



WHITE PAPER

INTEGRATED PERFORMANCE
MANAGEMENT





Introduction

Introduction

Most of us have a car. That car will consist of a number of components that were designed to work together. From the engine through to the chassis and the controls by which the vehicle is directed, although created separately, they are implemented so that they operate as a single unit.

It's true that in the past, cars could be bought as a kit, (as some enthusiasts still do today), with the buyer selecting the components they want. This approach requires knowledge on what works with each other and how to 'integrate' them, but in general the result is never going to be as good as buying a complete packaged designed for the task.

It's also true that in the past, cars were not that versatile and so you had to choose your model carefully and accept that there would be compromises. For example, speed meant that the cabin interior would be slim without much luggage space; fuel economy would mean a small vehicle without any luxurious extras. But today that isn't the case. Today you can buy a fuel efficient electric car with the performance of a Porsche, or a luxury car that can climb mountains and double up as a pickup van.

What has made the difference is that manufacturers have designed the components and combined them with an end view in mind. Tesla started out with the aim of building an electric sports car that would perform as well as a supercar. Land Rover designed the Range Rover as a luxurious, off road vehicle that can carry a huge amount of cargo.

Few people would ever have been able to design and built these cars on their own, and those that could will have to invest enormous amounts of time and effort, far more than would be required for the 'ready-made' model.

And so it is with today's performance management applications - or is it? It seems that despite the marketing hype, only 50% of organisations have bought into these systems. And the one cry from users who have invested is one for 'better integration'. This is echoed by the major software vendors who also promise 'better integration' in future releases.

In this short paper we will explore reasons for this state of affairs and offer ways in which an integrated software solution should be evaluated.

Content:

1.	Understanding performance management applications.....	2
2.	Components of an Integrated Performance Management Application.....	3
3.	Applications Vs. BI tools and spreadsheets.....	7
4.	Requirements of integrated applications.....	8
5.	Evaluating potential solutions.....	9
6.	Summary.....	12
7.	About.....	12





Understanding PM applications

1. Understanding performance management applications

Performance management applications are those that help an organisation to achieve its objective by supporting the planning and monitoring of things under management control, in a business environment that is both unknowable and uncontrollable.

All organisations have a purpose that is often defined by long-term objectives. To achieve this desired future state, the only things that can be managed are:

- The way in which the organisation's business processes are configured (ie how products are made, sold, delivered and supported).
- The resources that are allocated to those business processes (e.g. money, people and assets).
- The workload that is involved in carrying out those processes, which can be a function of the resources applied.
- The outputs that are generated throughout business processes (e.g. the volume and quality of product produced), which again can be a function of how business processes are configured, resources assigned and workload conducted.

These four areas are set by anticipating target customer requirements and what competitors and other agents (e.g. Legislation, social media, ...) may do that could affect what the organisation hopes to achieve in the long-term.

Gartner places performance management applications into the 5 separate categories of Financial Consolidation and Close Management; Financial and Management Reporting and Disclosure; Budgeting; Strategic Planning and Forecasting; and Profitability Modeling and Optimization.

But as we have discussed in our paper 'What Makes a Strategy Execution System', the way in which software vendors develop and market solutions to fit with this categorisation is confusing and not helpful for a number of reasons:

- **Functionality overlap.** Each application area contains functionality that is similar if not the same. E.g. The ability to define business dimensions and members; that load data from external files, that support manual data entry, that create reports and end user analyses.
- **Problem definition.** The applications outlined lack detail on the business problem to be solved, particularly in relation as to how they can improve business processes or make them more competitive. As a result the end user can not be sure that the software being offered will work for their situation.
- **Multiple products.** The larger vendors offer multiple products, often with different architectures and set up procedures. This makes the choice difficult - which products have what capabilities? What happens when an organisation outgrows the capabilities of the 'lesser' product?
- **Disintegrated product suites.** Because performance management requires multiple product categories, vendors offer multiple products as a 'suite'. The issue is then how well these applications work with each other, and how they can be configured so they operate as a single solution.

Because of the above, potential customers whose problems span multiple categories, will find it hard to evaluate cross-product solutions. And those products that are selected are destined to have duplicate functionality, multiple set-up procedures and databases; and will be more expensive and complex to maintain than a purpose designed single solution.

However, more and more vendors are placing integration at the heart of its development efforts. They recognise that customers want a single solution, that covers all aspects of performance, and that is fully integrated. So...

- How do organisations judge an integrated application?
- What things should they look out for?
- What is actually meant by 'integration'?





Integrated PM application

2. Components of an Integrated Performance Management Application

In our experience, performance management applications can only be judged by the components they contain and how well those components 'fit together' to support their purpose. No matter what the application is to support – budgeting, forecasting, management reporting – all will consist (or should consist) of the following 11 integrated components.

2.1. Multi-user database

At the heart of every application is a database that is typically used to hold both the setup details (e.g. business structures, measures, how results are calculated) and the data itself. Data should be accessible by multiple people and allow information to be easily manipulated and reported.

To accommodate this there are three types of database that can be used:

- Relational – this is where data is stored as tables of information, similar to what you would expect to see in an Access database.
- Multi-dimensional – this is where data is stored in a format that makes it easy for analysis. The 'look and feel' is similar in concept to pivot tables in Excel (although in Excel the data can be stored in both relational and multidimensional formats)
- Unstructured – this is where data doesn't fit into either of the above categories or includes non-textual data such as pictures and videos. This category is becoming more popular when it comes to analyzing trends in social media sites such as twitter and facebook.

Applications may use one or more of the above formats. In our paper on strategy execution systems we made the case that all need to be accommodated. How the software vendor achieves this is down to the way in which the product is architected.

2.2. Security

Security within a multi-user database can be split into the following areas:

- Data access. This determines what data can be seen or manipulated by which people. Typically data that is outside of a person's responsibility will be hidden, while general reports may be accessible to everyone, but without the underlying detail.
- Role. A person's role should determine what they can do. For example, an administrator is allowed to change an application (or parts of it), while a manager is allowed to submit budget and forecast data, as well as analyse (but not change) results. This role can change depending on the process being supported.
- Time/process-dependent. This area recognises that data and role security is dependent on time. For example, a budget holder may be able to submit a budget, but once done then they are locked out from making changes unless authorised to do so. Similarly, year end results may be 'invisible' to everyone except key finance staff, until they have been announced at which point 'normal' data access security is re-established.

Some systems go further and include escalation paths. i.e. the person who takes over the role should the original person not meet the requirements. For example, if a budget holder does not complete a submission in time, then their manager can be automatically informed and inherit that budget holder's responsibilities.

2.3. Model builder

All applications will have business structures that allow the organisation to be modelled. Some of these structures will be hierarchy based e.g. Department, Product, Channel and so on, while others will have 'flat' or non-hierarchy structures such as Version.

These structures determine what data is held and how it is consolidated. Quite often these structures will already exist in other systems such as the ERP and so a way of importing these is helpful as it reduces potential 'copying' errors. It also makes it easy to keep the current structures up to date and in synch with any operational systems.

As time passes, or to evaluate alternate business strategies, these structures will change. The system will need to cope by providing a simple graphical interface that allows administrators to re-organise the model, while still keeping historical structures for analysis purposes.

For this reason, data should not be attached to a structure and should allow both forecast and historic data to be viewed or reported by multiple structure versions.





Integrated PM application

2.4. Business rules

As well as having consolidation paths as defined by the model builder, there will also be the need to specify more complex 'rules' that determine how data is processed. Unlike a spreadsheet these rules should:

- Be named based. i.e. they do not use cell references but instead use the business name allocated. E.g. Revenue = Volume x Price.
- Contain functions for calculating things like ratios, percentages, and summations.
- Able to access data in any business dimension. Most rules that affect measures typically apply to every dimension. In the above example for calculating revenue, the rule will be applied to every department, product, version – or whatever dimensions exist. However, some calculations such as those that allocate the cost of the finance department to all other departments will need the ability to target specific values in different dimensions of the model.
- Provide control on how the calculation operates at different levels within a structure. Again, in the above revenue example this calculation should take place at the lowest level of the structure (also known as 'leaf nodes') with the result being accumulated through the organisation or product structure. However the rule for calculating average price (= revenue / volume) must take place at every level within a structure i.e. it cannot sum up individual average prices as this will give the wrong answer.

Some systems rely on the administrator using the rules supplied by the database being used eg MDX. These 'languages' can be complex to learn that then give rise to errors due to misunderstandings on syntax.

2.5. Data acquisition

This component is used to load (or export) data that exists in operational systems. Often referred to as ETL (short for Extract, Transform and Load), it should have the following features:

- **Choice of source data format.** Data exists in many different systems and formats and so ideally the component should be able to read these data sources directly. However this is not always practical so as a minimum the system should be able to 'read' a CSV (comma separated file) that most systems are able to produce.
- **Transform.** This relates to the ability to choose the placement of data within a data source. For example whether time and other dimension members are stored as records or as columns.
- **Summarise.** Quite often the data stored in a transactional system is at a far lower level of detail than is required for the planning application. In this case the data can be summarised by the data acquisition component which then just loads the totals.
- **Match.** When data is collected from multiple systems, the codes used to store the data internally is often different. This is particularly true when collecting data from multiple general ledgers that have different charts of accounts. To be able to load the data correctly there needs to be a map from the internal code of the source system to the corresponding dimension members. Quite often multiple sets of 'maps' are required to load data as each source system may use different codes.
- **Load.** Once the above has been set, the data can be loaded. The system should report when data doesn't match or the format is different from what was expected. In these cases there should be an option to reject the complete load or to load what can be read. Also, there should be options on whether the data to be loaded is to replace any data that may be present in the planning system or if it should be accumulated.

As well as loading data electronically, two other features are required for any performance management application:

- **Manual data entry.** This presents users with a data capture screen where they can enter data. This should support single or multiple periods such as 12 months when entering a budget or the next 6 months when entering a rolling forecast. Some systems also provide 'short-cuts' for entering data such as phasing and spreading a total amount over selected rows/columns.
- **Audit trails.** When data is loaded, whether electronically or manually, there should be a full audit trail of what was loaded along with the source data file, any maps used, details of the user loading the data, the date and time. This feature ensures that every number loaded into the system can be traced back to its source and author.





Integrated PM application

2.6. Financial / application 'intelligence'

This component relates to specific data processing capabilities that are required by the application, which in general is typically hard or impossible to code using standard business rules. This includes the following features:

- **Currency conversion.** If the system collects data in multiple currencies, then it will need to be converted to a base currency (or multiple base currencies) for consolidation and reporting purposes. Currency conversion can be complex where balance sheet data is involved as the system will need to cope with multiple rate types (e.g. opening, closing, average, historic, ...), which in turn can lead to exchange gain/losses.
- **Debit/Credit understanding.** When manipulating financial data, especially if it has come from a general ledger, then the system will need to 'know' what data is deemed as a 'credit' (Cr.) or a 'debit' (Dr.). The reason is that any rules such as variance analyses or summations involving multiple Dr/Cr items, will be affected – for example a Dr added to a Cr, is actually a subtraction.
- **Account type.** As with Dr and Cr items, some financial accounts can be summed over time (e.g. revenue and expenses), some should not be summed (e.g. stock and cash balances), while others should not be translated (e.g. ratios). This 'understanding' can greatly simplify the rules of the application.
- **Time aggregations.** Data will often be collected on a weekly or monthly basis. For reporting purposes this will need to be aggregated into quarters or as 'year-to-date' summaries. For retail, the time aggregation may be according to fixed dates e.g. the summer season, and it's not uncommon for them to have 13 periods in a year. Although defining time may be straight-forward, what is difficult is ensuring that financial data is converted and aggregated correctly as has been outlined above.

All of the above could be 'hard-coded' using standard business rules (as happens when using a spreadsheet, but the maintenance effort will be huge and can easily lead to integrity errors. For this reason, planning and reporting systems should have embedded 'financial intelligence').

2.7. Data manipulation

Once a business model has been defined, users will want to interact with the data to produce various analyses. This 'manipulation' is more than converting numbers to a base currency and 'adding-up' them up. The manipulation will depend on the analyses required and should be provided by the system as an 'on-demand' function. Examples include the following:

- **Copy data.** As well as loading data, it's sometimes useful to be able to copy data that already exists within the system. For example, to provide a starting point for a budget, last years 'actual' results could be copied into next years budget version. Or to create a second pass of a budget, the first 'pass' could be copied and stored separately so that the second pass can then be compared against it.
- **Archive / Delete data.** At some point, data will need to be deleted from the application. This will be done via an administrator command that could place the deleted data into a reloadable archive file should it be required at a later time.
- **Allocations.** Sometimes it's easier to have allocations performed not as a business rule, but 'on-demand'. This can greatly simplify the setup of an application.
- **Intercompany matching and elimination.** This relates to applications where the consolidated results must take out the affects of intercompany trading. This feature allows intercompany transactions to be compared and when 'matched' to be automatically eliminated on consolidation.
- **Journal adjustments.** As with intercompany trading, organisations that have multiple ownerships or where there is a need to adjust data to International Accounting Rules, a specific journal adjustment module can greatly simplify the business rules required.
- **Top-down planning.** Some organisations like to set high-level targets at a company level and have the system allocate the values to departments based on some set variable, such as last year's actual results.
- **Goal Seek.** More sophisticated systems enable managers to set a target at a summary level and then allow the system to adjust low-level values until that target is met.
- **Scenario planning.** This feature allows the model to run different simulations, whose results can then be compared. For example, running three sets of drivers that represent 'good', 'bad', and 'expected' levels of performance. More sophisticated systems allow the models structures to be changed between scenarios, but still allow the results to be compared.
- **Initiative planning.** This capability allows users to propose a range of projects that could be used to improve performance. The data collected includes proposed start dates, people responsible, etc. The system then allows these projects to be assessed singly and in combinations as part of a scenario. The idea is to allow the best combination to be selected which then become 'live' projects whose implementation can be tracked.

There are many functions that could be listed here. The important point is to note which data manipulation capabilities you really need and that the system chosen is able to support them. Also, make sure the system being used is able to keep an audit trail of all data and how it is transformed from entered data into its final reported state.





Integrated PM application

2.8. Presentation tools

Critical for any system is its ability to produce meaning information in a way that users can understand and take action if required. Information to be displayed can be presented as:

- **Static reports** where the user cannot change the report orientation although the security system may filter out any data the user is not allowed to see.
- **Dynamic reports** that allows the user to change report layouts (rotate); select lower levels of detail making up a report (drill-down); go back to the underlying source data (drill-thru).
- **Ad-hoc analyses** where the user has complete control over content, layout and formatting. When a report the user likes has been created it can then be saved (and shared with others) as a personal 'view' that can be recalled at any time.
- **Sorted lists** where data is printed in a selected rank order – e.g. the top 10 clients by revenue
- **Exceptions.** Only the data that meets the exception criteria is printed e.g. list the departments whose actual expense performance is over budget.
- **Alerts.** This is where users are notified (usually by email) only if data is outside of the defined criteria
- **Charts and other graphics.** As well as the 'normal' bar and line charts, advanced graphics can include support for waterfall charts and strategy maps.
- **Report-books** that consist of multiple reports and formats whose content is controlled by a single selection – e.g. the user department.
- **Period referenced.** With this type of report the system 'knows' about the current period and can use this information to display the appropriate data each month but without someone having to modify the report. Period referencing is also used when displaying or entering rolling forecasts, e.g. display the current month actual data and the next 6 months of forecast data.
- **Notes and comments.** This provides textual data that the user may have entered as a comment or a note when submitting data.
- **Structural information.** This is where the setup details are printed as part of the report. For example, a list of initiatives along with the owner name and the departments affected

The above is typical of what most users would expect and a good reporting system should be able to 'mix' any combination of the above

2.9. Data / User control

This area of capability is concerned with how users are directed through a particular management process, and the order of any data flows. In any management process there is an order of events. For example, the budget typically starts off with senior management setting high level targets. Next, sales will attempt to meet the revenue goals by working with marketing to budget the appropriate level of resources to attract new customers. Once this has been done, production and customer support can plan their level of resources to meet the anticipated demand and so on. Similarly when it comes to reporting results, data will first need to be extracted from the general ledger and loaded. Departmental managers can now review actual and make adjustments before finalising the reports for senior management to approve and then circulate.

In both cases, users need to be told what is required of them and by when. For those waiting for others it's important that they wait until previous activities have been completed and any data is 'locked' so that it can't be changed.

To ensure that these processes operate in an efficient and effective order, the system must provide some form of workflow that automatically provides individual's with their own personal 'To do' list.

2.10. Collaboration

Collaboration can come in many forms. First by involving people in a process such as budgeting where participants are informed as to what the goal is, why the company has set them and their role in ensuring they are achieved. This involvement can be triggered through email and in providing relevant documentation throughout the process.

Another form can be in how users respond to their involvement. Most planning and reporting systems deal in numbers, although the business area they represent such as department, version, time and measure defines those numbers. But figures do not tell the whole story and so systems need to allow a range of textual data in the form of notes, comments, and backup documents including pictures and video to be attached to requests and submissions. These should be available whenever an associated number is reported, so that users and managers alike get the complete picture.





Applications vs BI tools & Excel

2.11. User access methods

This final component relates to the way in which users access and use the application. Today, most systems are web-based, i.e. the only software that is required by a user is a standard web browser. But increasingly users want access via a range of personal devices such as tablet and smart phones. These users are more comfortable with using 'apps' that have been developed specifically for the medium.

Unlike a spreadsheet that is fixed to a machine and requires the right software to use it, performance management applications need to become more like interactive web sites that are independent of machine, location and application software.

3. Applications Vs. BI tools and spreadsheets

Having outlined the components of a performance management application, it's worth considering how these capabilities compare to what's typically available from Business Intelligence (BI) tools and spreadsheets. There are many BI tools available today with many claiming to support enterprise wide planning and reporting. In fact most performance management applications are built on top of a BI platform. However the real difference between the two is the BI tool's lack of pre-built functionality to cope with 'financial intelligence', 'data manipulation' and 'data/user control'.

Applications Vs. BI Tools	
<ul style="list-style-type: none">• Multi-user database (data, set-up): Architectures: Relational, Multi-dimensional, Hybrid, Big Data• Security: Data, Role, time-dependent• Model builder: Hierarchy based: Department, Product, Channel, Non-hierarchy (flat): Version• Business rules: Named based, Ratios, percentages, summations,• Data acquisition: Transform, summarise, match and load• Collaboration: Notes, comments, alert, ...	<ul style="list-style-type: none">• Financial 'intelligence': E.g. Currency, Dr/Cr and account-type understanding, Time: Aggregation of days, weeks, ...• Data Manipulation: E.g. Copy, Archive, Intercompany, journal adjustments, top-down, goal-seek, scenario planning, initiative planning• Presentation tools: Static / dynamic reports with charts, grids Drill-down, drill-through, slice/dice/rotate Period reference, notes, comments• Data / User control: Workflow: order-of-events / activities, approve• User access methods: Web, mobile, spreadsheet, ...

Comparison of a Performance Management application capabilities with a typical BI tool.
Items not supported 'out of the box' are shown in red.

Most of these capabilities can be built using the tools supplied but it means that these will need to be maintained by the administrator and it will greatly increase the complexity of the business rules. For example, controlling how different types of account are accumulated within the time dimension or how to convert them from a local to a base currency.

Similarly, controlling what a user can do during a management process, and how submissions are approved and then locked will require a lot of bespoke 'coding'.

Spreadsheets fare much worse than a BI tool as can be seen from the following graphic. Most of the functionality required for planning and monitoring performance is missing.





Applications vs BI tools & Excel

Applications Vs. Spreadsheets

- **Multi-user database (data, set-up):**
Architectures: Relational, Multi-dimensional, Hybrid, Big-Data
- **Security:**
Data, Role, time-dependent
- **Model builder:**
Hierarchy-based: Department, Product, Channel, Non-hierarchy (flat): Version
- **Business rules:**
Named-based, Ratios, percentages, summations,
- **Data acquisition:**
Transform, summarise, match and load
- **Collaboration:**
Notes, comments, attachments, alerts, ...
- **Financial 'intelligence':**
E.g. Currency, Dr/Cr and account-type understanding,
Time: Aggregation of days, weeks, ...
- **Data Manipulation:**
E.g. Copy, Archive, Intercompany, journal adjustments, top-down, goal seek, scenario planning, initiative planning
- **Presentation tools:**
Static / dynamic reports with charts, grids
Drill-down, drill-through, slice/dice/rotate
Period-reference
- **Data / User control:**
Workflow: order of events / activities, approve
- **User access methods:**
Web, mobile, spreadsheet, ...

Comparison of a Performance Management application capabilities with a spreadsheet.
Items not supported 'out of the box' are shown in red.

With a spreadsheet there is no central database and can only be used by one person at a time. There is no model builder, no financial intelligence, no user control and so on. Although some of these restrictions (not the multi-user aspect) can be overcome with the use of formula and macros, the end result falls far short of what is required to manage performance. As those using spreadsheets are aware, maintaining systems built from them can soon become an unmanageable nightmare where integrity and planning capabilities are greatly compromised.

For the above reasons, organisations using purpose built applications are less likely to run into problems. But that doesn't mean to say they provide the perfect answer. A key issue relating to the usability of a performance management application is the functionality provided by the components mentioned in section 3 and, just as important, the level of integration between those components.

4. Requirements of integrated applications

There are many software solutions that claim to be integrated and yet administrators find that linking some of the components and getting them to act as a single solution is difficult. For example, the component used to create a dashboard or a report, may have a different architecture to that used for setting up a budget data capture screen. Because of this, data has to be physically moved between the budget and reporting application, and more important, when a change to a structure or measure is required, it needs to happen in two places. This not only doubles the workload but can easily be 'missed' (or worse performed wrongly in one of those applications), leading to undetected integrity issues.

A true integrated application does not suffer from this level of 'dis-integration'. Instead the 11 components mentioned above, is built on an architecture that includes the following:

Single interface. No matter whether it's setting up a budget or creating a report, all capabilities are maintained through a single interface. By this we mean that the 'look and feel' is the same and does not appear to be two different products. The benefits of a single interface is that learning time is reduced for administrators; it's easier to know where to look when modifying an application; and it makes security of roles and user access easier to maintain.





Requirements of Integration

Central definition of measures and structures that are used by all components. These definitions are then accessed when creating a forecast template or setting up a report. Any change made here automatically updates any component where those measures and structures are used. The benefit of having a central definition is consistency over what a measure means or what a structure represents. As with the single interface, there is only one place to define a measure, modify a structure or create new data versions. In effect there is a controlled, central data dictionary for all aspects of monitoring performance.

Single data store. All data for planning and monitoring performance is held once. To do this will require support for multiple data formats such as relational tables and multidimensional models. How data is stored is irrelevant however, end users will want information presented in the most relevant way to support the analysis being requested - e.g a top 10 list of the largest customers, the volume of products bought by customers over time, etc. The benefits of a single data store is that all results are based on the same version of data no matter how it is reported.

Single 'To do' list. For users, irrespective of the process, there should only be one place for them to look as to what they need to do next. Menu's do not perform this task as they tend to be common to many people, and have no sense of task order or importance. It's also hard or impossible for administrators to gauge where the bottle-necks are in the process and which users/activities are holding up others. The only way is through a true workflow that operates across the whole of the application and all components that are in use. This should generate dynamic 'To do' lists for every user that automatically change as management processes advance. It should also provide administrators with a moving timeline of all activities within all 'live' management processes.

Single security system. There should just be one place through which users are defined and their role/access requirements for any management process. It should be possible to view everything:

- From a user viewpoint i.e. their role and access for every process.
- From a management process viewpoint i.e. everyone involved in a particular process such as budgeting.

Common capabilities. The capabilities of all components are available to all other components. For example there should be one load data module, one reporting module, one data manipulation capability, etc., that respects all business dimensions / members and whose functionality can be used to access data in any management process.

5. Evaluating potential solutions

5.1. Define the scope of the evaluation

Most software vendor demonstrations are akin to a beauty pageant. They are hosted by an 'actor' that plays out various scenarios all carefully scripted to show the product in the best possible light. As a consequence it's easy to imagine the product being shown is both 'easy to use' and fully integrated. But how can you be sure? Underneath the cosmetics may lurk another story. For an evaluation to provide value and that will lead to the right selection, it must be conducted in a logical fashion with clear objectives and criteria.

Solutions are very rarely required for just one person – they will impact others, so you need to make sure that the solution will meet all the needs and will not have an adverse affect on anyone else.

Before any vendor is selected or any products reviewed, the following items must be agreed:

- What problem(s) are we trying to solve?
- What causes those problems?
- What must we have in the first release?
- What is essential for the future?

Once the organisation's requirements and resources are fully understood, it is time to research vendors.





Evaluating Potential Solutions

5.2. Research suitable vendors

There are many vendors offering performance management solutions. A quick search on the Internet will reveal those who are prepared to spend on search engine adverts, while the lower pages will cover the rest.

In looking at individual websites determine how much experience each vendor has and their view on how an organisation should manage performance. After all you want a solution and not just a software product. Also, find out what product is being offered. Many of the bigger vendors have multiple products that were bought through acquisition. Sometimes this collection is offered as a suite but beware as this could mean multiple setups that could suffer from integration problems, and be more expensive than a single solution. It could also mean that one of the products on offer is due to be superseded by the other.

Ask to see the product road map. Find out the vision of the vendor for the product being proposed and its anticipated life span. Get them to outline the customer profile they have for how the product is developed. This information can help when assessing whether the product fits in with your current and future requirements.

5.3. Conduct detailed evaluation

Beware of the 'canned' demonstration. Vendors will typically only show you what they want you to see. A better way is to inform the vendor about your basic need and ask for a demonstration on how they would meet those specific needs. A good vendor will respect your time and get straight to the point saving both your and their time.

Some vendor solutions may involve multiple products that are hidden by a slick web frontend. To reveal whether this is the case perform the following checks on a realistic, full-sized application in a hands-on session:

Maintenance:

- Ask to see how an administrator sets up a data entry screen. Go through the process and see how this gets to the end user.
- Add a new measure and see what's involved in getting it into an existing data entry screen and then onto the reports. Check this by entering a number and seeing it work its way into the reports
- Enter a number during a forecast process and then check that the same number can be compared to actual and budgets in an existing report. If you don't see the number, the system you are looking at could have multiple databases and not be the single system you require.
- Add a new unit and perform a reorganization. Verify that the changes show up in the data entry system, the reports and the analyses. If this cannot be done, then the application either isn't integrated or it's too complicated for the vendor to show you.
- Add a new user (e.g. a budget holder) and a new department. See what is involved in getting them to be able to access the system to enter a budget, have it approved and appear in the consolidated result.
- Make sure the system will allow users to budget, track actual, and collect forecasts. If you need rolling forecasts, test how the system accommodates this.

Measure flexibility:

- Make sure the measures being collected are realistic: Have some accounts collected in more detail by a different business dimension, such as by product and customer.
- See how business rules are developed that can change by

version. For example, set up the rule 'Revenue = volume * price' for the budget and forecast version, and 'Price = revenue / volume' for the actual version.

- See how measures are converted from local currency to a base currency. What happens to exchange gain/losses?
- Check how measures can be placed in a cause and effect structure as required by the Balanced Scorecard.
- Check how measures can be grouped into resource, workload, and outcome types and then assigned to specific departments.
- See how high level objectives can be related to business processes and activities.

Date handling:

- See how data is entered at different levels – e.g. by day, week, month, etc.
- See how data can be summarized e.g. into seasons, quarters, etc. Create a new time summary during the presentation and see how this can be used to report results. Does the new summary respect Balance sheet accounts and the way ratios are calculated?
- See how data at a detailed level (eg sales by product/customer) is linked to summary data (e.g. total revenue account). Change the values of some of the detail and ask to see the effect in the summary P&L.
- Is there an audit of how results were generated? i.e. when and who entered or modified the underlying data.
- See how structure changes are dated. Can old data be re-consolidated under new structures while still preserving old results under old structures?





Evaluating Potential Solutions

Planning flexibility:

- Driver-based budgeting: See how budgets can be generated by entering a few driver values. Then see how the system can calculate those driver values from actual results
- Initiative Planning: Ask to see how budgeting can be separated into 'business as usual' and 'initiatives' or projects. See how initiatives are proposed and selected.
- Scenario planning: Ask to see how different combinations of initiatives or the results of using different driver values, can be compared in side-by-side scenarios.
- Generate a new pass of the budget. Look at the work needed to keep an old pass and to generate a new one. Make sure this control over versions is rolled out to users.

Workflow:

- Investigate what capabilities are provided for workflow. Does it allow individual parts of the planning process be triggered by events and exceptions, as well as a date?
- Check the facilities provided to track and approve budget and forecast submissions.

- Check out what happens if someone misses a deadline. How is this escalated?

Reporting:

It should be noted that some applications do not come with reporting capabilities and instead will rely on a third-party product or the user connecting to the database via Excel. If this is the case then any evaluation should check out what reports can be produced via those methods and the maintenance effort required as the system changes. Things to watch out for in an evaluation are:

- How 'intelligent' is the report selection. E.g. can the system bring back the assigned measures to individual departments without having to select them individually?
- How flexible is the report layout. E.g. is there complete control over column and rows?
- What charts styles are supported?
- Can data be sorted on the grid?
- How are exceptions in the data handled?
- Can the data be securely accessed from within a spreadsheet?

5.4. Establish cost of ownership

Cost of ownership is the cost to the company of the complete solution including its initial purchase, implementation, on-going running and any other hardware/software costs that were required to make the system work. This of course should be less than the benefits that are to be accrued. Things to watch out for are:

- **Initial software purchase price.** This is often set by the number of processors used by the application server. Find out what happens if the server is upgraded as you could be in for an additional charge when that happens. It is also usually dependent on the number of users and the role they will play. Software vendors will want you to buy more than required 'in case you need them later'. Find out what the cost is for upgrading the number of users at a later time. This could be more expensive than the per user charge you'll pay initially.
- **On-going annual software maintenance price.** This is often set as a percentage of the list price, which of course can change without notice. As with the initial software purchase, it will often be dependent on the server configuration.
- **Additional software requirements.** For example, some solutions rely on the customer providing the underlying database technology. The chosen application could use some of the 'seats' covered by the underlying technology purchase, which may then incur additional fees.
- **Hardware costs.** Does the application need its own hardware in order to run efficiently? New hardware could invoke additional software license fees.
- **User access costs.** How will the users gain access and what software do they need locally on their own devices.
- **Initial implementation cost.** How much will it cost to get the initial application working and rolled out to all users? This will be broken down into training and implementation consultancy. The latter tends to be quoted as a day rate but with no guarantee on how much time will be required. Try to get this part 'fixed' so you'll know exactly how much to budget. In the past the general 'rule of thumb' was that implementation costs are around 50% of the software purchase price.
- **Ongoing support.** Helpline support is typically included with software maintenance. But if your question is related to better use of the application, you may be asked to pay consultancy rates for this advice. Also check what times the helpline is available and the response time you can expect. Sometimes there is an additional charge for quicker response times.

The above costs should be gathered for the life of the project or at least the next 5 years. This is because some initial costs may be low but you'll pay for it in subsequent years as the system is upgraded or the number of users grows.

5.5. Vendor Proposal

The final thing to consider is what the vendor is actually proposing. Make sure that this is for a complete working solution as dictated by the specification, with the inclusion of any points uncovered in the detailed evaluation.





Summary

6. Summary

Performance management systems are at the stage transactional systems were in the 1990s. Then you would purchase separate accounts receivable, accounts payable, and stock control systems, although some vendors offered them as an 'integrated suite'. But as the needs of organisations grew for better forecasting and production planning information, new types of transactional systems appeared.

Enterprise Resource Planning (ERP) combined the traditional systems for accounting, stock and purchase ledgers and surrounded them with workflow. This did two things: first it allowed many more people to be involved who were part of the supply chain, and second it automated the different parts of the process which could now be triggered by events and exceptions. You no longer had to wait until the end of the month to review sales forecasts, order materials and schedule production. It could now happen on a continuous basis.

Integrated performance management applications should do the same thing for management. They should allow all the people involved in managing the business to work together; and for planning and monitoring activities to take place on a continuous basis. To do this they must combine all aspects of planning, budgeting, forecasting, reporting and analysis with workflow that automates who does what and when, based on events and exceptions.

Integrated systems are not about packaging up older applications. As with ERP, it requires an architecture that is designed for this combined purpose. However, very few software vendors have achieved this, although one of them - CorPeuM - is closer than most.

With CorPeuM we have built our system from the ground up with an architecture that supports continuous planning and monitoring in a single system. It has many unique features that you won't find in other products including:

- Dynamic workflow that provides tailored 'To Do' lists for all participants in a process, that are triggered by a combination of dates, events and exceptions.
- Automated escalation procedures should users not complete their tasks on time, or if done incorrectly.
- Intelligent attributes that automatically link specific measures to business departments and processes that makes it ideal for modelling business processes.
- Multi format data model that supports different levels of and types of detail that can be automatically combined by the system
- Support for multiple calendars and time aggregations.
- Central definitions for defining business dimensions, members and measures that are used throughout an application .
- Extensive data modelling capabilities that fully support initiative and scenario planning .
- Complete audit trail of all data and structural changes.
- Intelligent reporting system that can combine and report data from any process into any format and layout.

If you're involved with an organisation that is looking to get more value out of its existing management processes, or that is looking for new ways of managing performance, then our approach could well be what you are looking for.

7. About CorPeuM

CorPeuM is made up of an international team of acknowledged experts in the field of corporate performance management. Our experience in designing, using and implementing 'best practice' systems for planning, budgeting, reporting and analysis in some of the world's leading organisations, led us to design and build a new and better breed of application that is more able to cope with today's uncertain business environment. The result is the CorPeuM approach that allows organisations to focus on the execution of strategy by redefining the way they manage corporate performance.

If you would like to know more about our approach to budgeting and corporate performance management, get in contact with us via our web site or by email to info@corpeum.com

More whitepapers: <http://www.corpeum.com/white-papers/>

Join our strategy group on LinkedIn

Don't miss any of our posting by joining our LinkedIn Group. Click on the image to redirect to linkedin.

