Assessing Business and IT Projects on Compliance with Enterprise Architecture

Ralph Foorthuis[1], Frank Hofman[1], Sjaak Brinkkemper[2] and Rik Bos[2]

[1] Statistics Netherlands, Henri Faasdreef 312, 2492 JP The Hague, the Netherlands,
{R.Foorthuis, F.Hofman}@cbs.nl
[2] Utrecht University, Institute of Information and Computing Sciences, Padualaan 14, 3584 CH Utrecht, the Netherlands
{S.Brinkkemper, R.Bos}@cs.uu.nl

Abstract. This article focuses on how to assess projects implementing business processes and IT systems on compliance with an Enterprise Architecture that provides constraints and high-level solutions. First, the core elements of Enterprise Architecture compliance testing are presented. Second, we discuss the testing process and four types of compliance checks (correctness check, justification check, consistency check and completeness check). Finally, an empirical case is reported in which a real-life project has been tested on conformance. The results show that our approach works. Furthermore, to increase the objectivity of compliance testing, operationalization of EA prescriptions is recommended.

Keywords. Compliance assessment, Projects, Enterprise Architecture

1 Introduction

When studying the literature, Enterprise Architecture (EA) can be said to have two major ideal type functions. One function is to provide decision-makers with a clear and comprehensive *descriptive overview* of the enterprise, or relevant aspects thereof. Such insights into the enterprise form the basis for making high-level management decisions [cf. 1, 2, 3], e.g. determining which programs or projects to initiate. This reflective function of EA mainly targets managers as its users. The EA can be expected to have a heavy focus on depicting the (problematic) as-is situation. Another function of EA is to provide a *prescriptive framework* that guides and constraints subsequent development of business and IT solutions [cf. 4, 5, 6, 7, 8]. This normative approach, focusing strongly on the to-be situation, should make sure that both enterprise-level and local initiatives within the organization are consistent with the overall strategy, and enable a coherent and integrated development of business, information and IT. This directive function of EA targets not only managers as its users, but also business analysts, system analysts, software architects and other roles in projects (re)designing the business and its IT-support. In this paper, we mainly focus on this latter function, a prescriptive EA providing constraints and high-level solutions to which business and IT systems – and in particular the projects implementing them – should conform.

Proceedings of GRCIS 2009
CAISE Workshop on Governance, Risk and Compliance of Information Systems

In this prescriptive context, an EA is mainly applied in projects. Although EA typically focuses on the entire enterprise and compliance is demanded at this high level, in practice it is not realistic for an entire organization to be EA-compliant at short notice. It can therefore be expected that conformance is reached incrementally at the local level, step by step – or rather, project by project [9]. However, philosophers have acknowledged for hundreds of years that, although compliance with 'contracts' might be better for the group as a whole and it might be in an individual actor's best interest to agree with contracts, it may not be in his interest to actually comply with them. In contractarian ethics this is one of the issues of the so-called *compliance problem* [cf. 10, 11]. Because of this potential conflict of interest, it should be tested whether actors actually conform to the contract. If we consider a specific project to be the actor, then an EA could be seen as the contract that needs to be complied with. Testing should be done at the level at which EA is applied, which is the project level. Testing at this level also allows for correcting non-compliant aspects, if it is carried out while EA is applied.

[12] defines compliance in the context of IT-projects as "the extent to which software developers have acted in accordance with the 'practices' set down in the standard." [13] defines compliance in this context as "an accordance of corporate IT systems with predefined policies, procedures, standards, guidelines, specifications, or legislation." In the context of EA we define *compliance* as corporate business and IT systems being in accordance with predefined Enterprise Architecture prescriptions. We will use the terms "compliance" and "conformance" interchangeably. Likewise, "assessing compliance" and "testing on conformance" are used synonymously.

In this paper, we aim to find an answer to the following research question:

> *How can projects and the business and IT solutions they deliver, be assessed on compliance with a prescriptive EA?*

A *project* in this paper refers to the 'regular' projects that need to comply with Enterprise Architecture and that usually have a more or less local scope (e.g. delivering a new business process and related IT applications for a department).

To answer the main research question, we divide it into several sub-questions:

1. *What concepts play a key role in assessing compliance with EA?*
2. *What kind of compliance checks can be utilized in EA compliance testing, and what are their respective evaluation criteria?*
3. *By what process can EA compliance testing be carried out?*

The goal of our research is to identify and explore core aspects of testing projects on EA compliance. It is our intention to stimulate additional research into the topic. A second, more practical goal is that the results should provide organizations with a model that can be used to develop their approach for testing their initiatives on EA conformance.

This paper will proceed as follows. In section 2 related topics and work are discussed. Section 3 positions our research specifically in the context of EA. Sections 4, 5 and 6 aim to find answers to the respective sub-questions. In section 7 we present our empirical research. Section 8 is for discussions and conclusions.

2 Related Topics and Work

Although we know of no academic work dedicated to the issue of assessing compliance with EA, the topic can nonetheless be linked to other work. In particular, EA conformance testing is related to the topics of *compliance management* and *software testing*. In terms of compliance management, several topics can be acknowledged that are relevant to our discussion. First, due to *legislation*, organizations are required to comply with certain regulations, which have consequences for their business processes and information systems. Non-compliance here may even have penal consequences for an organization's management [14]. In Europe, important drivers in this respect are Directive 95/46/EC, i.e. the Data Protection Directive, and Directive 2002/58/EC, i.e. the Privacy and Electronic Communications Directive [16, 18]. Examples of laws in the United States demanding compliance are the Sarbanes-Oxley Act, the Gramm-Leach-Bliley Act and the Health Insurance Portability and Accountability Act [13, 17, 26]. Basel II, being relevant for Europe, America and Japan, is an example of a global regulatory framework [20]. As a result of the world-wide credit crisis that started in 2008, it can be expected that financial frameworks like Basel II will demand even more compliance in the future.

A second topic in compliance management is being consistent with international and industry-wide *standards* for processes and products, such as ISO 9001 for quality management and IEC 61508 for safety. There are several reasons to conform to such best practices, for example clients or strategic partners demanding certification for assurance reasons, and using best practices to improve the organization's processes and products. Conformance to standards is especially important in large and critical systems engineering projects in e.g. the defense, aerospace and telecommunications sectors. See [12, 21, 22] for more about compliance with standards. We will use some of the concepts in these publications in our research.

A third relevant topic is *security and risk management*, which aims to protect the organization's assets, such as valuable information. Compliance here has an important role in avoiding intentional and unintentional harm to the organization, e.g. by imposing access restrictions. See for example [23, 24, 25] for more on this topic.

All these topics are relevant to our discussion, as an EA can feature constraints and high-level solutions based on any of the above. Needless to say, these topics are not mutually exclusive. For example, security and risk management are central concerns of Basel II and of international standards such as ISO/IEC 27000.

Assessing projects and their products on compliance with EA can also be related to software testing. Several core elements can be distinguished [27] in this discipline. First, the *test items* refer to the items that require testing, e.g. a document or a version of an application. Second, the *features* are the specified properties that the test item is required to possess. Third, *acceptance criteria* are needed to decide whether the software is ready for successful usage in the business setting. This is relevant because features are not sufficient for testing, as not every feature is equally important and features may be only partially implemented. Lastly, a *test approach* is needed to define the testing techniques to be used in determining whether the test item possesses the features to an acceptable degree. In this paper we will translate these software testing concepts to the domain of EA conformance testing.

3 Positioning the Research

The diagram below shows the different levels involved in working with EA, and how the processes at these levels are related. The output of each level is input for the lower level. A rounded rectangle represents a process, a square rectangle the input and output of a process. A continuous line denotes the process flow, a dashed line an information or product flow.

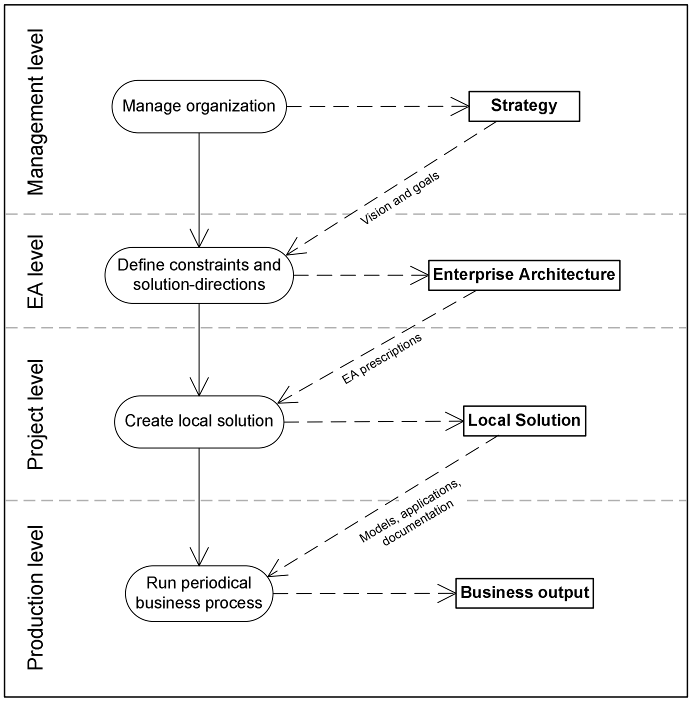

Fig. 1. High-level overview of the processes related to working with EA

The diagram should not be considered as modeling one single process, but rather as identifying four distinct processes, each at a different level. The model explicitly shows that the output of each process is the input for the level below. Feedback can certainly flow from lower to higher levels, but in order to focus on the essence we have abstracted from that in this diagram. The output of each of the processes will be described in more detail in the next section.

This paper focuses on testing whether the Local Solution indeed conforms to EA. In other words, on assessing whether the project has correctly applied the EA prescriptions in creating the solution. We will therefore concentrate mainly on the project level, as we expect a Strategy and Enterprise Architecture to be given, and the production process generating Business Output can only be run after the Local Solution has been delivered and judged to be compliant with EA.

4 Fundamental Concepts in EA Compliance Testing

This section presents an overview of the fundamental elements of EA compliance testing. We have represented these core concepts in the EA Compliance Model of Figure 2 using a UML class diagram. The bold-lined classes are the four output products of Figure 1. The double-lined class (the Compliance Check) will be described in more detail in section 5. The triple-lined class (the Baseline) is described in more detail in [29]. Since the model will function as the basis for the remainder of our paper, its contents will be supported with literature where relevant.

Four high-level concepts can be acknowledged in compliance testing, represented by the grey areas. These are inspired by section 2's core elements of software testing. First, analogous to software testing, there is an *assessment item* that needs to be tested. This is the set of project artifacts, in which the EA prescriptions should have been applied. An artifact here is a deliverable or intermediate work product, such as a software architecture document. Second, a set of *compliance norms* is needed. These are the EA's prescriptions, possibly complemented with local acceptance criteria. Third, an approach or *compliance test* will be used to establish (non-)compliance of the items. This comprises several types of compliance checks. Lastly, the *EA-compliant business* represents the desired result. We will discuss the model in more detail below. The individual classes of Figure 2 will be directly referred to using Capitalized Names, while properties will be referred to using *Italic Capitalized Names*.

An enterprise's Strategy will provide the input for its Enterprise Architecture, as an EA is a governance instrument intended to facilitate the translation from corporate strategy to daily operations [30]. The resulting EA consists of Views and Prescriptions [7]. A View typically provides insight into the context and meaning of a system (e.g. an entire enterprise, an IT system or a business service), and its fundamental organization, components and their relationships. As such, a View can depict both the as-is and the to-be situation. It can be utilized as a cognitive aid, in the form of an overview (e.g. a context model), a frame of reference (e.g. a structuring mechanism for analysis purposes), or a shared vocabulary (for communication purposes). A Prescription, focusing solely on the to-be situation, has an explicit guiding function and is required to take the form of a Principle (textual statement), Model (visual diagram) or Policy Statement (exposition containing text and possibly diagrams). These types of Prescriptions explicitly provide constraints or directions and are therefore more directly related to compliance than a View.

A Prescription is a relatively stable fundamental guidance that has to be complied with. As the Prescription is the central element in the model, it is presented with its properties (which will be used in section 5 to identify and define types of checks). These properties are based partly on the template for describing principles, as defined by [31]. The first property is the *Name*, which should succinctly and identifiably refer to the essence of the Prescription. Second, the explicit *Definition* is the compliance requirement, presented as clearly as possible in the form of a *Statement*, *Diagram* or *Exposition*[1]. A third important property is the *Rationale*, providing the reasons behind the Prescription and thereby elaborating on the business benefits achieved by adhering

[1] For didactic purposes, we used overriding of the *Definition* property here, which is unusual in OO.

to it. It should make explicit why and when the prescription can be effective, and could as such motivate compliance [12]. Fourth, the *Implication* describes the (potential) impact and consequences of applying the Prescription in terms of costs, resources and activities. This is input for a cost-benefit analysis when deciding whether to apply it or not and can provide information about how to apply it in practice. The fifth property, the *Illustration*, is valuable because examples can clarify Prescriptions that are inherently ambiguous as a result of their generic nature [9]. Lastly, the *Priority* indicates the importance of the prescription, stating e.g. whether it is mandatory or merely recommended.

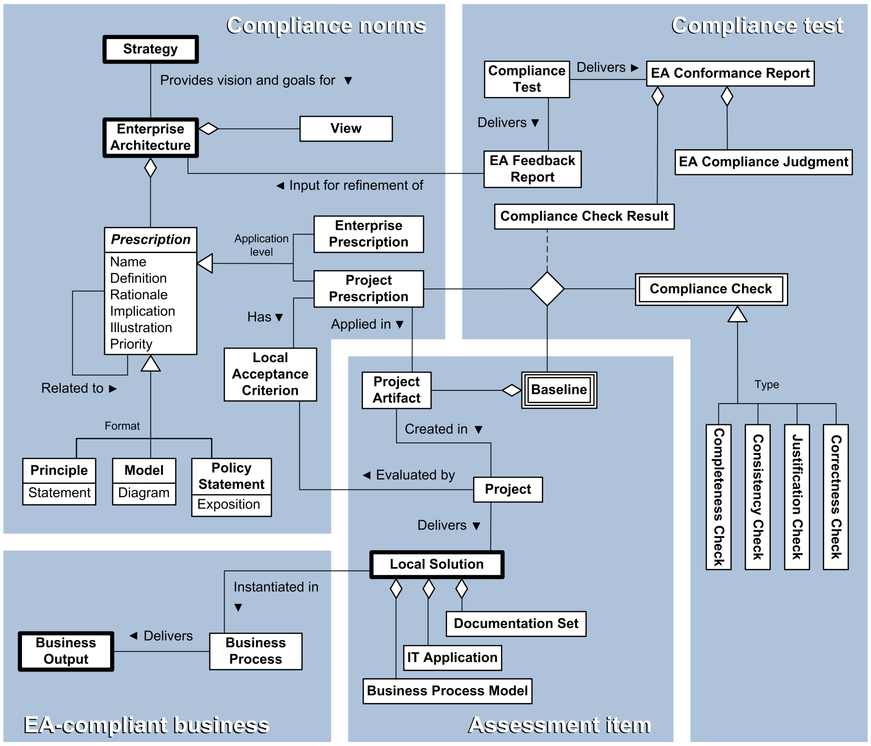

Fig. 2. The EA Compliance Model

A Prescription can be related to other Prescriptions. For example, prescriptions can be ordered *hierarchically* (which is relevant if the EA framework features abstraction levels). The *counterpart prescriptions* described in [9] are another example. This refers to business Prescriptions that have closely related IT counterparts, and vice versa. It is a mechanism to improve business-IT alignment. Business Prescriptions with IT implications should therefore have related IT Prescriptions, and vice versa.

A Prescription can be an Enterprise Prescription, which provides generic constraints (boundaries) and directions (high-level solutions) for an entire enterprise. Prescriptions applied at this level can as such guide the outlining of the enterprise's policy or direct the development and evolution of high-level enterprise-wide services. A Prescription can also be a Project Prescription, which provides generic constraints and directions for localized Projects (or rather, their products). Projects and compliance testing might also need to take Local Acceptance Criteria into account. The specific situation that is assessed might call for ad hoc variations, e.g. exempting the project from certain prescriptions in the case of urgency.

Project Prescriptions are applied by Projects in their Project Artifacts, i.e. deliverables such as software architecture documents. A Baseline collects several Project Artifacts that are reviewed and agreed on by their immediate stakeholders and that form the basis for further development [28]. Through their explicit or implicit application in Project Artifacts, Project Prescriptions can guide the development and evolution of local initiatives by providing constraints and directions that the Projects implementing them should adhere to. The Project delivers a Local Solution, which can consist of a newly designed Business Process Model, a newly developed IT Application and a Documentation Set (i.e. manuals and the final Baseline). Generating Business Output means instantiating the Local Solution in a Business Process, which means planning and running a real-life instantiation of the designed process and the IT application. First, however, compliance with EA needs to be assessed.

Key elements in performing the Compliance Test are Compliance Checks, norms (EA prescriptions and Local Acceptance Criteria) and a resulting EA Conformance Report (cf. [7, 12, 22, 27]). A Baseline provides an ideal opportunity for this compliance assessment, as it describes the agreed-upon basis for the remainder of the project and still allows for intervention in the case of non-compliance. As the ternary association class shows, the Compliance Check Result is the product of the EA's Project Prescriptions, the Project's Baseline to be tested and the types of Compliance Checks. In other words, given a Baseline that is to be tested, several compliance checks are performed for each Prescription, resulting in a number of Compliance Check Results. See Table 1 in section 7 for an example of such test results for a given Baseline. Each individual (non-conformant) Compliance Check Result will be an entry in the EA Conformance Report. Four types of Compliance Checks will be identified in section 5 of this paper. The EA Conformance Report also contains a final EA Compliance Judgment, which is the test conclusion stating whether the assessed item (i.e. Baseline) complies or not. Lastly, the Compliance Test may yield an EA Feedback Report that provides valuable information to the enterprise architects to update the EA.

4.1 EA compliance testing and auditing

It is interesting to mention the difference between the compliance test discussed here and an audit. According to [28], an *audit* is "an independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria." Therefore, if the goal of an audit is to assess compliance of designs with an Enterprise Architecture, then our EA Compliance Model applies. However, if an audit is to assess whether a business unit does in practice what is intended, then a conformance test on the Business Process is needed, in which the compliance norms are described in the documentation that describe the Local Solution. Furthermore, an audit may be a compliance assessment after *run-time* (also taking actual execution traces such as process logs into account), while the assessment of a project described above is carried out in *design-time* (taking artifacts that describe designs of processes and systems into account).

5 Types of Compliance Checks

Based on the EA Compliance Model of section 4 we can distinguish between several types of *compliance checks*. These are mechanisms to assess the current state of compliance [19]. When testing projects on EA conformance, several types of such checks can be distinguished, each assessing a specific aspect. The types of checks will be described below. For each type, the specific elements of the norms that are required for assessment will also be mentioned (in terms of the EA Compliance Model).

- *Correctness check*: verifies whether a given prescription is applied by the project in a way that is in accordance with its intended meaning, rationale and usage. In other words, this check verifies whether the application of the prescription deviates from the prescription as it was intended by the enterprise architects.
  In terms of the EA Compliance Model, the criteria needed for performing the correctness check can mainly be found in the Prescription's *Definition* and *Illustration* properties, as these serve to communicate its intended meaning. However, the *Rationale* and *Implication* might also be relevant here, as they elaborate on its value and usage.
- *Justification check*: verifies whether the (lack of) application of a given prescription is justified, depending on its relevance in the specific situation. The justification check's actual execution is dependent upon certain conditions. First, if the application of a prescription *deviates* from its intended application (which is determined by the correctness check), it needs to be ascertained whether the alteration is justified. Second, if a prescription is *not applied*, it needs to be ascertained whether it is justified not to apply it. Third, if a prescription is *applied correctly*, it needs to be checked whether it is indeed justified to apply it. This last sub-check aims to avoid 'blind' conformance that could harm project or enterprise goals in the specific situation. In short, the justification check verifies whether the project has made the appropriate choice when deciding to apply, alter or not to apply a given prescription.
  In the EA Compliance Model, the justification check's evaluation criteria can be found in the Prescription's *Rationale*. The rationale describes the prescription's benefits (which should be consistent with the local situation's objectives) and when it should be applied (which should be consistent with the nature of the local situation). In addition, the *Implication* may be relevant here, since the impact in terms of costs, resources and activities can play a role in the cost-benefit analysis. Furthermore, the *Priority* should state whether prescriptions are mandatory or merely guidelines.
- *Consistency check*: verifies whether, if a given prescription is applied, required related prescriptions are also applied. Some prescriptions, especially those at lower abstraction levels, might need to be implemented as a package. For example, the counterpart prescriptions mentioned in section 4. Another focus of

the check is to verify whether the prescriptions' applications do not contradict each other, but instead culminate in a consistent and balanced result.
The consistency check's evaluation criteria can be found in the prescription's relationship with other prescriptions (i.e. the self-reference of the Prescription).

- *Completeness check*: verifies whether all the prescriptions are applied. Minimally, the prescriptions that have been marked as mandatory (perhaps dependent on specific project situations) need to be applied.
The completeness check's evaluation criteria can be found in the Prescription's multiplicity with the Enterprise Architecture. It is the number of Prescriptions (that are of type Project Prescription) represented by the "*" symbol in a specific instantiation of the model. Or more simply: the total number of (mandatory) prescriptions relevant for projects.

The completeness and correctness check types are also mentioned in [22] in their discussion of compliance with standards. We have adapted them here to fit the EA context. The justification check has been added because the relevance of prescriptions can be conditional [21] and local acceptance criteria might need to be taken into account. The consistency check, which is especially relevant in the context of Enterprise Architecture, was added since EA aims for a coherent development of business, information and IT, but at the same time has to deal with potential conflicting stakeholder interests and requirements [31].

The correctness and justification checks are performed at the level of an individual Prescription. The completeness check is done at the level of the entire collection of Project Prescriptions. The consistency check is performed at the level of a group (package) of individual Prescriptions.[2] This is demonstrated in Table 1 of section 7.

Given an applied Project Prescription, each individual check can have one of three outcomes:

- *Passed*: the applied prescription passed the respective compliance check.
- *Failed*: the applied prescription failed the respective compliance check.
- *Needs attention*: the applied prescription might be (or become) compliant. However, it is applied partially or its application is ambiguous (i.e. there is not sufficient information to determine the outcome of the check).

6 The Process of Testing

In this section we will present several requirements for a process in which compliance testing is carried out. A first requirement is the separation of duties (i.e. checks and balances). An actor testing himself on compliance might not always produce true and objective results [23]. An EA compliance assessment or audit should therefore be performed by other individuals and preferably other organizational units than those carrying out the project that needs to conform to EA. In the context of this paper, this

[2] Note that the EA Compliance Model of section 4 has been simplified for didactic purposes, as it can only contain the checks at the level of an individual Prescription. To model the consistency and completeness checks technically correct, a Project Prescription Group, containing one or more Project Prescriptions, should be added between the Project Prescription and the ternary association.

means that if an enterprise architect actively participates in a project, he or she should not be the one performing the conformance test.

A second requirement is that assessing EA compliance should not be done solely at the end or later stages of the project, since the architectural decisions have already been made by that time. Because such decisions are fundamental, they will be difficult to reverse at a later stage. Therefore, compliance testing should be carried out at moments in the project lifetime when fundamental analysis and design decisions have matured and have been explicitly stated, but not yet implemented. This way, deviations from the architecture can be identified while there are still opportunities to correct them. Therefore, there should be multiple Baselines. Ideally, when creating these Baselines, the project has already consulted an enterprise architect [7].

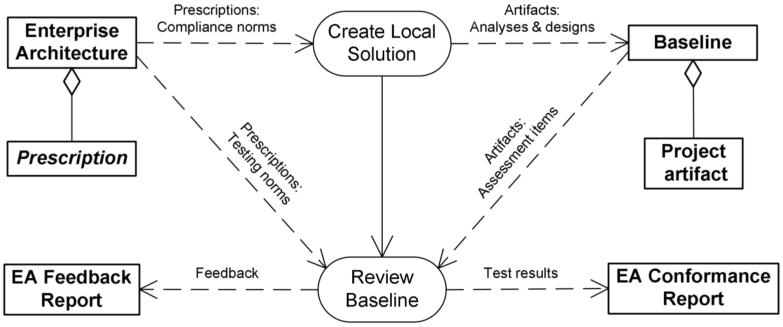

Fig. 3. Role of EA and project artifacts in carrying out projects and compliance assessments

Figure 3 shows the relationships between creating the project artifacts and assessing them on compliance. It shows prescriptions having two roles, as steering norms and as testing norms. The process “Create Local Solution” is described in [7]. “Review Baseline” represents the compliance assessment process. We have modeled the steps of this testing process in detail. Due to space restrictions, this can be found in [29] as the “Review baseline” action, which meets the requirements mentioned above. The assessment is performed by the enterprise architect role, which is considered external to the project. In addition, the use of three baselines shows that the assessment can be carried out at three moments in the project’s lifetime: after business analysis and design, after specification of functional requirements and software architecture, and after delivery of the final product. These baselines are defined in terms of the project artifacts of a standard software engineering approach.

7 Empirical Validation

To validate and illustrate the compliance checks of section 5 and explore the EA testing process, we tested two real-life projects on compliance with EA. The assessments were carried out at Statistics Netherlands (SN), the Dutch central statistical office.

SN has been developing its architectural practice since 2006. Conformance to its EA is relevant to projects, since it provides them with free IT-resources (including an adjunct team of experienced redesign architects cooperating with the project members).

To identify the arbitrary aspects of testing, both assessments were carried out independently by the two principal researchers. Contact between the testers only occurred before and after a test (to compare results and update the operationalization), not during it. The *assessment items* in both projects consisted of a business analysis and design baseline. Therefore, only the business and information prescriptions were used as *compliance norms*. The prescriptions here took the form of textual principles.

As part of the preparation of the *compliance test*, the two testers discussed the EA principles and elaborated on the rationale or implication where needed. Following this, the first test was carried out. As an example, Table 1 presents an excerpt from one of the tester's reports (note that one full report comprised 21 cells).

Prescription		Compliance Check Results			
		Correctness	Justification	Consistency	Completeness
1	The statistical production shall be output-focused and cost aware.	!	!	▨	×
4	Processes concerning the management function shall be distinguished from all other processes.	×	×	▨	
6	Data suitable for re-use shall be identified and stored in enterprise-wide steady state data stores.	!	✓	!	
7	Metadata and (anonymized) data stored in steady state data stores shall be standardized, easily discovered and publicly accessible within SN.	✓	✓		
9	Quality versions of steady state data stores shall be identifiable as versions of one data store.	✓	✓	▨	
Final judgment: Not passed yet. Especially regarding management, key elements are missing.					
Symbols: ✓ passed ! Needs attention × failed ▨ not applicable					

Table 1. The compliance check results per prescription.

Despite the joint preparation, the first test yielded the surprising result that, with only 3 identical scores, there was even less agreement between the two testers than could be expected on the basis of chance alone.[3] In addition, 6 scores showed "extreme" differences, i.e. passed versus failed values. For further analysis, Cohen's Kappa was calculated, which is a statistical measure for determining the agreement between two raters. It has a value between -1 and 1, with the former representing total disagreement and the latter total agreement. With the first assessment's Kappa having a value of -0.086 and a p-value of 0.383, we have to conclude the two testers agreed no more or less than if they had performed the assessment randomly. Post-assessment discussions revealed that the inter-rater differences were due to prescriptions,

[3] Using a binomial distribution and no empirical data, the number of expected agreed-upon ratings can be calculated as: $E = n \cdot p = 21 \cdot 0.25 = 5.25$ expected identical scores.

compliance checks and the business analysis artifact all being interpreted differently. Although a conclusion was that strict operational definitions were necessary, the four types of compliance checks were deemed useful. No additional compliance check types were required to be able to perform the assessment.

Following the first test, improved operationalizations of both the compliance checks and the prescriptions were created. The second test consequently resulted in a significant increase of agreement, with 14 identical scores, no "extreme" differences, a Kappa value of 0.520 and a p-value of <0.001. Although statistically significant and thus not attributable to chance, this value for inter-rater reliability is still far from total agreement and therefore only moderate [32]. While discussing the results, it became clear that the deviating scores could still be attributed to the remaining subjectivity of the prescriptions and business analysis artifact, but no longer to a different interpretation of the compliance checks (the most recent operationalizations of which can be found in [29]). Hence, our approach was applied successfully in this real-life project.

7.1 Discussion of research results

Our research sheds light on the aspects of compliance testing that are specific to Enterprise Architecture. The results indicate that testing on compliance with EA is inherently subjective and interpretive in nature, similar to judicial decisions and academic peer reviews. There are several reasons for this. First, EA prescriptions often prove to be inherently *abstract*, which is a consequence of their strategic nature and of them aiming at a partly unknown future. This renders prescriptions open to interpretation. Creatively interpreting and translating EA prescriptions to fit them in the specific situation is inherent in working with EA. Second, since EA prescriptions and project artifacts have to be read and applied by human actors (analysts, testers, programmers, managers and other stakeholders), *natural language* is the most appropriate format. Natural language, however, is always open to interpretation. Third, when discussing the tests we discovered that we (unconsciously) had used not only the information provided by the artifacts and the EA, but also *personal and contextual knowledge*, e.g. previous experiences with the domain in question that helped understand the assessed baseline. In short, testing requires sense-making, intuition, experience and knowledge of the business context. For example, principle 6 not only requires knowledge of existing and potentially re-usable statistical data and IT systems (inside and outside SN), but also of the goals and requirements of the specific project in order to make a match between these re-usable resources and project needs.

There are indications that these issues causing subjectivity in EA compliance testing are not solely present in the organization in which we did our empirical research. Take for example almost any of TOGAF's example set of architecture principles [31] to see the above-mentioned abstract and vague characteristics (e.g. "Data is an asset that has value to the Enterprise and is managed accordingly.").

What can be done to mitigate the effects of the interpretive nature of abstract EA prescriptions? First, our results suggest that the prescriptions need to be as operationalized as possible, similar to making concepts in social science research measurable. This renders prescriptions less prone to individual interpretation. The pseudo-formalizations can be based on real-life tests, limiting operationalizations to relevant issues. Second, the (important) projects could be assessed by two testers, and their joint EA Compliance Report could be reviewed by the lead enterprise architect. This is not only recommended to increase the objectivity of the assessment, but also to boost acceptance of the test by the project members (who will be aware of the interpretive nature of the prescriptions, as they have applied them). Finally, the EA Conformance Report itself should be reviewed and discussed with the authors of the baseline, in order to prevent erroneous check results and judgments. This step has been added in the "Review Baseline" process model of [29].

The results of our study also have ramifications for automated compliance testing. This is a popular topic in many publications on compliance with standards and legislation [cf. 12, 14, 15, 19, 22]. Indeed, it is feasible to perform all sorts of checks on documents, models and datasets. Especially when the mere existence of properties can be objectively measured, e.g. compliance with the standard "each user requirement includes a measure of priority" [cf. 12, 22]. However, our research leads us to suspect that EA prescriptions are less suitable for automated compliance testing. The above-mentioned characteristics of prescriptions severely hinder automated checking. Prescriptions are written in natural language, they are often inherently abstract and have been translated to the local situation. Furthermore, testing them often requires knowledge that is out-of-scope for machines, for example domain knowledge or information about the non-automated or non-modelled business or its context. Formalizing this might prove impossible or not worth the effort.

We consider it therefore likely that tools (at least in the short-term) will not be able to meaningfully test a substantial part of business processes and IT systems on EA conformance automatically. However, there definitely are areas in compliance testing that could be supported by tools. For example, the operationalization of the checks in [29] defines strict constraints for determining the compliance checks' values. These 'meta checks' can be carried out by a tool for recording the values. Furthermore, tools could provide valuable assistance for registering compliance issues and automated calculation of 'compliance scores'.

Another discussion altogether is the question of whether all four types of checks should always be reported for each prescription (set). If an EA contains many prescriptions, then it might be practical to see the checks as aspects to be kept in mind, and only report about an aspect if it has compliance issues. Also, it might be possible to perform the correctness and justification checks at the same aggregated level as the consistency check, allowing for a more superficial test when time is an issue.

8 Conclusion

We set out to explore how projects, and the business and IT solutions they deliver, can be assessed on compliance with EA. Our research has yielded several contributions. First, we presented the EA Compliance Model, which identifies the high-level core elements of compliance testing. Second, we discussed the testing process and the compliance checks performed therein. We empirically demonstrated that our approach can be used to test real-life projects. Lastly, we discussed the inherently subjective nature of EA compliance testing. Due to the abstract and strategic nature of EA prescriptions, the use of natural language and the need for contextual knowledge, formalized, objective and automated assessments are not to be expected in the short term. We expect more from operationalizing compliance norms for human-based compliance tests. However, we presented several options for tools supporting such assessments. This is an area for further research.

As our empirical research used principles, another topic for further investigation is studying whether our conclusions for principles also hold for models. Other aspects that deserve further research are arriving at optimal operationalizations for human-based compliance assessments. As our empirical research has focused on assessing business analysis artifacts, yet another interesting final topic is testing real IT baselines. It could be, for example, that EA prescriptions focusing on IT are less abstract. A final topic would be to investigate the role of tacit knowledge in testing, which could focus on developing shared implicit meanings regarding prescriptions, rather than on explicit operational definitions. Whatever the topics in future studies, our research clearly shows that objective compliance testing cannot be taken for granted.

Acknowledgements. The authors wish to thank Abby Israëls, Lieneke Hoeksma, Marlies van Steenbergen, Wiel Bruls, Tjalling Gelsema, Robbert Renssen and Peter van Nederpelt for their valuable remarks.

References

1 Johnson, P., Ekstedt, M., Silva, E., Plazaola, L.: Using Enterprise Architecture for CIO Decision-Making: On the Importance of Theory. In: Proceedings of the Second Annual Conference on Systems Engineering Research (2004)

2 Riempp, G., Gieffers-Ankel, S.: Application Portfolio Management: A Decision-oriented View of Enterprise Architecture. Information Systems and e-Business Management, Vol. 5, Issue 4, pp. 359-378 (2007)

3 Gammelgård, M. Simonsson, M., Lindström, Å.: An IT management assessment framework: evaluating enterprise architecture scenarios. Information Systems and e-Business Management, Vol. 5, Issue 4, pp. 415-435 (2007)

4 Kaisler, S.H., Armour, F., Valivullah, M.: Enterprise Architecting: Critical Problems. In: Proceedings of the 38th Hawaii International Conference on System Sciences (2005)

5 Op 't Land, M., Proper, H.A.: Impact of Principles on Enterprise Engineering. In: Proceedings of the 15th European Conference on Information Systems (2007)

6 Bommel, P. van, Buitenhuis, P.G., Hoppenbrouwers, S.J.B.A., Proper, H.A.: Architecture principles – A regulative perspective on enterprise architecture. In: Proceedings of the EMISA 2007 workshop, Lecture Notes in Informatics, Vol. 119, pp. 47-60 (2007)

7 Foorthuis, R.M., Brinkkemper, S., Bos, R.: An Artifact Model for Projects Conforming to Enterprise Architecture. In: Stirna, J., Persson, A. (Eds.). The Practice of Enterprise Modeling. Proceedings of PoEM 2008, IFIP WG 8.1 Working Conference, LNBIP 15, pp. 30-46. Springer, Berlin (2008)

8 Hoogervorst, J.A.P., Dietz, J.L.G.: Enterprise Architecture in Enterprise Engineering. Enterprise Modelling and Information Systems Architectures, Vol. 3, No.1, pp. 3-13 (2008)

9 Foorthuis, R.M., Brinkkemper, S.: Best Practices for Business and Systems Analysis in Projects Conforming to Enterprise Architecture. Enterprise Modelling and Information Systems Architectures, Vol. 3, No. 1, pp. 36-47 (2008)

10 Gauthier, D.: Why Contractarianism? In: Vallentyne, P. (ed.). Contractarianism and Rational Choice, pp. 15-30. Cambridge University Press, Cambridge (1991)

11 Hartman, E.M.: Organizational Ethics and the Good Life. Oxford University Press (1996)

12 Emmerich, W., Finkelstein, A., Montangero, C., Antonelli, S., Armitage, S., Stevens, R.: Managing Standards Compliance. IEEE Transactions on Software Engineering, Vol. 25, No. 6, pp. 836-851 (1999)

13 Kim, S.: IT compliance of industrial information systems: Technology management and industrial engineering perspective. The Journal of Systems and Software, Vol. 80, No. 10, pp. 1590-1593 (2007)

14 El Kharbili, M., Stein, S. Markovic, I., Pulvermüller, E.: Towards a Framework for Semantic Business Process Compliance Management. In: Proceedings of GRCIS 2008, Caise Workshop on Governance, Risk and Compliance of Information Systems (2008)

15 Sadiq, S., Governatori, G., Naimiri, K.: Modeling Control Objectives for Business Process Compliance. In: Proceedings of the 5th International Conference on BPM, Springer (2007)

16 Massacci, F., Prest, M., Zannone, N.: Using a Security Requirements Engineering Methodology in Practice: The Compliance with the Italian Data Protection Legislation. In: Computer Standards & Interfaces, Vol. 27, No. 5, pp. 445-455 (2005)

17 zur Muehlen, M., Indulska, M., Kamp, G.: Business Process and Business Rule Modeling Languages for Compliance Management: A Representational Analysis. In: Proceedings of the 26th International Conference on Conceptual Modeling, Auckland, New Zealand (2007)

18 Nouwt, S.: Reasonable Expectations of Geo-Privacy? SCRIPTed, Vol. 5, No. 2, pp. 375-403 (2008)

19 Emmerich, W., Finkelstein, A., Montangero, C., Stevens, R.: Standards Compliant Software Development. In: Proceedings of the ICSE Workshop on Living with Inconsistency (1997)

20 Barr, M.S., Miller, G.P.: Global Administrative Law: The View from Basel. The European Journal of International Law, Vol. 17, No. 1, pp. 15-46 (2006)

21 Pfleeger, S.L., Fenton, N., Page, S.: Evaluating Software Engineering Standards. IEEE Computer, Vol. 27, No. 9, pp. 71-79 (1994)

22 Chung, P.W.H., Cheung, L.Y.C., Machin, C.H.C.: Compliance Flow – Managing the compliance of dynamic and complex processes. Knowledge-Based Systems, Vol. 21, No. 4, pp. 332-354 (2008)

23 Solms, S.H. von.: Information Security Governance - Compliance management vs operational management. Computers & Security, 24, pp. 443-447 (2005)

24 Drew, M.: Information risk management and compliance – expect the unexpected. BT Technology Journal, Vol. 25, No. 1 (2007)

25 Vroom, C., Solms, R. von.: Towards Information Security Behavioural Compliance. Computers & Security, 23, pp. 191-198 (2004)

26 Lankhorst, M. et al.: Enterprise architecture at work. Modelling, communication and analysis. Springer, Berlin (2005)

27 Baresi, L., Pezzè, M.: An Introduction to Software Testing. Electronic Notes in Theoretical Computer Science, Vol. 148, No. 1, pp. 89-111 (2006)

28 IEEE: IEEE St. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, New York (1990)

29 Foorthuis, R.M., Brinkkemper, S., Hofman, F.: A Process Model for Project Members Conforming to Enterprise Architecture. Technical Report, Utrecht University (2009) URL: http://www.cs.uu.nl/research/techreps/repo/CS-2008/2008-023.pdf

30 Jonkers, H., Lankhorst, M.M., Doest, H.W.L. ter, Arbab, F., Bosma, H., Wieringa, R.J.: Enterprise architecture: Management tool and blueprint for the organisation. Information Systems Frontiers, Vol. 8, No. 2, pp. 63-66 (2006)

31 The Open Group: TOGAF. Version 8.1 "Enterprise Edition" (2003)

32 Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics, Vol. 33, No. 1, pp. 159-174 (1977)