

Enterprise Modeling in an Agile World

Harald Wesenberg

► **To cite this version:**

Harald Wesenberg. Enterprise Modeling in an Agile World. 4th Practice of Enterprise Modeling (PoEM), Nov 2011, Oslo, Norway. pp.126-130, 10.1007/978-3-642-24849-8_10 . hal-01572388

HAL Id: hal-01572388

<https://hal.inria.fr/hal-01572388>

Submitted on 7 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Enterprise Modeling in an Agile World

Harald Wesenberg
Statoil ASA, Forusbeen 50, N-4035 Stavanger, Norway
hwes@statoil.com

Abstract. As the pace of business increases, the speed at which enterprise models must be delivered increases accordingly. Enterprise modelers cannot spend years in an attic developing perfect models, but must deliver models that are useful in time to be used. In this paper I will take a look at the experiences we have with enterprise modeling in Statoil, a global oil company headquartered in Norway.

Keywords: Enterprise modeling, Agile, PoEM2011

1 Introduction

This paper accompanies my keynote from the Practitioners of Enterprise Modeling 2011 conference (PoEM 2011) held in Oslo, November 2011. In the keynote, I take a look at the experiences we have had with enterprise modeling in Statoil, and how our practices has changed with the emergence of agile software development practices and business models. As befitting a keynote, this paper is not based on years of scientific research. It is a summary of observations, experiences and reflections I have made over the years, forged into an opinionated keynote at a conference. In no way does this paper reflect official Statoil policies on software development, enterprise modeling or anything else.

The paper is structured as follows:

- **Section 2** defines what an enterprise model is in the context of this paper and compares this to other models typically found in an enterprise
- **Section 3** looks at some of the lessons we have learned from the past years of enterprise modeling in Statoil
- **Section 4** offers a conclusion and some recommendations

2 Defining Enterprise Models

Not all models are enterprise models. In a large enterprise, dozens, if not hundreds or thousands of models are created every day on whiteboards, paper and presentations to illustrate some aspect of the real world.

An enterprise model, on the other hand, has certain characteristic that differentiates it from other models:

- **Enterprise models are for communication through time or space.** An enterprise model exists over a longer period of time, and is distributed widely throughout the enterprise. It is often found in a corporate repository of sorts, and is accessed through web portals or similar. Although many software developers create models as part of their development work, few of these models are used over a longer period of time, or outside that team of developers. Hence, such models are not enterprise models.
- **Enterprise models are abstractions.** Although all models are abstractions of the real world [1], the concept of abstractions is particularly important for enterprise models. They often cover an enormous amount of complex enterprise knowledge that cannot be easily transferred to a model. Therefore, abstractions are necessary when developing enterprise models, else they will be too complex to be of use. If the wrong abstraction is chosen a model will never become an enterprise model, but instead gather dust in a drawer somewhere.
- **Enterprise models are managed.** Enterprise models are (or at least should be) managed properly. They are often subject to strict versioning routines, configuration management practices and release plans. In many ways enterprise models are similar to source code, and should be subject to the same professional practices. If the models are not managed properly they will not be trusted and they will subsequently fail to achieve their full potential as enterprise models.
- **Enterprise models must have the right quality.** Model quality can be measured in many ways. I have found that it makes sense to talk about three dimensions of model quality: Syntactic quality (how well the model uses the modeling language), semantic quality (how well the model reflects the real world) and pragmatic quality (how well the model is understood by the target audience) [4,5]. In enterprise models the balance between these dimensions becomes very important; else the model will not be used by its intended target audience.

Although not all enterprise models exhibit all of these characteristics equally, they are useful as a means of distinguishing enterprise models from other models.

2.1 Introducing the Agile World

Agile means a lot of things to different people. In the context of this paper (and keynote) I define Agile to reflect the way the pace of change increases in many areas of life, forcing individuals and organizations to change with it. Software that were released in a yearly release cycle 10 years ago is now released twice a day [2], business models change frequently, and companies are on a path of continuous change as they acquire, merge and divest at an increasing speed.

For enterprise modelers, this increasing pace of change means that enterprise models must be developed faster and faster to be useful. They must be managed to stay current, and they need to be in the right place at the right time to have an impact.

3 Enterprise modeling in Statoil

Statoil is a global oil company headquartered in Norway. It has more than 20.000 employees in more than 30 countries worldwide, and has spent significant resources in various forms of enterprise modeling over the years. We have achieved a fair success with enterprise modeling in its corporate management system [3] where workflow models are used extensively to communicate requirements and best practices throughout the enterprise. The current management system contains some 1500+ business process and workflow models with associated requirements and best practices, all available through a corporate web portal anywhere in the company. The models are used daily in many parts of the organization, and are a significant contributor in reducing operational, environmental and safety risks etc. In this section I will review some of the experiences and observations we have made over the years of developing enterprise models.

Every model is a journey. All enterprise models develop over time. As the world it reflects changes, the models changes with it. This journey lasts for the lifespan of the model, and throughout the journey effort must be made to ensure that the model is current and useful.

When setting out to develop an enterprise model, be sure to have someone with thorough knowledge of the real world (domain) it reflects as part of the development team. This ensures that the right details are kept as the real world is abstracted into an enterprise model. Another important part on the team is played by the youngster that is current on best practices for modeling and is capable of challenging the domain experts on their assumptions and beliefs. Only when they work together will the journey be a successful one.

Every model tells a story. A model tells a story to its target audience. This story will change over time as the model is developed, but the core of the story must be known before modeling starts. I have seen many models fail to meet their purpose because the core story has not been clearly understood by the development team.

Every model has an audience. The story of the model and the abstractions used in developing the model is determined by the needs of the stakeholders for the model, in particular the audience. Many times, I have seen models usefulness virtually destroyed by the need of the model development team to include too many details. The development team lacks the ability to see which aspects of the model that makes it useful to the target audience, and instead makes a model that is useful for them.

Everything changes with scale and time. When the number of enterprise models grows into the thousands and models must be kept for many years, a lot more effort

must be put into model administration and maintenance. Models must be managed through versions, branches and applicability so that the model library is current and useful for the end users.

All models are not enterprise models. As a development effort (software, business or organizational development) is running, a lot of models are developed during the course of the effort. All of these models are not enterprise models, and very few of them should be brought back into the library of enterprise models. Only the models that will be widely used throughout the enterprise for a period of several years should be transferred to the enterprise model library, while the rest should be discarded or archived. This reduced the effort that needs to be spent on maintenance and administration of the enterprise models.

4 Recommendations

Based on my experiences with enterprise modeling in Statoil and elsewhere, I have a few recommendations to model development teams, as they set out to develop a model.

Know your audience and tell them the right story. A model is developed to tell a story to a set of end users. This story governs the entire life cycle of the model, and a model developed to tell one particular story will rarely tell another story just as well. Thus, the end user audience must be known and studied before modeling begins, to ensure that the model tell the right story.

Be pragmatic. When determining what story to tell and which abstractions to make during the development of a model, it is important to keep the most important aspect of a model in mind: A model is useless unless it is understood by its target audience. Therefore, the development must be pragmatic in its modeling approach both when it comes to abstractions and content. Only include model elements that are useful to the target audience, and hide details that only serve to clutter the model. Even though these details may seem important to the development team, they may confuse the target audience and thus make the model useless.

Don't get lost in the details. Tied to the previous point, it is more important to get the level of details right. With too many details, the development will take too long, and the model will be outdated before it is finished. Better to have a conceptual model that is understood by most target audiences and delivered on time for it to be useful, than have a detailed logical model developed too late and outdated from the start.

Retire models when they are no longer useful. An enterprise model must be kept current to be useful. This means that all models have an inherent debt, i.e. resources that must be spent at some point in the future. A model that is not current reduces the confidence in the model library, and all models in the library should be reviewed periodically to ensure that they are current and still in use.

5 References

1. Jackson, M.A.: Problem Frames: Analysing & Structuring Software Development Problems. Addison-Wesley Professional, 2000
2. Bosch, J.: Architecture in the age of composability, Keynote SATURN2011, San Francisco, California May 2011, <http://www.sei.cmu.edu/library/abstracts/presentations/bosch-saturn2011.cfm>
3. Wesenberg, H., Landre, E.: Using the corporate Management System to communicate architecture, Presentation SATURN2011, San Francisco, California, May 2011 <http://www.sei.cmu.edu/saturn/2011/abstracts-presentations.cfm#25>
4. Lindland, O. I., Sindre, G., and Sølvsberg, A., "Understanding Quality in Conceptual Modelling," IEEE Software, vol. 11, no. 2, pp. 42--49, 1994
5. Carlsen, S., Krogstie, J., Sølvsberg, A., and Lindland, O. I., "Evaluating Flexible Workflow Systems" Hawaii International Conference on System Sciences (HICSS-30), Maui, Hawaii, 1997