BI Projects

“The NewFrontiers Approach”

Development Cycles and Managing Change

Patrick Teunissen (CEO and Founder)

August 2007 V2.5

How to run a BI project?

Notice the word "project". At NewFrontiers we believe that there is a major difference between classical business transaction projects (like CRM and ERP implementations) and BI/Reporting projects. The difference is that the objective of a business transaction project can be specified exactly, whereas for BI/Reporting it is much more difficult to articulate your information needs. The explanation is simple: the end result of processing business transactions will lead to knowing exactly what happened previously, whereas for your reporting requirements the challenge is to look into the future.

To explain the above think of the following example It should be quite straightforward to describe what the sales order handling process looks like at a very detailed level. Forms in use can be easily accessed so the ERP software can be configured based on the "as is" information, dealing with process handling, printing and exchanging of data between multiple parties.

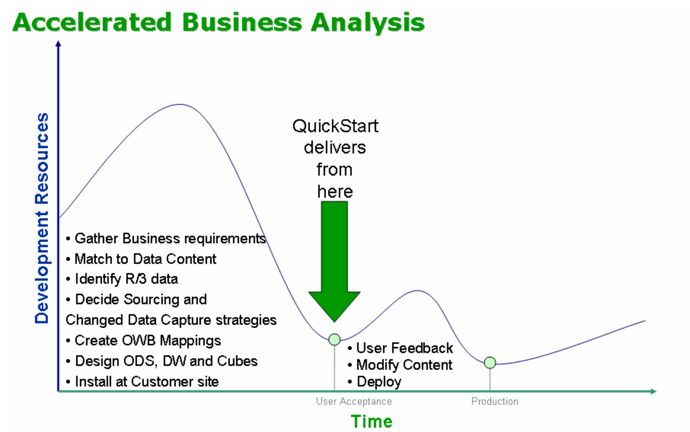
Contrast this with the analytical needs of an organisation, they are much more difficult to predict. Your customers may be taken over, new products and markets may need to be explored or maybe your own organisation changes. This also explains why only looking at the reports currently in use is a big mistake!

So how to approach BI projects today?

At NewFrontiers we recognised the trend of larger companies to standardise their transaction processes on pre-packaged business software some years ago (1999). Although our focus has always been on the delivery of information we understand that today's ERP software plays a key role. Understanding the ERP transactional software helps us enormously to take the first step when implementing a proper management information system.

How do we at NewFrontiers propose tackling BI projects?

Instead of running long investigatory exercises in which we try to specify the requirements of an organisation at a very detailed level we propose you install a "QuickStart". "QuickStart" is the name we have given to our pre-packaged data warehouses, loaded with smart business content and analytics. QuickStart is a well thought through application, with the in-built flexibility to grow and change. What we deliver out-of-the-box are measures of unambiguous facts, e.g. "revenue" as the value on the invoice or "days paid (too) late", which can be reported against a set of base dimensions. QuickStart delivers fast results and within a week key metrics can be viewed by business users.



Fast results enable business users to deliver fast feedback. From this feedback the QuickStart can in most cases be quickly added to and extended, such that within a few weeks the business users can have a production reporting application tailored to their specific needs, based on their feedback, derived from real-world data.

Starting a QuickStart project does not mean that after the implementation there will be no more change. In fact we expect needs and requirements to evolve and we call them cycles. In most cases QuickStart goes through 3 cycles until the formal "go-live".

Cycle one (1) – Pilot, "try and proof". In this phase the focus is on the "technology", i.e. can the QuickStart deal with multiple sources; can the application easily change and grow; how flexible are ad-hoc querying capabilities; do the deliverables fit with the high level business requirements.

Cycle two (2) – Validate, "check and understand". In this phase the focus is on the actual usage of the "technology", i.e. make sure that the business users validate the reports which should include their first round of feedback.

Cycle three (3) – Implementation, "fit for production use". In this phase the detailed reports are reviewed, data and analysis errors are corrected and the users sign off the end-to-end design.

What kind of changes can you expect?

The measures delivered by QuickStart are often not subject to much internal discussion, however the accountability (internal) for what is being measured may be. Often organisations put filters on the measures or group them in line with their specific views and structures. The dimensions delivered via the packaged QuickStart are very basic, so often users will want to have attributes added. QuickStart has been designed to allow that to happen. A typical example is an attribute on customers: companies want to total up separate 3rd party customers and also aggregate sales from group customers.

Appendix - Explaining the phases

Gather Business Requirements

Information/reporting requirements are often articulated by business managers in a very generic way at a very high level of abstraction, e.g. “I need sales information”, “we do not have a view on the efficiency of our logistics”, “we need a decision support tool to lower the cost of our net capital employed”. In this phase IT often tries to document and agree the “deliverable” with the business users. The classic way is to do that at a very low level of detail and agree on all the measures. At NewFrontiers we believe that the first requirements gathering phase should only be done at a very high level. Almost nobody in the business knows exactly what they want to look at in the future and even if business users do, they find it extremely difficult to articulate. Business users rarely explain exactly what they need because the analyst that they talk to often does not know what their job is really about.

At NewFrontiers we like high level requirements such as “we want a tool to monitor the efficiency of our purchasing process and have a view on our stock levels to reduce our working capital requirements”. We recognise the needs of this phase and in our view it is much more efficient that business users give feedback on a real application using real data as the method to gather business requirements.

Match Data Content

When business users explain to IT what they want to look at in analytical application(s) they mostly do not have enough background to understand what is available in the underlying transactional application (e.g. SAP ERP). Hence the first next step is to match the user needs to the application and then to the data in those applications. This requires that the people that do that work understand the business requirement as well have a good understanding of the way the systems (like SAP ERP) have been built.

At NewFrontiers we have pre-matched (mapped) many high level business requirements to the modules of SAP ERP. We have done this based on our experience with more than 25 customers and on an end-to-end transaction processing model. That means that we have anticipated the source transactions that will be needed to provide the data to fit and complete the (data) chain.

Identify R/3 data

After you have figured out what data comes from what module in SAP R/3 you need to identify the underlying transaction tables and master data. This in all cases requires a more technical SAP consultant (e.g. ABAP, Base consultant). Obviously these people communicate and look at systems differently than BI consultants (let alone business users).

At NewFrontiers we have pre-mapped the transaction data stored in the tables of the different transaction processing phases to subject areas for analytical purposes. To explain in simple English, when you want to do sales order analysis we have already mapped the ERP sales order tables.

Decide on (sourcing) a changed data capture strategy

After the go-live you will likely want to update the analytical application with new and changed data, e.g. update revenue reports daily with the latest billing data. Since transactional systems like SAP ERP can easily have millions of billing documents in the database you need to find an efficient way to only update the BI application with new ones, as selecting "all documents" would simply take too long and probably put to much load on your ERP transaction system.

Finding the right triggers is one of the most complicated things to do, but at NewFrontiers we have figured out what works best for every type of transaction.

Create process mappings

This could also be read as "write the program that physically extracts the data from SAP ERP, changes it and than loads it into the BI application". NewFrontiers has a lot of experience with all the different (ETL) tools that can do this task. We have even played an important role in the development of the current SAP integration of the tool we most often use, Oracle Warehouse Builder (OWB). In QuickStart you will find all the programs and processes that take the data from SAP and store it in the BI application.

Design, ODS, Cubes and Report writing

The next "technical" thing to do is to design/build up the Data Warehouse (DW) application. The ODS is the Operational Data Store and contains the transaction data at a detailed level. The cubes are specific subsets/aggregates (a.k.a. data marts) that are used for the on-line analytical processing, in user terms drilling, slicing and dicing etc. Reports are the way data is most often delivered to users.

Needless to say that the ODS, cubes and report design are all included in QuickStart and they will be populated with your data as soon as the OWB jobs (see above) start running.

Based on the user feedback you or your system integrators can start to enhance and modify the application, so that it works for you and delivers exactly what you need!

