Where Agile Meets BI

professional advantage

Table of Contents

Introduction

Business Intelligence (BI) is an ideal candidate for the agile approach to software development: it can deliver very quick business value, and it benefits from agile built-in philosophy of continuous improvement.

However, agile software development can seem at odds with what business managers and IT departments are comfortable with. Business and IT alike are used to drawing up requirements, creating a detailed design document, allocating a budget, and receiving a finished system at the end.

In a very real sense, business intelligence is never "finished", because business is never static. The requirements you stipulate this year could well be obsolete next year: your product lines might be different, your business units might be different. You may have entered new markets not included in the original system design, or restructured your business.

To keep offering value, business intelligence needs to be as flexible and dynamic as your business.

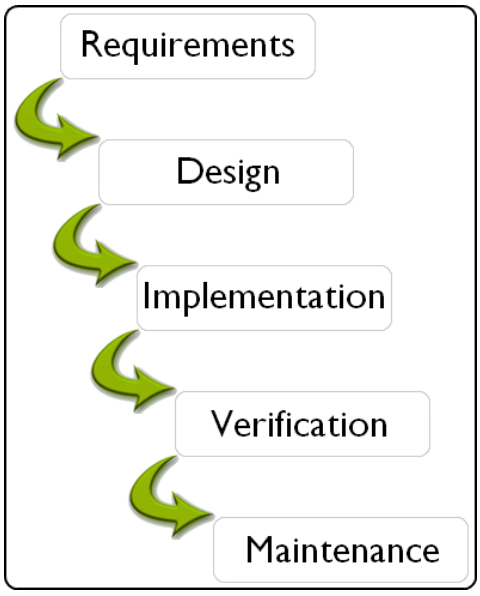
In the software discipline known as "agile development" the aim is to be iterative and incremental. Instead of spending one or two years creating a finished system that corresponds perfectly to a design document, the agile development project seeks to put a partial deliverable in front of users as quickly as possible, and use their feedback to help refine the design and define the features.

That approach is just as well suited to large implementations like business intelligence. In a BI implementation, the software is a stable product in the implementation, the two things that have to happen are the "plumbing" (creating the connections between the BI software and its myriad data sources), and the user experience.

Since the user experience is a large part of the value of BI, it makes sense to get the users as deeply involved in the implementation as possible. The challenge is to put together a plan that convinces the sceptics to get behind an agile implementation.

Agile development

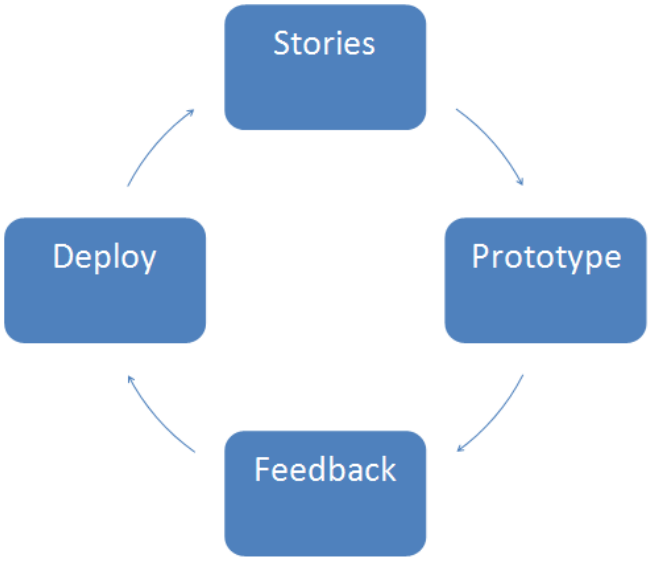
The traditional software development methodology, known as "Waterfall Development", is still relevant and mandatory for many projects. It's especially well-suited to key transactional systems, which perform a defined and relatively stable set of functions and have a long lifetime.



This has problems for more dynamic systems like BI, however:

- **Inflexible:** if your business changes during the implementation, it's hard to respond in the BI rollout.
- **Front-loaded costs:** you've spent most of your budget before you see any results (and before you identify problems).
- **Documentation conflict:** end users can't write BI specifications that mean anything to developers, and developers' documentation doesn't mean anything to end users.
- **Ownership:** for transaction systems, there are long-standing processes for developers to hand over a finished system to those that will maintain it. BI systems are handed over to end users as much as the IT department, which creates an abstracted ownership that gets in the way of future modifications to the system.

The Agile approach is more of a conversation. It invites the users into the implementation process early, gathers their feedback and input into the process, and adjusts the implementation based on the conversation with users.



The characteristics of the Agile BI approach are:

- It's an incremental approach instead of a "big bang" project.
- The focus is on developing prototypes instead of specifications.
- Business users are involved throughout the process.
- Agile is reactive, rather than master-planned.

The Agile BI implementation process helps resolve the problems with the Waterfall Development, and in addition provides:

- **Fast time-to-value:** with a usable system in front of end users early in the process, you get to see returns with only a small proportion of the budget spent.
- **User engagement:** users with a sense of ownership over the system are more likely to integrate it into their day-to-day working lives, maximising the value of the BI system.
- **Flexibility:** if your needs change during the rollout, you'll have the chance to make sure your new requirements are reflected in the BI system.

The Agile BI Checklist

1. Develop and deploy quickly

To keep the conversation going, you first have to get the conversation going. What you specify now is going to be obsolete in a year or two, and quick prototyping of the first deliverables gets runs on the board early in the project.

Select something that users are asking for, and look to deploy it in 15-20 days. That lets you begin the Agile BI cycle of collecting user feedback to produce the next iteration. While that process happens, your implementation team can continue working in the background, integrating the BI system with the data sources it eventually has to work with.

2. Be Agile in your project management

Abandoning the "Waterfall Development" project management style is confronting, but important. Remember, your first deliverable is not meant to be a finished system.

Instead, at each iteration through the Agile BI cycle, encourage project managers to ask "what did we do well? what worked? what needs to be changed?"

Encourage the conversation, and encourage users to suggest changes. Even if you don't adopt every user suggestion, you will be adopting enough, to get a genuine sense of ownership over the implementation.

3. Have Agile infrastructure

Since you're not delivering a finished system on day 1, it makes no sense to try and build your entire infrastructure at the outset.

A "Waterfall Development" project will provide very detailed specifications of the system it needs to run on: processors, memory, storage, transaction capacity and so on.

An Agile BI implementation will work better, and will cost less, if you deliver the infrastructure as and when required. This is a great place to consider cloud computing, since that will let you deliver the infrastructure on a just-in-time basis.

4. Agile IT departments

Rather than planning for the next 12 months, manage the IT department so that it is ready to respond to smaller, shorter-term requests.

Some IT functions are unchanged: they provide the infrastructure, manage the data, provide the security and conduct backups. The management skill needed is the ability to distinguish between whole-of-project IT department roles, and the more immediate, responsive, Agile roles.

5. Data architecture matters more than system design

While Agile BI doesn't demand the detailed design and specification of a "Waterfall Development" project, you need a vision of where you're heading, a high-level data model.

The data model will identify the data sources you're working with: the financial data, sales data, forecast and pipeline data, CRM data, and external "big data" sources such as Google Analytics and so on. The data model also identifies the relationships between the different information you're working with.

You won't be implementing analytics to all of these sources from the start. However, the data architecture provides the context in which the implementation is taking place. It defines the structure you're working towards, hopefully in a comprehensible form that fits on a single page.

The aim of the data model is two-fold: not only does it define your target, it also alerts you to any necessary changes to the model, that emerge as the implementation proceeds. That clear vision will make each of your iterations progressively more productive.

professional advantage

Call: 1800 126 499
Visit: www.pa.com.au
Email: enquiries@pa.com.au

About us

Professional Advantage is one of Australia's most awarded solutions providers, with over 20 years experience in helping organisations improve their business systems through industry leading software solutions. Our 250-strong team in 6 offices across Australia and internationally has successfully worked with over 1000 organisations.

© Professional Advantage® Pty Ltd. All rights reserved. All other trademarks and copyrights are property of their respective owners. BM3626-05-2013