Exploring Normalized Systems Potential for Dutch MoD's Agility

(A Proof of Concept on Flexibility, Time-to-Market, Productivity and Quality)

Martin Op 't Land[1,2,3], Marien R. Krouwel[1],
Edward van Dipten[4], and Jan Verelst[5]

[1] Capgemini Netherlands, P.O. Box 2575, 3500 GN Utrecht, The Netherlands
{Martin.OptLand, Marien.Krouwel}@capgemini.com
[2] Delft University of Technology, Delft, The Netherlands
[3] Technical University of Lisbon, Lisbon, Portugal
[4] Command and Control Support Centre, Defense Material Organization, Ministry of Defense, Bernardkazerne, Amersfoort, The Netherlands
EG.v.Dipten.01@mindef.nl
[5] Department of Management Information Systems, University of Antwerp, Prinsstraat 13, B-2000 Antwerp, Belgium
Jan.Verelst@ua.ac.be

Abstract. Both the Command and Control Support Centre (C2SC) of the Dutch Ministry of Defense, and Capgemini are constantly exploring better ways of building and maintaining information systems which effectively support strategy and operations of an enterprise. In a Proof of Concept conducted in March and April 2011, together with University of Antwerp, they evaluated the possibilities of the Normalized Systems approach for building information systems. In 50 man days not only a system was built that would normally be built in 100-150 man days, also a major technology change had taken place (in week 2 Cocoon and EJB2 were replaced by Struts and EJB3) and changes in the (data) models were instantly processed, even in the last week of development. With these results, Normalized Systems is considered a key enabler in rapid enterprise transformation. Moreover, because Normalized Systems makes it possible to have a continuous link from enterprise (process) modeling to software development and a short feedback loop from system development back to enterprise modeling, it is also considered a key enabler for agile enterprise engineering. This will also enable shortening the typical cycles of enterprise governance.

Keywords: DEMO, Normalized Systems, Agile Enterprise Engineering, Ministry of Defense, case study, SCRUM.

1 Situation

In today's dynamic environment, competitive success depends on being able to quickly respond to changing market, economic and regulatory conditions. Because every enterprise currently is supported by IT, enterprise IT has to be

F. Harmsen, K. Grahlmann, and E. Proper (Eds.): PRET 2011, LNBIP 89, pp. 110–121, 2011.

able to quickly respond and change directions to support critical strategic initiatives and requirements. The increased required flexibility as well as the ever increasing complexity of both enterprises and supporting IT make that software development and maintenance budgets have to grow every year, as was already stated by Lehman in 1980 [1]: "As an evolving program is continuously changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it." Enterprises that decrease — or (even) keep constant — the IT budget will be faced with less satisfactory IT, decreased support of organizational changes, decreased business-IT alignment and decreased *situational awareness*. There is a definite need for a software delivery approach with which high quality IT solutions can be developed and continuously changed quickly.

As an answer to this need and the challenges, University of Antwerp developed the Normalized Systems (NS) theory [2] on *evolvable complexity*. The ambition of the NS concepts and its associated tools, called *expanders*, is to create transparency and controllability in the development and maintenance of software and business processes, ultimately leading to productivity improvements, comparable with the effects of industrialization in other sectors. The theory prescribes a fine-grained modular structure such that one type of anticipated change only impacts one module in the information system. This results in manageable and evolvable complexity; for common maintenance actions it is formally proven that the impact on the system is the same when it is performed now, two years ago, or two years from now when the system has grown and become more complex. Over the last years, the Normalized Systems approach and tools have been used in 12 real-life projects in Belgium, creating a similar number of information systems of which four are mission-critical. University of Antwerp now wanted to share their knowledge and to receive feedback on NS and its supporting tools in a robust way. Such a controlled NS adoption would include applying intake and audit sessions for each real-life project.

As a global leader in consulting, technology, and outsourcing, Capgemini is always interested to collaborate on innovations with potential to create business value; a reason to actively participate in, e.g., the international CIAO! Research Network[1]. When University of Antwerp end of 2009 first explained the concepts of NS to the CIAO! network, Capgemini decided to start exploring the potential of Normalized Systems [3], in combination with the (earlier) proven [4, 5] business agility delivered by using the Design & Engineering Methodology for Organizations (DEMO[2]) [6]. In fall 2010, Capgemini division Custom Software Development (CSD) recognized that NS could increase productivity, flexibility, quality, and maintainability, and could also decrease the time-to-market of IT solutions, which fits in their ambition to make projects highly industrialized and predictable. Capgemini decided to adopt NS, to educate a team in NS, and to start seeking for an opportunity to see hands-on (1) to what extent NS already offers the expected improvement, and (2) what still can be improved on the NS

[1] http://www.ciaonetwork.org/
[2] http://www.demo.nl/

approach and tools. A dialogue between Capgemini and the Dutch Ministry of Defense (MoD) was started.

The Command and Control Support Centre (C2SC) of the Dutch MoD builds mission critical systems for land based operations. In doing so, it acts as supplier for other parties within the Ministry of Defense. Often, COTS and MOTS solutions are integrated with self developed software solutions. In that way, they are also demand party for external vendors. Since every operation differs in terms of people involved and their information and communication needs, C2SC must be able to adapt their existing applications and services or create new ones at high speed, both for current and future operations. In order to do so, C2SC is always looking for new ways to build information systems which (1) offer the freedom to be adapted to changing needs, (2) can be realized quickly, offering a short time to market, and (3) provide high quality, as bugs should be prevented and solved before the operation of the system and not during the operation.

2 Tasks

Both C2SC and Capgemini were curious to test the claims of NS in their own practical situation. Therefore a Proof of Concept (PoC) was conducted, a collaboration between C2SC and Capgemini, supported by University of Antwerp, in which the added value of Normalized Systems in terms of flexibility, time-to-market, productivity, and quality was evaluated, and experience in building a Normalized System was gained, finding indicators for improvement on the Normalized Systems approach and tools. In order to perform the evaluation, an application was build to support the domain of Information Management.

This domain of information management, earlier modeled by C2SC in an enterprise ontology according to DEMO (Figs. 1 and 2) operates as follows. For a successful execution of military operations it is of utmost importance that information products (reports, order, plans) are available valid and on time to planners, decision makers and executors. Easiest is if everyone involved in a military operation has access to all information. However, because of confidentiality but also because of limited available means for information exchange, for every single deployment the information exchange requirements must be determined:

- who offers which information (products)? – InformationProductOffer (IPO);
- who needs which information (products) from whom? – InformationExchangeRequirement(IER);
- which mean (medium) is required to transfer the information?
- which limitations in confidentiality apply for an information exchange?

The application should demonstrate NS capabilities in four main areas:

A1. CRUDS (Create, Read, Update, Delete, Search) screens;
A2. custom screens;
A3. workflow;
A4. export/visualization.

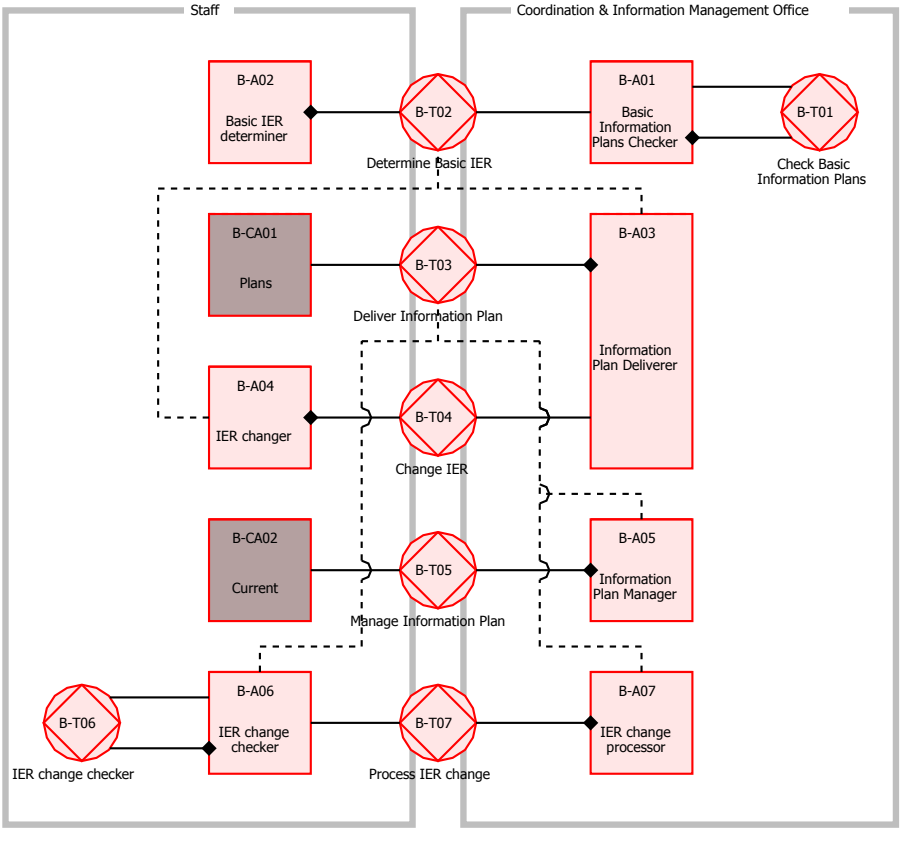

Fig. 1. Information Management MoD: DEMO Construction Model (input version)

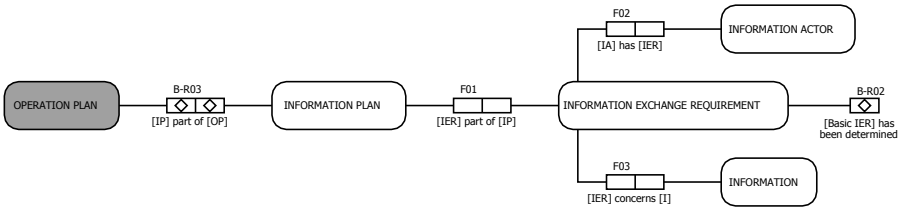

Fig. 2. Information Management MoD: DEMO Fact Model (input version)

Table 1. Planned activities and budget (man days)

Activity	C2SC	Capgemini	UA
Business Modeling	2	1	
Requirements specification	4	9	
Architecture and Design	2	6	
Development	0	8	
Testing	2	7	
Deployment	2	2	
Project Management		7	
Report NS benefits and concerns	2	6	
List of requirements for improvements		2	
Intake / audit / technical support			2
Normalized System training			2
Total	**14**	**48**	**4**

Next to an application, a report about the findings on using the Normalized Systems approach and tools should be delivered. Finally, at the end of the project, University of Antwerp would audit the created application.

Table 1 shows the planned activities and budget in terms of man days expressed in RUP terminology. System development of the project was scheduled from March 16thuntil April 13th, 2011. Deadline for the final report of the findings was scheduled for May 1st(Fig. 3). The developers from Capgemini that worked on this project were only recently trained in the Normalized Systems theory and had no experience with the supporting toolset yet.

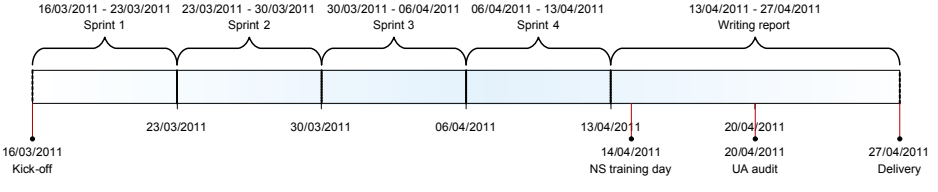

Fig. 3. Timeline

3 Approach

The development part of the PoC was executed with an Agile/SCRUM[3] approach with four *sprint*s of one week (Fig. 4). Every sprint started with a planning session in which the goals, in terms of user stories, for the sprint were defined. At the end of every sprint the realized application was reviewed by subject matter experts of C2SC.

[3] `http://www.scrumalliance.org/`

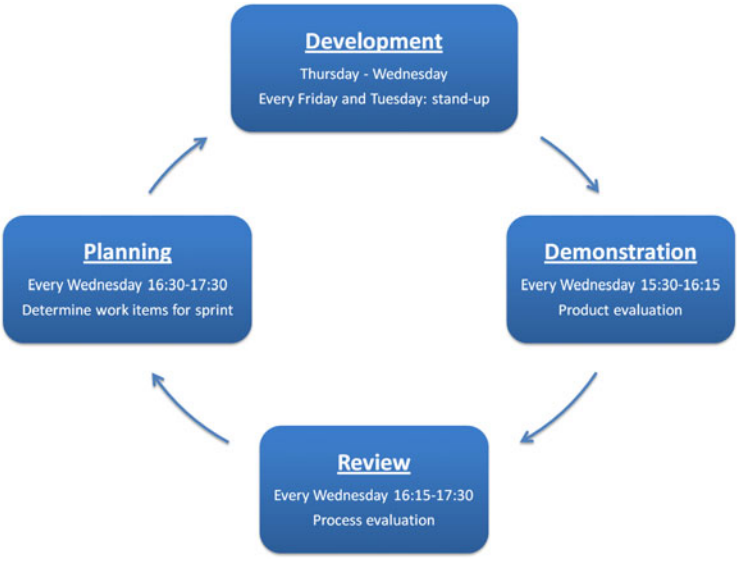

Fig. 4. Sprint activities

Input for the first sprint comprised a DEMO Construction Model (an organization implementation model of a chain of actors, delivering each other services) (Fig. 1) and a DEMO Fact Model (a semantic model of facts created by actors; used as a non-normalized data model) (Fig. 2), complemented by examples of the current IT implementation (Excel sheets). Already on the first day it was discovered the models were incomplete and partly inconsistent with the current operations of Information Management. As soon as the models were clarified, the data model was implemented instantly by the Normalized Systems tools, providing a first application with mainly CRUDS screens (A1) and components for user and profile management. In the second week, custom screens (A2), screens that were not provided by the Normalized Systems tool set, were realized. Because of the clear interfaces the toolset provides, it was easy to build these screens. The NS workflow (A3) mechanism was tested by realizing a basic workflow with which an actor was able to view a list of IERs and accept or reject them. Setting up the workflow turned out to be straightforward as well. An export functionality (A4) was realized to export a set of IERs to XML. Because of the clear interfaces provided by the tool set, it was easy to integrate external packages for the mapping to XML.

In every demonstration session it was discovered the application did not meet the real needs of C2SC yet; often the models had to be adapted and re-implemented in the following sprint. However, changes in the data model were implemented as easily as the first time, without impacting the rest of the application. The impact on custom screens brought about by the changes in the data model proved to be small and traceable. Even changes in the models, made in the last week (see Fig. 6 for the final — normalized — data model), did not

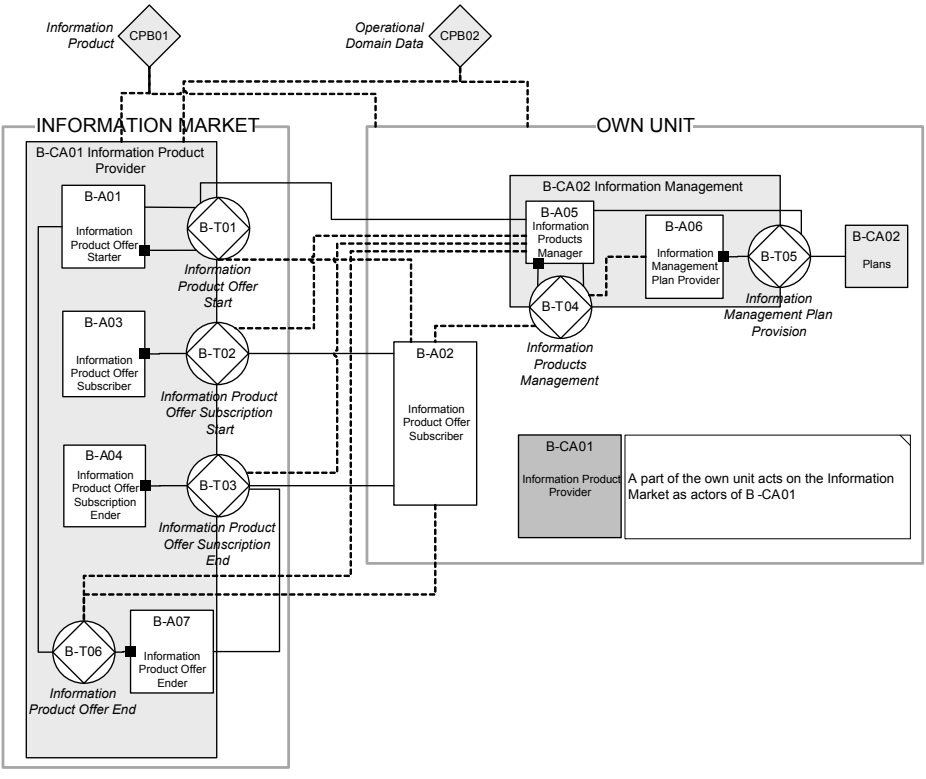

Fig. 5. Information Management MoD: DEMO Construction Model (final version)

impact the realized application with its custom screens, workflow, and export functionality.

Another field explored during this PoC was the migration of the application to another technology stack: within one day the application was migrated from Cocoon/EJB2 to Struts/EJB3, something that usually would not be done during a project, or would have cost at least a man month.

During the project, University of Antwerp supported the Capgemini team of developers in using the tool set for three Fridays, for them also a way to receive feedback on the tool set. At the end of the project UA performed an audit on the realized application, double checking the application is built according to the Normalized Systems principles. They confirmed the application was indeed built according to the Normalized Systems principles, the customizations are well maintainable, and the application thus is able to evolve. Also, UA provided a training on Normalized Systems for some C2SC decision makers in order for them to understand the results of the audit and PoC, and also to enable them starting to visualize the organizational and personnel impact of a — at that moment: hypothetical — Normalized Systems adoption.

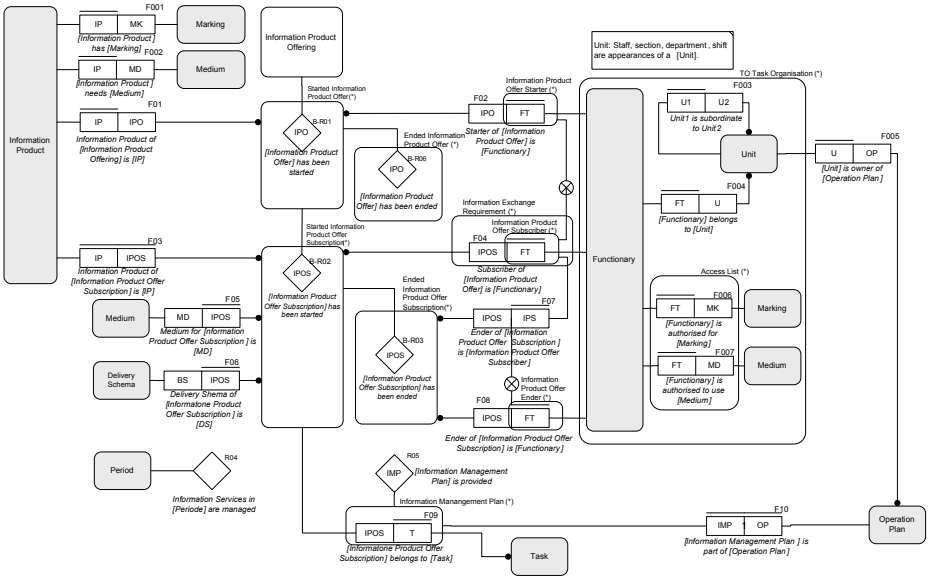

Fig. 6. Information Management MoD: DEMO Fact Model (final version)

In order to measure the productivity gain, both C2SC and Capgemini Accelerated Delivery Center (ADC) performed a NESMA[4] Function Point[5] Analysis (FPA); both parties agreed upon approximately 260 function points. Based on their FPA and their own best practices, both parties provided a time estimation (Table 2).

Table 2. Realized activities by Capgemini (man days)

Activity		**NS (planned)**	**NS (realized)**	**Capgemini (estimated)**	**C2SC (estimated)**
Business Modeling		1	1		
Requirements specification		9	2.5	7	0
Architecture and Design		6	3	34.5	42
Construction	Expanding	4	2.5	44.5	43.5
	Customizations	4	12		
Testing	Unit	2	1	5	5
	Other	5	1	27.5	10.5
Deployment		2	0	3	1
Project Management		7	17.5	16.5	20
Report + List		8	5		
Learning the tool		0	13.5		
Total		**48**	**59**		

[4] http://www.nesma.nl/section/home

[5] http://en.wikipedia.org/wiki/Function_point

4 Results

Although the application was not actually deployed in the organization, its possibilities for the organization can be valued; the current implementation with Excel sheets is definitely less scalable than the created application. Also, the Excel sheets are difficult to maintain and to trace back to the owner. Next, while the Excel sheets only supported the gathering of data, the realized application also provided support for the processes.

The flexibility of the realized application has proven to be good, both for functional changes, such as last-minute changes in the data model, and technical changes, such as a stack change from Cocoon/EJB2 to Struts/EJB3; the impact of changes is small and predictable, implying that maintainability of the application is also ensured. Greater flexibility and maintainability has a positive effect on the time-to-market. Moreover, the quality of the software satisfies the characteristics of evolvable software as described by the Normalized Systems theory.

The productivity gain is hard to measure since the approach — RUP, SCRUM, etc. — highly influences the time needed for an activity. However, there are activities that can be compared, as indicated by a grey shade in Table 2: As the Normalized Systems has its architecture defined in the tool set, only for custom components, not provided by the NS tool set, time has to be spent on Architecture and Design. And since the NS tool set provides standard functionality for, i.e., persistency, remote access, input validation, logging, and user management, only for custom components time has to be spent on Construction. Also, since the provided architecture and standard functionality is already tested thoroughly, only custom components need to be unit tested. On the other hand, less time was spent on Requirements Specification — instead, every requirement was almost instantly implemented. Also, because the project was a PoC, no real functional tests were performed — only during the demonstration sessions. In total, the productivity gain for an entire project, from business modeling until deployment including project management, is estimated at a factor of 2 to 2.5. With these results, Normalized Systems is considered a key enabler in *Rapid Enterprise Transformation*; an interesting next step after Mulder's DEMO-based *Rapid Enterprise Design* [5])!

Short feedback loops are a main feature of iterative (software development) approaches such as SCRUM. However, feedback loops are only useful when feedback can be processed within a reasonable amount of time. Because the Normalized Systems approach uses expanders for code generation, changes in requirements originating from feedback can be processed almost instantly. The short iterations between business modeling and implementation, proved to provide better insight in the organization (see Fig. 5 for the final process model), and enable almost simultaneously (re)engineering the organization and its supporting IT. Some changes cannot always be processed instantly but require more work. However, as with maintainability and flexibility, with the Normalized Systems approach it is very clear what the impact of a change is. Normalized Systems therefore is considered a key enabler in agile system development.

5 Reflection

At the start of the project, it was thought the provided DEMO models represented the organization's needs and the requirements of the information system to be built. However, as soon as implementation started, it was clear the models lacked information. Samples of the current implementation of the enterprise (Excel sheets) were helpful in clarifying the DEMO models and identifying additional needs. However, because the Normalized Systems approach enables quick implementation and demonstration, enabling the organization to provide feedback, it was found the DEMO models were not valid and needed to be adapted. Does this mean DEMO models do not suffice in creating information systems? Or does it mean DEMO models can only be validated by means of implementation, where with Normalized Systems these models can be implemented and thus validated within a significantly shorter period?

A significant part of the budget (13.5 man days out of 59) was spent on getting to know the toolset or solving issues with the toolset, and also the reason for the project to run over budget (11 man days over budget). A major issue in development was the lack of testing possibilities, of both the expander and generated code. A list of recommended improvements on the tool set is composed and is currently being processed by UA and Capgemini in productizing the NS tool set. To be fit for large-scale use in MoD, the NS expanders should also support geographical positioning and editing of data, tabular data entry, and possibilities for generating mobile platform specific code. It is estimated these features can be implemented within six months.

Also, the sprint length of one week was found to be rather short, making it difficult to reach target when activities take some longer than planned and the overhead of collective project sessions and project management activities too large. It is expected that with a team with more experience with the NS tool set, more mature expanders, and longer sprints — of say two weeks — the productivity improvement is even higher. However, the willingness of all people involved to communicate at high frequency and with short response time, was also rated as one of the success factors of this project and also had a major positive influence on the productivity gain. Without commitment of all parties involved, the productivity gain will be much lower.

Based on this evaluation of Normalized Systems, we think in future NS has to be incorporated in existing enterprise transformation methods. We will now illustrate the potential impact of NS in four areas of enterprise transformation.

First, using Normalized Systems itself requires an enterprise transformation as it impacts the skills required from the people in the organizations: As the tool set already provides standard functionality, less programmers will be required for realizing a single application. Instead, business modelers will need analysis skills as well be able to use the NS tool set in creating a basic application. Some of the programmers can then focus on the application-specific interfaces and connections that are not provided by the NS tool set (custom components) while other programmers focus on maintaining or extending the NS tool set. Since the customizations in this PoC appeared to be small and can be

well-defined, this perfectly fits in the trend of right sourcing. Also, less application testers will be needed; instead of focusing on 'did we build the things right', the focus can be more on 'did we build the right things', as can be determined by modelers and subject matter experts.

Second, Normalized Systems can ease the splitting and allying of organizations. Since it is easy to identify the impact of business changes on IT that is built according to the NS theory, the impact of outsourcing parts of the organization on the IT becomes well-predictable. This may change insight in which parts are or are not eligible for outsourcing. Moreover, as the Normalized Systems theory ensures easy change of the IT system, including IT splits, the process of outsourcing will be sped up, significantly improving the agility of organizations in splitting, but also in allying with new partners [4, p.102]. Also, the Normalized Systems theory might provide insight in how to engineer the organization in such a way that organization splitting can be done much faster.

Third, the typical cycle of enterprise governance can be shortened by using Normalized Systems. Although the typical cycles of enterprise governance are already shortening, it is often the IT that is believed to cause the most delays in these projects because of its complexity. Normalized Systems is considered to enable shortening these projects, thus shortening the typical cycles of enterprise governance.

Finally, Normalized Systems highlights the importance of data foundations and the need for structural renewal of these data foundations of many organizations. As organizations change, their information needs in terms of information products change as well. However, if the underlying data services are well-designed, the data services do not have to change. Let's illustrate this with an example of Rijkswaterstaat (RWS), the Dutch Agency which is a/o accountable for Road and Traffic Management. At RWS, until 2005 two roles a/o existed, viz. 'road quality observer' and 'dead animal remover', each implemented by different people. The road quality observer was supported by an asset observance list and geographical data, and the dead animal remover was supported by a dead animal list and geographical data. The three information products were each supported by its own data service. From 2006, the two roles were combined into one functionary type 'road inspector', fulfilling both roles, now supported by three data services. Because the data services were well separated, instead of combining the information products, the three data services could be combined into one information product, offering live data about road inspections and dead animals as well as a map. This shows that well-designed data services support evolvable information needs and enable evolvable information products. In this PoC, NS has shown its strength in stable data-driven applications. We think organizations should consider rebuilding their data foundations with NS, as it offers interfaces through which the data services can be combined in a maintainable way, being able to follow the changing information needs of the organization.

Acknowledgments. We want to acknowledge Colonel Jean-Paul L. Duckers, Commander C2SC, for his sponsorship for this Proof of Concept Normalized

Systems, and for generously making available the materials for this project. A thanks for cosponsorship goes to Professor Herwig Mannaert, Vice-Dean Faculty of Applied Economics at University of Antwerp, and to Nico Brandt, Director Custom Software Development at Capgemini. Finally we want to thank all Ministry of Defense and Capgemini experts and team members involved, for their wonderful cooperation and valuable input.

References

1. Lehman, M.: On understanding laws, evolution, and conservation in the large-program life cycle. Journal of Systems and Software 1, 213–221 (1980)
2. Mannaert, H., Verelst, J.: Normalized systems: re-creating information technology based on laws for software evolvability. Koppa, Kermt, Belgium (2009)
3. Krouwel, M., Op't Land, M.: Combining DEMO and Normalized Systems for developing agile enterprise information systems. In: Albani, A., Dietz, J.L.G., Verelst, J. (eds.) EEWC 2011. Lecture Notes in Business Information Processing, vol. 79, pp. 31–45. Springer, Heidelberg (2011)
4. Op't Land, M.: Applying architecture and ontology to the splitting and allying of enterprises. PhD thesis, Delft University of Technology (2008)
5. Mulder, J.B.F.: Rapid Enterprise Design. PhD thesis, Delft University of Technology (2006)
6. Dietz, J.L.G.: Enterprise Ontology – Theory and methodology. Springer, Heidelberg (2006)