Adapting Agile Practices for Analytics Projects
(Research-In-Progress)

Dinesh Batra
Rajiv Dahiya
Florida International University
Miami, FL 33199
batra@fiu.edu

ABSTRACT

Business surveys indicate that fewer than 30% of business intelligence (BI) and data warehousing (DW) projects meet the stated goals on budget, schedule, and quality. Specifically, it has been noted that such projects are marred with delays from environment, and changes in user requirements during this period may render the project irrelevant. Usually projects are available to end users after they have been fully implemented, which can take several months or even years. Other times, end users do not see the business value of the information afforded by the projects. Agile practices can address the changes in end user requirements but because of the large size of the typical analytics project, it may difficult to apply the agile values and principles. In this study, the following research question is raised: how can agile practices be applied to analytics projects to achieve customer value and project success? Based on the agile, and analytics literature, an empirical study is proposed to examine the competing maxims for balancing agility with structured practices.

Keywords: Agile, Analytics, Business Intelligence, Data Warehousing, Customer Value
INTRODUCTION

Business intelligence and Analytics (BIA) has emerged as an important area of study for both practitioners and researchers reflecting the magnitude and impact of data-related problems to be solved in contemporary business organizations (Chen, Chiang, & Storey, 2012). Data warehousing (DW) provides the foundation of this decision support infrastructure (Ariyachandra & Watson, 2010). Business analytics helps to understand the information contained in the data and to derive insights that are most important to future business decisions (Sharda, Delen, Turban, & King, 2015).

In a business survey by Gartner, it was found that about 59% of BI projects fail and fewer than 30% of business intelligence (BI) projects meet the objectives of the business (http://www.silvon.com/blog/bi-initiatives-fail/). Sen, Ramamurthy, and Sinha (2012) ascribe the failures to lack of maturity of data warehousing processes. However, Takecian et al. (2013) assert that the traditional process for DW construction does not allow rapid and partial deliveries of functional features. and one of the most important causes for high failure of DW projects is the long development time, which leads to delays in delivery of functional features to the end user. Often, when DW systems are finally available, some of the features implemented are already obsolete, while newer needs end up being postponed until future phases of development. Barrett and Barton (2006) state that the a “big bang” approach to data warehousing almost always ends in disaster, primarily because data warehouse projects do not scale well (in terms of team size). The BIA area also faces challenges because of the need to respond quickly to the large amount of external data that may need to be analyzed sometimes on a daily basis (Davenport, Barth, & Bean, 2012).

There is considerable evidence that agile approaches lead to higher project success rates in software development (Ambler, 2012; Sarker, Munson, Sarker, & Chakraborty, 2009; Sheffield & Lemétayer, 2013). In recent years, practitioners have claimed that agile methods can be employed to foster agility in analytics in general and BI/DW in particular (Collier, 2011). However, the specific mechanisms and the theoretical underpinnings of the agile use in analytics are still not clear. To address this issue, the proposed study asks the following research question: how can agile methodology practices be applied to analytics projects to improve project success?

COMPETING MAXIMS IN DATA WAREHOUSING AND ANALYTICS

In the earlier years of analytics, the platform investments were largely in IT-led consolidation and standardization projects for large-scale systems reporting. These projects tended to be highly governed and centralized, where IT-authored production reports were pushed out to inform a broad array of information consumers and analysts (Chen et al., 2012). Data warehousing projects have typically been large and have always been difficult to develop and implement (Sen et al., 2012). In recent years, the development of the analytics projects is facing several challenges (Collier, 2011; Davenport et al., 2012) such as: size of the projects becoming even larger, more variety in the type of data stored some of which are handled using NoSQL systems (Sadalage & Fowler, 2012), a wider range of business users demanding access to better predictions and more interactive styles of analysis and insights, and a dynamic environment leading to volatile demands.

Agile software development has been the prescribed method for addressing volatile requirements (Dingsøyr, Nerur, Balijepally, & Moe, 2012; Fowler & Highsmith, 2001; Nerur & Balijepally, 2007). However, agile development has been primarily used for small projects (Dybå & Dingsøyr, 2008). Agile
software development has not been subject to the analytics domain, which is characterized by simultaneous demands of volume, variety, volatility, and variety (Phillips-Wren & Hoskisson, 2015). In his book, Collier (2011) has prescribed the Agile Manifesto as the best solution to such challenges. In fact, (Collier, 2011) (page 6) has proposed the following manifesto, which is a variation of the original Agile Manifesto (http://agilemanifesto.org/), for data warehousing and business intelligence:

“We are uncovering better ways of developing data warehousing and business intelligence systems by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working DW/BI systems over comprehensive documentation
- End-user and stakeholder collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

(Collier, 2011) acknowledges that DW/BI systems are fundamentally different from application software but “nonetheless, the Agile core values are very relevant to DW/BI systems development”. However, there is a contradiction between being simply relevant and adhering to a manifesto, where the latter represents a fundamental approach to addressing a problem or policy. Furthermore, the manifesto ends with a mitigated statement “while there is value in the items on the right, we value the items on the left more”, which adds to the confusion on how much the left side values should be weighted over the right side. According to Glazer, Dalton, Anderson, Konrad, and Shrum (2008), the Agile Manifesto is frequently read in such a way that the things on the right, which are items commonly found in too many plan-driven, contractually-driven, standards, and audit-driven environment, are not merely valued less, they effectively have zero value. The DW/BI systems are large systems and it is inconceivable that there is no contract or no plan to implement a system that can have a strategic impact while incurring a very high cost perhaps to the tune of millions of dollars (Krishnan, 2013). In the DW/BI context, therefore, one could argue that both sides – agility and planning - are about equally important. However, a detailed assessment of the four agile values indicates that the left and right sides are not mutually exclusive anyway.

First, the comparison between individuals and interactions and processes and tools is somewhat tangential. Handling large amount of data is dependent on proficient tools. For example, if testing and integration are to be automated, which one could argue will significantly foster agility, the use of tools is inevitable. The choice of DBMS for NoSQL data need to be based on the type of data (Sadalage & Fowler, 2012). The size and criticality of DW/BI systems suggest that processes that facilitate configuration management and versioning need to be documented and shared so that any failures in implementing changes can quickly be reversed, which can mitigate fears of attempting to make changes. The term “process” seems to be associated with bureaucracy but ultimately, even agile methods such as Scrum and XP (Beck, 2000; Schwaber, 2004) follow processes and workflows. Processes that enhance agility should be welcomed and adopted. For example, DW/BI systems need effective processes and tools that automate testing and integration, and manage versioning. Designers and developers can take more initiative if failure is buffered by processes that can reverse changes that have negative side effects. Thus, both tools and processes are perhaps as important as individuals and interactions, and to pit them as mutually exclusive may be inappropriate because the two sides may be complementary.

Second, DW/BI are data heavy, not working software heavy projects and it is more difficult to change database structures than to change front-end or application programs because the database tier is at the
back end and changes in the database tier can have effects on several applications. Furthermore, working DW/BI systems need a certain degree of documentation because metadata are not self-documenting. It is easy to dismiss the argument for “comprehensive” documentation (see agile value 2) because nothing is comprehensive in any software or system. Conversely, terms such as “barely adequate documentation” are subjective and may be misleading because the reader might think that little or no documentation need to be done. Suppose the originating data for a DW/BI project has hundreds of tables and possible thousands of fields; in this case, the volume of metadata requires reasonable amount of documentation, which may need to include description, provenance, and versioning.

Third, a multimillion dollar project needs a contract to protect the interests of both the customer and the vendor. The contract is as important as end-user and stakeholder collaboration; in fact, the contract may have stipulation regarding such a collaboration. Opportunistic behavior is more likely in DW/BI than in routine applications because decision support is a nebulous and unconstrained construct. A transaction like sales order has a reasonable structure but forecasting, predicting, and flexible reporting can expand at the whim of the end users. It is well known that users have difficulty specifying what they want from a decision support scenario (Naumann & Jenkins, 1982). Certainly, an agile approach can be helpful; however, it is easy to visualize how the inevitable scope creep can be detrimental to the financial interests of the vendor bound by a fixed price and fixed schedule. Conversely, a time and labor contract may hurt the interests of the customer. Contracts that balance the competing interests of the client and the vendor may need to be effectuated (Larman & Vodde, 2010; Pilios, 2015).

Finally, responding to change over following a plan is certainly a laudable goal and is at the heart of agility analytics. However, large projects do require some degree of planning, which may include designing a flexible architecture and agreeing on a shared vision with the stakeholders, and coordination among various members and sometimes among different teams.

ACHIEVING AGILITY IN ANALYTICS PROJECTS: POSSIBLE SOLUTIONS

There is very little amount of formal literature on agility in analytics, and many of the solutions presented in this section may be considered as hypotheses or educated speculations. One issue seems certain though: the waterfall approach is ill-equipped to handle the contemporary issues of volume, variety, volatility, and veracity. The DW/BI is normally represented as a sequence of stages from source data systems, data staging, data and metadata storage, and end user presentation tools (see Figure 1). A waterfall approach will likely follow this sequential approach. A problem with this approach is that by the time a complete data and metadata storage area is populated, the user needs may have changed or the project could have incurred budget or schedule issues.
There is evidence that a good proportion of the typical data warehouse is never used. A structured approach may thus overdesign the data storage and this may result in issues relating to budget and schedule. A plausible solution may be to derive the data storage from two sides: the source data side and the end user side (see Figure 2). The process actually starts from the end user side with the organization unit strategy as an initial vision. The next step is to align information requirements that provide a high customer value with the strategy. Because the highest customer value requirements get prioritized, the scope of the project gets limited during a given time period and there is little risk regarding budget and schedule. The restricted information requirements guide the presentation tools, which then governs the data needed in the data storage. The steps needed to provide the data storage are technical and need to be automated, if possible.

The solution presented here is a filtering approach with the premise that a very large system cannot respond to environment volatility. Furthermore, the most critical and the most value-adding parts of the system need to be developed first.
Figure 2: A Customer Value-focused View of Data Warehousing and BI

**Step 1 - Outline the Strategy**

Based on the organizational business strategy, the DW/BI strategy needs to be defined. The DW/BI system cannot be all things to all stakeholders. A business strategy not only involves defining the major objectives, but implicitly specifies “what not to do” (Porter, 2008). The DW/BI strategy narrows the scope further so that the entire collection of attributes do not become the focus of the data warehouse. The definition of a DW/BI strategy should be conducted at the commencement of the project by bringing all the stakeholders together for a day or two and getting them to agree on the major objectives as well as have a senior manager such as the CEO to prioritize the objectives. The collaboration and communication mechanisms need to be defined to gain user and stakeholder involvement and commitment.

**Step 2a – Prioritize Requirements based on Customer Value**

A data warehouse can take several months or sometimes years to build. During this period, the organization may not have decision support and business intelligence. Also, the requirements can change during this time. Thus, a data warehouse should start off by picking a few strategic areas and defining specific information needs that provide the most customer value (Gowan, Mathieu, & Hey, 2006; Kimpel & Morris, 2013). Successful implementation of key areas can install confidence in the project. In fact, the project may be continued as long as the marginal utility is greater than the anticipated rate of return and although this threshold point may be hard to determine. If the costs are high, instead of a failed project, one will have a partial project that delivers high customer value.
Step 2b – Build Interfaces and Presentations

The customer may not care much about technical terms such as corporate information factory and data marts. The end user interacts with the interface. Determining what the end user needs is a difficult task and is best done interactively or iteratively. To obtain early feedback, simulated rather than real data can be used. The high customer value requirements need to be explored in depth so that the project can be initiated. The prototype interfaces need to be continually developed as the project expands and real data is substituted.

Step 3 – Data and Metadata Storage

The data and metadata storage layer provides the intersection ground for iterative activity involving customer needs and database structures. During the commencement, an architecture should be proposed. During the initial iteration, one or a few data marts that can support high customer value requirements should be designed and implemented with partial or full data. A process should outline how changes will be incorporated and how versioning will be instituted and preferably automated. The storage should operate with the minimum amount of data necessary to support decisions. When performing statistical procedures such as regression and correlation, a sample of the complete data may be enough. Eventually, a corporate information factory may be developed, if necessary.

Step 4 and 5 – Data Staging and Data Sourcing

These are conventional steps except that the iterative nature of agile development will require that the stages are continually visited and modified. The use of ETL and other tools becomes essential because without automation scripts, incorporating changes will be difficult.

BALANCING AGILITY WITH STRUCTURE IN ANALYTICS

We can evaluate the proposed solution in view of the four challenges of analytics systems: large size projects, variety of data stored, the wide range of applications, and dynamic environment. Although the DW/BI project size in theory is large, the scope can be mitigated at any given period of time by developing the project iteratively and by not including every element in the source data. Furthermore, for many applications, a sample instead of the complete data set may be adequate. There is a tradeoff between speed and veracity, and although veracity may be important in some applications, it can be traded for speed in many applications (e.g., percentages) and those that require statistics. When the sample size is increased beyond a threshold, the results may not always be more useful.

Additionally, by focusing on customer value, the scope is limited to some and not all applications. (Takecian et al., 2013) note that It is a hard task to understand all the details of the available data that come from multiple sources at once and favor a modular approach, which can be complemented with the customer value focus. Gowan et al. (2006) propose earned value management, an iterative approach that focuses on continual evaluation of budget and schedule. Thus, agility and speed can be achieved without losing value, and the smaller implementation would favor budget and schedule concerns.

Despite the need for agility, the focus on structure cannot be lost because a large project will inevitably require a contract and some degree of planning and coordination. In terms of agile values, the left and the right sides may need to be deemed complementary rather than mutually exclusive. Thus, in addition to individuals and interactions, tools need to be considered to automate the process so that changes can be implemented quickly and reversed if issues such as errors or performance arise. Customer collaboration...
should be considered independently of contracts. Working DW/BI systems need to considered along with
the documentation that keeps track of changes. Some degree of planning and coordination is necessary as
changes are implemented.

PROPOSED RESEARCH METHODOLOGY

The essential research issue here needs to be dealt at the theoretical level because there is no software
development theoretical framework that guides how projects that need to consider volume, volatility, and
variety can be developed. When a theoretical framework is missing, the preferred approach is to conduct a
grounded theory study (Corbin & Strauss, 2014; Glaser, 1992). The Agile Manifesto and practices can
serve as an initial guidelines. When there is some background literature, a variation called the evolved
grounded theory approach (Corbin & Strauss, 2014).

The initial step in following this methodology is to identify agile projects in the DW/BI area. An open-
ended questionnaire can be developed by employing the literature as a sensitizing guide. The interviews
can be transcribed and by using the open, axial, and selective coding (Corbin & Strauss, 2014; Glaser &
Strauss, 1967), a theoretical framework can be proposed.

CONTRIBUTIONS OF THE RESEARCH

A recent practitioner book by Collier (2011) indicates that there is evidence that the agile values and
principles can be applied in the Analytics and DW/BI context. However, it is not clear how the agile
practices actually link with DW/BI development. Other researchers such as (Davenport et al., 2012)
emphasize the importance of agility in big data projects but do not indicate the specific practices that are
essential in completing such projects and responding to continual changes. The study is intended to
provide specific guidelines and to propose a theoretical framework for achieving agility in Analytics
projects.

REFERENCES

Wiley & Sons.

Ariyachandra, T., & Watson, H. (2010). Key organizational factors in data warehouse architecture

Intelligence Journal, 11(4), 37.


Chen, H., Chiang, R., & Storey, V. C. (2012). Business Intelligence and Analytics: From Big Data to Big

warehousing: Addison-Wesley.

Corbin, J., & Strauss, A. (2014). Basics of qualitative research: Techniques and procedures for

Review, 54(1), 43.


