# Do IT Architecture Principles Contribute to IT System's Requirements Realisation?

## *Research Definition of Measuring the Contribution of IT Architecture Principles to the Realisation of IT System's Requirements*

Michiel Borgers
*School of Business and Economics, Maastricht University, Tongersestraat 53, 6211 LM, Maastricht, The Netherlands*

## 1 RESEARCH PROBLEM

We want IT systems achieve their objectives, hence during development we realise the IT system requirements. The aim of IT architecture is to focus on those essential requirements so the IT system will fit for its purpose (The Open Group, 2011), (Department of Defense, 2006), (Greefhorst and Proper, 2011). Architects guide the development of IT systems using IT architecture principles. Because IT architecture principles have a key role in guiding the design and hence the implementation of the IT system's requirements (Greefhorst et al., 2013), it is important to investigate their contribution. We did find literature about IT architecture principles theory, but no literature about empirical research as to how IT architecture principles contribute to realising system requirements.

In the current practise we do see an increase of IT systems. The enormous impact of IT technology on our organisations is addressed in many publications as in Gartner's Executive Reports, by Westerman (Westerman et al., 2014) and many others. Also the Dutch government, for example, introduced the Digital Government strategy to be implemented by 2017 (Plasterk, 2013).

Unfortunately, the current practice shows that IT projects fails in 34% of the cases worldwide, with requirements issues as an important reason (El Emam and Gunus Koru, 2008). For instance the Dutch Parliament reports ICT project failures with a loss of 1 to 5 billion euro's a year, with lack of good architecture as one of the perceived reasons (Elias, 2015).

So, although theory claims IT architecture principles are the cornerstone of the IT architecture and therefore important for IT systems achieving their objectives, we define the following research question:

"Do IT architecture principles contribute to IT system requirements realisation?"

## 2 OUTLINE OF OBJECTIVES

To answer our research question we use the following assumption:

- If the (essential) requirements, defined by stakeholders who are involved in using the IT system, are realised;
- And in the development of the IT system the IT Architecture principles are defined and used;
- Then there is a positive correlation between the use of IT architecture principles and the IT system requirements realisation. So then we can state that the IT architecture principles do (directly or indirectly) contribute to the IT system requirements realisation.

To investigate both the implementation of the IT system's requirements and the use of the IT architecture principles, we have to measure this using empirical research.

Therefore we have derived our main research question into three sub-questions:

1. How can we define and measure the implementation of the IT system requirements?
2. How can we define and measure IT architecture principles?
3. What is the correlation between IT architecture principles and the realisation of the IT system's requirements?

So the objective of this research is to answer these three sub-questions. In order to answer sub-questions one and two, we need to develop instruments for measuring both the IT system's requirements realisation and the IT architecture principles. With these instruments we can do empirical research and with the collected data we can try to have insight in the correlation between



Borgers, M.
Do IT Architecture Principles Contribute to IT System's Requirements Realisation? - Research Definition of Measuring the Contribution of IT Architecture Principles to the Realisation of IT System's Requirements.
In *Doctoral Consortium (DCEIS 2016)*, pages 3-8

those subjects of research.

# 3 STATE OF THE ART

In this section we will define and explain the current state of literature about IT systems, requirements and IT architecture principles.

## 3.1 IT Systems Defined

As it is a central concept in our research, we first define the concept of IT system in more detail. We are aware of the idea that IT systems can be seen as an integrated part of an ecosystem, and related to the actor-network theory described by Latour (Latour, 2005). In this research, though, we use the systems theory of Churchmann (Churchman, 1971), where an IT system is a separate entity affected by its environment. We choose this approach because we want to measure to which degree the IT system meets its requirements. Therefore, we need a defined scope of the IT system. With the system theory in mind we pose that the influence of the outside world is traced back to the requirements and the IT system itself.

Exploring the IT system in more detail, we distinguish three dimensions: the functionality of the IT system, its physical components, and its IT life cycle (De Leeuw, 1990), (Dietz, 2001).

In the functional dimension we focus on what the IT system should do from an end user perspective. The functionality of the IT system consists of the functions and the characteristics of the IT system, like response times or security levels.

If we look in an analytical way to an IT system, we consider the physical components of the IT system and how they are structured. Dietz calls it the construction of the system (Dietz, 2001), (Dietz, 2010) and in architecture frameworks like Zachmann (Zachman, 1987) and IAF (Van 't Wout et al., 2010) it is the physical level focusing on "with what" products and technologies. We define a component as an implementation unit of technology that provides a coherent unit of functionality, equivalent to the definition of a software component by Clements (Clements et al., 2002). In IT systems those components are hardware and software components.

The IT life cycle of an IT system is the change of the IT system over time. There are many IT life cycle models in literature (The Open Group, 2011), (ISO 2008), (Looijen, 1998). All IT life cycle models have phases like initiation, design, implementation, utilization, maintenance, and destruction. Definitions of phases are of course also depending on the level of elaboration of specific models, like Agile, Waterfall, etc. Key in this research is the distinction between the development stage and the usage stage. In the development stage an IT system is initiated, designed and implemented, while in the usage stage the IT system is maintained and utilized. The distinction is important because in the development stage the requirements are defined and implemented, while only in usage stage one can identify whether the objective is achieved (Thorp and Leadership, 2003).

So, in this research we define an IT system as "an entity with functions and characteristics, consists of hardware and software components and being in the development or in the usage stage to fulfil requirements to achieve an objective".

## 3.2 Requirements

An IT system should meet certain requirements to achieve the objective of the IT system. Requirements describe what kind of properties the IT system must have, directed by the stakeholder. Therefore a requirement can be defined as: a property that (a part of) the IT system must posses, consistent with Greefhorst (Greefhorst and Proper, 2011) and Paper (Paper and Wand, 2007) and is an abstract of the definition of IEEE (IEEE, 1990).

Although there are all kinds of requirements, within the scope of an IT system there are three coherent types of requirements: user requirements, system requirements and transition requirements.

Users define the functionality of the IT system through user requirements. User requirements can be divided into two types of requirements: functional requirements and Quality-of-Service (QoS) requirements (IIBA, 2015). A functional requirement defines a function of an IT system. QoS requirements describe the preconditions and characteristics of the IT system.

System requirements are the specifications of the IT system "how to intent to build it" (Zachmann, 2015) and describes what constructional properties the components should have from the perspective of the people who build the IT system (Greefhorst and Proper, 2011).

Transition requirements are "a classification of requirements that facilitate transition from the current state to the desired future state, but that will not be needed once that transition is complete' (IIBA, 2015).

During the development stage those

requirements are defined and implemented. The objective of the development stage is to implement those requirements as accurately as possible so in the usage stage the IT system will meet its overall objective. Requirements can be prioritised, so during IT system development the most important requirements will be implemented first. A well-known way of prioritising requirements is the MoSCoW method (IIBA, 2015).

Although we want the most requirements to be implemented, the current practice shows that the implementation (partly) fails in 34% of the cases worldwide (Emam and Koru, 2008) with requirements issues as an important reason. Especially for large government software projects the failure rate is 29% (Standish Group, 2015) and in the Dutch government it is a failure rate of 36% (Elias, 2015). So we want to measure to which extent requirements of the IT system are implemented, to get a better insight and control over the implementation of the requirements.

In the development stage testing is the instrument to indicate the extent to which an IT system meets its requirements. The objective of testing is to measure the quality and risk of failure of a specific IT system before the usage stage (Kaner et al., 1999).

Although testing results are a good indicator, there are some disadvantages. Testing cannot identify all the defects because not all input or output scenarios can be tested in advance (Pan, 1999). Moreover, not all conditions possible during the usage stage can be simulated during development (Kaner et al., 1999). And only during the usage stage the stakeholders can actually use the IT system, and able to confirm the IT system is meeting their requirements. In general, during this stage no measurements are taking place to which extent the IT system meets the defined requirements.

DeLone and McLean (DeLone and McLean, 2003), however, define variables, like system quality and user satisfaction, which are important to achieve the net benefits for the business in the usage stage. Some of those variables are directly related to the IT system as defined in this research. Other research (Ravichandran and Lertwongsatien, 2005) found that Information systems have a direct effect on firm performance. In this research too, one variable is directly related to the IT system itself.

So we conclude there are many ingredients to measure to some extent the realisation of the IT system requirements, but an overall measurement model is still lacking.

## 3.3 IT Architecture Principles

Our research focuses on the essential requirements and the fundamental organisation of the IT system, in order to achieve fitness for purpose. The essential requirements are those requirements, which are key to achieve the purpose of the IT system and are defined by the key users and/or management. IT Architecture is used to describe key requirements in relation to an IT system that is fit for purpose (The Open Group, 2011), (IEEE, 2000), (Slot, 2010).

IT architecture gives guidance to the design of the IT system. The design of an IT system is "a specification of an object, manifested by a design agent, intended to accomplish goals, in a particular environment, using a set of primitive components, satisfying a set of requirements, subject to constrains." (Ralph and Wand, 2007). Or in more popular words of the Oxford dictionary "the plan or drawing produced to show the look and function or workings of an object before it is made" (Oxford, 2015). So, as mentioned by Dietz (Dietz, 2008), (Dietz, 2004) "architecture is the normative restriction of design freedom" because its guiding the design into a specific direction.

To do so, a set of architecture concepts is available, like models, views, frameworks and architecture principles (Land et al., 2008). Architecture principles, first introduced in the PRISM report in 1986 (PRISM, 1986), are the interface between the objectives of the key users and the design of the IT system. They have a key role in guiding the design and therefore they are the cornerstones of the architecture as described by many authors and summarised by Greefhorst (Greefhorst and Proper, 2011). Because IT architecture principles have a key role in guiding the design and hence the implementation of the IT system's requirements, it is important to investigate their contribution.

Equivalent to Greefhorst's generic definition of architecture principles, we will define an IT architecture principle as: "A declarative statement that normatively prescribes a property of the design of an IT system, which is necessary to ensure that the IT system meets its essential requirements."

Although we now have defined the concept of IT architecture principles, we still have to specify them. Both in theory and practice there is no universal agreement on how to specify IT architecture principles (Greefhorst and Proper, 2011). IT architecture principles at least consists of the three attributes statement, rationale, and implications (The Open Group, 2011), (Greefhorst and Proper, 2011),

(Greefhorst et al., 2013). Besides these three basic attributes there are many other attributes and dimensions to address IT architecture principles, like applicable situations, owner, related principles, quality criteria, etc. (Greefhorst & Proper, 2011). So, no universal structuring will help in categorising IT architecture principles.

Moreover, research about architecture principles in general shows that there is not yet quantified empirical evidence about the effect of the use of IT architecture principles in realising the IT systems requirements (Stelzer, 2009), (Greefhorst et al., 2013).

So we now can conclude that in theory the use of IT architecture principles is an important instrument to contribute to the implementation of the IT system's requirements. But no empirical evidence is available IT architecture principles have indeed added value in implementing the most important requirements.

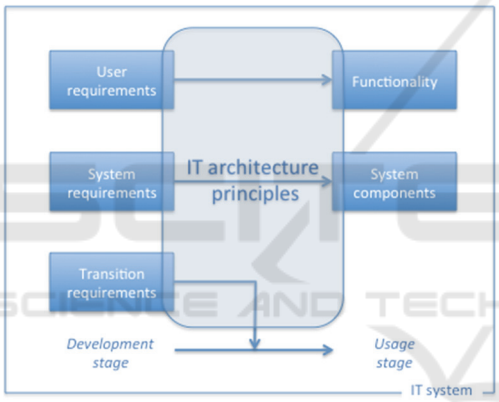


Figure 1: IT architecture principles related to the dimensions of the IT system.

## 4 METHODOLOGY

In order to answer our research question we will use the following research approach:

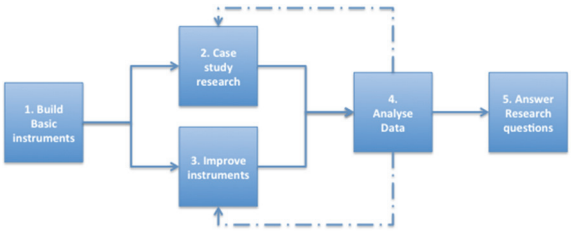


Figure 2: Research approach.

In the first step we build basic measurement instruments for both the implementation of the IT system requirements as IT architecture principles, based on literature review.

We do case study research to collect empirical data in step two. In this step we use the case study method because theory on IT architecture principles is limited. Case study research is in this situation a suitable research method (Yin, 2013), (Eisenhardt, 1989), (Harris and Sutton, 1986), (Gersick, 1988).

Meanwhile we improve the measurement instruments in step three based on additional literature review and the experimental data out of the case studies.

In step 4 we analyse and challenge the quality of the empirical data out of the different case studies. We expect to do a couple of iterations back to step two and three to achieve the right quality level of the data.

In the final step we answer the research questions based on the analysis made and draw final conclusions, limitations and recommendations for further research.

If possible, we choose a quantitative approach to obtain a more objective insight in both research subjects. With a quantitative approach we can easier compare and show easier correlations between variables than with a qualitative one (Babbie, 2015), (Muijs, 2004). If necessary we combine the results with qualitative data to address the context of the results (Mark and Caputi, 2001), (Bryman, 2009).

## 5 EXPECTED OUTCOME

We expect a positive correlation between IT system's requirements realisation and IT architecture principles in general, and we are interested in the in-depth details of this correlation. For instance, do architecture principles that are described in a SMART way have a higher success rate in terms of realising requirements? Or: Is it harder to realise QoS requirements than system requirements using IT architecture principles?

## 6 STAGE OF THE RESEARCH

We started this research with literature review to define the appropriate research question. Secondly we used the literature review to develop the measurement instruments for both the IT system's requirements and the IT architecture principles, defined as step one in our research approach.

Our next step was to do some empirical research to test our measurement instruments. We did four case studies at the Dutch Tax Agency at the end of 2015, to measure both the IT system's requirements and the IT architecture principles. We are now investigating the results of those case studies and the results will be described in a paper in the coming months.

# REFERENCES


Van't Wout, J. et al., 2010. *The Integrated Architecture Framework Explained - Why, What, How*, Springer Berlin Heidelberg.

Babbie, E., 2015. *The Practice of Social Research*, Cengage Learning, Inc.

Bryman, A., 2009. Integrating quantitative and qualitative research: how is it done? V. L. Pl. C. J. W. Creswell, ed. *Qualitative Research*, 6(1), pp.97–113.

Churchman, C.W., 1971. *The Design of Inquiring Systems: Basic Concepts of Systems and Organization: Charles West*, New York: Basic Books.

Clements, P. et al., 2002. Documenting Software Architectures: Views and Beyond.

DeLone, W.H. & McLean, E., 2003. The DeLone and McLean model of information systems success: A ten-year updated. *Journal of Management Information Systems*, 19(4), pp.9–30.

Department of Defense, 2006. Department of Defense Chief Information Officer Desk Reference. *Chief Information Officer Desk Reference*, I(August).

Dietz, J.L.G., 2008. *Architecture - Building stategy into design*, Academic Service.

Dietz, J.L.G., 2001. DEMO: Towards a discipline of organisation engineering. *European Journal of Operational Research*, 128(2), pp.351–363.

Dietz, J.L.G., 2004. Extensible Architecture Framework, The NAF Program GAF.

Dietz, J.L.G. (editor), 2010. Enterprise Engineering Toolset. *Engineering*, (January), pp.1–2.

Eisenhardt, K.M., 1989. Building Theories from Case Study Research. *Academy of Management Review*, 14(4), pp.532–550.

Elias, T., 2015. *Parlementair onderzoek naar ICT-projecten bij de overheid*, Tweede Kamer der Staten Generaal, (In Dutch).

El Emam, K. & Gunus Koru, A., 2008. A replicated survey of IT Software Project Failures. *IEEE Software*, pp.84–90.

Gersick, C., 1988. Time and Transition in Work Teams: Towards a New Model of Group Development. *Acedamy of Management Journal*, 31, pp.9–41.

Greefhorst, D. & Proper, E., 2011. *Architecture Principles the Cornerstones of Enterprise Architecture*, Springer Berlin Heidelberg.

Greefhorst, D., Proper, E. & Plataniotis, G., 2013. The Dutch State of the Practice of Architecture Principles. *Journal of Enterprise Architecture*, p.6.

Harris, S.G. & Sutton, R.I., 1986. Functions of Parting Ceremonies in Dying Organizations. *Academy of Management Journal*, 29(1), pp.5–30.

IEEE, 2000. *IEEE SA - 1471-2000 - IEEE Recommended Practice for Architectural Description for Software-Intensive Systems*.

IEEE, 1990. IEEE Standard Glossary of Software Engineering Terminology. , pp.1–84.

IIBA, 2015. A guide to the Business Analysis Body of Knowledge (BABOK Guide).

ISO, ISO/IEC 12207:2008 - Systems and software engineering -- Software life cycle processes.

Kaner, C., Falk, J. & Nguyen, H.Q., 1999. *Testing Computer Software* 2nd Editio, Wiley.

Land, M.O. et al., 2008. Enterprise architecture: creating value by informed governance. Springer.

Latour, B., 2005. *Reassembling the Social: An Introduction to Actor-network-theory*, Oxford University Press.

De Leeuw, A.C.J., 1990. *Organisaties: Management, analyse, ontwerp en verandering*, Assen: Van Gorcum & Comp.

Looijen, M., 1998. *Information Systems: Management Control and Maintenance* 1st edition, SDU.

Mark, B. & Caputi Peter, 2001. Introduction to quantitative research. *SAGE publication Ltd*, p.272.

Muijs, D.D., 2004. *Doing Quantitative Research in Education: with SPSS*, SAGE Publications.

Oxford Dictionaries - Dictionary, Thesaurus, & Grammar. Available at: http://www.oxforddictionaries.com/ [Accessed December 21, 2015b].

Pan, J., 1999. Software Testing.

Plasterk, R., 2013. Visiebrief digitale overheid 2017.

PRISM, 1986. *Dispersion and Interconnection: Approaches to Distributed Systems Architecture*, Final Report. Technical Report, CSC Index, Inc and Hammer & Company, Cambridge MA.

Ralph, P. & Wand, Y., 2007. *A Proposal for a Formal Definition of the Design Concept*, In Design Requirements Engineering: A Ten-Year Perspective, Volume 14, pp.103–136.

Ravichandran, T. & Lertwongsatien, C., 2005. Effect of Information Systems Resources and Capabilities on Firm Performance: A Resource-Based Perspective. *Journal of Management Information Systems*, 21(4), pp.237–276.

Standish Group, 2015. *Haze*, Available at: https://www.standishgroup.com/sample_research_files/Haze4.pdf [Accessed January 18, 2016].

Slot, R., 2010. A method for valuing Architecture-Based Business Transformation and Measuring the value of Solutions Architecture. University of Amsterdam.

Stelzer, D., 2009. Enterprise architecture principles: literature review and research directions. *Proceedings of the 2009 international conference on Service-oriented computing*, pp.12–21.

The Open Group, 2011. *TOGAF® Version 9.1*, Van Haren Publishing.

Thorp, J. & Leadership, F.C.C. for S., 2003. *The Information Paradox: Realizing the Business Benefits*

*of Information Technology*, McGraw-Hill Ryerson.
Westerman, G., Bonnet, D. & McAffer, A., 2014. *Leading Digital Turning Technology into Business Transformation*. Harvard Business Review Press.
Yin, R.K., 2013. Applications of case study research. *Applied Social Research Methods Series*, 34, p.173.
Zachman, J.A., 1987. A Framework for Information Systems Architecture. *IBM Systems Journal 26, No.*, 3(3), pp.276–292.
Zachmann, John A., Conceptual, Logical, Physical: It is Simple Available at: http://www.zachman.com/ea-articles-reference/58-conceptual-logical-physical-it-is-simple-by-john-a-zachman [Accessed December 18, 2015a].