancing between a top-down management style with a consensus based culture. A crucial factor also turned out to be the ability of the sponsors of the improvement process to realise that they themselves needed to mature as well with regards to their role in the transformation governance processes.

The top-down enforced creation of standards and reference architectures also emphasises the importance, in this case, of balancing the execution of a transformation authority's tasks with the progress of its maturation process.

7 Case 2 – Orchestrated improvement programme

7.1 Situation

A large Netherlands-based international financial institution (34000 fte), in an attempt to better align business with IT, is looking for ways to improve the governance of the information systems development process, encompassing a fairly large IT organisation of about 2000 fte in total. The institution wants to achieve better alignment between the business needs and the process changes and IT solutions delivered by the individual projects (about 500 projects are in portfolio). Earlier, relatively isolated, attempts focusing on improvement of the portfolio management process, development of a software factory and development of a corporate IT architecture failed or were very ineffective. The institution is facing a dilemma: to spend even more money "to get things right" in an integral manner or muddling on using the results of earlier improvement initiatives. After much deliberation, the choice is to start a *software process improvement program*. The overall objective of this programme is the alignment of business and IT. Sub-goals include development productivity improvement and increasing (perceived) quality, speed and professionalism. The productivity improvement turns out to be the main management trigger, as it enables a significant cost reduction while improving the quality of the project results.

7.2 The improvement program

The software process improvement programme consists of several change projects: introduction of an iterative enterprise transformation method, the implementation of a CMM level 3 compliant quality system, the introduction of a measurement dashboard and associated measuring instruments, special attention to professional attitude and, moreover, *harvesting* products of earlier initiatives. One of these initiatives is the top-down development of a large scale enterprise architecture. This architecture consists of a large number of guidelines, principles, rules, standards and reference models meant to guide and jump-start projects. These initiatives failed, largely because of the disconnection between the theory "invented" by the architects and the real need of the projects. However, a large body of knowledge was delivered.

The software process improvement programme uses a combination of top-down and bottom-up to implement the results of the various projects and initiatives. Bottom-up, because individual projects are used to gradually introduce the results. Top-down, because the boundaries of the implementation are very clear to all stakeholders, with a very high commitment of senior management wanting to achieve the programme goals. The combination proves to be very successful. In a time span of 3 years, the entire development organisation is transformed into a CMM level 3 organisation, using one, user-centred iterative development method, measuring its processes and their outputs and, for the first time ever, real *control* over the development project portfolio. This results in a good alignment between business and IT on both strategic (portfolio), tactical (project) and operational level.

The enterprise architecture is gradually introduced in the projects. A relatively small group of enterprise architects, headed by a Chief Architect, is formed, aiming to support the individual projects as much as possible using the components of the enterprise architecture. These architects act as consultants, paid by the software process improvement program. Once the individual projects and programs, and more importantly their *business owners*, see the added value of the enterprise architecture and the architects, the management of the financial institution decides to make those parts of the enterprise architecture compulsory that have been successfully applied in projects and programs. In this decision, the chief architect acts as a consultant. Over the course of the three years, large parts of the enterprise architecture are introduced in this manner.

7.3 Conclusion

Especially the gradual introduction of enterprise architecture in the context of the larger software process improvement programme proves to be successful. Projects and programs are not confronted with large piles of documents, but only with those parts of the Enterprise Architecture that really guide and help them. A relatively large period of time (3 years) proves also to be of importance: things have to “sink in” and much persistence is needed for that. Another critical success factor is the attitude of the architect: a supportive consultant, not an arrogant know-it-all. Harvesting is the fourth success factor: re-use existing products instead of inventing the wheel again. Last but not least is the level of alignment of introducing an enterprise architecture with other initiatives to improve governance & projects such as the implementation of the enterprise transformation method, as well as the incorporation of many related initiatives and projects into one single programme.

The big challenge in these initiatives is sustainability. During the three years of the program, management attention is quite high. Shortly after the end of the program, the financial institution is involved in a new change initiative, which redirects management attention. Sadly, this impairs most results in quite a substantial way.

8 Analysis of the cases

Obviously, the failure or success of a maturation/implementation of a transformation authority is largely determined by the characteristics of its execution process and the characteristics of the process aiming to mature this execution process, including the actors, products, et cetera involved. Moreover, the interaction between execution of the transformation authority's processes and the progress of the maturation process is an important determinant for success, as stressed in the case 1. The road to success would then be, to create execution and maturation processes with the 'ideal' characteristics. This, however, poses some problems:

- It is not possible to define 'ideal', as each situation is different and constantly changing;
- Suppose it would be possible to define 'ideal', it is hard to assign the 'right' values to characteristics.

A more practical way of looking at this, is to consider the *road blocks* for successful implementation of a transformation authority. Success is, in this view, achieved by *removing* the roadblocks. In the cases, several roadblocks were considered:

- The inconsistency between management culture and the culture of the 'shop floor'. In case 1, the management culture is very Anglo-Saxon and directive, whereas the culture of the rest of the organisation is more of a consensus-based nature. These cultures clash, especially in the case of transformation and its governance process;
- The lack of real management attention and support. In case 2, it turns out that, after the software process improvement program has ended, and management redirects its commitment to other initiatives, earlier achievements of the maturation program are eroded. It seems that the organisation has forgotten how well transformation governance went when management had the right attention;
- The inconsistency in communication, not "walking the talk". Management in case 1 used only one directive communication style, but the directives were not or hardly implemented. People were not informed about the rationale of certain management decisions, resulting in mistrust and vague implementation processes. Another related, roadblock was the lack of setting the good example by management: "practice what you preach". Management made an exception for itself not to follow the directives that were applicable to the entire organisation;
- The inability of architects to show results and add real or perceived value. The social and communicative competencies of the architect and other important stakeholders in the transformation governance process are of utmost importance. In case 2, the architects that designed the instruments to improve the systems development process learned how to act as a consultant, helping the projects that were using the development process. In case 1, architects used a more traditional ivory tower approach with resistance from the actors that were subject to the transformation as a consequence;
- The inability of actors in the transformation to play several roles, connected to different interests. In case 1, several actors, especially important ones like the sponsor, played roles in both the operational processes, the projects and programs changing those operational processes *and* the program to implement/mature a transformation authority. It requires special skills to be able to separate the interests related to the different roles, especially with respect to prioritisation and the correct behaviour dependent on the role one plays;

- The ineffective way signals from the shop floor are received by management. In case 1, they were more or less treated as threats, whereas in case 2 management did not want to listen anymore after the program ended. The workers in the projects that were subject to change often know best how to improve the way of working, and when these signals are not picked up, potential improvements are lost;
- The political forces and hidden agendas in an organisation. Often, the issue is dealt with in a rational way, whereas quite a number of irrational factors play a much more important role. In both cases, politics and irrationality were roadblocks.

Notice that roadblocks are typically *not* to be found in the artefacts of the various transformation initiatives, but in their context. Their removal is part of the change management required to implement the artifacts.

9 Transformation maturity framework

Based on the analysis as discussed in the previous section, this section discusses a broad maturity framework for the maturation of transformation authorities, taking several important aspects into consideration that may lead to blockages, and maybe even lead to erosion of already achieved results.

9.1 Implementation strategies and situational factors

The cases show some best and worst practices regarding the implementation of an transformation authority. Some strategies work, some do not. In general, we have seen that the success of an implementation is determined by:

- The characteristics of the implementation process;
- The characteristics of the situation in which the transformation authority is implemented.

The process of implementing/maturing an transformation authority is not very different from processes aiming at organisational changes (see, for instance [26]) in general. These processes boil down, to put it very simplistically, to a combination of top-down vision setting, initiating, managing, communicating, measuring and supporting versus the bottom-up ability to execute, competencies, willingness, enthusiasm, learning and showing success. Irrespective of using a top-down or bottom-up approach when introducing an transformation authority, the awareness of the importance of architecture as a means to steer developments, needs to grow – it is not viable to implement a full-fledged transformation authority right from the start. It will need time to learn and to be able to support the enterprise transformation projects and programs in the best possible way. We have seen that architecture always needs to balance longer-term interests to short-term interests. In this present day and age and faced with day-to-day pressures, one tends to forget about the longer term interests. Especially management is very focused on achieving short-term results, which poses a huge risk on the potential success of a transformation authority.

Implementing/maturing an transformation authority therefore requires a deliberate strategy in which the key players are made aware of the interplay between short-term and long-term interests, while gatekeepers are introduced to safeguard the long-term interests against erosion by short-term considerations. To be able to develop such a deliberate strategy a more explicit understanding of the role of architecture as a means of steering is needed, as well as the underlying reasoning. We have seen in the cases that this role is very much dependent on situational factors, such as:

- The competencies of the architects, especially with regard to communication and other personal skills;
- The historical perspective of earlier attempts to implement an transformation authority;
- The focus, attention and support of the management;
- The extent to which a "burning platform" exists: is it really necessary to implement a transformation authority?
- The extent to which the architecture implementation is connected to other improvement initiatives;
- The culture of the organisation and its parts, especially management culture versus "shop floor" culture;
- The maturity of project and program management;
- The scope, depth and size of the architecture itself.

There is obviously no silver bullet for successful implementation of a transformation authority. One conclusion we can draw from the cases is, however, that the "maturity" of the organisation plays a crucial role. We will elaborate this in the next sections.

9.2 Removing roadblocks: the transformation maturity concept

In both cases we found that the extent to which roadblocks can be removed is dependent on a concept we call "transformation maturity". We define transformation maturity as the capability to achieve effective governance of transformation processes, where "effective" is obviously situation-dependent. We use the number of roadblocks as a proxy to the transformation maturity level: the higher the number of roadblocks, the lower the transformation maturity level. A high number of roadblocks is a symptom of low transformation maturity, and this can hamper the transformation significantly.

For the time being, we consider transformation maturity as a relative notion, so we can only *compare* the transformation maturity of two distinct systems. To enable comparison and provide a more generic analysis of the case, we use the levels identified in Figure 9: *execution level*, *transformation level* and *maturation level* in relation to enterprise transformtions. At each of these levels, we can refer to transformation maturity:

1. On the execution level, we consider the *operational enterprise* system. In this system, the operational processes of an enterprise are executed and governed. This system encompasses the execution and governance of primary and secondary processes. In case 2, for example, issuing loans or trading stocks.
   At this level, transformation maturity is related to the system's ability to undergo transformations.

2. On the transformation level, we consider the *enterprise transformation* system. This is the system that transforms the *operational enterprise* system. It consists of transformation processes and the governance of those processes. Typically, these processes are information system implementation projects and programs, for instance a large ERP implementation or a custom development project.
   At this level, transformation maturity refers to the quality and effectiveness of the transformation execution and the activities of the transformation authority.
3. On the maturation level, the *transformation maturation* system is considered. This is the system that aims to transform the *enterprise transformation* system, in order to improve its transformation maturity. Again, the system consist of the actual maturation processes and the governance of these processes. Examples of maturation processes are the software process improvement program in case 2 and the project to implement the transformation authority in case 1.
   At this level, transformation maturity is concerned with the ability of the *transformation maturation* system to continuously improve the maturity of the *enterprise transformation* system.

The transformation maturity concept and the distinction of three systems in the transformed enterprise enable us to analyse the cases in a different way.

In case 1, the transformation maturity of the *operational enterprise* system is at about the same level as the transformation maturity of the *enterprise transformation* system: the number of roadblocks is more or less comparable in both systems. The transformation maturity of the *operational enterprise* system is improved by the *enterprise transformation* system. However, the transformation maturity of the *transformation maturity* system is lower than the other two systems, because there are (relatively) much more roadblocks in this system. Due to this, the *transformation maturity* system is hardly able to transform the *enterprise transformation* system.

In case 2, there are two situations: the states during and after the software process improvement program.

During the Program, the *transformation maturity* system has a relative high transformation maturity level compared to the *enterprise transformation* system. A lot of roadblocks in the Program are not present or have been removed. The Program is quite able to transform the systems development and project management processes, i.e., the *enterprise transformation* system. The latter, on turn, becomes, as a result of the Software Process Improvement Program, more mature than the *operational enterprise* system.

This changes after the end of the Software Process Improvement Program. A roadblock is introduced in the governance of the *enterprise transformation* system: lack of management attention. Because the *operational enterprise* system has matured due to the improved transformation ability of the *enterprise transformation* system as a result of the Software Process Improvement Program, the relative transformation maturity of the Enterprise transformation Program is decreasing as compared to the *operational enterprise* system. This, on turn, decreases transformation ability of the Enterprise transformation Program, resulting in lower project productivity and lower alignment between business and IT. This is exactly what happened in case 2. In a comparable way, the transformation maturity of the *transformation maturity* system also decreased, as a consequence of the addition of some roadblocks on this level.

9.3 The transformation maturity framework

The case analysis in relation to the introduction of levels in Transformation Governance and the Transformation Maturity concept has provided us with the following important condition for successful transformation governance: System A, transforming System B, must have at least the same transformation maturity as System B. Applied to the three relevant systems we identified in a semi-formal notation (where TM stands for Transformation Maturity):

$$\begin{aligned}&\mathsf{TM}(\mathsf{Transformation\ maturation\ System})\\ &\geq \mathsf{TM}(\textsf{\textit{enterprise transformation}}\ \mathsf{system})\\ &\geq \mathsf{TM}(\textsf{\textit{operational enterprise}}\ \mathsf{system})\end{aligned}$$

This is principle 1 of our *transformation maturity framework* (TMF). Enterprise transformations complying with this principle are called “ideal”.

As a consequence, it does not make much sense (or, at least, it is ineffective) to transform processes with a relative immature system. Before starting a transformation, roadblocks in the transforming system have to be removed to ensure a sufficient transformation maturity level and “ideal” enterprise transformations.

A second observation is, that transformations are always an interplay of bottom-up and top-down actions. Typically, the “pain” comes from below, i.e., the systems that are transformed. “Healing” this pain is supposed to come from above, the transforming system. Healing consists of, for example, providing an artefact or removing a roadblock. If the cure is structural, it can be considered a transformation. In non-ideal enterprise transformations, sometimes a lot of pain is generated and in reaction, the transforming system tries to heal this pain in an ad-hoc fashion. As we have seen in case 1, there is often a trade-off between pain and healing. In much the same manner, a trade-off exists between preventing and healing: preventing being a pro-active activity and healing as reaction to pain. In an ideal transformation governance, everything can be foreseen and thus prevented - but reality is different, so there is still healing to be done.

The second principle of the TMF is that in enterprise transformations, there is always a trade-off between opposite factors, often in terms of healing vs. pain and healing vs. preventing.

The second principle implies that an enterprise transformation is an iterative process, aiming at an “ideal” situation with the right transformation maturity levels and constantly seeking the trade-offs between healing and pain and between preventing and healing. In the cases we observed that it does not make much sense to produce all the artefacts in advance, because reality is changing constantly and artefacts have to connect to this reality instead of being theoretical. On the other hand, it is not possible to start completely without artefacts, so a certain minimal set has to be there before the journey to the ideal situation can start.

The third observation relates to the actors in the enterprise transformation. As mentioned in the list of roadblocks, many actors play different roles. If these roles are played on different levels, the actor involved has to be aware of this. He has to be able to deal with the different interests and the different objectives and priorities in systems on different levels. For example, a sponsor is often a line manager on a fairly high organisational level. As a line manager, he might have other interests, priorities and objectives than as a sponsor. If he cannot separate the various roles he is playing, the number of roadblocks can increase. This is not necessarily bad, as long as it occurs on the 'lower' levels, notably the Execution level. If it occurs on higher levels, for instance the Maturation level, the consequence is a lower maturity and a lower chance of achieving the "ideal" situation.

We translate the observation in the third principle of the TMF: An actor has to be able to play multi-level roles.

10 Conclusion

The driving interest of this chapter was the implementation of mature *transformation authorities*. To this end, we took both a theoretical and a practical perspective on enterprise transformation and their governance.

We started by defining more precisely what we mean be *enterprise transformations* and their *governance*, as well as the need to use *architecture* to achieve *informed governance*. The concept of a *transformation authority* was introduced as the function in an organisation which is responsible for the governance of enterprise transformations.

We then continued with a discussion of two cases drawn from our industrial practices. Using the insight from these cases, we then refined our theoretical considerations in terms of the *transformation maturity framework* (TMF).

As a next step we intend to further refine our definitions on the basis of practical experiences. More specifically, we aim to further refine the maturity model in terms of situational factors (such as the one discussed in Section 9) influencing the aptness for different strategies to mature/implement transformation authorities, as well as concrete roadmaps to indeed grow/mature transformation authorities.

References

1. Op 't Land, M., Proper, H., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture – Creating Value by Informed Governance. Springer, Berlin, Germany, EU (2008). ISBN-13: 9783540852315
`http://www.springerlink.com/content/978-3-540-85231-5`
2. Leeuw, A.d.: Organisaties: Management, Analyse, Ontwikkeling en Verandering, een systeem visie. van Gorcum, Assen, The Netherlands, EU (1982) In Dutch.
3. Leeuw, A.d., Volberda, H.: On the Concept of Flexibility: A Dual Control Perspective. Omega, International Journal of Management science **24** (1996) 121–139.
4. Ashby, W.: An Introduction to Cybernetics. Chapman & Hall, London, United Kingdom, EU (1956). ISBN-10: 0412056704
5. Beer, S.: Diagnosing the System for Organizations. Wiley, New York, New York, USA (1985).

6. Hagel III, J., Armstrong, A.: Net Gain – Expanding markets through virtual communities. Harvard Business School Press, Boston, Massachusetts, USA (1997).
7. Horan, T.: Digital Places – Building our city of bits. The Urban Land Institute (ULI), Washington DC, USA (2000). ISBN-10: 0874208459
8. Mulholland, A., Thomas, C., Kurchina, P., Woods, D.: Mashup Corporations - The End of Business as Usual. Evolved Technologist Press, New York, New York, USA (2006). ISBN-13: 9780978921804
9. Tapscott, D.: Digital Economy – Promise and peril in the age of networked intelligence. McGraw–Hill, New York, New York, USA (1996). ISBN-10: 0070633428
10. Government of the USA: Sarbanes-Oxley Act of 2002. Number H.R.3763. (2002).
11. Porter, M.: What is strategy. Harvard Business Review (1996).
12. Treacy, M., Wiersema, F.: The Discipline of Market Leaders – Choose your customers, narrow your focus, dominate your market. Addison Wesley, Reading, Massachusetts, USA (1997). ISBN-10: 0201407191
13. Graaf, E.v.d.: Architectuurprincipes en clustercriteria voor de afbakening van out-sourcebare kavels. Master's thesis, Radboud University Nijmegen (2006) In Dutch.
14. Op 't Land, M.: Towards Evidence Based Splitting of Organizations. In Ralyté, J., Brinkkemper, S., Henderson-Sellers, B., eds.: Proceedings of the IFIP TC8 / WG8.1 Working Conference on Situational Method Engineering: Fundamentals and Experiences (ME07), Geneva, Switzerland. Volume 244 of IFIP Series., Springer, Berlin, Germany, EU (2007) 328–342. ISBN-10: 9780387739465
15. Op 't Land, M., Dietz, J.: Enterprise ontology based splitting and contracting of organizations. In: Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC'08), Fortaleza, Ceará, Brazil. (2008).
16. Rijsenbrij, D.: Outsourcing zonder enterprise architectuur lijkt op autorijden zonder veiligheidsgordel. Technical Report NIII–R0404, Nijmegen Institute for Information and Computing Sciences, University of Nijmegen, Nijmegen, The Netherlands, EU (2004).
17. Oxford University: Oxford Dictionary of English. Oxford University Press, Oxford, United Kingdom, EU (2005). ISBN-13: 9780198610571
18. Bunge, M.: A World of Systems. Volume 4 of Treatise on Basic Philosophy. D. Reidel Publishing Company, Dordrecht, The Netherlands, EU (1979).
19. Deming, W.: Out of the Crisis. MIT Center for Advanced Engineering Study,, Massachusetts, USA (1986). ISBN 0911379010
20. The Open Group: The Open Group Architecture Framework (TOGAF) Version 8.1.1, Enterprise Edition. (2007).
`http://www.togaf.org`
21. Nabukenya, J.: Collaboration Engineering for Policy Making: A Theory of Good Policy in a Collaborative Action. In: Proceedings of the 15th European Conference on Information Systems. (2005) 54–61.
22. Nabukenya, J., Bommel, P.v., Proper, H.: A theory–driven design approach to collaborative policy making processes. In: Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS-42), Los Alamitos, Hawaii, USA, IEEE Computer Society Press (2009).
23. Nakakawa, A., Bommel, P.v., Proper, H.: Quality Enhancemenets in Creating Enterprise Architecture – Relevance of Academic Models in Practice. In: Submitted to: Practice-driven Research on Enterprise Transformation. (2009).
24. Raadt, B.v.d.: Normalized Architecture Organization Maturity Index. Technical report, Capgemini (2006).
25. Department of Commerce, Government of the USA: Introduction - IT Architecture Capability Maturity Model. Government of the United States of America (2003).

http://ocio.os.doc.gov/groups/public/@doc/@os/@ocio/@oitpp/documents/content/prod01_002340.pdf

26. Senge, P., Kleiner, A., Roberts, C., Roth, G., Ross, R., Smith, B.: The Dance of Change – The Challenges to Sustaining Momentum in Learning Organizations. Doubleday, New York, New York, USA (1999). ISBN 9780385493222