Towards Agile Enterprise Data Warehousing

Mikko Puonti
and Timo Lehtonen

Solita, Tampere, Finland
Email: puonti@iki.fi
timo.lehtonen@solita.fi

Antti Luoto
and Timo Aaltonen

Department of Pervasive Computing,
Tampere University of Technology,
Tampere, Finland
Email: antti.l.luoto@tut.fi
timo.aaltonen@tut.fi

Timo Aho

Yle, The Finnish Broadcasting Company,
Helsinki, Finland
Email: timo.aho@iki.fi

***Abstract*—Traditional business intelligence and data warehouse projects are very much sequential in nature. The process starts with data preparation and continues with the reporting needed by business measurements. This is somewhat similar to the waterfall model of software development and also shares some of its problems: the work is done in serial manner and the reaction time for possible design changes is often long. Agile principles are not well supported by the traditional serial workflow. By making the data preparation and reporting tasks parallel, it is possible to gain several advantages, such as shorter lead time and shorter feedback cycle. The solution proposed in this paper is based on enriched conceptual model that enables the business intelligence implementation process of different teams to change from serial to parallel workflow.**

Keywords*–*data warehouses; business intelligence; agile software development; scrum.

I. Introduction

Business Intelligence (BI) projects are traditionally following a pattern, where the work is actually done in serial tasks, which are strongly dependent on each other. This leads to long development cycles where some tasks need to be done before the next tasks can be even started. The problems of this approach include long feedback times and inefficient working process. The working method does not support the agile process models, such as scrum [1].

Scrum is an iterative project management approach to deliver software in incremental development cycles called *Sprints* that usually last from two to four weeks. Its benefits come from the ability to respond to the unpredictable environment changes as every sprint is planned separately.

In this article, we propose a process improvement to avoid the dependency of serial BI development tasks. The core of the idea is to rearrange serial development sprints to parallel ones by using a conceptual data model as a basis for a dimensional data warehouse (DW) model. The dimensional model is, on the other hand, an agreement between different development teams with different skills and, thus, a basis for communication between them. Research literature about combining BI with agile mindset exists but to the best of our knowledge none of them concentrate on how to organize work of teams in parallel way in agile BI project.

The expected benefits of our approach include shorter sprint cycle lengths, which leads to shorter customer feedback time. Also, it helps the DW modelers and BI reporters to concentrate on their work by reducing the fragmentation of development sprints, because of easier allocation of work. As a result, more development iterations can be done in the same time frame as with a serial workflow.

The proposed process improvement can be seen as a first step towards agile practices in BI projects and it can be later on combined with other agile practices.

The rest of this paper is structured as follows. In Section II, we introduce the necessary background for the paper by addressing the related work in agile BI processes. Section III presents the current and target states of the data warehousing and reporting process while Section IV introduces the approach from the viewpoint of data modeling. Finally, we draw some concluding remarks in Section V and outline our strategy for validating the expected benefits of the proposed approach in Section VI.

II. Related Work

The chosen related work concentrates on bringing miscellaneous agile practices to DW and BI processes. In general, incremental and iterative approaches are seen as beneficial in them but to the best of our knowledge, other authors have not discussed about organizing different teams' work in parallel so that traditionally done serial work could be done simultaneously. This is a gap we are trying to fill by improving the DW modeling process.

In [2], the authors categorize different agile BI actions in their literature review. Their categorization is based on previous work presented in [3] and identifies four agile BI action categories which are *Principles* (rules and assumptions derived from extensive observation and evolved through years of experience [4]), *Process models* (guidance to coordinate and control different tasks systematically which must be performed in order to achieve a specific goal [4]), *Techniques* (a way or style of carrying out a particular task) and *Technologies* (tools). The ideas presented in this paper fit to category *Process models* as the idea is to parallelize DW design tasks. The work in [2], also noted that agile principles are often discussed in a relation to agile process models, and in *Process models* category, Scrum can be seen as the most popular research topic between the years 2007 and 2013. We go through some of this previous work in the following paragraphs.

A process model called Four-Wheel-Drive (4WD) introduced in [5] utilizes six agile DW design practices (incremental process, iteration, user involvement, continuous and automated testing, lean documentation) that are based on software engineering methods. According to them, the impacts of an iterative and incremental process are better and faster feedback, improved change and resource management, clearer requirements and early detection of errors. They discuss incremental techniques in the light of risk analysis that balances between the value to users and the risk of releasing early. Similarly, our approach aims to enable ways of working more iteratively and incrementally while also making customer feedback easier but they don't have the viewpoint of parallelization which would also shorten the required time for DW projects.

In addition to direct process improvement, the work in [6] presents an optimization model for sprint planning in agile DW design, which is based on the team's ability to estimate a set of development constraints. In contrast to our work, we do not concentrate on the planning phases of sprints even tough the planning should be also easier in our parallel workflow where teams are working more in close collaboration. They aim to optimize the sprints by planning whereas we optimize time usage with work parallelization.

The work in [7] gives a description of a DW project that was executed in an agile manner. The lessons learned include successful usage of agile Enterprise Data Models, tools integrated to version control and continuous integration of the database. Even though their usage of Enterprise Data Model improved communication and collaboration by shortening feedback loops between different teams, they don't explicitly mention about making the workflow parallel, which is our goal. Our approach similarly improves the communication and collaboration between teams.

III. Data Warehousing and Reporting Process

According to [8], BI is a process that consists of two main activities: getting data in and getting data out. The first activity, i.e., (DW), is about collecting data from source systems to a single DW that combines the data. The data is then extracted to a useful form for decision support. Getting that data out is the part that receives the most attention as it eventually brings out the value even though the DW part is considered to be more laborious.

The skills and the tools needed for the two activities are different. Thus, the competence is diversed in DW and reporting teams. DW implementation work consists of modeling in addition to Extract, Transform, Load (ETL) loads and data integration with an ETL tool. An ETL developer needs technical knowledge of databases and data transformations while a report specialist makes visualizations and needs understanding of the data. The naming of the data items in report meta model utilized for analysis is done using business terms. Hence, a reporting specialist needs understanding of the customer's business process.

The data is the driver for the whole implementation of the reports. For analytical purposes, data is stored in a dimensional schema of a data mart [9, Chapter 1] by the DW team. Report implementation consists of two steps. In the first step, a meta model of data entities and the structure of the data is created, while in the second step, the actual report is created with a reporting tool. Testing of the reporting functionalities is commonly done by an end-user with the actual customer data. Thus, a prerequisite for the report development is an existing DW utilizing dimensional schema which is populated with the customer's data.

The diverse expertise of the different teams and the need of an existing DW before starting the report development results in lengthy workflow in current BI processes.

A. Current State

Currently, the way of working divides the design and implementation process of BI report into two teams, in which one team finishes the DW design work and another team continues by producing the specified report. Only after both the teams have finished their serial sprints, it is possible to gain feedback from the customer and start fixing the problems, starting again from DW work and continuing to reporting. This is presented in the Fig. 1.

Fig. 2 presents the current state of the workflow in a timeline. In the figure, *DW Sprint* includes actions, such as data integration, ETL and DW modeling (dimensional model) while *Reporting Sprint* consists of actions, such as creating a meta model for the report and creation of the actual report. The specification describes the business requirements and the visual guidelines for the report.

The result of the work in *DW Sprint* is a data mart that utilizes dimensional schema. The data mart and data loads in the data mart are done by an ETL developer. In the scrum process model, the DW implementation is done first in a *DW sprint* as can be seen in the Fig. 2. After the *DW sprint* deliverable (the data mart with customer's data) is available, the report implementation will be able to start. This dependency leads to a situation where there is first a *DW sprint* after which a *Reporting sprint* will follow. Implementation of a report requires at least two sprints, since in the first sprint the data comes available to the DW (DW sprint) and the actual report for the end user is implemented in the next sprint.

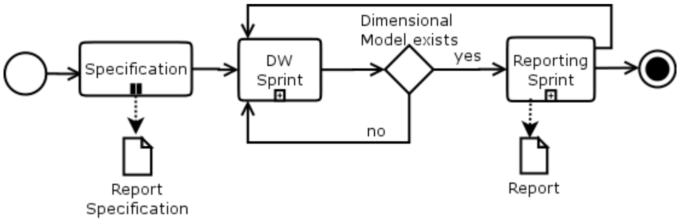

Figure 1. The current state of the process.

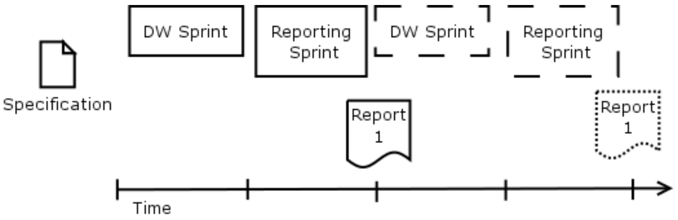

Figure 2. Workflow presented in a timeline.

B. Problem: Sequential Working

As a result of the diverse expertise in the teams and the need of an existing DW before reporting work, the full report development in *Reporting sprint* will not start before the first *DW Sprint* is finished, as it is presented in Fig. 1. The situation leads to a dependency between the DW implementation and report implementation.

The main problem of the current state is that getting feedback from the customer, which is based on the report, requires finishing both the sprints before it is possible to get feedback. After the feedback is received, the teams can start fixing the problems with new iterations of *DW sprint* and *Reporting sprint*. This also leads to fragmentation of work and excess waiting time between the sprints. Moreover, even though the workload is not as big as in the first iteration, it is still serial work and takes two sprints. If each sprint lasts for two weeks then completing both the sprints takes four weeks which multiplies to eight weeks after the feedback has been received and the corrections have been made. This is also illustrated in Fig. 2.

C. Solution: Parallel Working Enabled

As a solution to shorten the customer feedback cycle length and to defragment the DW and reporting work, we are targeting to parallelization of the serial sprints. The parallel team working is presented in Fig. 4. The parallelization is enabled by dimensional model based on conceptual model that contains information of the source systems. Based on the source system information in conceptual model, the dimensional model can designed at an attribute level with the support of interface specifications. A conceptual model presents associations between the modeled entities while the interface specification presents the attributes related to that association. The target state of the DW development process is presented in Fig. 3. The following aspects rise when comparing the current state to the target state.

1) Dimensional Model Based on Conceptual Model: Dimensional model represents *facts* which are business measures of the *dimensions*. The *dimensions* are grouping the business. Conceptual model consists of business entities and relationships between those entities. By adding information about a source system for an entity in a conceptual model, it is possible to get enough information of that entity without doing an exact logical data model. For creating the dimensional model, it is vital to know all the attributes of the *fact* and *dimension* tables. The attributes of each entity in a conceptual model can be solved out by looking at the interface of that entity. Each entity needs an interface from the source system to the DW and it the interface has to exist before the *DW Sprint* can start. The interface has the attribute information of the conceptual model entity, which makes it possible to create a dimensional model based on a combination of a conceptual model and an interface documentation.

2) Parallel Work of Different Teams: In the current state, the way of working was divided to serial sprints of different teams. The result of the completed DW sprint was a dimensional model which was utilized by reporting team. Thus, it would be beneficial, if the team could receive the dimensional model earlier to utilize it as a specification between them and the DW team. With the help of a dimensional model that is based on a conceptual model, it is possible to arrange the work so that the reporting team can to start developing the meta model for the reporting at the same time as the DW team starts the ETL work. In addition, the parallel way of working makes it easier for the teams to communicate with each other since they are concentrating on the same main goal, and further, the report can be produced in the end of the parallel sprints enabling customer feedback.

3) Shorter Feedback Cycle and Shorter Delay of Modifications: Since end-user is using the reports, getting useful feedback based on the report requires the report to include actual business data. Parallel working in *DW sprint* and *Reporting sprint* enables finishing the report in one sprint of calendar time. End-user can give feedback based on the report to both teams directly after the sprint. This is a huge difference to the DW team, which will get the feedback immediately after the sprint when compared to serial work in current state when the feedback was available only after the *Reporting sprint* was finished. This is beneficial because receiving feedback is more relevant when it is received directly and without delay. Faster feedback will also shorten the delay of starting the modification work. Therefore, making the modifications is easier since it requires less fragmented work and context switching.

Furthermore, parallel working shortens implementation time which also shortens the time that the end-user waits from giving the business needs to getting a report. In addition, the end-user is likely to be more participating in the process since the implementation time is shorter. According to [10], the end-user participation is such customer collaboration, which makes the product better. As an example of the effects in time, if a sprint lasts for two weeks, the parallel work ensures that delivering a new version of the report takes only two weeks. This is a notable improvement when compared to current state when delivering a report needed four weeks.

Data modeling is the key for communication between the teams and therefore it enables the parallelization of the work.

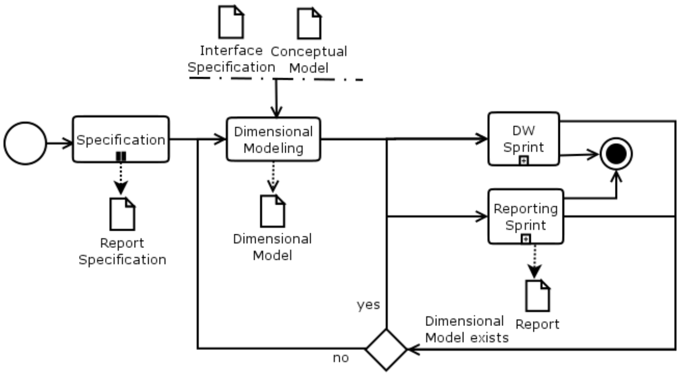

Figure 3. The target state of the process.

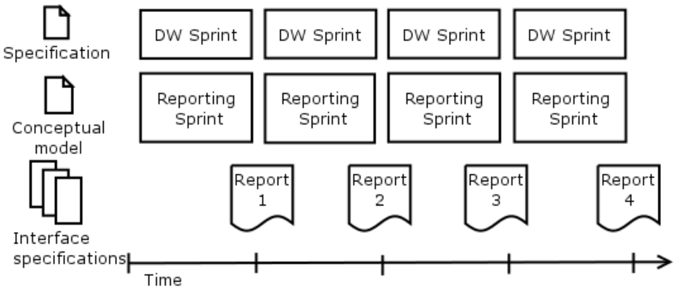

Figure 4. Sprints are parallel and feedback is faster.

IV. Data Modeling

Well managed data modeling is a crucial task for a DW project. Data modeling is about gathering the customers' data requirements and satisfying them with a DW solution.

According to [11], data modeling work is done on three design layers: logical, conceptual and contextual (by bottom-up order). Out of those layers, in this article, we are mostly interested in the conceptual and logical data modeling.

A. *Conceptual Data Modeling*

Conceptual data modeling is about modeling the user's data requirements in a conceptual manner using common concepts, such as entities and relationships. It describes the data and relationships between different data entities. Conceptual data modeling is a quick way to create a model of the problem domain with business representatives in a workshop, because the main entities come from the business domain and thus they have a business meaning. The collaboration between business stakeholders and data modelers is very important in order to tie the data intensive solution to the business processes.

Conceptual model can be utilized to ensure that all the participants share the same conceptual understanding of the modeled area [11]. In addition, it is a base that evolves to logical data model.

B. *Logical Data Modeling*

Logical data model presents all entities and their attributes. Each entity which has a primary key is marked in the model. Many-to-many relationships between entities are specified by creating an association entity between the entities. Creating a logical data model requires the following steps [12]:

- Specifying primary keys for all the entities.
- Finding the relationships between different entities.
- Finding all the attributes for each entity.
- Resolving many-to-many relationships.
- Normalisation.

The purpose of the logical data model is to provide a detailed specification for the physical relational database design [11]. In our context a logical data model is a tool for DW designers to produce a DW.

C. *Dimensional Modeling*

A dimensional model consist of fact and dimension tables in which the main items generally are *facts* and *dimensions* [9]. A *fact* represents a business measurement and is linked to several *dimensions*. A *dimension* groups and labels the measurements while it is also used to restrict the data set of measurements. Dimensional modeling is widely used modeling technique to offer data from DW to reporting tools.

D. *Granularity of Data Modeling*

The conceptual data model is important for communication between each participant in the project, especially for the business stakeholders, but it does not cover the detailed information needed in the implementation. The logical model, on the other hand, is more detailed but requires more work as it is relatively slow to model all the attributes and relationships of each entity.

The kind of data modeling described so far, is missing one critical piece of information as it does not tell where the data actually exists. The source system information is the most vital information in the reporting project. The needed granularity of data modeling is a mix of conceptual and logical data modeling enriched with information about the location of different entities. The combination of conceptual entities marked with the primary key attributes and information of source systems is the minimum required granularity of needed data model. A model should be enriched with the vital attributes, but the amount of attributes depend on how well the modelers know the domain. When the available information is well known and the business entity is clear, it is possible for everyone to understand the information even if it is not modeled in detail.

V. Conclusions

In this paper, we presented an idea to shorten the feedback cycle of BI projects. The proposed method consists of parallelizing DW and reporting team sprints by using a dimensional model as an agreement between the teams. Since modeling plays a crucial part in BI process, it is important to provide the dimensional model as early as possible. In this paper we claim this to be possible by developing dimensional model based on a combination of a conceptual model and the interface documentation of a source system.

Traditionally, reporting team starts working after DW team has offered a dimensional model with actual data. In our approach, reporting team can start working in parallel with DW team but initially without any actual data. The DW team implements ETL processes with small increments which gives then increasing amount of actual data to reporting team. It is worth noting that making the specifications in the new approach does not increase the overall process time. This is because interface specification is created implicitly anyway and conceptual model is very light weight to create.

As a result of the approach, the customer feedback cycle shortens which moreover makes the feedback more direct. Furthermore, because of parallel working, the communication between teams is more efficient and reaction time to feedback between teams is shorter. This is a step towards agile enterprise data warehousing where a bigger team consists of two separate teams with diverse competence.

VI. Future Work

As a future work, we are planning to conduct a case study in which we will utilize our ideas in an industrial BI project in a mid-sized Finnish software company. Moreover, we are eventually aiming at integrating the different teams (DW team and reporting team) so that the expertise of a person working in a BI project would cover both the required perspectives. That way, it is possible to reduce the amount of persons needed in a project.

The proposed idea is our first step towards agile BI projects, since it can be adopted with other agile principles, as well. To make the BI process even more agile and faster, we are studying how to shorten implementation time by generating ETL processes automatically based on modeling principles [13]. To get full advantage of these improvements, we also aim at creating release management practices to get our BI project closer to the continuous delivery.

Acknowledgment

The work was financially supported by TEKES (Finnish Funding Agency for Innovation) DIGILE Need for Speed program. We would also like to thank Solita and Yle for the possibility of doing this research.

References

[1] K. Schwaber, “Scrum development process,” in the Proceedings of the Workshop on Object-Oriented Programming Systems, Languages and Applications Workshop on Business Object Design and Implementation, OOPSLA ’95, Austin, Texas, pp. 117–134, October 1995.

[2] R. Krawatzeck, B. Dinter, and T. Duc Ang Pham, “How to make business intelligence agile: The agile bi actions catalog,” in System Sciences (HICSS), 2015 48th Hawaii International Conference on, pp. 4762–4771, January 2015.

[3] R. Krawatzeck, M. Zimmer, and S. Trahasch, “Agile business intelligence - definition, maßnahmen und herausforderungen,” HMD Praxis der Wirtschaftsinformatik, vol. 50, no. 2, pp. 56–63, January 2014.

[4] F. Tsui, O. Karam, and B. Bernal, Essentials of software engineering. Jones & Bartlett Publishers, 2013.

[5] M. Golfarelli, S. Rizzi, and E. Turrichia, “Modern software engineering methodologies meet data warehouse design: 4wd,” in 13th International Conference, DaWaK 2011, Toulouse, France. Proceedings, pp. 66–79, August 2011.

[6] M. Golfarelli, S. Rizzi, and E. Turricchia, “Sprint planning optimization in agile data warehouse design,” in Proceeding DaWaK'12 Proceedings of the 14th international conference on Data Warehousing and Knowledge Discovery, pp. 30–41, 2012.

[7] T. Bunio, “Agile data warehouse – the final frontier: How a data warehouse redevelopment is being done in an agile and pragmatic way,” in Proceeding AGILE '12 Proceedings of the 2012 Agile Conference, pp. 156–164, August 2012.

[8] H. Watson and B. Wixom, “The current state of business intelligence,” Computer, vol. 40, no. 9, pp. 96–99, September 2007.

[9] R. Kimball and M. Ross, The data warehouse toolkit: the complete guide to dimensional modeling. John Wiley & Sons, 2011.

[10] M. Fowler and J. Highsmith, “The agile manifesto,” Software Development, vol. 9, no. 8, pp. 28–35, 2001.

[11] A. Sharp and P. McDermott, Workflow modeling: tools for process improvement and applications development. 685 Canton Street Norwood, MA 02062: Artech House, 2001.

[12] 1keydata, “Logical data model,” http://www.1keydata.com/datawarehousing/logical-data-model.html, accessed: 2016-01-18.

[13] M. Puonti, T. Raitalaakso, T. Aho, and T. Mikkonen, “Automating transformations in data vault data warehouse loads,” in Proceedings of the 26th International Conference on Information Modelling and Knowledge Bases, EJC 2016, pp. 219–235, June 2016.