The SMB’s Blueprint for Taking an Agile Approach to BI™

The people, process and technology necessary for building a fast, flexible and cost-effective solution
The Agile Approach to Business Intelligence

The word “agile” is used as a buzzword in many contexts and in different ways. We are using the word in its traditional definition: the ability to move quickly and easily, in a nimble and well-coordinated way.

So, by Agile BI, we mean ‘the ability to create BI solutions quickly and easily, in a nimble and well-coordinated way’.

Using an agile approach improves the success of BI projects, and enables you to start more projects. It does this by changing the economics, the technical solution, and the execution of the projects.

AGILE AND LEAN PRINCIPLES

In recent years organizations have been increasingly using agile and lean software development methodologies and tools. This rise in popularity is spurring the adoption of agile philosophies in other domains.

Adapting the principles of the Agile Manifesto to work with BI leads to these:

- Satisfy the customer through early and continuous delivery of valuable data and features.
- Welcome changing requirements, even late in development.
- Deliver a working solution frequently and measure progress by this.
- Foster a closer working relationship between business people and developers throughout the project.
- Build projects around motivated and knowledgeable individuals.
- Decide late, deliver fast.

The frequent delivery of a working solution will obviously solve some of the problems BI projects face including:

- Communication and vision gaps will be reduced in each iteration as end users see the working results.
- Development latency will be significantly reduced.
- Shortcomings of the top-down or bottom-up approach will be alleviated as rapid iterations allow a hybrid approach that combines or alternates them.

LEAN DELIVERY

You can reduce development tasks and costs by using the “decide late” principle. By treating the first delivery of a BI solution as temporary until proven otherwise, you avoid extra work and cost. Some examples of savings are:

- Use manual flat-file extracts from source systems instead of fully-automated data flows.
- Extract a partial (but still useful) set of data. The data can be limited by a time range or can be restricted to a subset of a geographical, organizational or other dimension. Make sure the extracted data is fully useful to a subset of users, not partially useful to all of them.
- Transform the data into simple fact tables instead of star, snowflake or other complex data schemas.
- Install the solution on existing hardware, or cloud-based hardware.
- Use open source databases, middleware, and front-end software instead of proprietary software.
- Don’t bother with automation, auditing, production controls, etc.

Monitor the usage of the system for a month or two.

Using an agile approach improves the success of BI projects, and enables you to start more projects.
AGILE HARDWARE

If you need to acquire computing hardware before a BI project can begin, you can run into trouble. In some cases it delays the start of a project, in other cases it is a contributing factor in a project’s cancellation.

To get a project going quickly, or enable a prototype to be conducted cheaply, you’ll find it advantageous to use one of the following:

- **User hardware:** Using existing desktops, workstations, or laptops means no procurement delays or budget spending. A desktop environment is great for a business analyst or technically-oriented end-user to get started on a project.

- **Cloud computing:** Cloud computing quickly and cheaply make a BI solution available to a distributed group of people. This includes both public clouds like Amazon EC2 and private clouds like Eucalyptus.

In some cases user’s hardware is locked down and only certain applications are available to them such as office productivity, email, web, and corporate applications. In these case cloud computing gives technically-oriented end-users a new option.

In most cases a BI solution will go into production on dedicated, on-premise hardware. But prototyping and development can be done on desktop machines and cloud environments. The ability to migrate easily from user hardware to cloud environments, and cloud environments to static deployments further increases the productivity of the team and the flexibility of the project.

For these hardware options to be viable, the BI software must be suitable (in terms of licensing and hardware requirements) for all those environments.

AGILE TEAMS

An agile BI team is typically made up of 4-5 people, each typically having one of these roles: IT Developer, Project Manager, BI Consultant, End User, Business Analyst, and/or a Database Administrator. Any of these people is capable of starting a project on their own.

Many spreadmarts in existence today are complicated and intricate. Most have been constructed by end-users because an officially sanctioned BI solution is neither available nor planned. This shows that there is a population of technically-oriented end-users who are willing and able to create BI solutions. Having these individuals on the team and giving them tools that enable them to experiment will help BI projects significantly.

Ideally the team should be based in the same location, and if they can work in the same room most of the time, that’s even better. Regardless of location, the team should be provided with tools to help them collaborate, such as forums, mailing lists, wikis, and a document/content management system.
The software must scale up to meet the demands of the production deployment, but it must also scale down onto laptops and utility hardware.

Cloud computing quickly and cheaply make a BI solution available to a distributed group of people.

**AGILE SOFTWARE**

An agile approach works best when iterations of the BI solution are frequently delivered to a group of end-users, who provide valuable feedback and changing requirements based on the progress so far.

This implies some requirements on the software used. The BI software used should:

- **Support quick iterations**: Iterations will take longer if the tools are cumbersome, hard to use, or do not work well together.
- **Offer full BI capabilities**: Even the quickest prototype or iteration is likely to involve data transformation, data quality, modeling, visualization, and content creation.
- **Make basic features easy to use**: The software should enable technically-oriented end-users to participate in or initiate development of a BI solution.
- **Allow delivery to a large audience**: Valuable feedback will be lost if the licensing of the software restricts the potential pool of end-users providing feedback. For this reason you should avoid software that is licensed per-user.
- **Allow prototyping**: The ability to perform prototypes or pilot projects at will, without the hindrance of software licensing issues, enables many more BI projects to be considered for development.

**AGILE BI**

Agile BI allows BI practitioners to behave in new ways:

- A BI project can be started by a single end-user, business analyst, IT developer, database administrator, or consultant.
- Different participants can be engaged sequentially, not simultaneously. An end-user, business analyst, or consultant can create a BI project, then the IT group can institutionalize it over time, as its usage dictates.
- A BI project can be developed on a laptop, on a hosted service, in the cloud, or in a data center. The project can be easily moved among these environments.
- A prototype can be completed for less than a few hundred dollars cash outlay, or no cash outlay.
- Spreadmart developers become BI developers, and have the advantages of both: control, flexibility, self-sufficiency, scalability, security, and reliability.

**AGILE BI USE CASES**

Agile BI can be used in different scenarios. Some examples of using Agile BI for projects that are driven by IT:

- **Fast Track**: Take your most important BI project, and be agile with it. Create a prototype using existing or cloud hardware. Iterate quickly - weekly, daily, or even hourly. Provide access to it for a large user community. Enable the users to communicate and collaborate together. Iterate until they are happy with the data and the content. Only at this point should you decide whether to bring the project on-premise or not. Don’t fully institutionalize the project until after 6 months of consistent usage have passed.
- **Backlog Shotgun**: Perform a quick bottom-up iteration of all your backlogged BI projects. Use cloud computing for the hardware. Always use real data: end-users cannot get excited by fake data, nor can they find data quality issues that exist with the real data. Let your users explore the solutions for a few weeks then let them decide which ones to develop further. In each iteration take their top requests and implement them in no more than 4 weeks. See which projects get traction and which ones fade away: institutionalize the successful ones.
• **Data Quality Hunt:** Provide bottom-up solutions of your operational systems to let the users determine where interesting data fields are not consistently populated. Alter the application logic or operational procedures so that those fields become suitable for future analysis.

Some examples of using Agile BI for projects that are driven by end users include:

• **Spreadmart Conversion:** Find your spreadmart authors, provide them the tools to turn their spreadmarts into scalable, secure, centralized solutions, and give them the ability to enhance and develop those solutions further. The central IT group can provide access to a ‘dimension store’ which contains standard hierarchies for the organizations main dimensions (products, geography, business units etc). Providing ways for developers to check the consistency of their data with these standard dimensions will improve quality, consistency and lower integration costs.

• **Scratch Space:** Provide some on-premise or cloud-based hardware to your technical end-users and let them create their own prototypes and solutions. Monitor them to see which are used frequently. Turn these into supported solutions.

### The Boundaries of Agile BI

So where are the boundaries of Agile BI? What is not ‘Agile BI’?

Agile BI is not a product - it is combination of technology, economics, and execution that enables new behaviors.

Agile BI is not an alternative to the Kimball Data Warehouse methodology. Agile BI provides new ways to approach BI projects. You can use Agile BI to create data-marts one at a time or in parallel, and then use the Kimball DW methodology to approach the creation of a data warehouse.

Agile BI, because of its iterative nature, it is not ideal for fixed-price, waterfall-style projects. As an alternative approach, some consulting companies offer their technical expertise on a ‘pay-per-iteration’ basis specifically to support agile projects.

Agile BI is not the same as BI delivered using a Software as a Service (SaaS) model. SaaS BI offerings are hosted, are typically focused on a specific domain, can be hard to customize, and are not easy to move out of their hosted environment.

A BI project can be developed on a laptop, on a hosted service, in the cloud, or in a data center.

Agile BI is not a way to falsely under-estimate the long costs of BI projects. It is a way to incrementally invest as the value is proven, and a way to make use of utility pricing if suitable.

### Summary

Agile BI changes our perception of BI projects by dramatically changing their economics and execution. Instead of regarding them as something that ‘the organization might start next quarter if they can line up the resources’, they can be viewed as something that ‘I can start this afternoon’.