AgileKDD:
An Agile Business Intelligence Process Model

Givanildo Santana do Nascimento
Federal University of Sergipe
São Cristóvão, Brazil
gsnascimento@petrobras.com.br

Adicinéia Aparecida de Oliveira
Federal University of Sergipe
São Cristóvão, Brazil
adicineia@ufs.br

***Abstract* — In a knowledge-based society, transforming data into information and knowledge to support the decision-making process is a crucial success factor for all the organizations. In this context, one mission of Software Engineering is to produce systems able to process large volumes of data, transform them into relevant knowledge and deliver them to customers, so they can make right decisions at the right time. However, companies still face failures in determining the process model used in their Knowledge Discovery in Databases and Business Intelligence projects. This article introduces the AgileKDD, an agile and disciplined process for developing systems capable of discovering the knowledge hidden in databases, built on top of the Open Unified Process. A case study shows that AgileKDD can guide projects whose goal is to develop Knowledge Discovery in Databases and Business Intelligence applications, increasing success factor as well as customer satisfaction.**

Keywords – Business Intelligence; Knowledge Discovery in Databases; Agile Software Development; Software Process.

I. Introduction

In 1996, the Organization for Economic Cooperation and Development (OECD) redefined knowledge-based economies as: “economies which are directly based on the production, distribution and use of knowledge and information” [1]. In knowledge-based economies, the global competition is increasingly based on the ability to transform data into information and knowledge in an effective way. Knowledge is equated with the traditional factors of production - land, capital, raw materials, energy and manpower - in the process of wealth creation. Thus, data, information and knowledge constitute key assets for all organizations working in this economic model.

Knowledge management, Data Mining, Knowledge Discovery in Databases (KDD) and, more generally, Business Intelligence (BI) are key concepts in a knowledge-based economy. BI applications have vital importance for many organizations and can help them manage, develop and communicate their intangible assets such as information and knowledge, improving their performance. For instance, Continental Airlines’ investments in BI have a Return on Investment (ROI) of 1000%, attributed to increased revenue and reduced costs [2].

However, companies still face problems in determining the process model used to develop KDD and BI applications. As business requirements become more dynamic and uncertain, the traditional static, bureaucratic and heavy processes may not be able to deal with them. Recent researches have demonstrated that waterfall lifecycles and traditional software development processes are not successful in BI because they are unable to follow dynamic requirement changes in a rapidly evolving environment [3]. As software process is mandatory for KDD and BI development, one possible solution is to use an agile process, which is typically characterized by flexibility, adaptability, face-to-face communication and knowledge sharing.

This article discusses the importance of using an agile software process in KDD and BI applications development. Thus, the main objective of this paper is to present AgileKDD, an agile process able to guide the KDD and BI applications development in a manner compatible with the current ever-changing requirement environments. The next sections of this article are organized as follows: Section 2 describes BI and Knowledge Discovery in Databases as techniques for transforming raw data into information and knowledge. The 3rd section presents the agile software development processes. Section 4 presents the AgileKDD, an agile KDD and BI process model built on top of the Open Unified Process. Section 5 presents related work. Finally, Section 6 presents the conclusions.

II. Transforming Data Into Information and Knowledge

The raw data evolve into information and knowledge as they receive degrees of association and meaning [4]. The knowledge gained from the interpretation of data and information drives the knower to action, so knowledge is an important asset for organizations that operate in knowledge-based economies and markets. BI, as well as KDD, has the goal of transforming raw data into information and knowledge, in order to support the decision making process.

A. Knowledge Discovery in Databases

Knowledge Discovery in Databases is a nontrivial process of identifying valid, novel, potentially useful, and understandable patterns in data [5]. The discovered knowledge must be correct, understandable by human users and also interesting, useful or new. In addition, the knowledge discovery method must be efficient, generic and flexible (easily changeable).

Data Mining (DM) is the main activity of KDD and consists of applying algorithms to extract models or patterns from data [5]. Data Mining is the process of searching for relationships and distinct patterns that exist in datasets but are hidden among the large amount of data. Its aim is to transform data apparently devoid of connection into relevant information for decision making and results evaluation. DM is used to find information without a prior formulation of hypotheses and search for something non-intuitive, transforming meaningless data into valuable strategic knowledge. DM tasks and tools include data classification, neural networks, clustering, regression analysis, correlation and predictive analysis. DM applications are characterized by the ability to deal with the explosion of business data and accelerated market changes. These characteristics provide powerful tools for decision makers. Such tools can be used by business users to analyze huge amount of data for patterns and discover trends [1].

The KDD systematization effort has resulted in a variety of process models, including the KDD Process [5] and the Cross-Industry Standard Process for Data Mining (CRISP-DM) [6], which are the most widely used in KDD projects, the most frequently cited and supported by tools. These two processes are considered the *de facto* standards in the KDD area. Several other process models were derived from KDD Process and CRISP-DM. Figure 1 shows the evolution of 17 KDD/BI process models and methodologies. KDD Process can be pointed out as the initial approach, and CRISP-DM as the central approach of the evolution diagram [7]. Most of the approaches are based on these process models.

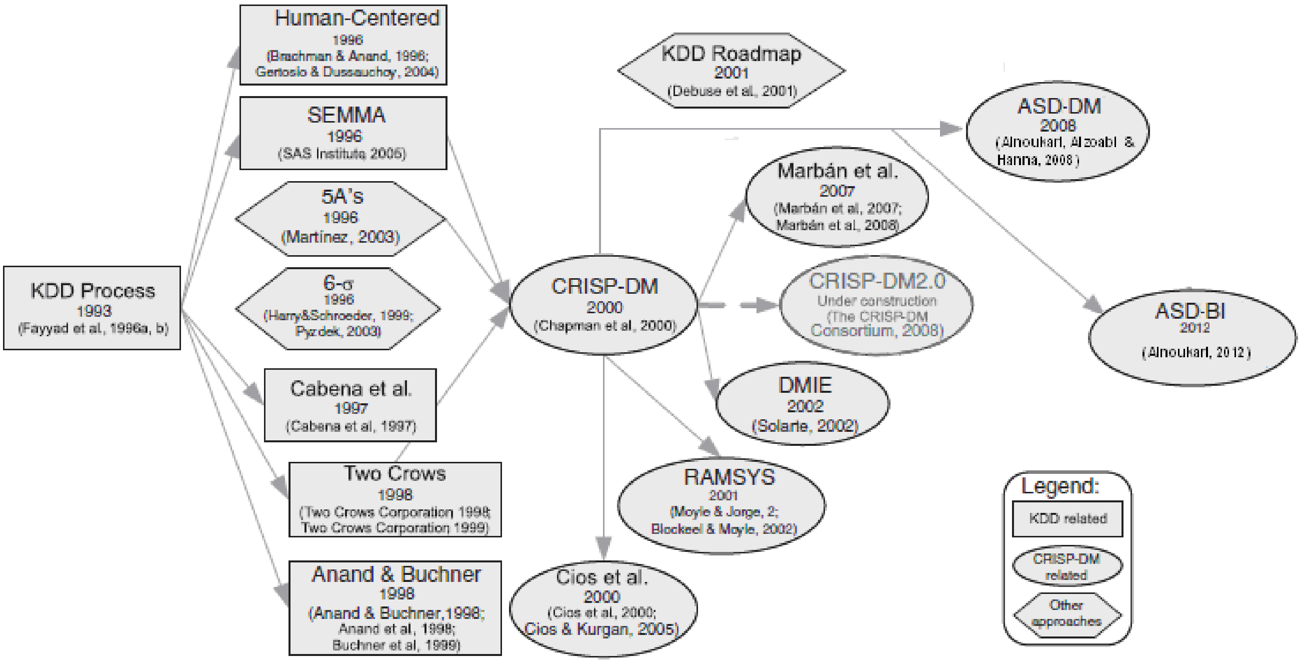

Figure 1. Evolution of KDD/BI process models (Source: Adapted from [7])

The KDD process models created between 1993 and 2008 were discussed in detail in a survey by [8] and then categorized by [7] into three groups:

- KDD related approaches – KDD Process (1993); Human-Centered (1996); 5A's (1996); 6-σ (1996); Cabena et al. (1997); Two Crowns (1998); and Anand & Buchner (1998).
- CRISP-DM related approaches – CRISP-DM (2000); Cios et al. (2000); RAMSYS (2001); DIME (2002); Marbán et al. (2007); and the CRISP-DM 2.0 initiative (not concluded).
- Other approaches – KDD Roadmap (2001).

Sometime later [9] continued the older surveys done by [8] and [7], and proposed a different categorization to the KDD process models:

- Traditional approach – Starting with KDD Process, many other process models used the same sequential steps: business understanding, data understanding, data processing, data mining, model evaluation, and deployment.
- Ontology-based approach – This approach is the combination of ontology engineering and traditional approach.
- Web-based approach – This approach is similar to the traditional approach, but it has some steps to deal with web log data analysis.
- Agile-based approach – Integrates agile processes and methodologies with traditional approaches. The process models in this category are Adaptive Software Development – Data Mining (ASD-DM) Process Model [10] and Adaptive Software Development – Business Intelligence (ASD-BI) Process Model [1].

Thus, the knowledge discovery process models are evolving from traditional to agile processes, becoming more adaptive, flexible and human-centered. However, these processes still lack software engineering capabilities such as requirements management, project management and changes management.

B. *Business Intelligence*

Business Intelligence assists in extracting information from the available data and using it as knowledge in developing innovative business strategies. BI is an umbrella term that combines architecture, tools, databases, applications, practices, and processes to organize, integrate and explore information, with the goal of developing understanding and knowledge, which can produce a better decision making process. Moreover, BI is an Information Technology (IT) framework vital for many organizations, especially those which have extremely large amounts of data, which can help organizations manage, develop and communicate their assets such as information and knowledge [2]. According to Mariscal et al. [7], BI is a broad category of applications and technologies for gathering, storing, analyzing, and providing access to data to help enterprise users make better business decisions, and DM is an important component of BI.

The number of BI projects has grown rapidly worldwide according to Gartner Group annual reports. BI has been on the list of the top ten priorities in IT since 2005 and was at the top of this list for four consecutive years, from 2006 to 2009. In a broader sense, companies understand that the information and knowledge provided by BI applications are essential to increase the efficiency and effectiveness, support competitiveness and innovation. Thus, investments into data mining BI applications grew by 4.8% from 2005 to 2006 and by 11.2% from 2007 to 2008 [7], [11], [12]. Continental Airlines' success case is the most expressive example of BI profitability. In 2006, the overall ROI of BI projects in that company was 1000% [2], [13] and in 2008 that initiative also reported very positive results [14].

However, not all KDD and BI results are positive. Regardless of the priority and budgets growth, neither all the projects results were delivered [7]. Further, the worsening of international financial crisis has led to significant cuts in IT budgets from 2008 on. In addition, many BI projects had failed to achieve their goals or were canceled because they were unable to follow the dynamic requirement changes in rapidly evolving environments. Because of this, BI left the top of the list of priorities in IT and, in 2010 and 2011, dropped to the fifth position. Technologies with higher productivity, lower risk and faster return on investment were priorized instead [12], [15].

The BI environment consists of the transaction processing applications, the Extraction, Transformation, Loading (ETL) and data integration processes, the Data Warehouse (DW) as well as the Data Marts, BI tools and analytic applications. The raw data are loaded by ETL processes into DW and data marts. During loading, the ETL processes also perform cleaning, completion, correction and integration of data. The DW and data marts are then explored by the user utilizing On-line Analytical Processing (OLAP) tools and data mining [2].

Many companies still develop KDD and BI applications without the guidance of a software process, but, as any software project, KDD and BI projects need a software process to succeed [16]. Also, the dynamic business requirements, the needs of quick ROI and fluid communication between stakeholders and the team led to agile process as one possible solution [3].

III. AGILE SOFTWARE ENGINEERING PROCESSES

A software process provides an ordered sequence of activities related to the specification, design and implementation as well as validation and deployment of software products, transforming user expectations into software solutions [17]. According to Pressman [18], the software processes set the context in which technical methods are applied, the work artifacts (models, documents, data, reports, forms) are produced, the milestones are established, quality is assured and changes are managed.

The traditional software development processes are characterized by rigid mechanisms with a heavy documentation process, which make it difficult to adapt to a high-speed, ever-changing environment [19]. Researchers have suggested that the complex, uncertain and unstable environment is pushing developers to adopt agile processes rather than traditional software processes. They claim that agile approach is the answer to the software engineering chaotic situation, in which projects are exceeding their time and budget limits, requirements are not fulfilled and, consequently, leading to unsatisfied customers [20].

The Manifesto for Agile Software Development [21] defines the values introduced by the agile software processes: individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan. Ultimately, by following these values, software development becomes less formal, more dynamic, and more customer-focused. Based on these values, agile processes are people-oriented and have the customer satisfaction as the highest priority through the early and continuous delivery of functioning software [20]. Also, the response to all types of changes and fluid communication between all projects participants become top priorities. In agile development, the main work product is the increment of functioning software, delivered to the customer within the fixed timeframes. Agile approaches are best fit when requirements are uncertain or volatile; this can happen due to business dynamics and rapidly evolving markets. It is too difficult to practice traditional plan-oriented software development in such unstable environments [19].

A. *Unified Process and Open Unified Process*

The Unified Process (UP) [22] is based on the Incremental Model [18], focuses on architecture and is use cases driven. Based on the use cases model, the analysis, design and implementation models are created to realize the use cases. The UP is focused on architecture, so it starts by the definition of an application skeleton (the architecture), which evolves gradually over development. The UP is also an iterative and incremental process because it offers an approach of partitioning the work into smaller portions or mini-projects. In UP, the architecture provides the framework to guide the system development into iterations, while the use cases define the targets and lead the work of each iteration.

Open Unified Process (OpenUP) is a variation of the UP that applies agile, iterative and incremental approaches within a structured lifecycle. OpenUP embraces a pragmatic, agile philosophy that focuses on the collaborative nature of software development. It is a low-ceremony process that can be extended to address a broad variety of project types [23]. OpenUP has compliance with the Manifesto for Agile Software Development, is minimal, complete and extensible. Moreover, it increases collaboration and continuous communication between project participants, more than formalities and comprehensive documentation [24].

The OpenUP process is divided into three layers, has four phases and six disciplines. The process applies intensive collaboration as the system is incrementally developed by a committed, self-organized team. OpenUP layers are illustrated by Figure 2. They are [23]:

- Project Lifecycle – structures the software project into four phases: Inception, Elaboration, Construction and Transition. A project plan defines the lifecycle and results in a released application.
- Iteration Lifecycle – OpenUP divides the project into iterations: planned, time-boxed intervals typically measured in weeks. Iterations focus the team on delivering incremental value to stakeholders in a predictable manner. OpenUP applies an iteration lifecycle that structures how micro-increments are applied to deliver stable, cohesive builds of the system that incrementally progresses towards the iteration objectives.
- Micro-increment – personal effort on an OpenUP project is organized in micro-increments. These micro-increments provide an extremely short feedback loop that drives adaptive decisions within each iteration.

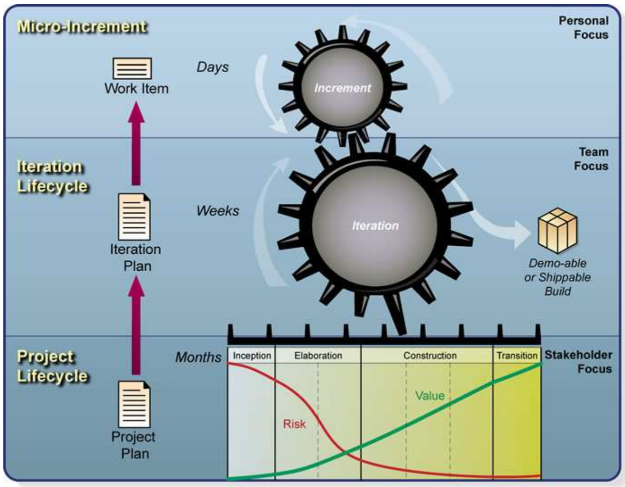

Figure 2. OpenUP layers and lifecycles (Source: [23])

The OpenUP phases are inception, elaboration, construction and transition. In inception phase the product vision is specified. The product architecture is defined in elaboration phase. The result of construction is a demonstrable or deliverable product version, deployed to the customer at the end of the transition phase. The OpenUP disciplines are requirements, architecture, development, testing, project management, and configuration and change management.

Again, the development of KDD and BI solutions must be guided by a software process. Therefore, it is mandatory to define processes that address aspects of KDD, BI, as well as the software engineering process disciplines, whose function is to order the software development. By the other hand, waterfall lifecycles and traditional processes are not successful in BI because they are unable to follow requirements in ever-changing environments [3]. Hence, one possible solution is to use an agile process, which is typically characterized by flexibility, adaptability, communication and knowledge sharing.

IV. AGILEKDD

AgileKDD is an agile and disciplined process for the development of KDD and BI applications. CRISP-DM and KDD Process provide the capabilities related to knowledge discovering. OpenUP provides to AgileKDD the lifecycle, phases and disciplines, which are requirements, architecture, development, test, project management and changes management. OpenUP also adds the agile software development core values and principles, without giving up the management disciplines. The personal effort on an AgileKDD project is organized in micro-increments. They represent small work units that produce measurable steps in the project progress, usually measured in days. The process applies intensive collaboration between the actors as the system is built incrementally. These micro-increments provide extremely short cycles of continuous feedback to identify and resolve problems before they become threats to the projects.

AgileKDD divides the projects in iterations with fixed time boxes, usually measured in weeks. The iterations drive the team to deliver incremental value to stakeholders in a predictable manner. Iteration plan defines what must be delivered during the iteration and the result is a demonstrable or deliverable piece of the KDD or BI solution. The AgileKDD lifecycle provides stakeholders and project team visibility and decision points at various milestones, until a working application is fully delivered to stakeholders. Figure 3 presents an overview of AgileKDD, highlighting its phases and activities.

The Inception phase has the aim of developing an understanding of the application domain and the relevant prior knowledge and identifying the goal of the BI project from the customer's viewpoint. In this phase the project vision and plans are defined and agreed by all project participants. Also, in inception the target data set, or subset of variables or data samples, is selected. The knowledge discovery processes will be performed on the selected target data set.

The Elaboration phase is responsible by the system's architecture, the data preprocessing and modeling. Data cleaning removes noise, collects the necessary information to model and decides on strategies for handling missing data fields. Data quality is a critical success factor for any KDD or BI project, so it is verified prior to the DW and data marts modeling.

Once DW and data marts are modeled, ETL processes are built to extract, integrate, transform and load the selected target data into DW and data marts. Thus, the data mining techniques that best fit to the data are selected and applied to the information stored in data marts.

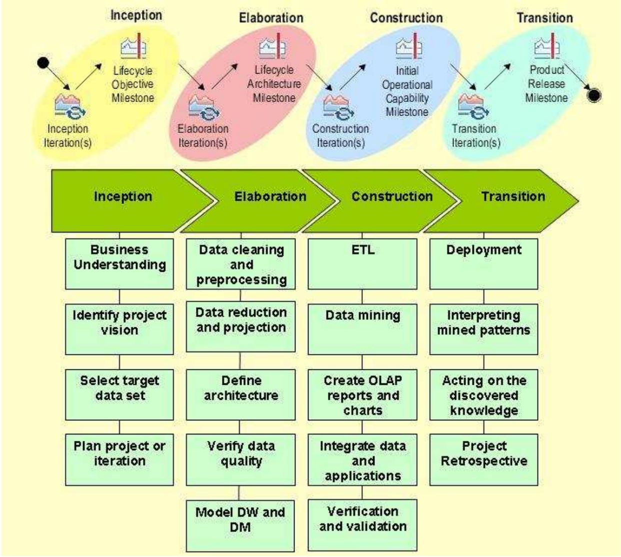
Inception
Elaboration
Construction
Transition
Lifecycle Objective Milestone
Lifecycle Architecture Milestone
Initial Operational Capability Milestone
Product Release Milestone
Inception Iteration(s)
Elaboration Iteration(s)
Construction Iteration(s)
Transition Iteration(s)
Inception
Elaboration
Construction
Transition
Business Understanding
Identify project vision
Select target data set
Plan project or iteration
Data cleaning and preprocessing
Data reduction and projection
Define architecture
Verify data quality
Model DW and DM
ETL
Data mining
Create OLAP reports and charts
Integrate data and applications
Verification and validation
Deployment
Interpreting mined patterns
Acting on the discovered knowledge
Project Retrospective

Figure 3. Initial AgileKDD lifecycle

Data mining tools search for meaningful patterns in data, including association rules, decision trees and clusters. The team can significantly aid the data mining method by correctly performing the preceding steps. The OLAP reports and charts as well as the dashboards are built to allow user data exploration. The verification and validation activities guarantee that the data was extracted, loaded and processed correctly, according to business objectives.

In Transition phase the deployment of both software and knowledge takes place, the discovered knowledge is discussed and interpreted by human beings, actions are created and the retrospective discusses lessons learnt during the project to promote continuous process improvement. Interpreting mined patterns involve visualization and storage of the extracted knowledge into knowledge bases, or simply documenting and reporting it to interested parties. This activity also includes checking for and resolving potential conflicts with previously believed knowledge. The AgileKDD process can involve significant iteration, interaction and can contain loops between any two phases.

A. *Case study and process refinement*

AgileKDD has been validated by a case study in oil and gas field. The process was applied to a KDD and BI project that dealt with Reservoir Evaluation data and afforded the early and continuous delivery of results to the customers. DM results in first iteration were delivered two months after the project kickoff. The second iteration delivered the performance indicators as a dashboard and the third iteration deployed the OLAP reports, graphs, and *ad hoc* exploration of the DW.

The case study showed that AgileKDD was able to guide the KDD and BI application development and helped to anticipate the project ROI. Moreover, the process was refined after the project retrospective. The refined AgileKDD lifecycle is represented by Figure 4.

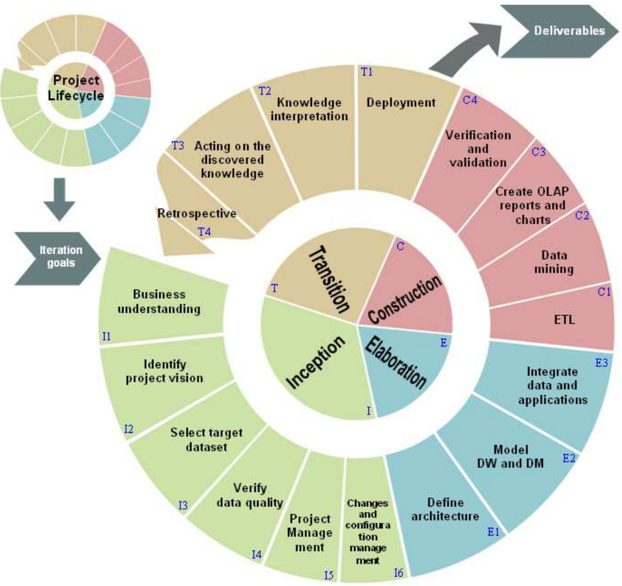
Project Lifecycle
Deliverables
Iteration goals
Transition
Construction
Elaboration
Inception
T
C
E
I
T1 Deployment
T2 Knowledge interpretation
T3 Acting on the discovered knowledge
Retrospective T4
C4 Verification and validation
C3 Create OLAP reports and charts
C2 Data mining
C1 ETL
E3 Integrate data and applications
E2 Model DW and DM
E1 Define architecture
I1 Business understanding
I2 Identify project vision
I3 Select target dataset
I4 Verify data quality
I5 Project Management
I6 Changes and configuration management

Figure 4. Refined AgileKDD lifecycle after case study

As data quality is a critical success factor for any BI project, the verify data quality activity was moved from Elaboration to Inception phase because its result influences the project management. Then, data quality is verified in Inception phase to indicate the project feasibility and quality constraints. Project management activity consists on high level project planning, risks management and governance concerns. Changes and configuration management activity is related to the version control of all the project artifacts, including documentation, sources and binaries.

Under the architectural point of view, the Data Mart Bus Architecture [25] fits perfectly in agile approach. In this architecture, the data marts are built incrementally in response to iteration requirements. The Data cleaning and preprocessing and Data reduction and projection activities defined in Elaboration phase were performed as part of ETL activity in Construction phase. Since they are activities related to the transformation of the data, the best place to them is ETL activity.

All the documentation artifacts needed to develop the three iterations were composed by the vision document and the data models of operational sources and DW. There was no need to use cases and additional diagrams.

The data mining results verification against the operational data sources was crucial for the knowledge discovered acceptance by business experts. The rules proof using operational data gave no room for questioning the correctness of DM methods and tools used. The documentation of the discovered knowledge in an electronic presentation was sufficient to support communication with knowledge users. No other form of knowledge representation and storage of was required.

Requirements changes directly affected the project planning, but did not harm the product objectives achievement because they were discovered early, between project iterations.

B. AgileKDD disciplines

The refined AgileKDD disciplines are the same of OpenUP: requirements, architecture, development, test, project management and configuration and changes management. Table I shows the AgileKDD disciplines, their purposes and suggested work products.

TABLE I. REFINED AGILEKDD DISCIPLINES

Discipline	Purpose	Work products[a]
Requirements	Elicit, analyze, specify, validate and manage the requirements for the system being developed.	Vision document. Initial project glossary. Prototypes.
Architecture	Define an architecture for the system components.	Software architecture description. DW and DM models.
Development	Design and implement a technical solution adherent to the architecture that meets the requirements.	Software components. Integrated software increment.
Test	Validate system maturity through the design, implementation, execution and evaluation of tests.	Plan and test procedure. Test record.
Project management	Instruct, assist and support the team, helping them to deal with risks and obstacles faced when building software.	Project plan. Feasibility and risk evaluation.
Configuration and change management	Controlling changes in artifacts, ensuring a synchronized evolution of the set of artifacts that make a software system.	Work items list.

a. All the work products are optional. Only the necessary artifacts must be produced.

During a full project cycle, most of the requirements discipline effort is concentrated in the Inception phase. The architecture is the main discipline during the Elaboration phase. In the same phase, the development is intensified from the definition of the system architecture and continues as the main discipline of Construction phase. The tests occur mainly in verification and validation activity of Construction phase. The project management discipline is concentrated predominantly in the Inception phase. The configuration and change management has greater prevalence in Inception and Transition phases. Each discipline can be related to a set of work products created during the process phases, according to the project needs.

V. RELATED WORK

There are not many works related to agile software processes appropriate to the development of KDD and BI applications. The main work that applies agile methodologies to BI is [1]. Alnoukari [19] discusses Business Intelligence and Agile Methodologies for knowledge-based organizations in a cross-disciplinary approach. Alnoukari [26] introduces Adaptive Software Development – Business Intelligence (ASD-BI), a BI process model based on Adaptive Software Development agile methodology. Likewise, Alnoukari et al. [10] defined Adaptive Software Development – Data Mining (ASD-DM) Process Model. The main difference between this work and these is the fact that AgileKDD is a process, not a methodology. As a process, AgileKDD defines what to do instead how to do KDD and BI development. Also, the process proposed by this work defines lifecycle, roles, activities, inputs and outputs regarding agile KDD and BI application development. Moreover, AgileKDD contains management disciplines like project, changes and requirements management, which were inherited from OpenUP. Even in an agile process like AgileKDD, management is a crucial success factor for any software projects.

Three surveys about data mining and knowledge discovery process models and methodologies are discussed and compared by [7], [8] and [9]. All the process models and methodologies presented by these works focus on data mining and knowledge discovery, and don’t consider databases like DW and data marts nor BI and OLAP. As BI is more comprehensive than data mining, this work focuses on an agile process modeled to address both KDD and BI software projects, in an adaptable, flexible and systematic manner.

VI. CONCLUSION

A software process is mandatory for KDD and BI developments; however traditional software development processes are not successful in KDD and BI because they are unable to fulfill dynamic requirement changes in an ever-changing environment. Agile processes fit in KDD and BI better than traditional processes because they are characterized by flexibility, adaptability, communication and knowledge sharing.

This work presented AgileKDD, a KDD and BI process based on KDD Process, CRISP-DM and Open Unified Process. AgileKDD has been validated by a case study and results indicate that software development organizations may apply AgileKDD in KDD and BI applications projects. The process bring benefits as more customer satisfaction through early and continuous delivery of functioning software, better communication between team members and reducing projects failures risks.

The main contribution of AgileKDD is its ability to guide the BI solutions development according to the practices present in agile software development processes. AgileKDD can increase the projects success factor and customer satisfaction through the early and continuous delivery of functioning software and useful knowledge. The process can be used to guide BI applications projects in scenarios of continuous requirements evolving and early ROI need.

Future work can validate AgileKDD by case studies in different areas and investigate the need of storing the knowledge discovered into ontology or knowledge bases.

References

[1] A. El Sheikh and M. Alnoukari, Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications. IGI Global, 2012. pp. 1-370.

[2] M. Alnoukari, H. Alhawasli, H. Alnafea, and Amjad Zamreek, "Business Intelligence: Body of Knowledge" in *Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications*. IGI Global, 2012. pp. 1-13.

[3] D. Larson, "Agile Methodologies for Business Intelligence" in Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications. IGI Global, 2012. pp. 101-119.

[4] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, Sixth Edition. Pearson, 2010.

[5] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From datamining to knowledge discovery: an overview" in *Proc. Advances in Knowledge Discovery and Data Mining*, 1996, pp. 1–34.

[6] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, CRISP-DM 1.0: Step-by-step data mining guide, 2000.

[7] G. Mariscal, O. Marbán, and C. Fernández, "A survey of data mining and knowledge discovery process models and methodologies". *The Knowledge Engineering Review*, vol. 25, 2010, pp. 137-166.

[8] L. Kurgan and P. Musilek, "A survey of Knowledge Discovery and Data Mining process models". *The Knowledge Engineering Review*, vol. 21, 2006, pp. 1-24.

[9] M. Alnoukari and A. El Sheikh, "Knowledge Discovery Process Models: From Traditional to Agile Modeling" in *Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications*. IGI Global, 2012. pp. 72-100.

[10] M. Alnoukari, Z. Alzoabi, and S. Hanna, "Applying adaptive software development (ASD) agile modeling on predictive data mining applications: ASD-DM Methodology" in *IEEE Proceedings of International Symposium of Information Technology*, 2008, pp. 1083–1087.

[11] M. McDonald, M. Blosch, T. Jaffarian, L. Mok, and S. Stevens, "Growing It's Contribution: The 2006 Cio Agenda". Gartner Group, 2006.

[12] Gartner Group, "Gartner says more than 50 percent of data warehouse projects will have limited acceptance or will be failures through 2007". 2005. Available: http://www.gartner.com/it/page.jsp?id=492112 [retrieved: Oct., 2012].

[13] H. J. Watson, B. H. Wixom, J. A. Hoffer, R. A. Lehman, and A. M. Reynolds, "Real-Time Business Intelligence: Best Practices at Continental Airlines". *Information Systems Management*, Bristol, vol. 23, n. 1, pp. 7-18, Dec. 2006. Available: http://dx.doi.org/10.1201/1078.10580530/45769.23.1.20061201/91768.2 [retrieved: Jul., 2012].

[14] B. H. Wixom, H. J. Watson, A. M. Reynolds, and J. A. Hoffer, "Continental Airlines Continues to Soar with Business Intelligence". *Information Systems Management*, Bristol, vol. 25, n. 2, pp. 102-112, Mar. 2008. Available: http://dl.acm.org/citation.cfm?id=1451615 [retrieved: Jul., 2012].

[15] Gartner Group, "Gartner Executive Programs Worldwide Survey of More Than 2,000 CIOs Identifies Cloud Computing as Top Technology Priority for CIOs in 2011". 2011. Available: http://www.gartner.com/it/page.jsp?id=1526414 [retrieved: Oct., 2012].

[16] O. Marbán et al. "Towards data mining engineering: a software engineering approach". *Information Systems Journal*, vol. 34, n. 1, Mar. 2009. Available: http://dl.acm.org/citation.cfm?id=1458745 [retrieved: Oct., 2012].

[17] I. Sommerville, Software Engineering. Addison Wesley, 2006.

[18] R. Pressman, Software Engineering: A Practioner's Approach. McGraw-Hill, 2005.

[19] M. Alnoukari, "Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications" in *CEPIS UPGRADE: The European Journal for the Informatics Professional*, vol. 12, pp. 56–59, 2011. Available: http://www.cepis.org/upgrade/media/III_2011_alnoukari1.pdf [retrieved: Oct., 2012].

[20] Z. Alzoabi, "Agile Software: Body of Knowledge" in *Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications*. IGI Global, 2012. pp. 14-34.

[21] K. Beck et al Manifesto for Agile Software Development. 2001. Available: http://agilemanifesto.org [retrieved: Oct., 2012].

[22] G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide. Addison Wesley, 1999.

[23] H. Hristov, Introduction to OpenUP. 2011. Available: http://epf.eclipse.org/wikis/openup/index.htm [retrieved: Oct., 2012].

[24] S. Santos, OpenUP: Um processo ágil. 2009. Available: http://www.ibm.com/developerworks/br/rational/local/open_up/index.html [retrieved: Oct., 2012].

[25] R. Kimball and M. Ross, Data warehouse toolkit: o guia completo para modelagem dimensional. Rio de Janeiro: Campus, 2002.

[26] M. Alnoukari, "ASD-BI: A Knowledge Discovery Process Modeling Based on Adaptive Software Development Agile Methodology" in *Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications*. IGI Global, 2012. pp. 183-207.