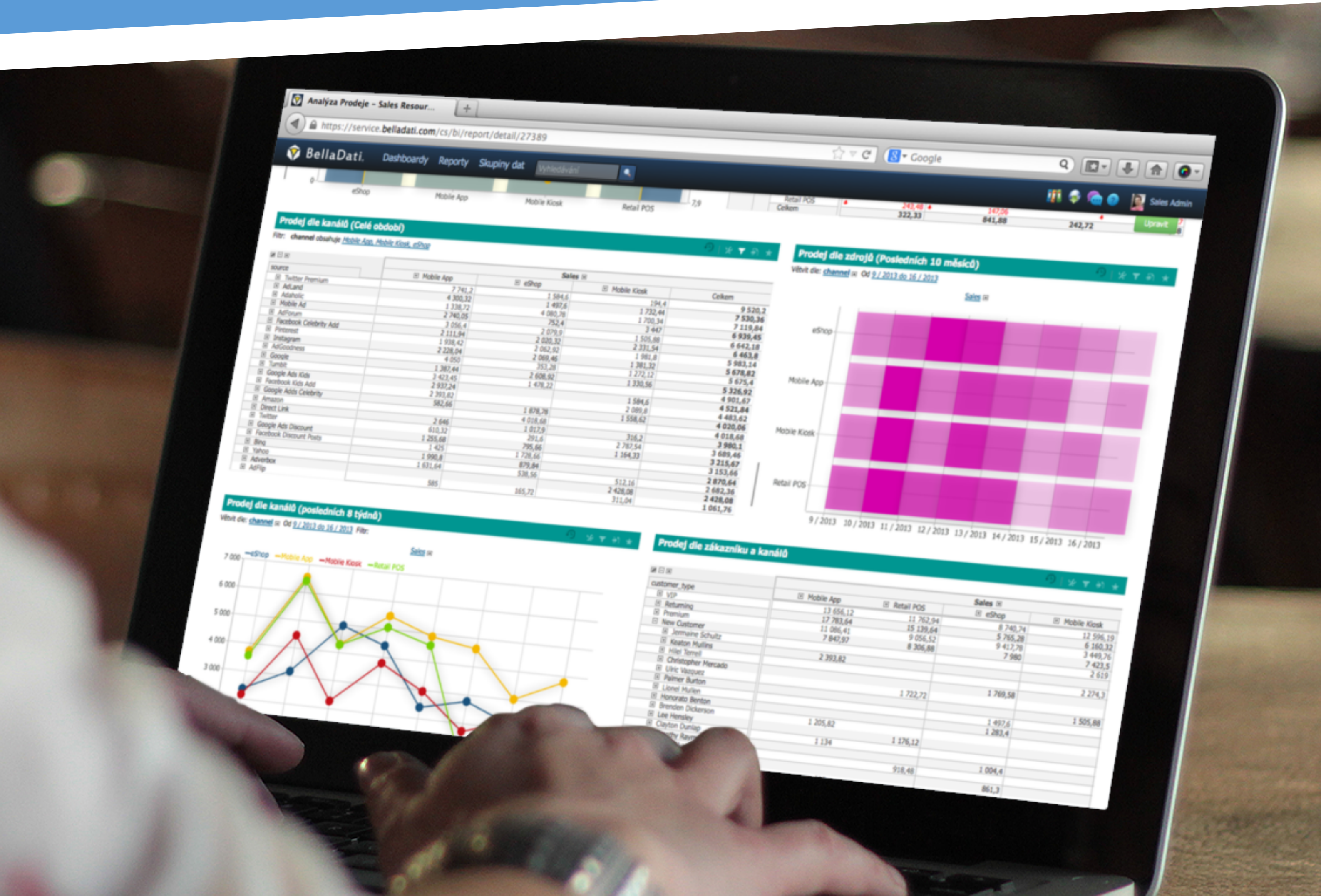
BellaDati

To “BI” or not to “OEM”?

A short discussion about the traps you might fall in when developing BI services for your product on your own.

Peter Kesch, EMEA Business Development, Vicepresident
(peter.kesch@belladati.com)


Analýza Prodeje - Sales Resour...
https://service.belladati.com/cs/bi/report/detail/27389
BellaDati
Dashboardy
Reporty
Skupiny dat
Sales Admin
Prodej dle kanálů (Celé období)
Prodej dle zdrojů (Posledních 10 měsíců)
eShop
Mobile App
Mobile Kiosk
Retail POS
Prodej dle kanálů (posledních 8 týdnů)
Prodej dle zákazníku a kanálů

To “BI” or not to “OEM”

In todays IT world Data is getting more and more important. There is a lot of discussion about Big-Data, how to manage data and how to apply it in order to gain benefits. This discussion, actually, is not so much about the data handling as such, but more about getting valuable information out of the available data.

Now, in order to provide such valuable information and to make it transparent to the users of IT systems, more and more Software Development companies have to make a decision if they want to add BI services to their own product by in house development or by integrating an existing BI solution (OEM) into their product.

The demand of the users is quite simple. They want to analyze the data which is stored in the IT systems, they want to play around with the information, combine different data-sets, visualize the data in different charts-types or tables, they want to interact with the results- filter and modify them and, finally, make their findings available to others via e-mails, Web and mobile.

Actually, what they demand is **BI**!

Before we are talking about the difficulties you might face when developing BI services on your own, lets first get a clearer picture about BI and what it actually means:

Extract from WIKIPEDIA:

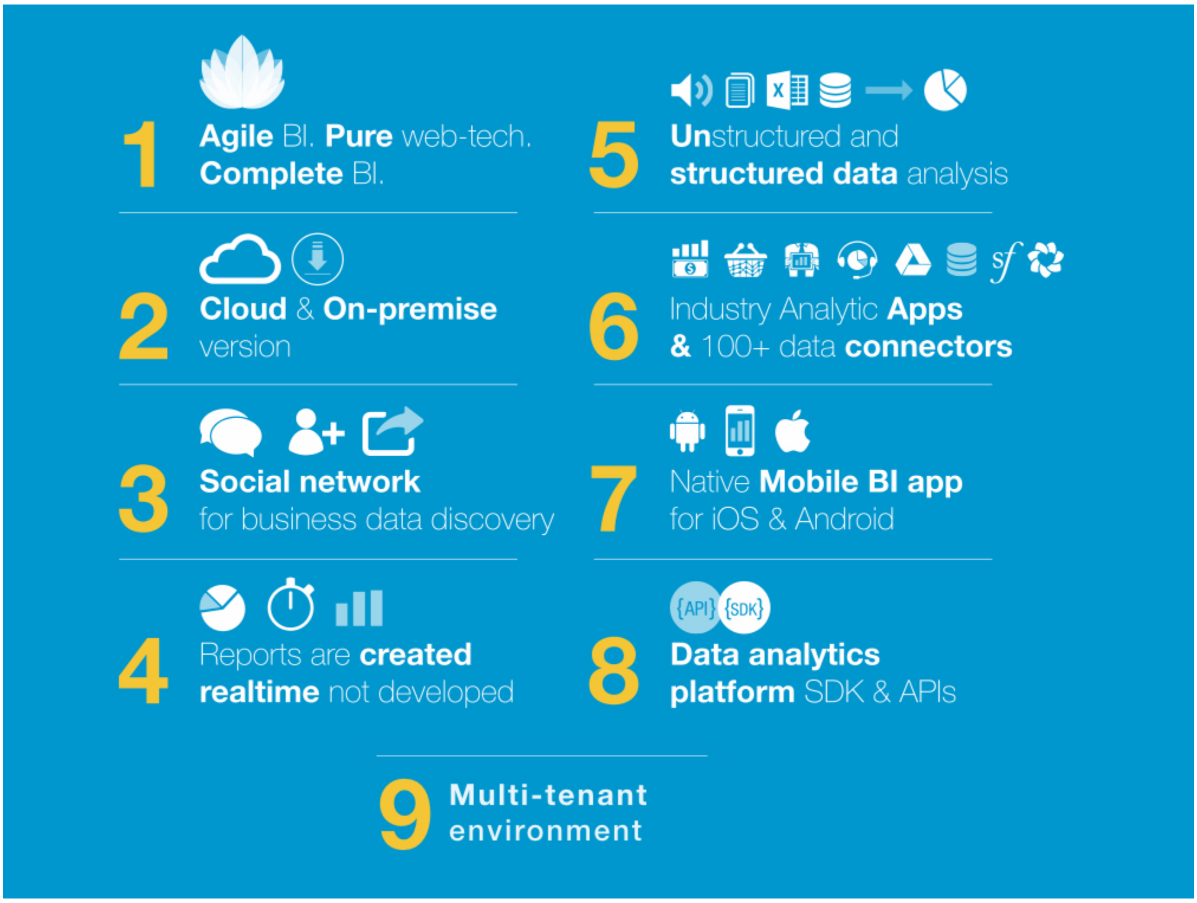
Business intelligence (BI) *is often described as "the set of techniques and tools for the transformation of raw data into meaningful and useful information for business analysis purposes"[. The term "data surfacing" is also more often associated with BI functionality. BI technologies are capable of handling large amounts of unstructured data to help identify, develop and otherwise create new strategic business opportunities. The goal of BI is to allow for the easy interpretation of these large volumes of data. Identifying new opportunities and implementing an effective strategy based on insights can provide businesses with a competitive market advantage and long-term stability.*

(source: http://en.wikipedia.org/wiki/Business_intelligence)

Having a look at the user needs it is quite clear that BI, as defined in Wikipedia, is the right solution to fulfil their demands.

Now, should those BI functionalities be implemented from scratch or should an existing BI solution be bundled with the own product?
This, indeed is a difficult question and lots of product managers struggle in finding the right answer.
Of course, we at BellaDati have a very clear answer to that question:

"Integrate an existing BI solution and select BellaDati ☺"



1 Agile BI. Pure web-tech. Complete BI.

2 Cloud & On-premise version

3 Social network for business data discovery

4 Reports are created realtime not developed

5 Unstructured and structured data analysis

6 Industry Analytic Apps & 100+ data connectors

7 Native Mobile BI app for iOS & Android

8 Data analytics platform SDK & APIs

9 Multi-tenant environment

However, at the end of the day, the decision will be made by you based on an extensive analysis and it is only up to you which way to choose.

But let us give you some hints about traps you might fall in when you tend to develop BI services for your product on your own:

Top 10 Traps you might fall in when you develop BI services for your product on your own:

Trap No. 1: simple charting vs. flexibility

Typically, one trigger which makes you thinking about BI for your product is the demand for charting. Users want to visualize the data which they have in their system(s). Adding charts to an existing system seems to be quite easy at the first instance. There are lots of charting libraries out there which can easily be integrated in any product and it is quite simple to generate nice looking charts with those libraries. The question you have to ask yourself is:

Is it sufficient for us to provide more charts only? Or do we need flexibility in charting, allowing the end-users to select the data which they want to analyse and the chart types? Do we want to allow modification on charts on the fly, such as filtering of data or adapting the look and feel by changing colours, fonts, styles? Or, even worse, should the end-user be enabled to build charts completely on their own and be able to add comments, discuss about charts within the chart?

Adding such interaction capabilities is quite a demanding task and could increase the implementation time by factors. Time to market is an important topic which should not be underestimated.

The "good old time" of IT where end-users have been happy when they saw a nice chart is over. Users nowadays expect flexibility and self-service in charting and you should provide that.

Recommendation:
If your customers and their end-users are happy with pre-defined charts and limited flexibility, go with a charting library and implement it on your own. If you (better said your customers) need flexibility, go with a ready made BI solution.

Trap No. 2: Data consistency

Nowadays you might think that data in the IT systems is consistent. Unfortunately, this is not the fact. In most cases data is created by humans and humans make mistakes. Double entry of data, different writing conventions, typos… all this are reasons which create inconsistencies (just to name a few of them). And, data cleaning / cleansing is not an easy task in IT systems as their data is normalised and

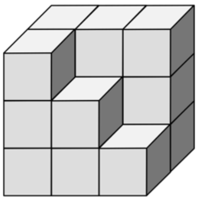
relational stored in databases and cleaning always also means re-structuring (re-building the relations) and in most cases also re-calculating application based summaries.
In transactional systems such inconsistencies are painful, but do not disturb the operation of the system as such. Talking about analysing data and getting information out of such systems, however, does.
Analyses of Data and charts are often used for decision making and if your decisions are based on inconsistent data they might be wrong.

Another way of having a look at data consistency is the timing of the reports. On-line real-time reports have advantages and disadvantages. The biggest disadvantage of online-reports is the consistency of reports among viewers. Let's imagine a user having a look at the actual sales analysis early in the morning and saves the report, another one checks the same report in the afternoon the same working day. Their numbers will be different. In order to provide consistent reports and the flexibility to play around with the data you need a snapshot, an extract which is used for analysis and is the same for all users. Also here the build in ETL functionality steps in, providing exactly this functionality.

Existing BI solutions typically (not all of them, however...) have an integrated ETL process. ETL stand for "Extract" "Transform" "Load" and by using the "Transform" functionality data can be programmatically modified, and cleaned. This allows you to make sure that the data is consistent and charts are based on real facts.

Recommendation:
If you intend to implement BI services on your own, make sure you also implement the "T" out of "ETL"!

Trap No. 3: Data enrichment



IT solutions store lots of data which is needed for their operation. For data analysis this is sometimes not enough and it is necessary to combine data-elements.
Lets have a look at a simple example:
In a customer database the customer has a field for the first name, one for the middle name and one for the last name. In a report, however, you need the full name.
Also here the "T" from the "ETL" can help out and you can easily create a new field which contains the fill name, a combination of first, middle and last name.

Adding such new fields is not an easy task in transactional, relational systems, for ready made BI solutions this is easy.

Recommendation:
Make sure you think about data-enrichment when implementing charting!

Trap No. 4: performance

Performance is a more technical topic but a very important one. Most of IT solutions store their data in a relational database and the data-structure is normalized. Normalization means that data fragments (attributes) are stored in such a way that the data is kept only once and changes of one data record impacts all other records which have a relation to that specific record.
Let's have a look at a simple example:

We have a customer database with the address of the customer including the town, the zip-code and additional classification attributes for the town. In the database the customer record holds a reference to the related town record where the classification attributes such as number of inhabitants are stored. If the number of inhabitants changes it only has to be updates once in the table where the records of the cities are kept and each customer which lives in that specific city keeps untouched. If a query for a specific customer is executed it can follow the link to the city and also return the number of inhabitants of this town.

This has huge advantages and helps keeping data clean.

But this also means, that extracting all details about a specific customer means following links to other tables and executing sub-queries to get all necessary data.
For transactional usage of a system this is not a problem. Databased optimize the queries and can provide this information quite quick. But if you think about analysing all customers based on their purchase habits in combination with their location this can create a huge workload on the database-side and have a big impact to the reaction time of transactions.
Actually, this was one of the main reasons why BI systems have been developed. Extensive analyses should not, in any case, have a negative influence to the system performance.

BI systems have a different approach to store data. Their data is stored in a way which is most performant for analyses. Data sets are kept redundant, which means in our example that each customer record also hold all fields from those related tables, which are necessary for the analyses.

Now, you might think this will result in much more data which has to be stored and you are right. However, hard disks are cheap and for the sake of performance and flexibility it is a must for agile BI. The cost factor can be neglected here.

Recommendation:
Do not create BI services based on your existing relational database. Always keep data for analyses apart from transactional data.

Trap No. 5: data security

Data security is a very sensitive topic and IT systems are typically well prepared to handle this when it comes to daily operation. When we talk about reporting we have to look at security from a different angle.

Each report can be a combination of different data with different permissions and a report should not necessarily reflect the permissions a user has on data in the transaction system. Just imagine a department manager who should not see the details about transaction in other department, but should be able to benchmark his department against other. So some of the data should be visible for him in all details, others only at a high level.

This means that the reporting module should provide a permission system which allows the report creator or the manager of the underlying data which data the creator or the report user should be able to see in what detail.

A typical example here is budget reporting. The implemented reporting system should be able to define only one budget report and each manager should be able to see only details from his department using the very same report.

Recommendation:
Make sure you implement a sophisticated authorization system for your reporting!

Trap No. 6: data sources

ERP systems cover a wide range of functionality but depending on the industry a customer works in there are lots of additional systems in use for the daily operation. Each system is specialized and covers parts of it. None of those system cover the complete operation.

This also means, that data is not only store in one location, data is distributed in different systems, stored in different databased or even Excel sheets and kept on different locations.
For BI and sophisticated data analysis, this also means that the reporting system has to be able to read data from different systems and combine it in one report.

A typical example here is Warehouse-Management and Customer Relationship Management. The first one is typically done in an ERP system, the latter one in a specialised CRM System.
If an analysis has to be done to show the purchase-delivery time for customers in a specific region (CRM) related with different types of products stored in different warehouses (ERP) using different shipment channels (external delivery company), the data from multiple systems is needed.

Therefore, even if we talk about implementing BI functionality in one specific product, the BI module has to be able to grab (extract) data from different systems and combine it in one report. And, the report creator should be able to add different data-sources on the fly, adding Excels from different departments or connecting to a local database.

Recommendation:
Keep your reporting module open and allow import of data from different sources. And, make sure you also include unstructured data from social networks or NO-SQL databases in your import capabilities!

Trap No. 7: Office Integration

Microsoft Office® is everywhere and represents the standard for business documents. This also means that reports and charts must have the capability to integrate in Office products.

Users expect that charts can easily be integrated in Word documents or PowerPoint slides and that tables should be available for export in Excel format (and we talk here about real excel format, no CSV-lists.

Recommendation:
Make your charts and tables available for Office Products!

Trap No. 8: Distribution channels

Distribution channels are a very important topic when we talk about BI. Creating reports and charts as such is not everything. Those reports also have to be distributed to the users and they can be very creative in finding new ways of distribution. Typical distribution channels are:

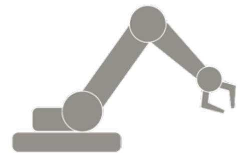
- Reports in E-Mails as attachment
- Reports in E-Mails as Mail-Body
- Available via Web
- Available on the smartphone
- Push to an app

But this is not everything. Here we are also talking about distribution groups for specific reports, notification services, periodically sending of reports.
Users want to access information everywhere and at any time without restrictions and this should be taking into account.
And, managing different distribution channels also means thinking about data security:

- Which reports can be distributed?
- Which ones should only be available online?
- Which ones can be pushed to a mobile?
- How the data is secured on the mobile device?
- And much more …….

Recommendation:
Ensure that different distribution channels are part of the product from the early beginning! And think about mobile first also for reporting!

Trap No. 9: Automation

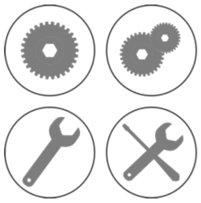


As we have seen in Trap No. 4 own data management for BI is essential. This also means that an integrated BI functionality should be capable of scheduled data loading and automating all necessary steps in the "T" part of ETL.
Also, if data is loaded automated it must be ensured that the data is kept consistent and it has to be managed that loading of data from one source should be able to trigger loading of data from each dependent source, including managing fall-back scenarios in case of problems and restart capabilities.

Loading data also means that the Extraction service has to be able to load only incremented records which means only those records which have been changed and / or updated since the last import.
This is a huge task and needs full attention when implementing BI services.
Only if the data export is consistent and reliable the reporting provides value.

Recommendation:
Implement a sophisticated and reliable Extraction and Automation service for your BI services!

Trap No. 10: Maintenance and new Features



Last but not least let's have a look at maintenance and new features. Once a first version of your BI service is available it will by far not be complete. As soon as end-users can play around with analyses they will find bugs and demand new feature.
There are lots of BI companies out there, developing their products for years and optimising the usage and functionality in order to satisfy their customers. Just have a look at the complexity and traps which are described in the pervious mentioned paragraphs. We at BellaDati know exactly what it means to implement a sophisticated BI solution which is powerful and flexible on the one hand, but also easy to use on the other hand.

Recommendation:
Make sure that you understand that implementing a flexible BI service for your product is not a task of several months, but a task for several years!

Summary:

Implementing BI services as an extension to an existing product is a very demanding task. In the first instance it might look easy, just adding some charts and allow some flexibility.
Having a closer look however, shows the complexity and reveals the effort and time which will be needed to do it right.
If some charting alone will be sufficient for you, there is nothing wrong with implement some charting on your own.

If charting alone is not your goal, you should think twice if you really want to go down that road and start implementing a complete BI functionality. Taking in account the time needed and the effort necessary we would not recommend to re-invent the wheel again.

There are lots of BI solutions out there, some of them more complex, others just basic and only a few of them prepared for white-labeling and complete integration by providing an SDK and API.

BellaDati is very well prepared to be integrated and provides a solution to avoid all the top 10 traps and even goes far beyond the typical needs.

But at the end of the day the decision is up to you only ☺.