



# Built to extend – Oracle Packaged BI Applications



## New Goals of BI

Allow self-service reporting and analytics so that business users are not totally dependent on IT

BI must be embedded in operational processes so that it becomes the driving workflow for operational decision making

BI is an application to solve targeted business problems as opposed to a collection of reports

BI projects cannot be multi-year engagements; today's business is too nimble and fluid to wait for such initiatives

## Overview

For the last decade, KPI Partners has been involved with building packaged analytical applications for a wide variety of enterprise software vendors. Throughout our business interactions, we're repeatedly asked the same question.

*"Should I build analytical applications or should I buy and extend?"*

Let's answer with an analogy. When you're moving to a new town, an important decision is whether you want to move to an existing neighborhood or choose to build a home from the ground up.

Let's say you've decided to build your own home. When you take this approach, there's a host of very specific tasks you have to fulfill to make it a reality. You need to hire an architect, clear off the land, lay the infrastructure, build the house, pave the driveway, connect with nearby services such as schools and businesses, and those are just the beginning stages.

Obviously this "custom-built" approach is very time consuming and expensive. The costs associated with building this infrastructure are all up front.

Now let's look at another approach. Most individuals don't have the time or financial resources to custom-build their living environment. They buy homes that meet 80% of their needs, and then work to customize the other 20%. Sure there might be some minor improvements to be made, but for the most part the infrastructure is in place.

Let's look at the two approaches side by side. It's obvious that the "buy and extend" approach is far less risky than the "custom-build" approach.

It's also less expensive and involves less effort since most of your needs are streamlined. Of course, one can choose to "build" from the ground up, but that approach involves much more "heavy lifting" across the board.

Deciding to "build" your home from the ground up or choosing to "buy and extend" depends on your objectives, available time and available budget.

So let's return to our question.

*"Should I build analytical applications or should I buy and extend?"*

While BI initiatives in the past were often custom, in-house development projects, trends like organization-wide BI deployments, more mature BI products, tighter integration with other business applications, and smaller budgets for custom development are all driving companies to invest in pre-built BI applications.





# Built to extend – Oracle Packaged BI Applications



## New Goals of BI

Allow self-service reporting and analytics so that business users are not totally dependent on IT

BI must be embedded in operational processes so that it becomes the driving workflow for operational decision making

BI is an application to solve targeted business problems as opposed to a collection of reports

BI projects cannot be multi-year engagements; today's business is too nimble and fluid to wait for such initiatives

Let's discuss the underlying concerns about deploying pre-built BI applications.

### **1. How can a packaged application work with my customized environment?**

Packaged BI applications are built against an un-customized source system (Siebel, Oracle EBS, PeopleSoft, SAP, etc.) that gets customized during deployment. Also, most companies do not rely on one source system to run their business. They traditionally have multiple source systems, some of which can be homegrown. So how does a packaged BI application designed to work against an un-customized source system work against a customized source system or worse, a non-standard homegrown source system?

### **2. How can a packaged application meet my custom analytics needs?**

Each organization has their unique business processes and analytical needs. How can the analytics and workflows that a business requires be provided by any packaged BI application without encountering considerable trade-offs?

These questions arise from marketing messages from the vendor, industry analysts, and the implementers as well. A true packaged BI application is really an extensible application framework as opposed to a "set-in-stone" application. The packaged BI application from Oracle provides a flexible framework along with industry leading tools that can quickly be tailored to meet any unique business or IT need within an 8-12 week deployment period.

## Extensibility of Oracle's packaged BI applications

The current set of Oracle BI applications share the same architectural foundation that KPI Partners' founders and key personnel helped shape while at Siebel Systems. Remember that Oracle inherited the BI application architecture from Siebel.

While at Siebel, we were given a dual charter.

- The packaged BI application should be able to be customized very easily.
- The packaged BI application should be able to be upgraded seamlessly as the source system changes.

We embarked on creating a packaged BI application framework that worked "out-of-the-box" against a non-customized source system, but that was tailored easily to meet any business need within 10-12 weeks.

How did we accomplish this?





# Built to extend – Oracle Packaged BI Applications



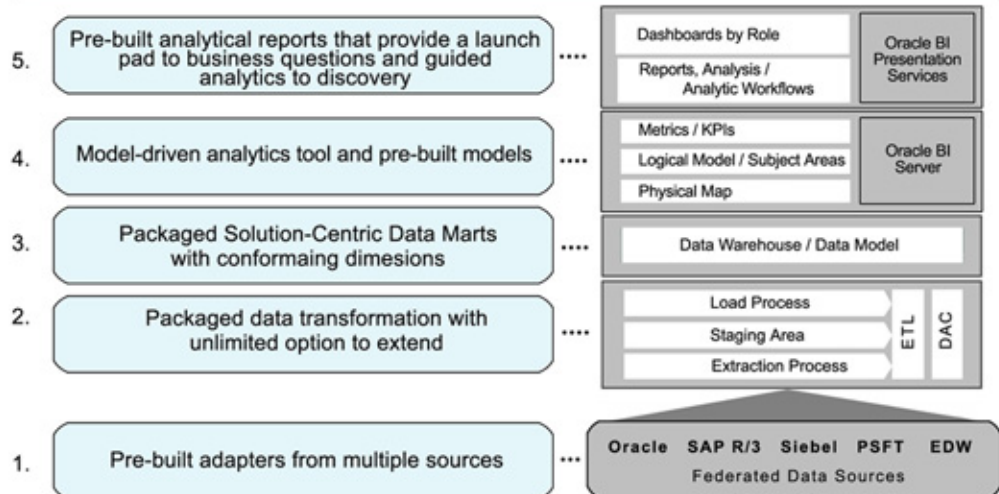
The goal of a packaged BI application is not to have a “one size fits all” approach but to provide an extensible framework that allows the organization to tailor the packaged application to its unique need without compromising on the architecture and performance.

## Extensibility of Oracle’s packaged BI applications

A BI application has essentially few basic building blocks:

1. An extraction module (often called adapters) that can get data from a source system like Oracle EBS or PeopleSoft or Siebel CRM
2. A transformation engine (often called an ETL tool) along with pre-built transformations that transform the extracted data from the source systems to the data structure of the analytical application
3. An analytical application data model designed to solve business analytics issues in a particular subject area (e.g. Receivables or Sales)
4. A business model or metadata model containing key business metrics, business hierarchies, and business processes
5. A reporting module that can deliver the data to the end user in a variety of reporting styles from interactive dashboards and scorecards to production reports, alerts and self-service reporting

### Oracle BI Application Framework Contents





# Built to extend – Oracle Packaged BI Applications



## Oracle BI Apps provide:

Best of breed data warehouse tool set

- Informatica
- OBIEE
- DAC

Pre-built domain specific data model

Pre-built extractors for all common sources and open adapters for custom source

Source-independent pre-built data transformation

Pre-built reporting metadata and pre-built dashboards

Extensible data warehouse framework for extending the pre-built apps

Now, let's discuss how a packaged BI application addresses the two key concerns of a customer, namely:

1. How can a packaged application work with my customized environment?
2. How can a packaged application meet my custom analytics needs?

A well-packaged BI application supports five critical criteria:

1. Extensible components—each component of the packaged BI application should be extensible to accommodate additional business requirements
2. Hot-replaceable components—each component should be hot-replaceable with a different component to accommodate business and IT needs
3. Easily configurable—system should be able to be tailored and deployed within 10-12 weeks
4. Leverage data warehousing and BI best practices
5. Excellent knowledge and leveraging of the source system to deliver maximum business value

All these are very lofty goals, but entirely achievable. Here are the reasons why.

### **Extensible components:**

For example, let's say you start doing a map-and-gap against the packaged HR analytical application and find out that there is a gap in a certain area. Now you'll need to create an additional data model (often called a new star), extraction routine, transformation routine, and reports.

You can create all of this within 1-2 weeks because the existing data model star already contains key dimensions such as employee, supervisor, geography, time, and others that don't have to be built from scratch. The speed of customization is fast because Oracle has already provided you with the infrastructure for each of the components and industry leading tool sets (often graphical in nature thus requiring very little programming). You only need to be cognizant about using the Oracle infrastructure and the customization guidelines so you can leverage what you already have because 80% of the building block for this "new requirement" is already there.

### **Hot-replaceable components:**

Let's say you're deploying the HR analytics applications, but your payroll data is not stored





# Built to extend – Oracle Packaged BI Applications



## Oracle BI Apps extension possibilities

- > New/or modified data model
- > New/enhanced source systems
- > New/modified data transformation logic
- > New/modified KPI and dashboards

in one of the standard ERP sources like Oracle EBS or PeopleSoft or SAP but may be in a custom application. The Oracle architecture is flexible for you to replace the existing extraction module with a custom extraction module for your source system without touching the rest of the application.

### **Easily Configurable:**

The biggest value a packaged BI application provides is the speed of deployment. In over 95% of cases, a source system is customized to add additional numerical and non-numerical attributes but the fundamentals of the source system rarely change.

Each business has their different business process but eventually they store data in the same set of table structure. A packaged BI application has pre-built maps from all the core tables and data extraction is not dependent on the operational business processes. The pre-built metadata and reports also have the templates for all standard analytical questions a business may need and are easily modifiable to match a particular company's process.

You can essentially install and deploy a packaged BI application in as little as 1-2 weeks out-of-the box, leaving you a 10-week period in an implementation cycle of 12 weeks for configuration and customization. Compare that against a custom data warehouse where you probably won't get your data model designed for a single subject area in 4-5 weeks.

### **Leverage data warehousing and BI best practices:**

When buying a packaged BI application, you're purchasing a solution that has been built according to data warehousing practices in critical areas such as performance, multi-lingual capability, multi-currency, change capture and other key features of an enterprise class data warehouse. These capabilities have been refined based on the experience of hundreds of customers and are therefore battle-tested. Contrast this with an in-house team dealing with many of these requirements for either the first time or even for the second time while at the same time trying to meet business requirements within a tight implementation schedule.

For example, one of our customers building a custom data warehouse for the first time essentially built a custom data mart for every report that business wanted. It worked in the short run but had to be abandoned very quickly.

### **Leverage the source system for maximum business value:**

Packaged BI applications contain the vendor's deep knowledge of source applications to extract maximum business value. For example, the Oracle BI apps have different change capture methodology for Oracle EBS and Siebel CRM because the engineers understand





# Built to extend – Oracle Packaged BI Applications

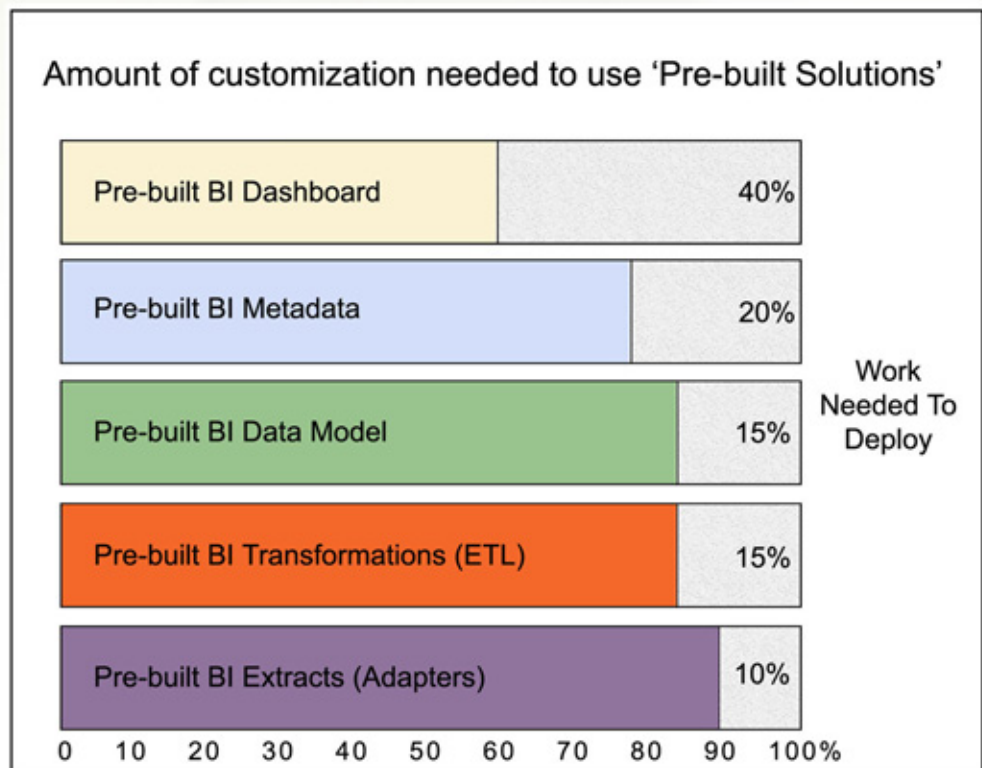


## Oracle BI Apps extension possibilities

- > New/or modified data model
- > New/enhanced source systems
- > New/modified data transformation logic
- > New/modified KPI and dashboards

the nuances of the various source systems. An in-house development team would need to understand the application model to a depth that would not be cost-effective. In addition, Oracle's BI apps have been designed to upgrade to the next version of the source application with both technology and methodology to minimize the cost of ownership for customers

The following graph demonstrates our experience in implementing BI apps and shows the level of tailoring a typical deployment goes through when BI apps are deployed against a standard supported source system.



If a company is deploying the packaged BI application on a non-standard/custom data source, the percentage of customization required for the pre-build adapters goes from 10% to 50%.





# Built to extend – Oracle Packaged BI Applications



## Build

If you have a very short term focus and are willing to live with throw away architecture and spaghetti code.

OR

If you have very large budget (>\$ 2 million per module) and 14 + months (per module) time before you can in production

## Buy and Extend

If you want quick ROI without compromising on your enterprise architecture

*Do not ignore the collective experience of the various deployments that come with a packaged solution. Oracle has over 1000 deployments of BI apps now that have constantly given feedback to Oracle engineers.*



## Build vs. Extend: Which One Provides Better ROI?

Build or buy-extend? Now we're back to the question we posed in the beginning. If your goal is to quickly get 2-3 reports up on a particular area of a source system and you don't care about long-term viability of the solution, "building" is obviously more feasible as well as less expensive. However, if you are really serious about BI initiative in your company and have heard about significant failures with data warehouse initiatives in other companies, then "buying and extending" is much less expensive in terms of dollar value as well as risk.

Here's an example. Earlier in our career, we were involved with several large companies trying to build data warehouse on MFG/Pro (QAD), SAP and PeopleSoft. The team composition included:

- **Data warehouse architects**—this individual didn't just gloss over Ralph Kimball's data warehouse design related books; they were data warehouse architects for large deployments. They understood the various types of designs--data warehouse data model, staging table, aggregate table, as well as optimal data transformation architecture, error handling, and so on. These people were very high-priced consultants and the available pool of such personnel was very small.
- **Source system experts**—these workers knew the source system and had extensive SQL background to understand the best way to extract data without compromising on the performance.
- **Data transformation designers**—these engineers were familiar with an ETL tool (we used Informatica) with all its features and shortcomings. They also understood data warehouse architecture because a simple tools developer occasionally made mistakes during development.
- **Analytics metadata and report developers**—these individuals were experts with the analytical tool (like Oracle BI Enterprise Edition)
- **Strong data warehouse DBAs**—these experts understood how a database could be tuned to handle data warehouse that is more query centric as opposed to transaction centric; a maintenance DBA of an organization usually doesn't have the skill set for tuning a data warehouse
- **Security architect**—this person was experienced in building data and report access security that met business and compliance requirements
- **Quality assurance team**—this group performed data validation
- **Load testing team**—a group that tested performance against data volume and users





# Built to extend – Oracle Packaged BI Applications



## Resource Required for Build

- a. Architect
- b. Source System Expert
- c. ETL Engineers
- d. Metadata and dashboard engineers
- e. Data warehouse DBA
- f. Security modeler
- g. Quality Assurance team
- h. Load testing team

## Software Required for Build

- a. ETL Tool
- b. Metadata and reporting tool
- c. Workflow tool
- d. Load Testing Tool

## Time Required for Build

35 weeks for each subject area like GL

It often took eight months to come up with the first application (one single subject area, such as General Ledger analytics). The production deployment took another two months. We had a very good team (most of them currently work at Oracle and SAP engineering and are responsible for Oracle BI apps and SAP Business Warehouse) but it still took a long time. Data validation, performance tuning and error management took time to develop. There was no single person who knew everything. We clearly realized that it's inconceivable that any IT organization has an appetite, capacity, or budget to have the entire team ready to build a proper BI application. All organizations have a jack-of-all-trades type of individual, but then you get a throwaway reporting application and not an enterprise BI framework.

Many industry veterans shared our frustration and the general rule was few million dollars and about nine months for deploying the first subject area (e.g. GL analytics) in an enterprise BI deployment. Companies then started proposing departmental data marts.

There was a plethora of analytical sources that didn't have shared dimensions, and this prevented businesses from analyzing cross-subject area business questions and limited the ability to truly understand their business processes such as Order to Cash. Compare this with buying a packaged BI application and customizing it.

One should also not ignore the risk of custom building an analytical application. There is no collective knowledgebase available in the market on building an analytical application. However, a packaged analytical application vendor has the benefit of collective feedback from hundreds of customers.

In summary, if you have an unlimited IT budget and more than 12 months to deploy the ideal solution, then "build" is your best bet. Otherwise the better option is to "buy and extend." The following diagram demonstrates the timesaving capabilities associated with a custom-built data warehouse versus customizing a packaged BI application.

When you do a cost comparison between different alternatives, first make sure that you are comparing apples to apples:

- Do not compare a data warehouse project with deploying a reporting solution on some views (e.g. Noetix views or DBI); this is not an enterprise analytics solution because they serve more operational reporting as opposed to analytical reporting.
- When you're trying to compare the software license cost of building a datawarehouse versus a packaged BI application, ensure that you're not ignoring the cost of acquiring critical tools like an ETL engine (Informatica, Ascential, Oracle Data Integrator) and an analytical platform (OBIEE or cognos or Business Objects)
- When you're comparing the services cost of building versus extending, don't ignore the cost you'll incur in a build scenario for all the roles we have outlined before





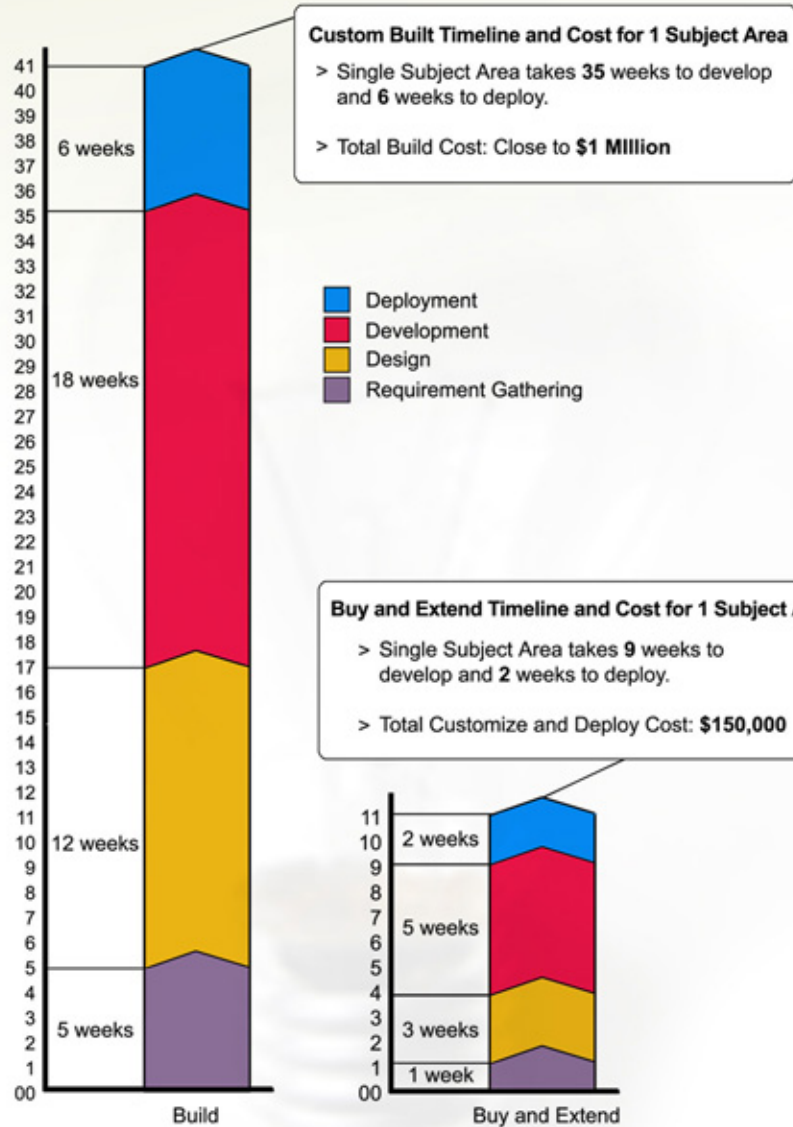


# Built to extend – Oracle Packaged BI Applications



The packaged BI application from Oracle provides a flexible framework along with industry leading tools that can quickly be tailored to meet any unique business or IT need within a 10-12 week deployment period.

Typically “build” comes to at least four times the cost of buying and extending. “Build” also takes about six times the timeline required to deploy a packaged BI apps.





# Built to extend – Oracle Packaged BI Applications



## Top-Down BI Applications Deployment

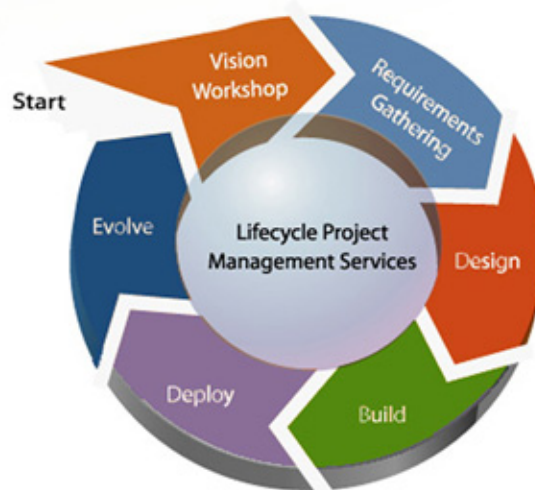
- Vision workshop
- Requirements gathering
- Design
- Build
- Deploy
- Evolve

## How Long Does It Take To Deploy ?

BI applications essentially have two methodologies for deployment. One is top-down and the other is bottom-up. In a bottom-up design, the objective is to bring in all possible fields from the source system that we can possibly report on in future. The design stems from the fear that a BI application is "set-in-stone" and the need to capture everything now.

A better and more business-centric approach is to deploy using top-down methodology. In this methodology, you identify the top business questions or drivers for your company and then tailor the application to meet those immediate needs. This is an iterative approach. You address your immediate business concerns and as your business users become more BI savvy, you iteratively extend the application by bringing in additional fields or as simple as creating new dashboards. A top-down approach gives you a quick "return-on-investment" without compromising on your enterprise architecture. This is called "build for today, design for tomorrow."

KPI Partners' BI Success Methodology can accommodate either a top-down or a bottom-up approach.



The following is a description of the various phases:

**Vision Workshop**—identify the success criteria of the project, meet the stakeholders, define roles and responsibilities, and share the timeline.

**Requirements**—install the application in a development environment and configure it for initial load; run the initial load of the BI application from the source system so that the out-of-the box dashboards show up; then identify the pressing business questions for the company and sketch out the reports, dashboards and the workflows using our proprietary templates





# Built to extend – Oracle Packaged BI Applications



## Top-Down BI Applications Deployment

- Vision workshop
- Requirements gathering
- Design
- Build
- Deploy
- Evolve

Build takes four times longer and costs at least five times more than buy and extend

Many first time build efforts result in failures. An enterprise BI application allows companies to meet departmental objectives without compromising on enterprise architecture



**Design**—perform a map and gap between the requirements and the out-of-the box application and then design extensions to the adapter, ETL maps, data model as well as the metadata and reports

**Build**—take the design documents and extend the application using the Oracle recommended methodology so customizations do not create problems during the future upgrade process; perform unit and system testing along with performance testing at this phase.

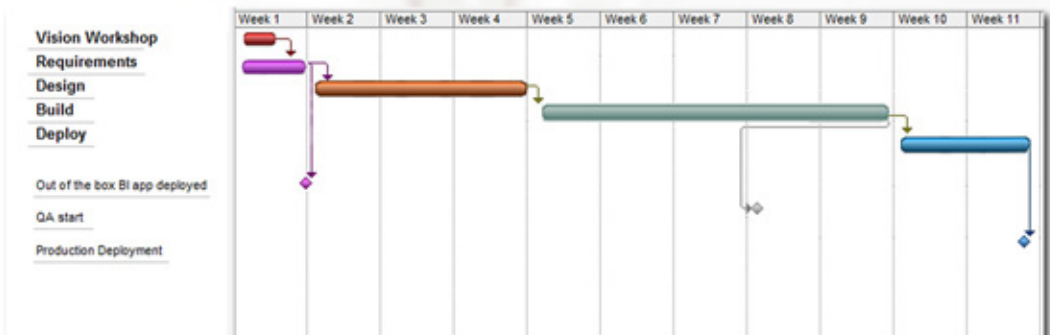
**Deploy**—perform user acceptance testing and then deploy the application on the production environment.

**Evolve**—train the internal IT team to maintain the application as well as train the business users with self-service reporting; conduct workshops to plan for the future phases of deployment

The team composition is as follows:

- Project Architect
- ETL consultant
- Metadata and report consultant
- Source System Techno functional analyst (part time)
- DBA experienced with data warehouse (part-time)
- QA (part time)

A typical phase for a top-down deployment using KPI methodology is around 12 weeks.



Choosing to “build” is significantly more time-consuming, much more expensive, and infinitely more risky than the “buy and extend” option. Packaged BI solutions provide so much out-of-the-box functionality that the cost to implement is a fraction of the price of using best-in-breed tools to build a solution.





# Built to extend – Oracle Packaged BI Applications



## HIGHLIGHTS

Oracle's most experience BI applications partner

Founders were responsible for architecting the current Oracle BI apps architecture

Consultants come from product management, engineering, expert services and analytics competency center of Siebel and Oracle

Strong data warehouse design background

Experts at deploying Oracle BI analytical applications (HR, Financial, Supply Chain and CRM) against Oracle EBS, PeopleSoft, Siebel CRM and custom sources

## ABOUT KPI PARTNERS

KPI Partners is Oracle's most experienced consulting partner founded based on Oracle's packaged BI applications. The founders were early pioneers in many packaged BI applications companies, including Siebel, Epiphany, PeopleSoft and Cognos. Our consultants were instrumental in driving the architecture of the packaged BI apps that Oracle sells today.

Our staff comes from product management, engineering, expert services and the analytics competency center of Siebel and Oracle. While many of other Oracle BI partners focus on reports and dashboards consulting, KPI Partners' heritage lies in packaged BI applications.

The company's employees have designed and deployed over 200 analytical applications on Siebel CRM, Oracle EBS, PeopleSoft, SAP and MFG/Pro using leading ETL tools and analytics toolset.

KPI Partners focuses on:

- Oracle BI Application deployment, customization and extension
- Custom Data warehouse development using Informatica, Oracle Warehouse Builder and Ascential on custom data sources, PeopleSoft, Oracle EBS and SAP
- Oracle BI EE tools deployment on custom data warehouse (metadata, dashboards, security)
- Migration of business intelligence structure from Business Objects, Cognos and Discoverer to Oracle BI Enterprise Edition
- Essbase deployment
- Siebel Marketing deployment

KPI Partners operates all across North America as well as in India. The company strongly believes in the theme "**Business Intelligence beyond reporting**" and is considered a thought leader in packaged BI application strategy and operational BI. The company is privately held and has approximately 45 consultants in USA and 50 consultants in India. The company was recently awarded "Rookie BI Partner of the Year" by the Oracle BI team.

For additional inquiries, contact us at [success@kpipartners.com](mailto:success@kpipartners.com)

Best regards,  
Kusal Swarnakar, Partner, KPI Partners

