

Questions to ask when evaluating an Enterprise Reporting or BI suite.

The world of reporting is becoming increasingly more complex and there a myriad of vendors that have solutions which address various aspects of the needs encountered in Reporting and Business Intelligence (BI).

This paper does not address in detail what Business Intelligence is, because it can mean several things to several people. The Business Intelligence market is in rapid growth phase as at the end of 2004 and is predicted to continue consolidating (Business Objects bought Crystal Decisions in 2004, Hyperion bought Brio, etc.).

So with all these various offerings and some very advanced capability and features, why is it that most companies still struggle with reporting, and that over 50% of Business Intelligence projects land up failed or cancelled ? IT would seem that lack of proper planning, failure to clearly define expectations and vendor bamboozlement are major contributors to the lack of success so rife in this industry.

When your organization is small with very few users and a few thousand records in a local or central database, you can afford for everyone involved with reporting to develop, deploy and run all reports. However once the organization grows beyond a certain critical size, a much more disciplined approach is required to ensure success, accuracy and user satisfaction.

Information (after cashflow) is the lifeblood of a business, and rapid access to real-time information can mean competitive advantage, which is becoming critical for survival in the modern information age. All too often, the way that information is delivered has not kept up with advances in technology and needs to be rethought.

This paper will address in detail, the list of questions that ought to be asked prior to embarking on an Enterprise Reporting initiative. It will also touch on some BI points without diving too deep into this particular subject.

Table of Contents:

Who	3
Describe the types and number of users in the organization that will run / receive reports	3
Security	3
User-based, role-based, position-based information	3
Ability to control where reports are distributed	3
Real-time or replicated information	4
What	4
Define the type of information required	4
Exception Based Reports	4
Management / Executive Reports	5
Data Sources	5
Architecture questions	6
Replicated data or live data ?	6
Replicated Data	6
Scalability	6
Size of datasets	7
Server, memory, disk and network usage	7
.Net or J2EE ?	7
Automated Scheduler Architecture	8
Features & Ease-of-Use	9
Interactivity	9
Parameterized reports	9
Drilldown and drill-across	9

Composite Reports and Dashboards.....	10
Annotation.....	10
Multi-lingual capability.....	10
Spreadsheet Integration.....	10
Totals: Report totals & group totals	10
Formatting & Templates.....	11
Master-detail reports	11
Integration.....	11
XML Integration.....	12
Consolidated Reporting across multiple systems.....	12
Administration.....	12
Configuration & Development	13
Change Management	14
Learning curve.....	14
End-User Training.....	14
Developer Training.....	14
Administrator Training.....	15
Performance	15
Long-running reports.....	15
Running from server cached datasources	15
Native RDBMS Access	15
Prepared statement / bind variable usage	15
Network Bandwidth Usage	16
Compliance & Auditing	17
Licensing Considerations	17

Who

Describe the types and number of users in the organization that will run / receive reports.

Just as there are various jobs / roles / positions within any organization, there can be multiple requirements for different kinds of information. The CEO of a large manufacturing concern would probably want daily profitability summary information and any other type of report that would show summary information about various aspects of the company.

Division or line managers in the organization would want daily or ad hoc information about market trends, supplier summary information, production summary information, sales & HR summary information.

Regular employees might require information about competitors, opportunity & sales information, Service Request information, parts & inventory, RMA and other detailed type of information based on recent time events (today, this week, month-to-date, etc.)

The Reporting system in the company should be able to address the needs of all of these types of users. The DSS & executive type of users would require information that is much more sensitive than the information required by the customer-facing clerical type users. The Reporting system should ensure that access to information is guarded, yet open to the next questions that will be asked.

Security

Most reporting systems these days are web deployed and viewable through a browser that connects to a web server. A robust and mature reporting technology will be integrated with single sign-on technologies, like LDAP, and should support most of the popular LDAP offerings.

If the Reports or a portion of the Reporting Infrastructure is exposed to the internet, then security should be a top consideration, and usernames and passwords might not be enough. In this case, a VPN connection makes a lot of sense, especially if sensitive information is to be transmitted. Also, for sensitive information, SSL (https) should be a configurable option.

User-based, role-based, position-based information

With various applications, the information that is visible to a user may be restricted to a subset of the whole, based on the user's login attributes (example: Siebel position-based visibility to data). This is normally only provided by means of using a COM-like API call to the application, which may have performance implications, as most COM interfaces handle records 1 at a time instead of in a batch-like mode. Also, the ability to drilldown, using COM like interfacing is almost non-existent outside the main application interface.

Ability to control where reports are distributed.

Most enterprise reporting products have the ability to schedule reports that are delivered via email or ftp. But the security aspect needs to go one further. Besides having audit trail capability, a definite advantage would be to have some controls as to where reports can be emailed. Where groups of companies are involved, then certain information should not be able to be emailed via the scheduler outside the list of these various domains. Many such email based schedulers are happy to accept an email address and to send the reports to this email address. This opens up the case of an insider using the automated scheduler to send for example, a fresh daily new opportunity report to someone

who works for the competition. A simple domain validation feature would prevent this type of industrial espionage from happening in the first place.

Real-time or replicated information

A very important question to ask right up front, that could determine a lot about the reporting infrastructure and architecture is deciding the urgency of having real-time information, or if the information can be periodically replicated to an offline source where it can be summarized and scrubbed in a way that does not affect the production system.

If Real-time information is sought, then this information should be as limited as possible. What is meant by this is that real-time information should be limited by date-time range, to only allow data to be accessed from whatever threshold is acceptable for the company. Should Real-time sales data be limited to 'today', 'This week', 'this month' or 'this quarter' ? The answer to these questions depends on the volume of data.

A large TV-based shopping network in Florida is known to have transaction burst rates of over 10,000 TPM (transactions per minute), with an average of over 1 million transactions per day. Information that pertains to 'This quarter' might span hundreds of millions of records, depending on the entity being looked at (Order line items). Having real-time access to this information might not be in the best interest of overall performance, as the online users will suffer and so will the result of the business. However, summary information that might mean on-the-fly summarization of today's orders by various demographic attributes might be an acceptable task, as the databases will be very powerful, but even then, there may be times, like during a peak-time promotion when there are hundreds of concurrent callers online placing orders, when this kind of information gathering for a report might drag down the overall performance of the system and thus hamper normal business activities. In the example stated above, there might never be an acceptable time to run aggregate-based summary reports against the OLTP live system. The same might be true for any truly 24 X 7 large-scale operation.

What

Define the type of information required.

There are various types of reports that can be generated by any number of Reporting Vendors. Typically, the most important of these is exception based reports, followed by executive level summary reports.

Exception Based Reports

If your enterprise handles 20,000 calls per day of which 100 are not resolved satisfactorily within the day, then the reports to be created should show the 100 calls not closed off, instead of the 19,900 that were handled fine. Other examples of reports might be to show a report of any customers who have had more than 2 complaint-type Service Requests in the last 30 days, as these would be the disgruntled customers that the company is in danger of losing and require the detailed attention of at least a follow-up call.

Management / Executive Reports

This often involves summary and aggregated data by various demographic attributes (sales per salesman, per office, per product line, per division, per day / week / quarter) where some attributes are related to each other in a hierarchy.

The questions to be concerned about are:

- Is this reporting done by querying a cube or the actual main source of the data itself?
- Is the data summarized with pre-computed summaries ? If so, how old is the data ?
- If the summary is against the main source (summarization done by reporting tool), is the summary computed by the rdbms using SQL and Analytically-enhanced SQL (fastest) or is all the data pulled from the database, across the network to the reporting tool summary engine to be computed using proprietary computation methods (slowest) ?
- How readily can the results of the report be verified against the actual data ?
- How much of the summarization for management reports can be done using SQL directly against the datasource ?
- How open is the reporting tool to parameterization ? This will determine optimal indexing too, unless all the data is dragged to the reporting engine, which will severely limit the scalability of the reporting product itself, leading to server farms where really a single server that lets the RDBMS perform the summarization, would have sufficed, but this way the parameterization is easier, leading to a trade-off that favours spending a lot more on getting server farms at the enterprise level.

Data Sources

In an enterprise environment, there is often more than just a single database or a single application. In a large multi-national with thousands of users, there are probably various types of RDBMS technologies too, not just the different business systems.

- Does the reporting system allow you to report from the various databases ?
- Is there a licensing concern for each new datasource ?
- What about legacy systems and mainframe databases (ADABAS, IMS, etc) ?
- Does the reporting software let you run reports based on data from ODBC datasources?

The reporting system chosen should be able to run reports against any of these data sources without severely impacting the licensing costs.

In an ideal utopian setting, the reporting system should be able to join information from one type of datasource to another without too much difficulty, in real-time, without having to drag everything over to a cube type structure. Thus if your Marketing system resides in Access, your contacts reside on Oracle, and your billing & payment information resides on SQL Server, a report that will show which outstanding marketing campaigns are targeting any contacts that have any late payment history in the last 12 months, this should be achievable without having to drag the contents of each of these databases into a data mart or cube.

Architecture questions

Replicated data or live data ?

This is one of the most fundamental questions to ask and will determine a lot of where the entire reporting infrastructure goes. There are pro's and con's to both approaches.

Replicated Data

In a reporting solution that uses replicated data, the data is taken from its original source and replicated either partially or wholly, to another environment, whether that environment is a data mart, a data warehouse or another type of OLAP / ROLAP / MOLAP structure. This approach has some distinct advantages and disadvantages, and the reader needs to make the choice of the trade-off.

Advantages of a replicated reporting solution:

- Generation of reports does not impact online business performance.
- The data can be replicated into an environment that is more suited for the types of reports to be delivered, for example: multidimensional analysis and cubing, RDBMS specifically configured for OLAP.
- Data from more than 1 single source can be brought into the same database / schema and thus consolidation of information from different systems can be achieved.
- Data can be checked and scrubbed in the ETL (Extract, transform and load) phase, so that data that gets to the target system is cleaned of erroneous data.
- Summarized data can be pre-computed, thus improving the speed of aggregate type queries on the data.
- Subsets of data can be isolated to different users by means of ETL constraints.
- Packet transmission sizes can be optimized to reduce the number of network round trips from OLAP source to client.

Disadvantages of a replicated reporting solution:

- Data is as recent as the last ETL run. If your requirement is for real-time information, this approach will be VERY costly as you will need to implement trigger-based real-time replication and the dependencies that accompany it in a relational model.
- Data is often denormalized, leading to large-scale redundancies.
- The storage requirements for a DW / Datamart are between 3 – 5 times the size of the original data.
- Another machine/s & database to maintain.
- More servers required for ETL, thus more backups, OS licensing costs, etc.
- Additional network bandwidth required for the ETL process / processes.
- Specialist OLAP skills required for proper and efficient functioning of these environments. OLTP DBA's do not automatically make good OLAP DBA's, it is a different mindset, a different skill set, a different focus and often comes at a premium. **Hint:** you get what you pay for. Many OLAP projects are severely hampered because of inexperienced / under-trained staff with OLTP type skills that are put onto OLAP projects without the necessary training or mentoring.
- Loading of data through ETL has its own set of dependencies (in a star / snowflake schema, you cannot load Fact tables until dimension tables are loaded) which adds additional points of failure

Scalability

Many books have been written on this one subject alone. When it comes to enterprise reporting, it really boils down to how many concurrent users can you get onto a single server with acceptable performance ?

Regardless of the Reporting vendor chosen, the number of servers required for an enterprise deployment should be the minimum. Scalability should also include things like the ability of reporting enterprise servers to load-balance across the server farm (if applicable).

A single dual CPU server (whether Intel or RISC based, whether Windows, Linux or Unix servers) with 2 GB or main RAM should be able to support at least 300 report requests per minute coming from various points within the logical network and be able to adequately.

Another important point to consider when thinking about scalability is related to a previous point, as to whether reporting is happening on a replicated data set or not. If so, the network scalability needs to be taken into account.

Size of datasets

The size of the datasets that are being processed and the size of the datasets that are being sent to the client application need to be considered.

This point also forces one to refocus on the idea of exception-based reporting. Do you really want to print out a 20,000 line (300+ pages) report (this is not a way to become popular with the Greenpeace movement) every day for several users?

Server, memory, disk and network usage

This topic overlaps with scalability, but the reader can appreciate that deploying a reporting solution to 10,000+ users will have a significant overhead. Whether this happens on a farm of 40 servers or on a single server or anywhere in between will make a huge difference to the selection process.

When it comes to network usage, this will be discussed in a little more detail in the 'Performance' section of this paper, but for now, the location of the reports server in relation to the location of the datasources it must extract data from is rather important. Minimizing the number of network hops is of prime importance, to reduce router and switch traffic. This is most easily achieved by placing the reporting server on a common subnet with the most popular datasource. Some organizations that have hundreds of different reporting datasources will no doubt undergo a complex planning process for this one aspect alone.

.Net or J2EE ?

This topic is bound to spark controversy, but is important when choosing the vendor. There are many papers on the internet discussing the comparison between the two, and this paper will not go down that road, but will prompt the user to ask a few pertinent questions:

- Which of these technologies is deployed as a standard at the corporation?
- Which of these technologies is seen as more secure?
- Which of these technologies is more open to a change of platform?
- Which of these technologies is embraced by the rdbms vendors?
- Which if these technologies has more qualified technical professionals in your area from which to harvest resources?
- Which web server have you chosen to run your mission critical applications, and then: is this web server supported by J2EE or .NET ?
- Do either of these technologies lock you in to a specific vendor for future upgrades?
- Licensing costs associated with either technology.

Licensed J2EE Web servers are:

- Web Sphere (IBM) – [expensive, but very stable with string international following]. IBM support.
- WebLogic (BEA) – [not as expensive, but probably the most stable of all J2EE app servers]
- SunOne (Sun Microsystems) – very robust, a ‘must-consider’ if your application server runs on Solaris.
- JBOSS – open-source (GPL no-cost license), support will cost, but a worthwhile investment for enterprise deployments.
- Tomcat – open-source (GPL no-cost license), based on Apache, the most widely deployed web server. Version 5 is very stable, and development continues at a rapid pace.

The J2EE consortium comprises the following corporate members: Oracle, IBM, Sun, HP.

.NET is a Microsoft supplied non-open-source product with a license fee and will only run the IIS web server.

Automated Scheduler Architecture

Most enterprise reporting offerings include an automated scheduler to enable unattended running and distribution of reports. Look more closely into this feature of any reporting tool. The questions to ask are:

- Is the reporting scheduler mechanism tied in with the operating system (windows ‘at’ or unix ‘cron’ based) ?
- Does it support multiple time zones?
- If the scheduler is down for a period of time, what happens to any reports that should have been run during the outage?
- Do users have the ability to schedule their own reports?
- What safeguards are there around abuse?
- If a scheduled report run fails for any reason (datasource unavailable, query error, delivery network down, etc.) what is the sequence of events for retrying and notification ?
- Is an audit kept of scheduled runs?
- What is the scheduler delivery mechanism (SMTP, ftp, http, web service, etc) ?
- How many points of failure are there?

Features & Ease-of-Use

One of the most important sets of questions to ask is of course, what features do you get and how easy are these to use?

Interactivity

- Are the reports generated static, or is there a chance to interact with the reports (filter data, sort data, summarize data)?
- Can you hide certain columns in a report?
- Can you drag column over as an end-user to customize the look of the report?

Parameterized reports

- How easy is it to set up parameterized reports (example: please choose the Region to report sales for) ?
- How does the reporting handle complex and multiple parameters ?
- Can the parameter list be a query in itself?
- Can one parameter drive the available options for the next parameter (example: please choose state, then please choose city [where the cities shown are contained within the state chosen for the first parameter).
- Does the report allow you to choose multiple values from a parameter list (please choose city / cities) ?
- How does the parameterization handle dates ?

Drilldown and drill-across

Many reporting packages allow one to drilldown to the next level of detail, but some of these have some severe limitations that need to be questioned.

Some reporting solutions allow you to expand rather than drilldown, that is where all the data is sent to the reporting engine, which will perform the aggregation and then present the aggregated data, and when drilldown is activated, it will open that particular node. This has performance implications both positive and negative, and will be discussed in performance.

- Can you drilldown from a summary based on one datasource to the detail in another ? What is meant by this is: If you have an aggregated cube and you are looking at sales per region per month, if you click on a particular month within a region, can you drilldown to a report that shows the details of those sales, where the detailed records reside in a separate and different datasource (example: the OLTP system) ? How is this achieved ?
- What is the maximum number of drilldown levels one can go to ? (Example: From sales by month → drilldown to sales by region for month X, → drilldown to sales by branch for Month X for Region Y, → drilldown to sales by sales rep for month X for Region Y for branch Z, → drilldown to sales by sales method (cash, credit card, voucher, etc) for Rep A for Branch Z for Region Y for month X, → drilldown to invoice summary for sales method B for Rep A for Branch Z for Region Y for month X, → drilldown to invoice line item from here, → drilldown to suppliers per product found on the invoice line item → drilldown to order history per supplier for supplier of product → drilldown to the order to expand order line item. How easy would a hierarchy of reports like this example be to configure?
- Could you filter any of the sub-reports as you drilldown?

Composite Reports and Dashboards

- What level of effort is required to create a dashboard where several summary reports are shown in a single page where each report can be based on a different division / business unit / measurement and each dashboard sub-report might even be from a different datasource or data type (example: one sub-report is from sales data on Oracle while another is from marketing campaign tracking on SQL Server) ?
- Can you drilldown on any sub-report of a dashboard (composite report) to expose further detail and the data that makes up the summary ?

Annotation

Reports are often sent further up the chain of command within a company, all too often with post-it notes stuck to the printout. When post-it notes are lost or discarded, often valuable information about the report is lost too.

If somehow the report can be annotated and comments about what should be looked at (see point on exception based reporting above) or what the report means, this will ensure that the urgency of what triggered the report to be sent further up in the first place, is not lost as it goes up the corporate hierarchy to individuals with decision-making power.

Multi-lingual capability

In true multi-national corporations, applications will be run in a variety of languages, not just English. If your corporation fits this description, ensure that the interactive reporting can accommodate all of the users for whom reports are created, in their particular language. This includes things like interactive dialogue, parameter prompts, help and error messages, column headings (how many people would understand 'verkaufs betrag' (German for 'sales amount') as a column heading? Or 'venditore' (Italian for 'sales person') as a heading ? It would be a really neat feature if the Italian, German and English users each ran the same report, but the Italian user saw 'Venditore' as the column heading for their sales report, the German user saw 'Verkäufer' for their sales report and the English user saw 'Sales Person' as the column header for the exact same report.

If your organization does not use multiple languages, then this feature is not a point of consideration.

Spreadsheet Integration

Spreadsheets (Excel) have become a very popular method for storing, manipulating and working with data. There are some products (TM1 by Applix) that enhance the power of spreadsheets, using the TM1 MOLAP engine to create very powerful BI functionality, including 'what-if' scenarios on reports.

Microsoft Excel opens itself to be a datasource in that it supports ODBC calls (and JDBC via the JDBC-ODBC bridge in Java) using the Jet engine to drive relational queries.

- Can the reporting technology easily save reports to .XLS or CSV (comma-separated values) file format?
- Can the reporting technology treat a spreadsheet or CSV file as a data source and create reports based on relational queries to the spreadsheet?

Totals: Report totals & group totals

- What level of effort is required to display group and grand totals in the report ?
- What about users who want to export the data and not have the totals? Can totals be turned off at the client level by the user?

- What sort of totals can be computed in the report (count, min, max, average, sum) ?
- What about statistical analysis functions (variance, etc).
- Does the reporting software easily lend itself to ROLLUP and CUBE operations?

Formatting & Templates

The old way of creating reports was to create a template of what the report is to look like, and then to create fields on the template and then let the reporting software populate the fields with the queried data.

The drawback to these templates is that they had to be loaded onto every server and there were some distribution issues with templates and prior versions of the template.

These days, the only time that templates are really required to be used is when the report is something like a certificate, an Invoice or credit note or some other business stationary or anytime that the output must conform to an absolute specification in terms of font, displacement, graphics.

To have to specify, design, create and modify templates for each and every report in an organization is counter-productive. Also, it does not allow the end-user to hide certain fields that might not be of importance to a particular user.

Dynamic templates (created by the reporting software itself implicitly) is the way to go, for greater productivity gains and flexibility.

If templates are to be used, are the templates based on an industry standard, like Adobe PDF Forms or is the template a proprietary format that locks you into a particular vendor ?

If you were to switch reporting vendors, what level of effort is required to recreate the templates / reports used by the previous vendor's software?

Master-detail reports

Often the drilldown from a parent report to the subordinate level would like to see a master-detail type report, where the summary information is displayed at the top of the report, with the details of what the summary information is based on, is displayed at the bottom.

What level of effort is required to create master-detail reports ?

Integration

These days, in large corporate enterprises, rarely does 1 single application hold all the information for a corporation. To get a real-time true picture of all of your business, various integration technologies exist, including queue-based message systems (like MQ Series, Vitria, Tibco, etc.). Most integration in the 21st century happens via XML, since XML is easy to learn, easy to generate, easy to process.

In addition to this, there is still a huge use of mainframes by large corporations, and mainframe databases store the majority of corporate information. The most popular mainframe databases are IMS, Adabas, DB2 (MVS) and Datacomm (to a lesser degree). Other legacy type technologies are AS400 systems, for which not a lot of middleware and reporting solutions exist.

Often applications will already perform some business logic and processing on data prior to displaying it to a customer. Rather than having to duplicate this logic through SQL or querying, if the application can create an XML dataset from the application and send this XML file to the reporting server as the

data which to process and create reports on, that would be far more expedient, and save the additional call to the database.

If your organization is one that uses mainframes or the AS400 type of computing technology, the questions to be asked of report vendors are:

- Does the reporting software allow a direct real-time connection to/from legacy applications? Or does it rely on ETL processes that need to pump the data from the mainframes to another OLAP type infrastructure?
- What requirement is there for real-time data retrieval from mainframes ?
- Can the legacy system that is built on these mainframe technologies make a call to the reporting system to provide reports?
- Can an application make a call to the reporting system to provide real-time information based on information that is based in another system / database ?

An example of this type of integration might be where a CRM application that resides in Oracle needs to get some billing and payment information which resides on IMS on an MVS mainframe system. A customer calls and would like to know why the credit they were promised 3 months ago has not been applied. In this case, the CRM customer service representative, who is proficient with the CRM application might need to open a separate application to access the mainframe data, or if the CRM application can make a request to the reporting system citing the customer's details, and force the reporting server to do the lookup and retrieve the billing and payment information in real-time from the IMS mainframe application.

XML Integration

Most modern applications use XML in some form to communicate with external applications. XML is no longer the future, indeed it is the present way of cross-application communication in large heterogeneous environments.

- Does the reporting software process incoming XML as a datasource?
- Can the reporting software produce XML output which could be fed to another application?
- How committed is the report vendor to XML technology?

Consolidated Reporting across multiple systems

Does the reporting software easily facilitate the relational joining of entities cross-platform and cross-rdbms?

An example of this might be to create a consolidated report of the top 10 selling products, and for each product, show the available suppliers for this product. This could involve the joining of product information to the sales system in a common database, and then from the products catalog, to join in supplier information which could potentially reside in a totally different system with a different rdbms technology. A truly advanced enterprise level reporting application would allow one to easily achieve this.

Administration

- Who administers reports?
- How are reports administered?
- What level of responsibility is assumed by the reports administration for the configuration?
- What level of effort is required to set up a user?
- Who creates privilege groups that reports can be assigned to?
- Are reports assigned to privilege groups or can anyone run any report?

- Are there certain times assigned to when users can run certain reports?
- Who or what determines the reports that a user can see and has access to?
- Who creates the connections to the various datasources? What level of effort is required?
- When templates are used, who creates, loads and migrates these?
- Who assumes responsibility for the scheduler?
- Who receives notifications of any error condition within the enterprise reporting system?
- Who is the point of contact for technical issue resolution with the reporting software vendor?

Configuration & Development

Some report vendors will insist that since end users know exactly what information they need, the end users should be the ones creating the reports they run. This usually involves a significant learning curve before an end-user can create a single report, and if the end user does not practice the skills, soon the skills are outdated or forgotten, thus leading to large-scale wastage of training, time, and perhaps forcing the user to do something that is not in their list of competencies. For instance, a senior accountant has significant value to the corporation as an accountant, but when this same person is forced to spend 2 days creating a report that performs poorly (because the accountant does not know anything and should not be required to know anything about optimal database access or SQL) that a competent developer could have done in 10 minutes, this is a much larger cost to the company than simply the cost of the software.

Would a small services company feel confident about sending in the CEO to defend a legal case brought against it? No, they would hire a professional attorney to represent them. So why then do so many report vendors insist on turning business professionals into reporting professionals? This approach costs the company in time, money, and often leads to poor overall performance, which in turn leads to diminished business opportunities, which costs even more.

Questions to be asked around report development and configuration are:

- Do end users create their own reports?
- How many of the end reports are based on user-specific data?
- Does the report creation allow for developer comments, which make maintenance and enhancement of reports easier?
- If a report is changed, what effort is required to restore the report to its previous version?
- Can format masks be applied to fields (example: date formats, thousand separators, rounding, left/right justification, X and Y axis labels, column heading alternate display names, etc.)?
- Can the look & feel of a report be easily modified or customized (different colours, fonts, grid line colours, etc.)?
- Can you change the order in which reports appear on a menu?
- What level of effort is required to switch from one graphic chart to another (switch from pie chart to vertical bar graphs or plot-point chart)?
- What types of charts do you mostly use at the corporation and does the reporting vendor support these? The most popular types of chart are pie charts, bar graphs, multi-series bar chart, plot-point charts, Gantt charts [for project management or timeline reports].
- How are reports tested in a non-production environment first?
- Can the reports easily include corporate logo's and other types of static graphic files?

Change Management

In large organizations where there can be literally thousands of reports, an organized process is required in order to stay on top of the many requests for new reports and report enhancements. This topic will raise a few questions in itself:

- How long does it take to get a new report created on average?
- What level of effort is required to migrate a tested report definition from one environment to the next ?
- Are the checks in place to ensure that two developers cannot be changing the same report concurrently?
- What happens if a developer working on a report is not available (vacation, sick, training, termination) and the work needs to continue?
- Can reports be 'accidentally' modified?
- If a report is being changed / worked on by a developer, how can other developers tell this? Are other developers prevented from making changes to this report too?
- How much effort is involved in rolling back a report definition to a prior version?
- What auditing is in place to see the list of changes made to a particular report definition?

Learning curve

Regardless of the reporting technology eventually chosen, there will be a learning curve involved for the end users, the Administrators and the Developers.

End-User Training

How long will it take to get the average user proficient in the use of the reporting tool? What sort of training is offered? Is it on-site classes, web seminars, CBT, recorded demo's, written manuals?

The effectiveness of the training is inversely proportional to how long it takes a user to start using the reporting software, the longer the training, the less likely the users will want to delve deep into it because of the complexity.

From an end-user perspective, the software should be as intuitive as possible, whereby the user could almost take themselves through the entire reporting system features in under an hour.

Developer Training

The longer the developer training, the more tied in the company becomes to a single vendor.

For example: if the developer training is 2 weeks long, that represents 2 weeks that the developer will be away from the office, and it makes the developer's replacement that much harder to cope with.

How much of the in-house database skills can be harnessed? Or does the developer who is experienced with 1 vendor's products (whether BI, reporting and database products) have to learn almost everything from scratch?

How easy is it to find skills for the particular reporting / BI suite in the job market?

Administrator Training

Once again, Administrator training should be as intuitive as possible. Some advanced administrator training courses run for weeks, which means that not only will the administrator be gone for possibly weeks at a time, but the company should have an understudy (or more) to back up the Administrator in case of staff turnover, vacation, sickness, training, etc. The usefulness of Administrator training can be inversely proportional to the length of time taken to train.

Performance

This is probably one of the most important considerations for large enterprises who are already stretching their hardware resources to the limit.

Long-running reports

Does the software provide a configurable 'governor' that will arrest runaway queries and terminate the sessions in the database?

How are timeout limits configured?

What about some 'more important' users who absolutely must have their reports? Can the governor settings be over-ridden for these users?

Can the timeout limits be varied throughout the day?

Once a user launches a report, can they stop it? If so, does this actually terminate the session in the datasource or just at the client level?

What information is given to the administrator in terms of who is running what and when?

Can administrators terminate any running sessions?

Running from server cached datasources

Does the reporting software have the ability to configure a report to be run from cache?

How long are cached data sets valid for?

What if a user wants a refresh from real-time data, is this functionality available?

How are cached datasets cleared out to prevent server storage requirements from growing endlessly?

Native RDBMS Access

Does the reporting software allow the query to be accessed with native rdbms query language?

Example of this is to use Oracle-specific SQL enhancements inside a query, Oracle hints, SQL Server-specific syntax when accessing SQL Server, use of PL/SQL or Transact SQL user-defined functions.

Is access to database via ODBC or JDBC? (ODBC is much slower and more cumbersome and much more limiting in what can be done).

Is the reporting software open enough to allow a performance tuning specialist with advanced knowledge of SQL to be able to modify the queries generated to allow for optimized SQL to be run?

Prepared statement / bind variable usage

Nothing limits database scalability more than the non-use of bind variables (prepared statements).

What is meant by this is that for parameterized reports, if sales are desired for a particular state, to limit the amount of data returned by the database, the query will have the state as a predicate of the where clause and the value for this is to be provided via a parameter.

For instance, the prompt for the parameter may ask the user to choose a state and then feed this state back to the SQL statement where clause:

SELECT from where STATE='CA'.

If another user runs the exact same report for the state of NY, this would translate to:

SELECT from where STATE='NY'.

In essence, this is the same statement, but due to the hard-coding of the values 'CA' and 'NY', the database optimizer cannot see this as the same statement, and will thus load the statement into the library cache twice, causing no reuse or sharing of the statement. Each of these statements will need to be fully parsed and then loaded, causing possible library cache overruns and very poor performance.

A prepared statement would look like this:

SELECT from where STATE=?

Or like this in Oracle:

SELECT from where STATE=:1

The '?' or ':1' (bind variable) will be then fed to the SQL engine, which will then reuse the statement and only do 'soft parsing' which involved only the very parts of statement parsing, that is where the bind variable placeholders are filled in with the values. This leads to good database performance.

All too often, where reporting products access the database directly, for parameterized reports and drilldown reports, these values are hard-coded into the statements / queries that are sent to the database. This should never be the case, because scalability in large environments is extremely important.

Network Bandwidth Usage

For large reports that travel over a LAN, or worse still, a WAN, network resource conservation becomes of paramount importance. Most modern software should be aware of the benefits that http compression provides.

Does the reporting software have a built-in ability to compress the data it sends from the reporting server to the client?

If so, what is the typical compression rate and what is the threshold before network (http) compression kicks in?

Compliance & Auditing

In November 2004, Sarbanes-Oxley will become a required standard in the USA. Similar compliance and auditing standards are being established world-wide in an attempt to provide better corporate governance and increased protection to stakeholders, where company insiders are being called to greater levels of fiscal and security responsibility.

If there is a requirement for compliance and auditing at your corporation, the questions to be asked of the report vendors are:

- Can the reporting software provide logs of who ran which report when?
- Does the reporting software provide logs of any reports that failed to run?
- Can the reporting software provide logs of who logged onto the system when and for how long and what their activities were?
- Does the reporting software provide audit trails of how long a particular report ran for?
- Does the reporting software provide an audit log of any changes to reports?
- Does the reporting software provide an audit log of any changes to users, datasources, migrations, updates to license keys, template changes, etc.?
- Is there a way to see if a report has been run by any user for a given number of days (possibly parameterized start-date)?

Licensing Considerations

Most Report software vendors will either implement a NAMED user or CONCURRENT USER licensing arrangement. Some license by the number of servers in the reporting enterprise and some license by the number of datasources connected to.

Ensure that you understand the implications of current and possible future reporting user growth and where this will take your company in terms of licensing cost.

Also ensure that you understand the cost of future upgrades. Some reporting software vendors will provide upgrades for the same number of users / servers / CPU's at no additional cost if their annual maintenance, support & upgrade options are purchased, some companies may provide discounts for existing customers to upgrade to new releases as they emerge.

Ensure that you understand fully the implications of your licensing, upgrades to newer versions and upgrading the number of users / seats that are required as the organization grows.