MARCH 2005

Development Techniques for Creating Analytic Applications

By Wayne W. Eckerson

TDWI REPORT SERIES

tdwi
THE DATA WAREHOUSING INSTITUTE
A DECADE OF EXCELLENCE

A 101communications Publication

Research Sponsors

ADVIZOR Solutions, Inc.

arcplan, Inc.

Business Objects SA

Microsoft Corporation

MicroStrategy

ProClarity Corporation

SAP America Inc.

MARCH 2005

Development Techniques for Creating Analytic Applications

By Wayne W. Eckerson

Table of Contents

About the Author

WAYNE ECKERSON is the director of research for The Data Warehousing Institute (TDWI), the leading provider of high-quality, in-depth education and research for data warehousing and business intelligence professionals worldwide. Eckerson oversees TDWI's research services and coordinates its BI Strategies education program.

As an industry analyst, Eckerson has written and spoken on data warehousing and business intelligence subjects since 1995. He has published in-depth reports and articles on a variety of data warehousing and business intelligence topics. In addition, Eckerson has delivered presentations at industry conferences, user group meetings, and vendor seminars. He also consults with vendor and user firms.

Prior to joining TDWI in 2000, Eckerson was a senior consultant at the Patricia Seybold Group in Boston, where he started the firm's data warehousing and business intelligence practice. He can be reached at weckerson@tdwi.org.

About the TDWI Report Series

The TDWI Report Series educates technical and business professionals about emerging issues in business intelligence (BI). TDWI's in-depth reports offer objective, vendor-neutral research consisting of interviews with practitioners and industry experts and a survey of BI professionals worldwide. TDWI's in-depth reports are sponsored by vendors who collectively wish to evangelize a BI discipline or emerging technology.

Acknowledgements

TDWI would like to thank many people who contributed to this report. First, we appreciate the many users who responded to our survey, as well as those who responded to our requests for phone interviews. Second, we'd like to thank our report sponsors who reviewed outlines, survey questions, and report drafts. We would also like to recognize TDWI's account and production team: Denelle Hanlon, Marie McFarland, Jennifer Agee, Angela Richard, and Deirdre Hoffman.

About TDWI

The Data Warehousing Institute (TDWI), a division of 101communications LLC, is the premier provider of in-depth, high-quality education and training in the business intelligence and data warehousing industry. TDWI offers quarterly educational conferences, regional seminars, onsite training, professional membership and certification, leadership awards, print and online publications, and a public and private (Members-only) Web site.

Executive Summary

Analytic Applications. An analytic application is not just a set of randomly created reports in a person's inbox or "My Reports" folder. An analytic application contains business logic that steps users through a series of interactive reports so they can access, analyze, and take action to optimize the activities in a specific business domain. Although some organizations currently deploy dozens of analytic applications to hundreds of users, most organizations in our survey are still inching their way toward maturity in this space.

Customization Challenges. Building intuitive analytic applications is not easy. Organizations spend too much time customizing and extending commercial products to meet user requirements. On average, organizations customize 33 percent of the final analytic application using mostly SQL and other hand-written code. The total process takes 7.5 months on average, too much time to meet fast-changing user needs.

In practice, organizations are most likely to build analytic applications around a BI tool, but then add substantial amounts of custom code (mostly SQL) to customize and extend the application. Most organizations also customize the BI tool itself, focusing on the GUI, calculations, and navigational elements. Developers spend significant time customizing ETL mappings and data models in packaged applications. Developers, who are mainly IT staff and application programmers, make frequent changes to analytic applications, and power users are often enlisted to change the front-end environment.

Buy and Extend Using ADEs. BI vendors have recognized the need to deliver "buy and extend" capabilities. Most are starting to deliver analytic development environments (ADEs), which expose BI components in a graphical drag-and-drop development interface. ADEs are the analytic complement of integrated development environments (IDEs), which are used to build transaction applications. ADEs promise to accelerate development time and reduce costs by allowing power users in every workgroup to tailor an analytic template for their immediate colleagues.

There are a wide range of ADEs on the market today. Pure ADEs, like those from arcplan, Business Objects, Microsoft, ProClarity, and SAP, give developers almost unlimited control over the look and feel of an application and the way users navigate through it. Newer report authoring tools, like those from MicroStrategy and ADVIZOR Solutions, give report developers greater flexibility to create a range of reports or dashboards to meet unique requirements of a broad range of users.

ADEs for Dashboards. Many ADEs are now used to build customized dashboards and scorecards, which are quickly becoming the most popular way for users to navigate analytical information. While most dashboards today are strategic in nature and enterprise in scale, the number and type of users supported indicate that we are still in the early stages of dashboard deployments.

Research Methodology

Report Scope. This report is designed for technical executives who wish to learn more about development techniques for creating analytic applications. The report describes the range of available techniques, analyzes trends in how organizations develop analytic applications, and briefly describes the development capabilities offered by its sponsors.

Methodology. The research for this report is based on a survey that TDWI conducted in the fall of 2004, as well as interviews with BI practitioners, consultants, industry analysts, and report sponsors. TDWI defines BI as an umbrella term that encompasses both data warehousing and business intelligence (i.e., analytical tools and applications). We will use the term in this context for the remainder of this report.

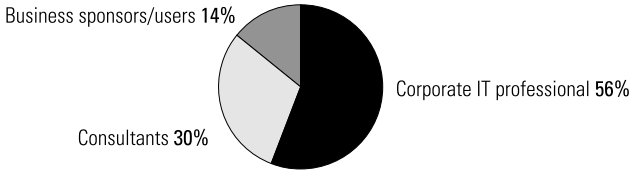
Survey Methodology. TDWI contacted BI professionals in its database and 101communications' database. (TDWI is a business unit of 101communications.) A total of 496 people responded to the survey. We did not count results from 23 respondents who said they work at vendor firms or are professors or students. So in total we had 473 qualified respondents.

Survey Demographics. A majority of the qualified survey respondents (56 percent) are corporate IT professionals. The rest are independent consultants (30 percent) and business sponsors/users (14 percent). Most of the qualified respondents (60 percent) work in corporate IT. The rest work in either divisional IT (21 percent), departmental IT (14 percent), or some other organizational unit (6 percent).

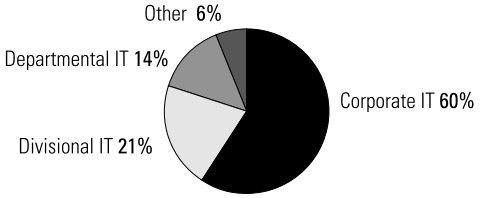
Most respondents are U.S.-based corporate IT professionals. More than one-quarter of respondents (28 percent) work at companies with revenues of less than $100 million. Almost a third (30 percent) work at companies that earn between $100 million and $1 billion in revenues. Another third (31 percent) work at companies with more than $1 billion in revenues. Most respondents are based in the U.S. and work in a range of industries, the largest percentage of which are consulting and financial services industries. Consultants were asked to fill out the survey with their most recent client in mind.

Demographics

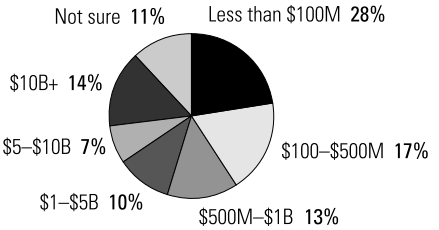
Position



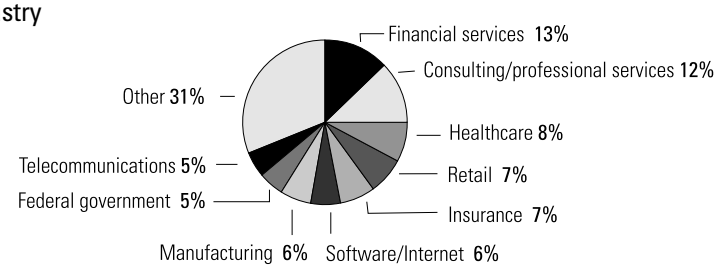
Departmental Affiliation



Company Size by Revenue



Industry



Geography

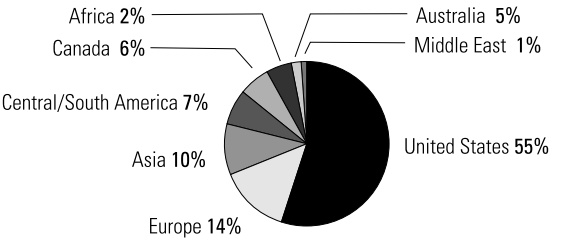


Illustration 1. Based on 473 qualified respondents out of 496 total. Vendor representatives, college professors, and students were not counted.

Overview of Analytic Applications

Characteristics of an Analytic Application

Business intelligence (BI) professionals throughout the world have one thing in common: no matter how they design or architect their systems, the end result is an *analytic application.* Yet, this phrase is vague and generally misunderstood by most data warehousing and business intelligence (DW/BI) professionals and their business counterparts.

The Definition of an Analytic Application Has Always Been Vague

This confusion exists partly because we, as an industry, tend to focus on the tools, technologies, and architectures we use to create analytic applications, rather than their output. But if we've made the term ambiguous through lack of attention, we've also bastardized it by giving it a multiplicity of definitions. Although no definition will satisfy every constituency and industry pundit, we use the following in this report:

> *An **analytic application** consists of a series of logically integrated, interactive reports, including dashboards and scorecards, that enable a wide range of users to access, analyze, and act on integrated information in the context of the business processes and tasks that they manage in a given domain, such as sales, service, or operations.*

As this definition implies, an analytic application is more than just a "bucket of reports," as Forrester Research analyst Philip Russom is fond of saying. An analytic application also provides more than just unfettered access to dimensional data, which has been the aim of BI tools until recently.

This definition contains four key elements that an analytic application comprises:

1. **Logically integrated.** An analytic application is first and foremost an *application.* That means that some degree of business logic is embedded in the application that helps users navigate through a series of tasks, among other things. In the transactional world, an application steps users through a series of data entry screens and validates their inputs. In the analytical world, an application steps users through a series of interactive reports or views of dimensional data, ideally leading to the point of action—a decision, plan, or request for more information.

   Guided Analytics Equal Navigational Logic

   While savvy power users may not need navigational logic, the bulk of users who view reports once or twice a week need *guided analytics* to help them effectively analyze data and take action. This navigational guidance can take many forms. In a custom-built application, users may push a button to move to the next report or logical view. Or they may traverse a hierarchical tree of reports, tabbed worksheets, or a briefing book, or they may select predefined filters in a parameterized report.

   In addition, some BI tools offer users context-sensitive recommendations for the next reports they should view or actions they should take. For example, dashboards and scorecards offer built-in navigational logic via graphical icons whose colors, shape, or direction inform users what metrics and data to examine in which order.

2. **Interactive reports that enable users to access, analyze, and act.** Whereas transactional applications usually consist of fixed screens, analytical reports are interactive. This means users can drill from a top-level view of a report to lower levels for additional information. In a dashboard, for example, users might click on a red stoplight to pull up a compound chart and table, then click on the chart to view transaction data. Interactive reports can be delivered

by any number of technologies: OLAP cubes, parameterized reports, linked static reports, advanced visualization techniques, dashboard/scorecards, numeric search, and so on.

A Subject-Oriented Slice of a DW Environment

3. **Integrated information.** Architecturally, an analytic application is a subject-oriented slice of an enterprise data warehousing environment. A single data warehouse may support dozens of analytic applications. For example, Continental Airlines, which won the 2003 TDWI Leadership Award, has 34 analytic applications running against its enterprise data warehouse. These applications range from tracking flight statistics and fraud detection to revenue management and analysis of pilot and flight attendant pay. Other organizations run analytic applications on separate data marts, although this distributed approach often creates problems when executives seek a consistent version of enterprise information.
4. **Address a business domain.** Analytic applications are defined by the information requirements of a business domain, such as sales, service, or manufacturing. For example, a sales analytic application may let a sales team monitor its pipeline of opportunities, analyze the performance of sales representatives and regions, and examine the sales and contact history of prospects and clients. These domains are often interconnected within a logical model, since multiple business areas reference the same entities (customers, products, geographies, and so on).

Summary. An analytic application is not just a set of randomly created reports in a person's inbox or "My Reports" folder. An analytic application contains business logic that steps users through a series of interactive reports so they can access, analyze, and take action to optimize the activities in a specific business domain.

Profile of Analytic Applications

Our definition of analytic applications comes from observing the way organizations deploy them in the field. The next section profiles the characteristics of analytic applications that organizations have already implemented.

Business Areas. According to our research, the most popular domain for analytic applications is finance and accounting, which was cited by 61 percent of our survey respondents. Since finance touches every area in an organization and financial data is the language of business, it's not surprising that the finance group creates more analytic applications than any other department. In a distant second is sales (45 percent), followed by marketing (38 percent), forecasting (37 percent), budgeting and planning (35 percent), and customer service (34 percent). (See Illustration 2.)

Groups Support One to Five Analytic Applications

Scope. Organizations have a surprisingly low number of analytic applications. Whereas we routinely come across organizations with dozens of analytic applications, more than two-thirds of organizations (70 percent) we surveyed have five or fewer analytic applications. Only 14 percent have more than 10 analytic applications. (See Illustration 3.)

Why the small scope of deployments? First, packaged analytic applications sold by vendors have not caught on as fast as some industry observers predicted. Second, there is probably a slight imbalance among our survey respondents toward those who are just starting to build analytic applications, dashboards, and scorecards—people who are taking the survey to learn more about the topic. Second, we required respondents to list only the number of analytic applications that their immediate working group supports—not their organization as a whole. Collectively, organizations may support dozens of analytic applications in multiple BI environments, but each group may support only a small portion of the total.

Analytic Application Domains

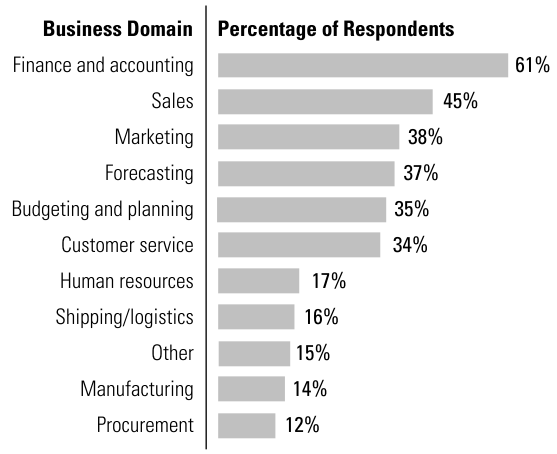


Illustration 2. Based on 473 respondents.

Number of Analytic Applications

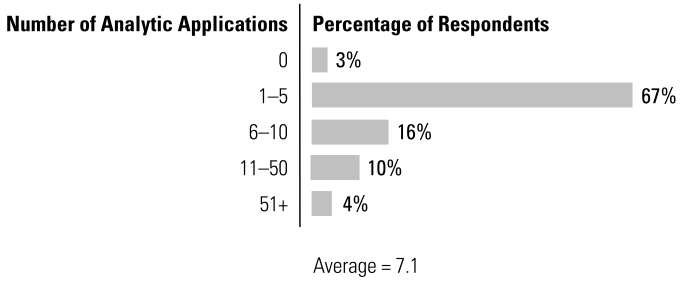


Illustration 3. Based on 473 respondents.

Number of Users. We next asked respondents to describe the number and types of users that access their analytic applications. On average, organizations support 320 users per analytic application and 528 users per data warehousing environment.[1] In actuality, these numbers are skewed by a small percentage of companies with extremely large user deployments. Closer examination of the data reveals that more than three-quarters (76 percent) of organizations have 100 users or fewer per analytic application. According to Roman Bukary, vice president of SAP xApps and Analytic Applications:

> In general, business intelligence and analytic application vendors have done an incomplete job in "democratizing" analytics. … Analytic application and tools need to focus not only on the traditional "business analyst," but really need to be redesigned to address the needs of the typical business user who is very knowledgeable about their business but who does not want to master arcane tool menus or key clicks. These users need "right information at the right time to make the right decision," and it is the analytic applications which must accommodate user requirements to access, analyze, and act on relevant, comprehensive data with ease.

[1] We obtained the first number (320) from a question focused only on dashboards and scorecards, which are a specific type of analytic application. The 528 number comes from preliminary results in TDWI's 2005 *Business Intelligence Salary, Roles, and Teams Report*, published in February 2005.

Number of Users per Analytic Application

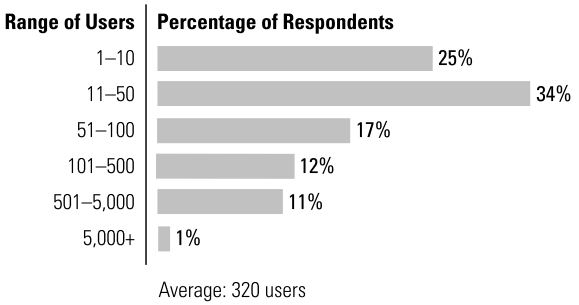


Illustration 4. Based on 473 respondents.

User Types. The types of users that access analytic applications can be broadly divided into four groups:

- **Consumer.** Views reports on a regular basis, but only occasionally explores the data in more depth. (Also called a "casual user.")
- **Analyst.** Regularly explores and filters data to find trends or answers to problems. (Also called a "power user.")
- **Author.** Modifies existing reports or templates developed by IT or professional application developers.
- **Developer.** Creates reports and templates from scratch; usually an IT person or application developer.

According to our research, most users of analytic applications are consumers (44 percent), followed by analysts (31 percent), developers (14 percent) and authors (12 percent). We suspect that the ratio between consumers and analysts will widen as organizations deploy more analytic applications to more users, including customers and suppliers. (See Illustration 5.)

User Types

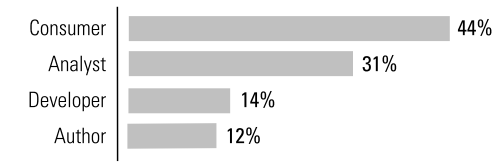


Illustration 5. Most analytic application users simply consume information with minimal exploration. Based on 473 respondents.

Access. Users access the analytic application through either a Web client or a desktop client. (See Illustration 6.) Leading BI vendors have recently brought the functionality offered by their Web-based tools in line with their desktop tools. Thus, in the next few years, we expect to see the percentage of Web users outstrip desktop users by a healthy margin.

Surprisingly few users access the analytic application through a portal or operational application. We suspect these percentages will also increase in the next several years.

Data Access Platforms

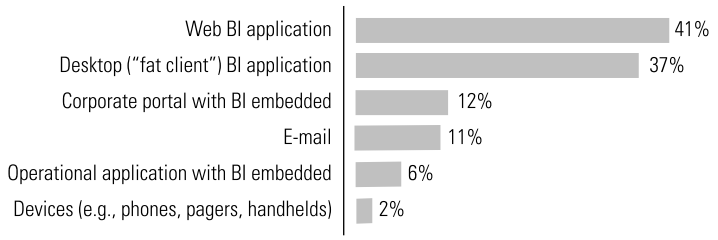


Illustration 6. Most users access the analytic application directly through a Web client.

Report Interactivity. Although interactive reports form the heart and soul of an analytic application, this doesn't mean static reports are dead. In fact, they are alive and well!

Interactivity Is the Key to Analysis

According to our survey, 87 percent of organizations still generate static reports. (See Illustration 7.) This is partly because old habits die hard, and partly because some reports, such as financial reports for investors and auditors, are best delivered in static form. However, almost three-quarters of organizations (70 percent) also produce interactive reports, and a majority let users create new reports from scratch (58 percent) and interact with charts and visual displays (53 percent).

Report Interactivity in Analytic Applications

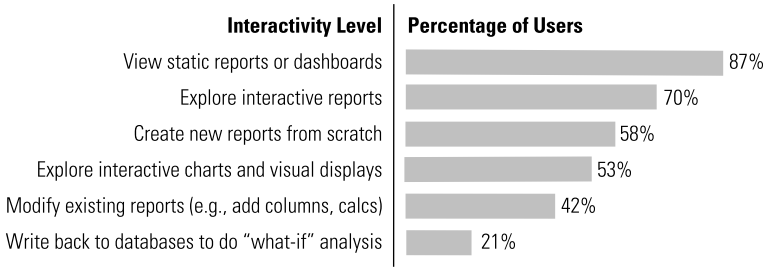


Illustration 7. Most analytic applications support a range of end-user interactivity. 473 respondents, selecting more than one answer.

Data Sources. Traditionally, analytic applications run against a single source: a data warehouse or data mart. But most survey respondents indicated that their analytic applications pull data from multiple sources. The most common reason for this trend is the growing prevalence of spreadmarts, or Excel-based analytic applications built by power users that extract data from multiple systems. Unfortunately, spreadmarts create silos of analytic information and undermine an organization's ability to deliver a single version of truth.[2]

Spreadmarts Undermine the Consistency of Enterprise Information

But there are other reasons for multi-sourced analytic applications, such as:

1. The organization needs to provide near-real-time information, which requires developers to combine transaction and analytic data using distributed queries and joins
2. The organization doesn't own the data and isn't allowed to store it locally

[2] See TDWI's report, *In Search of a Single Version of Truth: Strategies for Consolidating Analytic Silos,* July 2004, www.tdwi.org/research.

3. The application is an older management reporting system that runs directly against operational systems
4. The organization needs to consolidate information stored in multiple data marts or data warehouses.

Our research shows that a significant percentage of organizations still multi-source data. Fifty-six percent of our survey respondents said their analytic applications draw data from a data mart or divisional data warehouse. However, only a slightly smaller percentage (53 percent) pull data from an enterprise data warehouse. Trailing are OLAP data sources (34 percent), legacy applications (27 percent), external data sources (23 percent), and packaged applications (18 percent). (See Illustration 8.)

Data Sources Supported

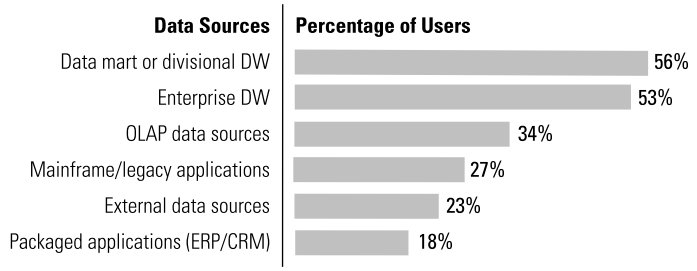


Illustration 8. Most analytic applications access data from data marts or data warehouses. Respondents could select more than one answer. Based on 473 respondents.

Summary: Although some organizations currently deploy dozens of analytic applications to hundreds of users, the majority of organizations in our survey are still inching their way toward maturity in this space.

Development Approaches

Build versus Buy

Now that we understand what analytic applications are and how they're deployed, let's explore how organizations create them.

There is a range of approaches for creating analytic applications. On one end of the spectrum—the "buy" side—organizations purchase packaged analytic applications that require minimal customization and little or no coding across a range of functionality. On the other end of the spectrum—the "build" side—programmers write the entire application from scratch using custom code. Between these two poles are hybrid options that blend both packaged and custom approaches.

Pros and Cons. Many organizations seek to purchase tools or packages to standardize their software investments, accelerate deployment, and reduce total cost of ownership. Unfortunately, as we shall see, many companies end up *over*-customizing the commercial software, undermining its potential to reduce costs and accelerate deployment. And although building applications from scratch provides complete flexibility, it is expensive to keep developers on staff to maintain and extend a custom application that may already exist commercially.[3]

[3] For a detailed examination of the pros and cons of each side of the "build" versus "buy" issue, see *The Rise of Analytic Applications: Build or Buy?* September 2002, www.tdwi.org/research.

The Need to Customize. Today, no single development approach delivers a complete analytic application out of the box. Even packaged analytic applications supply only 60 to 80 percent of requisite functionality, depending on users' requirements, the organization's existing infrastructure, and the data sources used. Thus, organizations must find ways to customize existing BI packages and tools using a variety of "build" approaches.

"Buy and Extend" Is the Most Common Development Approach

Buy and Extend. It's no surprise, then, that the majority of respondents to our survey (52 percent) said their organizations prefer to both build *and* buy application components—the "buy and extend" approach. Here, organizations purchase a BI tool or analytic package and then customize it. (See Illustration 9.)

"Does Your Group Prefer to Build or Buy Its Analytic Applications?"

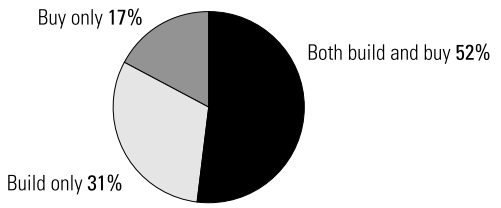


Illustration 9. Based on 473 respondents.

The remaining 48 percent of respondents took a hard line, preferring to either build or buy analytic applications. Nearly twice as many of these "hard-liners" prefer to build rather than buy (31 percent to 17 percent). (See Illustration 9.) This two-to-one ratio mirrors the findings in our 2002 study, *The Rise of Analytic Applications: Build or Buy?*

Summary. While the industry is beginning to coalesce around a buy-and-extend approach to creating analytic applications, some organizations are still committed to build-only or buy-only methodologies.

Architectural Layers

Before purchasing a tool or writing code, an organization needs to architect the environment in which the analytic application will run. Three architectural layers support analytic applications:

1. **BI Architecture.** The BI layer enables users to access, analyze, and act on information contained in back-end databases—usually a data warehouse or data mart. The major elements of a BI architecture include: (1) a graphical interface that provides the look and feel of the application, (2) a semantic layer that provides a layer of abstraction between end users and back-end data sources and presents complex data objects in simple business terms, and (3) an analytic server that processes end-user requests for reports and data and handles security, administration, performance, and other things.
2. **Data Architecture.** The data architecture is most often implemented using a data warehousing environment, which integrates data from multiple back-end systems by mapping, transforming, cleaning, and moving the data into a single, consistent data model. While most BI tools run against a data warehouse or data mart, some can also run directly against source systems, bypassing the data warehouse. However, these applications' data

architecture is very thin, usually driven by an enterprise information integration (EII) tool, a transaction system, or an old-style reporting system.

3. **Technical Architecture.** This layer comprises the technical infrastructure required to support and run the organization's applications. The technical architecture is usually delivered by the information technology (IT) department. It consists of servers, storage systems, networks, desktop computers, wireless devices, backup and restore systems, and so on.

Although we've seen that most analytic applications are built using BI tools and Excel, there is much more to an analytic application than what these tools provide. To get a sense of the overall distribution of work required to build an analytic application, we asked survey respondents to tell us how much time they spend creating these architectural layers. On average, organizations spend only 42 percent of their time on the data architecture, 38 percent on the BI architecture, and 20 percent on the technical architecture. (See Illustration 10.)

Distribution of Development Workload

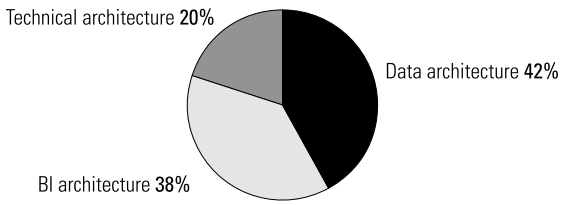


Illustration 10. The percentage of total development time devoted to creating each type of architecture. Based on 473 respondents.

However, these averages may be misleading. When organizations create their first analytic application, they often need to build a data architecture from scratch, a task which can easily consume 60 to 80 percent of overall development time. On the other hand, organizations that have already implemented a data warehouse supporting several analytic applications will spend considerably less time on data sourcing and management issues, accelerating overall development.

The Spectrum of Development Techniques

Once an organization lays the architectural foundation for its analytic applications, it must select the tools and techniques to develop them.

Below is a list of development techniques that organizations use to create analytic applications. Illustration 12 (see page 15) shows the extent to which each can deliver a complete analytic application. In the next section we will examine several of the more prominent techniques in detail.

1. **Packaged Analytic Applications** are predefined reports and BI objects (i.e., semantic layer) that run against an integrated data model. This enables users to carry out a comprehensive set of analytical processes, potentially delivering 60–80 percent of the functionality needed.
2. **Packaged Data Marts** bundle an ETL (extract, transform, and load) tool, source adapters, a target data model, and source-to-target mappings to accelerate the development

of the back end of an analytic application.[4] Most ERP (enterprise resource planning) vendors offer packaged data marts that work with their own packaged operational applications. Some packaged analytic applications contain packaged data marts, a combination that delivers up to 95 percent of a finished application.

3. **BI Starter Kits** are templates in a BI tool that contain style sheets, metrics, reports, and some business logic (e.g., semantic layer, data model) tailored to a business domain in a vertical industry, such as category management in the retail industry. The templates accelerate development, and also give end users sample screens and reports to help them articulate requirements.
4. **Microsoft Office Tools** consist of Microsoft Excel and desktop databases (e.g., Microsoft Access, contact managers) that provide end users with a powerful interface for manipulating and displaying analytical information. These tools either query source data directly or via a BI tool's analytic server. They provide excellent graphics, powerful modeling, and a macro language to insert business logic. Excel has been, and always will be, the most popular "BI tool" in the industry. Unfortunately, its popularity has led to the proliferation of analytic silos.
5. **BI Tools** were traditionally deployed out of the box with little customization. Customizing a BI tool required writing custom code against the tool's software development kit (SDK) or tweaking administrative options to make small changes, such as font type, colors, and backgrounds. However, many BI vendors have recently redesigned their tools using a service-oriented architecture, making it easier for developers to customize a tool's functionality or look and feel.

   At the same time, many BI tools now offer extremely flexible *report design environments*. Many now use a desktop publishing paradigm that lets them drop various objects (tables, charts, visualizations, graphical icons) into a report to create a custom look and feel. These report design environments, as well as the component-based tools that support them, are beginning to rival the capabilities of full-blown analytic development environments (see ADEs).
6. **Analytic Development Environments (ADEs)** are the newest development technique on the block. ADEs are typically component-based extensions of BI tools that let developers and power users create sophisticated analytic applications by dragging analytical objects onto a graphical workbench, where they can be connected and configured to create an analytic application without writing much code, if any at all.

   ADEs are the analytical complement of IDEs (integrated development environments) such as Microsoft's Visual Studio.NET. ADEs go beyond BI report designers by allowing developers to insert business logic into an application to control the user's experience at a more granular level. (See Illustration 11 for a sample screenshot of an ADE.)

[4] Adapters are predefined interfaces that connect to specific systems or packaged applications. Source-to-target mappings contain transformations required to move selected source data into the target data model.

Analytic Development Environment

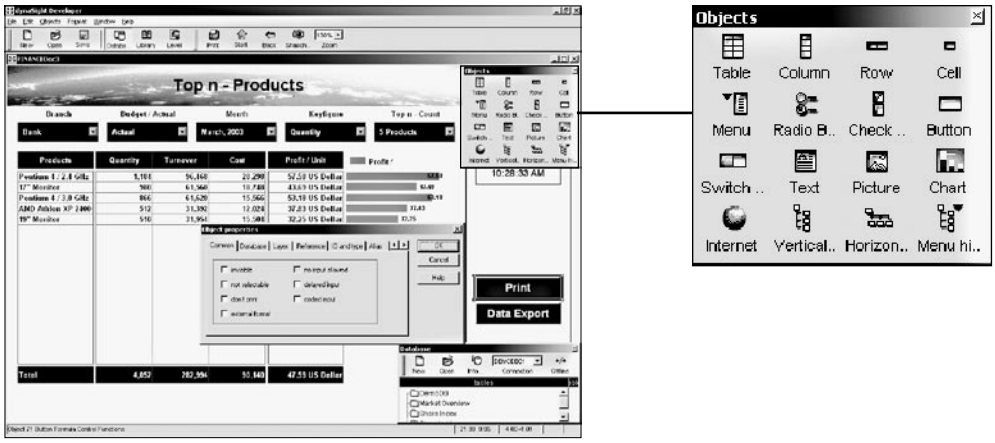


Illustration 11. In an ADE, developers drag and drop graphical objects and widgets onto a screen—such as "table," "column," "cell," and "button"—and configure them in a WYSIWYG (what you see is what you get) environment. (Courtesy arcplan, Inc.)

7. **Scripting.** Developers can modify a BI tool's look and feel, output, or interactivity by changing the templates or scripts that generate Web pages. These scripts include active server pages, Java server pages, CGI scripts, and XML configuration files. Alternatively, developers can write Visual Basic or other types of scripts to insert custom functionality into a BI tool, such as adding a custom toolbar or a menu screen that pops up prior to loading the BI tool. The problem with custom scripts is that they are difficult to maintain and often must be rewritten whenever an organization upgrades its BI tool, Web browser, or Web application server, for example.
8. **Portal Integration Kits.** These kits let companies embed BI reports and controls in a portal offered by a third party, such as Microsoft, SAP, IBM, BEA, or Oracle. The kits expose basic BI functionality in the format required by the portal API (application program interface), such as HTML tags, URL strings, Java servlet extensions, or .NET objects. Unfortunately, these "portlet" connections are often brittle; they don't support sophisticated process handling conventions such as error recovery, error messaging, timing, and message queuing. In addition, BI vendors must commit to keeping their kits in sync with portlet APIs offered by portal vendors. Some vendors, such as SAP with its Visual Composer tool, are embedding portal integration kits right into ADEs to accelerate portal integration.
9. **BI Software Development Kits (SDKs).** Most BI tools provide SDKs that expose most of their product's functionality in one or more APIs. Some BI tools publish thousands of calls to their API, which may come in many flavors: Java, .NET, COM, and XML. A good SDK thoroughly documents all the calls, provides sample code, and offers online help and a community of developers to share tips and ideas. As vendors move to service-oriented architectures, low-level API calls are being replaced with higher-level calls (or methods) for accessing services. And ADEs are replacing SDKs for considerable custom development. Today, developers mostly use SDKs to call BI tool functionality from within a third-party application.

10. **Custom Code.** Many Web developers customize reports and applications by writing custom code, using 3GLs, 4GLs, Web scripts, and SQL. When building applications from scratch, developers often use Java- or .NET-based integrated development environments (IDEs) attached to one or more Web application servers (e.g., IBM WebSphere, BEA's WebLogic, or Microsoft's IIS). Today's Web application servers accelerate application development by handling common server processes, such as database connectivity, application integration, security, clustering, Web delivery or project management, that developers previously had to create from scratch.
11. **Modeling Tools.** Organizations can purchase modeling tools to create a data warehouse or data mart model. Or they can create logical models the old-fashioned way with paper and pencil, and then deploy them using database definition language (DDL).
12. **ETL.** Organizations can write extraction, transformation, and loading routines with custom code or purchase a commercial ETL tool. ETL and modeling tools are necessary to build the data architecture of an analytic application.

Application Completeness. An analytic application comprises many elements. At a high level, these include: (1) look and feel, (2) reports and analytics, (3) analytic server, (4) business logic, (5) data model, and (6) source mappings. (See sidebar on page 17, "Major Elements of an Analytic Application.") These elements must be either built or purchased; even if purchased, developers will need to modify or extend them to meet business requirements.

Illustration 12 shows the degree to which each development technique just listed supports these six analytic application elements. Clearly, no technique alone creates an analytic application, although packaged applications and custom coding go the farthest toward delivering comprehensive functionality.

Major Application Elements

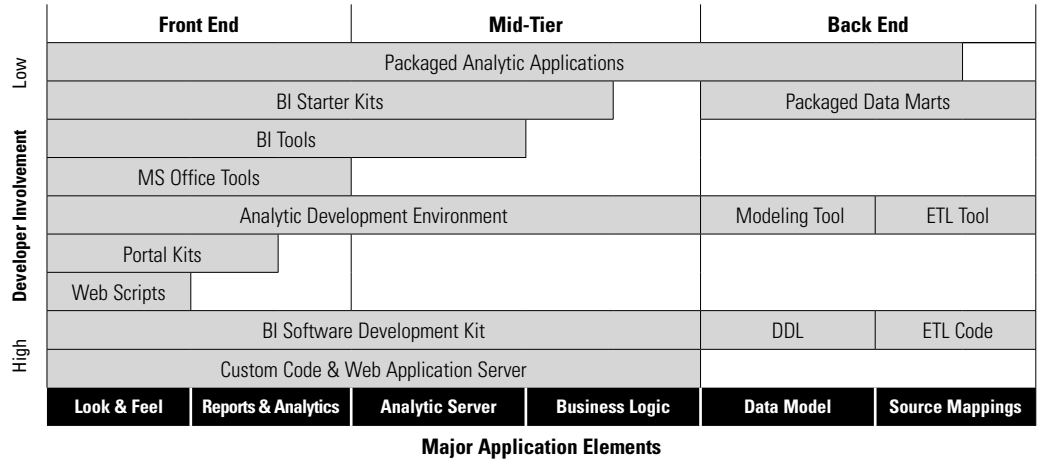


Illustration 12. No development technique supports all major elements required to deliver an analytic application.

If an organization is large enough, it can assign separate teams to build the front-end, mid-tier, and back-end elements, as pictured in Illustration 12. (These tiers roughly correspond to the BI architecture, data architecture, and technical architecture.)

However, the teams must cooperate extensively, since there is considerable overlap between the architectures. For instance, which team should build the components represented in the mid-tier? Should the metrics be calculated in the data model, the ETL programs, the analytic server, the reports, or a combination of all four? What facility should manage authentication and access control: the reports, the BI tool, the database, an LDAP server, or all four? And so on.

"We've been successful because we've put all three development groups [ETL, BI, and Web] in one room so they can more easily communicate about what they're working on and optimize development," says Kevin Lam, manager of business performance at Telus Corp., a Canadian telecommunications company.

Development Techniques in Practice

Primary Technique. Given the spectrum of development techniques for building analytic applications, which techniques or combinations of techniques do organizations employ in practice?

BI Tools Are the Starting Point

First, we asked survey respondents to identify the primary development technique that serves as the foundation for their analytic applications. Respondents could only select one technique, forcing them to choose the one they use as a starting point.

Not surprisingly, the largest percentage of respondents (36 percent) selected BI tools. Less than half as many selected custom code (16 percent), Microsoft Office tools (15 percent), or packaged analytic applications (11 percent). (See Illustration 13.)

Foundation of Analytic Applications

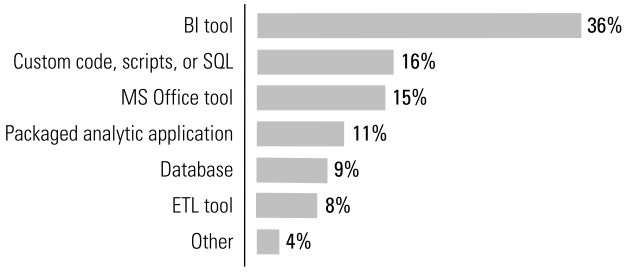


Illustration 13. Respondents were asked to select the item their group prefers to use as the basis for its analytic applications. Based on 473 respondents.

Custom Coding Is the Most Prevalent Technique

All Techniques. Next, we asked respondents to select all the development techniques they used to create their single largest or "primary" analytic application.[5] Their answers would tell us which techniques organizations employ most often, as well as the combination of techniques in use.

Although organizations prefer to build their primary analytic application with BI tools, on the whole, they use custom coding more than any other development technique. More than half (52 percent) of respondents use custom code when building an analytic application. Next, BI tools were selected by nearly the same percentage as in the previous question (35 percent), followed by Microsoft Office tools (28 percent), packaged analytic applications (22 percent), BI tool SDKs (21 percent), and starter templates (11 percent).

[5] Asking respondents to answer questions about the single largest application managed by their group provides more accurate data than asking them to estimate organization-wide averages.

Major Elements of an Analytic Application

Analytic applications consist of six major elements. Almost any of these elements can reside on any tier. (See Illustration 12.)

- **Look and Feel**
  The "look" consists of the graphical user interface (GUI), which defines the colors, fonts, and background screens. The "feel" consists of the navigational framework—how components are laid out on the screen and how users activate them.

- **Reports and Analytics**
  Reports consist of predefined (and usually formatted) views of information that can be delivered as static or interactive documents. Interactive documents let users drill down to more detail or filter the data to get different views. Analytics incorporate sophisticated calculations, statistical algorithms, or advanced visualizations to help users analyze large volumes of complex data.

- **The Analytic Server**
  The analytic server is largely responsible for processing queries using run-time metadata defined in a design environment and stored in a server repository. The analytic server also performs administrative functions such as security, report delivery, usage tracking, clustering, and file storage.

- **Business Logic**
  We've already discussed one type of business logic: navigational rules that step users through a series of reports, views, actions, or workflows. However, there are other types of business logic: (1) calculations that specify how groups define metrics and KPIs; (2) a semantic layer that provides a business view of back-end data objects, generates syntactically correct queries on behalf of users, and insulates applications from changes in back-end systems; (3) query logic that constrains runaway queries and prioritizes query requests; (4) prompts that expose contextually sensitive filters for a report or data set; and (5) dimensions, hierarchies, and metrics defined in multidimensional models. Business logic can exist in reports or programs that run on clients, mid-tier servers, or databases.

- **The Data Model**
  The data model functions as the brain of the analytic application. It often works in conjunction with a BI semantic layer—or set of query objects—that represents data elements in common business terms. Data models are used to define the design of an OLAP cube, the logical view of distributed databases, or the conceptual, logical, and physical structure of a data warehouse or data mart.

- **Source Mappings**
  Source-to-target mappings define the transformations required to load source data into a target data model. ETL tools enable developers to create, manage, and execute these maps.

Profile of an Analytic Application. Extrapolating from this data, the "typical" analytic application consists of a BI tool that has been customized using custom code, scripts, or SQL, is written partially against the SDK, and supports Excel as an alternative front end. (See Illustration 14.)

Development Techniques Used in Primary Analytic Applications

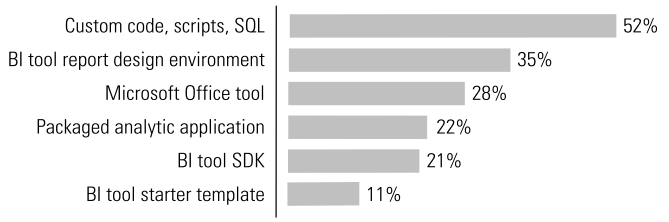


Illustration 14. Techniques that groups used to build their primary analytic application. Respondents could select multiple choices. Based on 473 respondents.

Most Custom Code Is SQL

Custom Code. With so many organizations using custom code in their primary analytic application, we were curious to find out what the most popular languages are. Not surprisingly, most coding is done using SQL, followed by Visual Basic, Web scripts, Java, and .NET. (See Illustration 15.) Many respondents who selected "other" use database programming languages, such as Oracle's PL/SQL.

"You may think you don't need to write SQL when you purchase a BI tool, but that's not true," says a database developer at a major credit card company, who asked not to be named.

Custom Coding

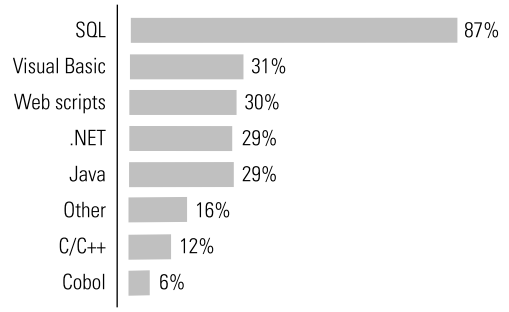


Illustration 15. Languages used to write custom code in an analytic application. Based on 294 respondents who said they use custom code in their analytic applications.

Custom Code Comprises 46% of a Finished Analytic Application

Composition of Analytic Applications. Custom coding is not only the most prevalent technique for building analytic applications; it also comprises the largest percentage of the finished product—46 percent on average. This is a surprisingly large percentage, given the maturity of most BI products and packages on the market today.

We wondered whether some development techniques require more custom coding than others. The answer appears to be *yes.* Our data shows that organizations write more custom code when their preferred development technique is a Microsoft Office tool (61 percent) or a database (52 percent). Custom code is 45 percent for both BI tools and packaged analytic applications, which is still a sizable portion. (See Illustration 16.)

Percent of Custom Code in an Analytic Application by Approach

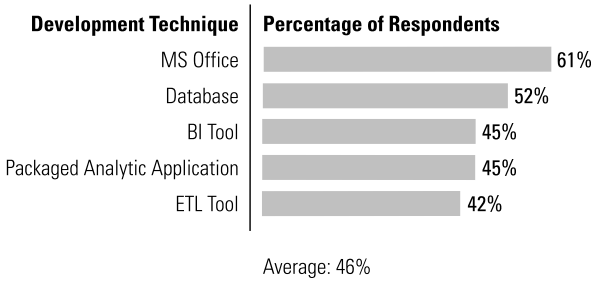


Illustration 16. This chart shows how much of a primary analytic application consists of custom code when using different types of BI products.

A Large Percentage of Excel-Based Analytic Applications Come from Small Firms

We also looked at the makeup of organizations using Microsoft Excel in their analytic applications. We discovered that a preponderance of Excel-based analytic applications exist in small- to medium-sized organizations. Specifically, a third of organizations using Excel in their analytic applications have less than $100 million in revenues, while the figure is 18 percent for organizations with revenues between $100 million and $500 million. (See Illustration 17.)

Microsoft Office Usage by Company Revenues

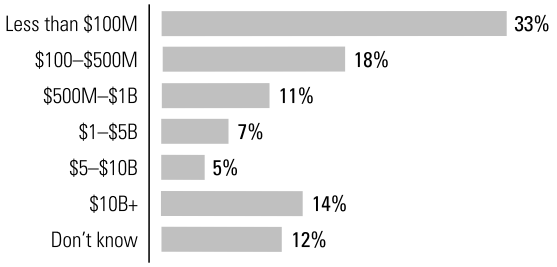


Illustration 17. Smaller companies are more likely to use Excel in their analytic applications. Based on 193 respondents who said Microsoft Office is a component of their primary analytic application.

BI Tools and Packages. We also examined where organizations sourced the BI tools and packages used in their primary analytic application.

Users Prefer Independent BI Vendors

Organizations are more than *four times* as likely to purchase a BI tool from an independent BI vendor as an ERP vendor (82 percent to 18 percent) as the basis for their primary analytic application. But ERP vendors close the gap somewhat in analytic packages. Organizations are more than three times as likely to purchase an analytic package from an independent BI vendor as an ERP vendor (77 percent to 23 percent) for their primary analytic application. (See Illustration 18.)

This shows that while ERP vendors have made inroads into BI, they still have a long way to go before they can dominate the market.[6]

[6] See the TDWI report, *Developing a BI Strategy for ERP/CRM Data*, October 2004, for an in-depth analysis of the approaches being taken to integrate BI environments with packaged operational data and ERP vendor data warehousing and BI offerings.

Sourcing Products

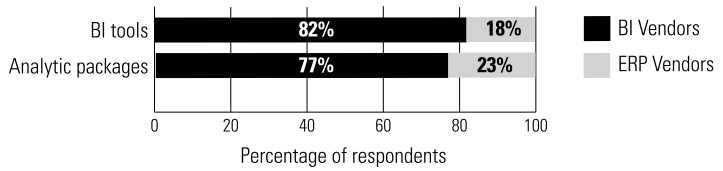


Illustration 18. Organizations are much more likely to buy BI tools and analytic packages from independent BI vendors than ERP vendors.

Most Organizations Customize or Extend One-Third of Their BI Tools and Packages

Multiple Tools. From this data, we are able to calculate the number of vendors that supply organizations with BI tools or packages for their primary analytic application. It's important to note that these numbers refer only to a single analytic application, not all applications in the entire organization.

On average, organizations purchase BI tools from 1.95 vendors for use in their *primary* analytic application. If they are using an analytic package, they purchase products from slightly fewer vendors (1.7 on average). So it's clear that even within a single application, organizations are leveraging multiple BI products or packages from different vendors.

Top Suppliers. The top vendors supplying survey respondents with BI tools for their primary analytic application are Business Objects (31 percent), Cognos (28 percent), Microsoft (20 percent), Crystal Decisions (now also Business Objects) (14 percent), Informatica (14 percent), Oracle (13 percent), and MicroStrategy (10 percent).

The top vendors supplying survey respondents with a packaged analytic application for their primary analytic application are Business Objects (31 percent), Cognos (28 percent), Microsoft (22 percent), Hyperion (15 percent), Oracle (14 percent), and SAP (10 percent).

Customizing Analytic Applications

If nearly half of every analytic application consists of custom code, what are developers creating with that code? How are they modifying an existing analytical product or package? Or are they supplementing those products with additional functionality?

Organizations Use BI Tools and Packages from More Than a Single Vendor to Create Their Primary Analytic Application

To answer that question, we investigated how organizations are customizing both BI tools and packaged analytic applications, two of the primary development techniques. (We didn't ask the same series of questions about Microsoft Office tools, although results would likely be the same as for BI tools.)

Degree of Customization. Our data shows that organizations customize or extend about one-third (33 percent) of their BI tools and packaged applications. This is a substantial percentage, increasing the overall cost of deploying and maintaining the resulting analytic application. However, a large percentage (44 and 43 percent, respectively) customize less than 20 percent of their BI tool. (See Illustration 19.)

What Percentage of the BI Tool or Package Did You Customize?

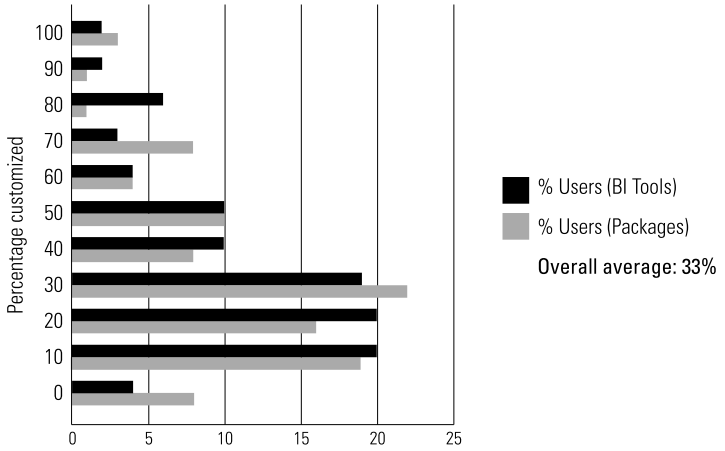


Illustration 19. On average, organizations customize 33 percent of their BI tools and packages, but most customize less than 20 percent. Based on 293 respondents for BI tools and 135 respondents for packages.

What Gets Customized? So what elements of the BI tool and analytic package do organizations customize or extend? Almost everything, according to our survey!

Look and Feel, Screen/ Layouts, and Calculations Get Customized the Most

For BI tools, the elements that developers customize the most are look and feel (58 percent), screen and layouts (57 percent), calculations (56 percent), and report navigation (47 percent). It appears that developers spend the most time customizing those elements that end users interact with most frequently. (See Illustration 20.)

The results for packaged applications are almost the same as BI tools, but with a few exceptions. First, developers are slightly less inclined to customize an analytic package's look and feel, layout of screens, and report navigation. But almost 50 percent of the time they customize data models and ETL mappings, two elements that do not exist in BI tools.

Companies spend a lot of time customizing the look and feel of dashboards and scorecards. Many firms we talked with want to recreate the look and feel of their graphically designed scorecard in software.

Chick-fil-A, a nationwide restaurant chain, created its electronic scorecard to resemble the five puzzle pieces in its paper scorecard. "Standard scorecard packages have limited display options, so we built our own application using PL/SQL, Java, and a charting engine," said Chris Levine, reporting architect at the company. "Rather than customizing a packaged application, we felt our company's unique business rules required starting from scratch."

Customizing Scorecard Home Pages

What Parts of the BI Tool Did You Customize or Extend?

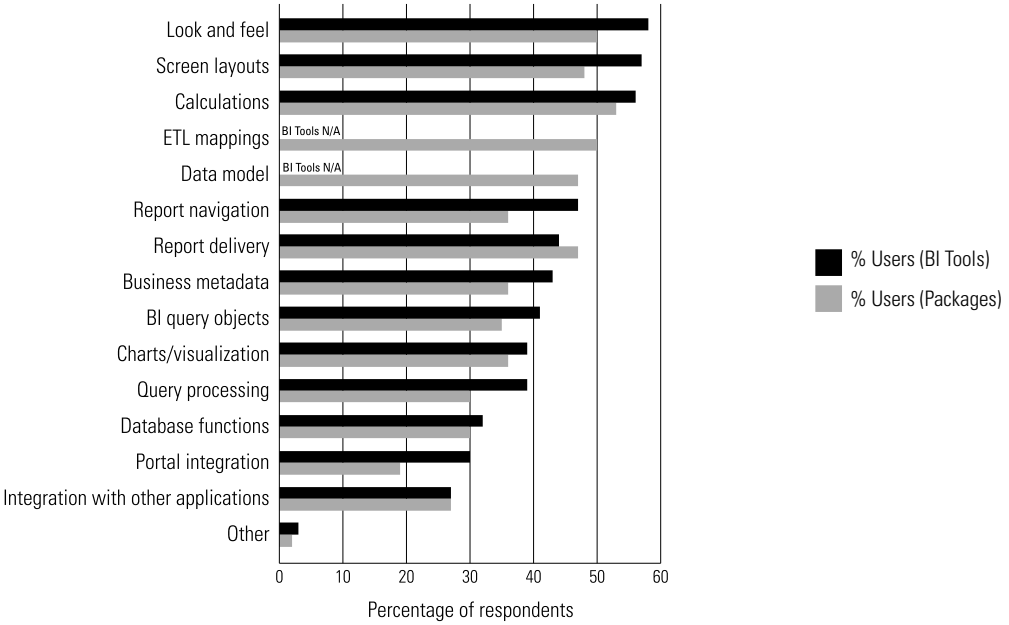


Illustration 20. Based on 293 respondents for BI tools and 135 respondents for packaged analytic applications.

Which Elements Did You Customize the Most?

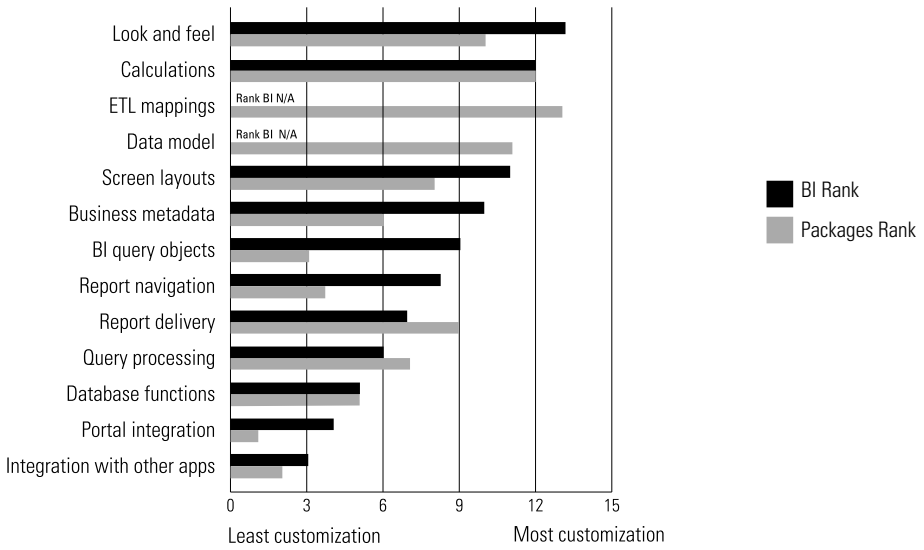


Illustration 21. Developers spent the most time customizing the look and feel and calculations in BI tools.

Ranking the Most Customized Elements. When we asked respondents to identify which application elements they customized *the most*—not just which ones got customized—we uncovered some interesting nuances.

Developers Spend the Most Time Customizing Things End Users Touch the Most

The results for BI tools are similar to the above question on which elements they customized. Organizations spend the most *time* customizing an application's look and feel, the calculation of metrics, and screen layouts. But after that, metadata overtakes report navigation and delivery as the next most customized element.[7] (See Illustration 21.)

For packaged analytic applications, ETL mappings, calculations, and data models topped the list in a cluster. This shows that the "data architecture"—even when packaged—still consumes the lion's share of development time.

Developers Customize the Data Architecture the Most with Analytic Packages

Development Processes

Development Time. Despite the advent of ADEs and packages, analytic applications take a long time to develop. On average, it takes organizations 7.5 months to design and implement an analytic application.

The actual duration varies by the development approach. For example, respondents who prefer to build their analytic applications in Excel finish the job in 26.5 weeks, whereas those who custom-code their analytic applications take an average of 37 weeks. (See Illustration 22.)

It Takes 7.5 Months to Design and Implement an Analytic Application

Time Required to Build an Analytic Application

Preferred Approach	Average Number of Weeks to Build
Microsoft Office tools	26.5 weeks
BI tool	28.7 weeks
Packaged analytic application	31.5 weeks
Custom code, scripts, or SQL	37 weeks
	Overall Average: 30 weeks

Illustration 22. Based on 473 respondents.

Of course, these averages say nothing about the richness and functionality of the resulting analytic application. A fully featured analytic application designed to support hundreds or thousands of users will take longer to build than a lightweight application built in Excel.

Also, it will always take organizations much longer to build their first analytic application than subsequent applications. That's because they need to build the BI architecture, data architecture, and technical architecture concurrently. After that, adding new applications to the infrastructure should take significantly less time.

Analytic Packages Don't Accelerate Deployment, on Average

Surprisingly, analytic packages—which vendors often price and sell at a premium because they "accelerate development time"—do *not* deliver a speed advantage. According to our survey, it takes organizations an average 31.5 weeks to deploy a packaged analytic application. The problem is that organizations spend too much time customizing the packages—due partly to a culture that permits over-customization, and partly to a lack of desired features. Organizations should avoid purchasing

[7] Business metadata and BI query objects translate technical elements into business terms that users can understand.

packages that lack required features, but many are constrained by corporate "buy-only" policies. They should also avoid over-customizing packages, although the temptation is often too great to resist, especially when users demand special features.

Modifying Analytic Applications. It's one thing to build an analytic application; it's another to maintain it. If it takes too much time and too many resources to modify an analytic application, the total cost of ownership rises and executives look for ways to sunset the system. Developers often need to modify both the front end and back end of an analytic application. The *front end* refers to the BI architecture, and the *back end* to the data architecture. Our survey shows that organizations change the front end of an analytic application a tad more frequently than the back end. (See Illustration 23.)

How Frequently Do You Modify the Analytic Application?

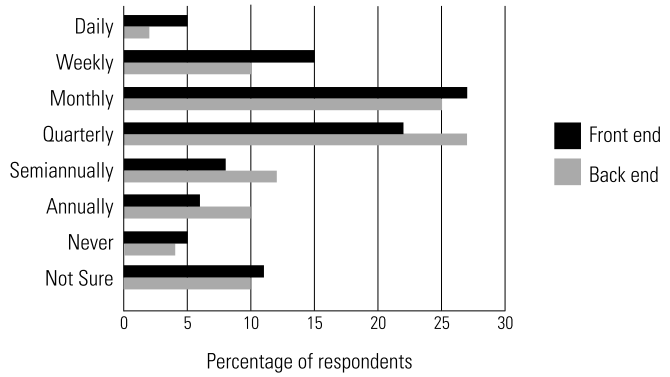


Illustration 23. Based on 473 respondents.

Time Required to Modify. A bigger disparity between front- and back-end development occurs when we examine the time it takes developers to make changes to an analytic application. More than one-fifth of respondents (21 percent) say they can modify the front end of an analytic application in less than a day, while only eight percent can modify the back end in the same time frame.

One-Fifth of Companies Make Changes to the Front End in Less Than a Day

This is not surprising given the rise of point-and-click report design environments and ADEs. In contrast, modifying data models, source-to-target mappings, and data quality validations are more manual in nature, and therefore take longer, even with commercial ETL tools. Back-end modifications require careful planning (i.e., impact analysis), testing, and debugging. Nevertheless, a majority of organizations (54 percent) can execute changes to back-end environments in a week or less.

How Long Do Modifications Take?

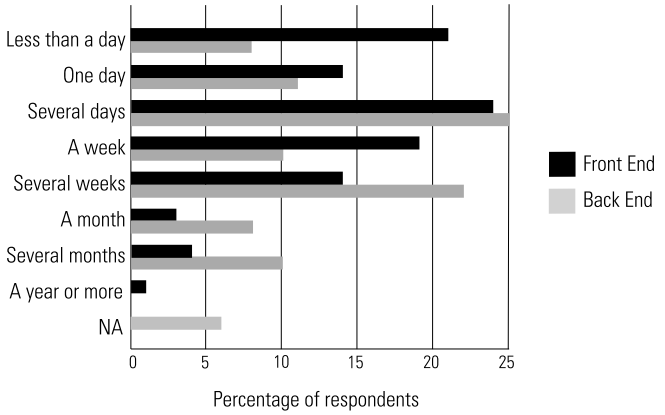


Illustration 24. Based on 473 respondents.

Developer Profile. When it comes to making modifications to either front- or back-end environments, the majority of the work is handled by IT staff and application developers. However, power users are four times as likely to modify the front end of analytic applications as the back end. Power users are more familiar with the business logic expressed in the BI environment than the data sources and access language required to modify the back end. Also, most of these modifications can be done without writing code.

Who Makes the Modifications?

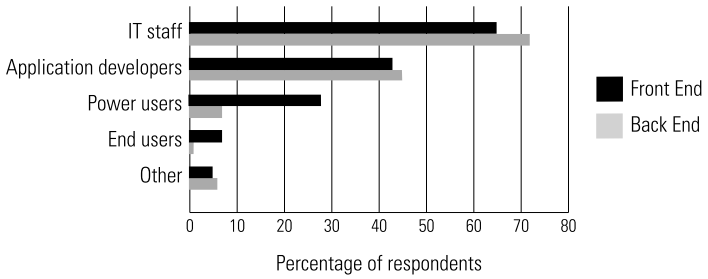


Illustration 25. Based on 473 respondents.

Summary. In practice, organizations are most likely to build analytic applications around a BI tool, but then add substantial amounts of custom code (mostly SQL) to customize and extend the application. Most organizations also customize the BI tool itself, focusing on the GUI, calculations, and navigational elements. Developers spend significant time customizing ETL mappings and data models in packaged applications. Developers—mainly IT staff and application programmers—make frequent changes to analytic applications, while power users are often enlisted to change the front-end environment.

Analytic Development Environments

As mentioned, there is an emerging category of tools that makes it faster and easier to create custom analytic applications. TDWI calls this new toolset an analytic development environment, or ADE.

ADEs Are the Analytic Complement to IDEs

An ADE is the analytic counterpart to the integrated development environment, or IDE, which developers have used for years to build operational applications. Examples of the more popular IDEs today are Microsoft Visual Studio.NET, Borland's JBuilder, Eclipse, IBM's WebSphere Studio, and BEA's WebLogic Workshop, to name a few. ADEs are the spiritual heir to IDEs, both in functionality and name.

A Promising Future. ADEs promise to accelerate the development of custom-built analytic applications, as well as make it easier and faster to customize packaged analytic applications. An ADE enables developers to drag and drop analytic components onto a screen to rapidly create analytic applications. More than report designers, ADEs give developers precise control over the look and feel, functionality, and workflow of an application. As a result, users will soon be using ADEs to "buy and extend" analytic applications, quickly customizing the last 20 to 40 percent of the front end.

ADEs Will Get IT Out of the Report Creation Business

In fact, the drag-and-drop nature of ADEs will further shift development responsibilities from IT developers to power users in the field. With an ADE, a power user can easily modify a packaged analytic application, flesh out a report definition, or create a new application or report from scratch (once IT has established data connections and BI query objects). Thus, ADEs will once and for all get the IT staff out of the business of creating reports so they can focus on what they are best at: building robust data architectures and abstraction layers for end users.

ADE tools will also accelerate the trend towards rapid prototyping. Developers and power users can use an ADE tool in a joint application design session to get immediate feedback from users on data, application screens, metrics, and report designs. This iterative process results in better designed applications that are delivered more rapidly. Many vendors are shipping ADEs for specific applications to facilitate rapid prototyping. For example, many dashboard and scorecard solutions are ADEs.

Reusable, Object-Oriented Components Power ADEs

Service-Oriented Architecture. The real power behind ADEs comes from the fact that vendors have componentized the functionality of their BI tools. In the past, vendors hard-wired presentation, logic, and data functionality together. But the advent of object-oriented programming and service-oriented architectures has enabled vendors to open up their products, componentizing functionality within a services-oriented framework. The upshot is that ADEs enable developers to create multiple instances of components, store them centrally, and reuse them in other applications.

Vendor Offerings

Tool Extensions. In most cases, ADEs are extensions of existing BI tools. Many BI vendors, under pressure from users, now offer an ADE version of their BI tool. (See the appendix at www.tdwi.org/adeapp for a list of criteria for evaluating vendor ADEs.)

ADEs: Replacement for SDKs

"We found that many users wanted more flexibility to create custom applications using the components in our toolset without dropping down into an SDK," says Clay Young, senior vice president of strategic marketing at ProClarity Corp. "So now we give users the option of getting our traditional BI tool out of the box or a developer's version [ProClarity Analytics Application Development Platform] in which we surface hundreds of our components in a graphical development environment."

ProClarity recently went one step further into the ADE camp when it released an Excel-based design and modeling environment that provides real-time access to multiple, distributed data sources, including ODBC/JDBC, CSV, and Web pages. IT professionals set up persistent data connections and then step aside, allowing power users to create reports and screens inside Excel (see Illustration 26).

Real-Time ADE

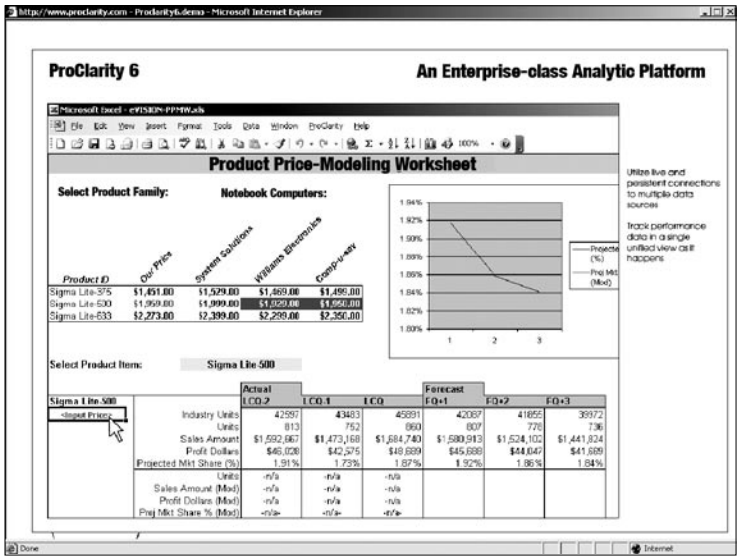


Illustration 26. ProClarity's new Live Server product empowers business users to rapidly create read/write analytic applications in Excel after IT establishes persistent connections to multiple data sources. Here, an analyst at a distributor compares product pricing with direct competitors (whose prices were extracted in real time from their Web sites), and then inserts new prices to model the impact of a price change on margins and revenues. (Courtesy ProClarity Corp.)

ProClarity introduced Live Server because it found that traditional analytic application development cycles were too long for many users. According to Young:

> For every business user who has been served by a traditional analytic application, there are 50 who haven't. Even tools like ProClarity [Analytic Applications Development Platform], which speeds development by an order of magnitude, is not fast enough to serve many users. With ProClarity Live Server, we put the tools in business users' hands so they can develop custom applications quickly.

But Young points out that ProClarity Live Server is not a replacement for most types of analytic applications. For example, companies will still want to use ProClarity's development platform if they wish to deploy Web-based applications (versus Excel-based desktop applications) that provide guided analysis, rich OLAP navigation, and sophisticated visualization and calculation objects.

Commercializing Internal Development Tools

Internal Requirements. Other ADEs have evolved from practical necessity. When Business Objects decided to enter the market for packaged analytic applications, it leveraged numerous analytic engines, visualization components, and query/reporting tools it had built and acquired over the years. It wrapped these components together into a graphical development environment to build its suite of packaged applications. After completing the applications, Business Objects decided to commercialize the development environment as well. Much to its surprise, the toolset, then called Application Foundation, became as popular as the packaged applications it helped build.

"We discovered that there was pent-up demand among customers to 'buy and extend' their BI tools and packaged applications, which is why [Application Foundation] has been one of our fastest-growing products in recent years," says Steve Wooledge, product marketing manager at Business Objects.

Business Objects now calls the ADE "Performance Manager" to reflect the way most organizations use it—as a specialized ADE for building dashboards and scorecards. Performance Manager contains metrics maps, strategy maps, goal setting, alerting, and built-in annotations and discussions—all the elements a developer needs to create any type of dashboard or scorecard. (See Illustration 27.)

Specialized ADE for Dashboards and Scorecards

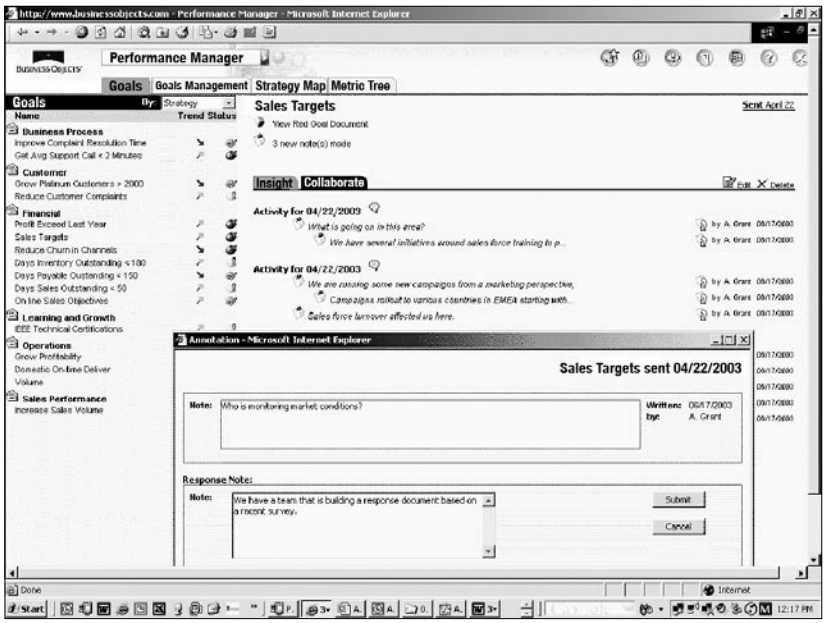


Illustration 27. BusinessObjects Performance Manager provides a host of components for building dashboards and balanced scorecards. This screenshot shows four tabs that let users define and view goals, thresholds, initiatives, strategy maps, and metrics. The body of the screen shows color-coded performance against goals and a threaded discussion attached to those goals.

ADE Pure-Plays. Outside of mainline BI vendors, some companies have emerged as ADE pure-plays. AlphaBlox was the first company to evangelize the benefits of component-based analytical development. But like many first-movers, Alphablox mistimed the market and never gained significant traction. It was recently acquired by IBM and will be embedded in various IBM development environments.

Another ADE pure-play is arcplan, an 11-year-old German firm with 1,800 customers. The company provides a "codeless" development environment that lets developers or power users rapidly build robust analytic applications against OLAP, relational, and legacy data sources. Initially, arcplan grew its business around Hyperion Essbase, which used Excel as its only front-end environment. Now, arcplan is gaining momentum as a front end to Microsoft Analysis Services. Unlike many ADEs, arcplan lets developers position objects anywhere on a screen to create a completely custom look and feel. (See Illustration 11 on page 14.)

Microsoft BI Development Workbench

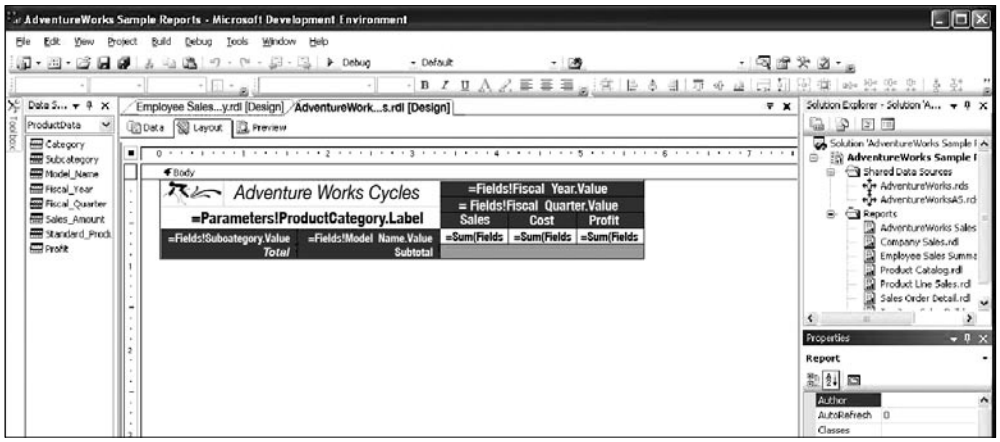


Illustration 28. Microsoft's ADE is a BI instantiation of Microsoft Visual Studio.NET.

Leveraging IDEs. Some ADEs are built inside IDEs (integrated development environments). For example, Microsoft's ADE, BI Development Workbench, is a BI-instantiation of Microsoft's IDE, Visual Studio.NET. With Microsoft's BI Development Workbench, developers can not only build reports but also create ETL programs, design relational schemas and OLAP cubes, and build data mining models. This is a one-stop-shop ADE, running on one of the most popular IDEs in the industry today.

SAP Web Application Designer

IDEs Integrate Analytics and Other Services

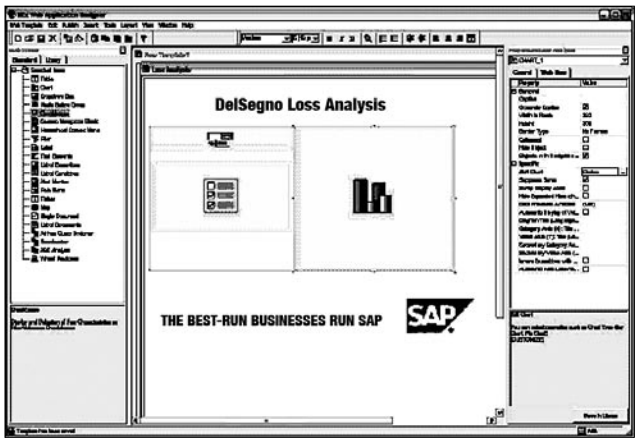


Illustration 29. SAP's Web Application Designer is an ADE that produces zero-client analytic applications that leverage the NetWeaver platform.

In a similar fashion, Crystal Decisions (acquired by Business Objects in 2003) differentiates its toolset by embedding Crystal Reports and report components inside third-party IDEs, specifically Microsoft's Visual Studio.NET and Borland's JBuilder. This lets application developers create and embed Crystal reports into their applications without having to leave the comfort of their preferred IDE.

Like Microsoft, SAP delivers BI and data warehousing capabilities through its broad-based NetWeaver platform, but with a twist. SAP's BI-specific ADE—a Windows tool called Web Application Designer—integrates with every other service supported by NetWeaver, including SAP and non-SAP services, and generates zero-client analytic applications. (See Illustration 29.)

Currently, the NetWeaver platform supports portal management, collaboration, knowledge management, messaging, master data management, and multicasting. This means, for example, that developers can add collaboration or document management services to an analytic application without additional coding, integration work, or cost.

"We are blurring the lines between analytic and operational applications," says Roman Bukary, SAP NetWeaver product marketing manager. "Ideally, analytics become invisible, embedded within transaction processes and workflows, assisting the user and guiding their actions."

Report Design Environments. Other vendors blur the lines between an ADE and a robust report or dashboard authoring environment. For example, ADVIZOR Solutions enables power users to create interactive visual applications without coding. ADVIZOR Solutions supports 12 visualization components that users can query, drill into, filter, and animate. These components include standard bar, pie, and line charts as well as advanced visualization techniques such as histograms, scatter plots, data constellations, and Paraboxes. (See Illustration 30.)

Drag-and-Drop Interactive Visual Dashboards

A Visual ADE

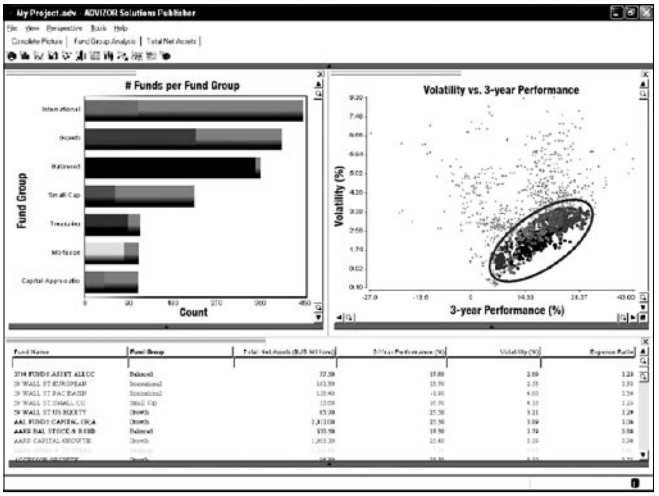


Illustration 30. ADVIZOR Solutions enables power users to create visual dashboards, as shown. In this example, their Visual Discovery™ technology enables users to circle a cluster of data points that represent strong mutual fund performance. The scatter plot highlights the selected data points and grays out others, while the dynamically linked bar chart and data sheet automatically highlight the selected data. Users can then click an icon to remove non-selected data and then drill down to make fact-based decisions.

"Our visualization components enable users to quickly identify patterns in complex data that would be nearly impossible to ferret out in traditional BI tools," says Doug Cogswell, chief executive officer of ADVIZOR Solutions. Since ADVIZOR uses an in-memory database, it excels when analyzing hundreds of thousands of records that are up to 40 columns wide.

Dancing on the edge of an ADE, MicroStrategy provides new report design tools that let users create a wide variety of report types, including dashboards and scorecards, master-detail management reports, invoices and statements, and compound reports that mix graphical and tabular elements. The finished HTML reports use drag-and-drop functionality, embedded OLAP grids, and links to other reports to deliver an interactive end-user experience. Reports can also be output to Excel and PDF.

The firm's newest release, MicroStrategy 8, provides two report development environments. The first, MicroStrategy 8 Web Design Mode, is geared to professional developers who need to create complex, Web-based reports that offer document navigation, query prompts, and OLAP grids to provide a high degree of interactivity for end users. The second, MicroStrategy 8 WYSIWYG Mode, is geared to power users who may take a report definition created in Web Design Mode and customize document formatting, grid objects and views, and other elements for a specific group of users.

"Our goal is to provide flexible report design without the traditional steps and backlog involved in designing and delivering a report," says Mark LaRow, head of marketing for MicroStrategy.

Robust Report Design Environment

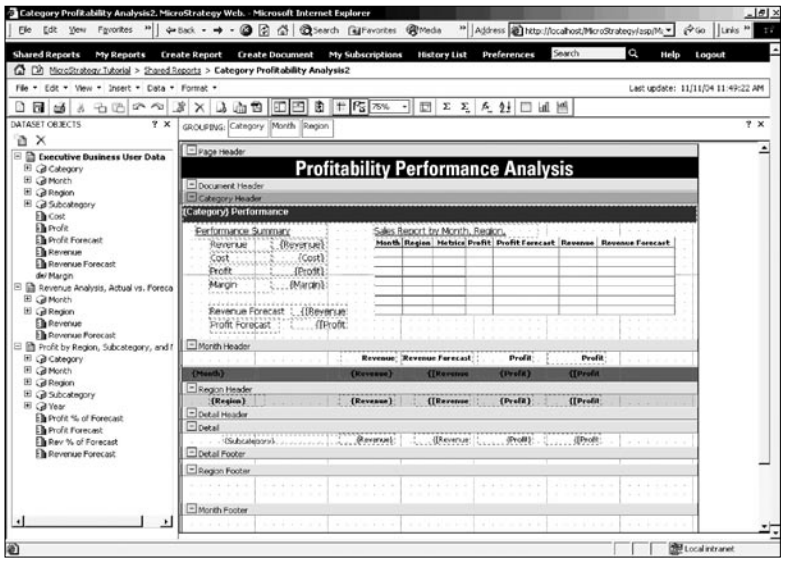


Illustration 31. The MicroStrategy 8 Web Design Mode provides both banded and zone-based (i.e., desktop publishing-like) report design capabilities that enable authors to create many different types of reports and imbue them with interactive features, such as embedded grids, prompts, and document navigation features.

Summary. Clearly, BI vendors have recognized the need to deliver "buy and extend" capabilities. Most are starting to deliver ADEs or ADE-like capabilities. Pure ADEs, like those from arcplan, Business Objects, Microsoft, ProClarity, and SAP, give developers almost unlimited control over the look and feel of an application and the way users navigate through it. Newer report authoring tools, like those from MicroStrategy and ADVIZOR Solutions, give report developers much more flexibility to create a wide range of reports or dashboards to meet the unique requirements of a broad range of users. In both cases, the tools provide an easier-to-use authoring environment, which is helping to finally move development out of the hands of professional developers and into the hands of power users and business analysts.

Using ADEs to Build Dashboards and Scorecards

Rapid prototyping using ADEs will soon become the predominant method for building dashboards and scorecards. Many BI vendors already offer specialized ADEs for creating dashboards. (See Appendix for criteria to evaluate vendor dashboard solutions: www.tdwi.org/adeapp)

A Dashboard Tells How You Are Doing; a Scorecard, How Well

Definition. A *dashboard* or *scorecard* is a graphical display that compares performance against predefined goals. Most people use these two terms interchangeably, although there is a subtle difference. A dashboard records actual performance or behavior—like an automobile dashboard—while a scorecard measures that performance against objectives or goals. In other words, a dashboard tells you *how* you are doing, while a scorecard tells *how well* you're doing. We will use the term "dashboard" from now on to refer to both types of analysis.

Popularity. Dashboards are increasingly popular; a majority of our survey respondents said their group uses a dashboard as its primary analytic application (31 percent) or has deployed one elsewhere (28 percent). Another 24 percent are currently developing a dashboard or scorecard. Thus, almost three-quarters of respondents either have a dashboard or scorecard or are developing one.

Dashboards Are Visual Exception Reports

Visual Exception Reporting. The popularity of the dashboard is growing because it provides an extremely intuitive interface to monitor and analyze data. With dashboards, users can glance at graphical, color-coded icons to determine whether a process, project, or task is on target, drilling into charts and tables only when necessary.

Proactive Notification. The best dashboards go one step further: if performance is off-target, they notify users automatically, even sending the data via e-mail or another channel. In essence, dashboards are visual exception reports that filter the morass of data and highlight critical elements for users. When implemented with "right-time" data, dashboards enable users to work proactively and intelligently, without getting bogged down collecting or analyzing data that has little immediate business value.

Types of Dashboards. There are three types of dashboards: strategic, tactical, and operational.

Operational dashboards monitor business processes and activity as it happens so users can intervene to fix a problem or capitalize on an opportunity. Operational dashboards leverage "right-time" information—operational data collected and displayed as it occurs or within a time frame deemed appropriate. For instance, a flight controller needs up-to-the-minute information about incoming and outgoing flights, while a truck dispatcher may only require hourly data to manage shipments efficiently.

Tactical dashboards measure the progress and performance of departments or project teams against goals established by a mid-level manager. Many tactical dashboards are really portals of commonly used reports and indicators. Since tactical dashboards facilitate monitoring *and* analysis, they often incorporate more sophisticated analytical capabilities than operational dashboards, such as visual mining and discovery capabilities. (Illustration 30 on page 30 provides an example of visual mining and discovery.) This is why these types of dashboards are sometimes called "analytical dashboards."

Strategic dashboards contain metrics that are linked to corporate strategy. These dashboards are "cascaded" to every level and group in the organization, each with a different version of corporate metrics tailored to its business domain. Strategic dashboards ensure that all parts of the organization march in lockstep toward a common goal. One way to implement a strategic dashboard is to use the Balanced Scorecard methodology created by David Norton and Robert Kaplan.

When we asked survey respondents to identify the type of dashboard they support, 41 percent chose a strategic dashboard. This was followed by 35 percent who maintain tactical dashboards, and 23 percent who support operational dashboards. (See Illustration 32.)

Types of Dashboards in use

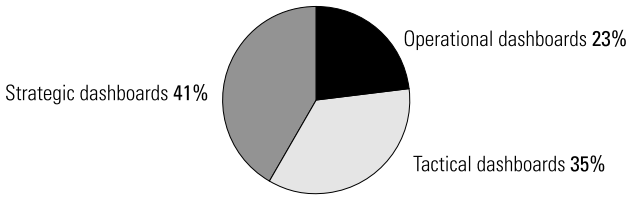


Illustration 32. A larger percentage of organizations are implementing strategic dashboards than any other type of dashboard. Based on 240 respondents who said their group has deployed a dashboard.

Scope. The average number of users per dashboard at these organizations is 320, but this number is skewed by a few organizations that have deployed dashboards to tens of thousands of users. In reality, three-quarters of organizations have fewer than one hundred dashboard users. Dashboards still have a way to go before they support a majority of an organization's users.

Three-quarters of Organizations Have Fewer Than 100 Dashboard Users

At the same time, almost a majority of survey respondents (48 percent) say their dashboard application is "enterprise" in scope. A much smaller percentage say their dashboard has a divisional scope (28 percent), departmental scope (16 percent), or inter-enterprise scope (7 percent). In our experience, most companies implement dashboards for a functional area, such as finance or operations, but on an enterprise scale. In other words, everyone in the functional area throughout the company uses the dashboard.

Levels. We also discovered that organizations deploy *three* levels of dashboards throughout the enterprise. We did not ask whether these dashboards are cascaded, but it's safe to assume most are.

Types of Dashboard Users

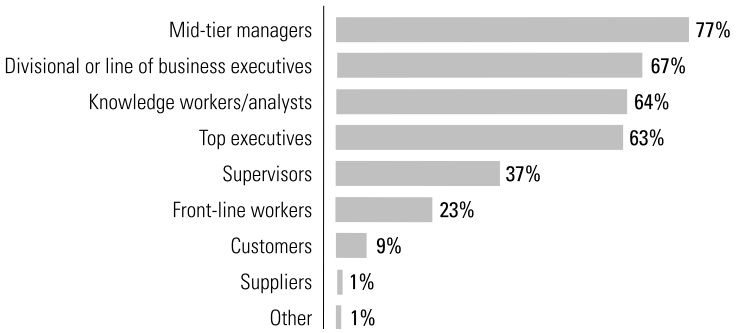


Illustration 33. Based on 240 respondents.

Users. The highest percentage of dashboard users are mid-tier managers (77 percent). I suspect these managers are using tactical scorecards to keep track of the processes, projects, and operations under their management. Other top users are divisional executives (67 percent), knowledge workers (64 percent) and top executives (63 percent). (See Illustration 33.)

Dashboard Tools. Most organizations created their dashboards from BI tools (41 percent), followed by custom code (22 percent) and Microsoft Office (13 percent). Only 17 percent have purchased a packaged dashboard solution from a vendor. We expect this percentage to rise in the next few years as more vendors provide robust dashboards solutions.

A performance dashboard delivers information to users in layers as they need it. The top layer is what we traditionally associate with a dashboard or cockpit, graphical indicators that alert users to exception conditions. The middle layers allows users to "slice and dice" the data, while the bottom layer provides transaction data in the form of a table or report. (See Illustration 34.)

Dashboard Functionality

Dashboard Levels		
	1	Graphical Indicators **72%** *(Dials, gauges, speedometers, maps)*
	2	Interactive Tables/Charts **82%** *(OLAP, parameterized reports, linked reports)*
	3	Transaction Detail **54%** *(Contained in a data warehouse or legacy systems)*

Illustration 34. Percentage of respondents who support each level. Based on 240 respondents.

Advanced dashboard implementations make it easy for executives to add or modify metrics. This ability is not overwhelmingly employed, although power users are enlisted to update dashboards in one-third of organizations. Currently, the IT and application development departments are the most likely candidates to update the dashboard to reflect new requirements. (See Illustration 35.)

Who Updates the Dashboard?

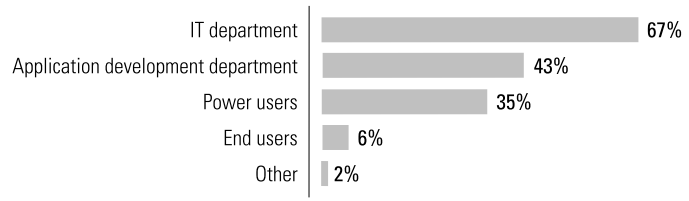


Illustration 35. Respondents could select multiple answers. Based on 240 responses.

Summary: Dashboards are quickly becoming the primary interface to business intelligence information because they conform to the way the majority of users wish to access, analyze, and act on that information. While most dashboards today are strategic in nature and enterprise in scale, the number and type of users supported indicate that we are still in the early stages of dashboard deployments in most organizations.

Conclusion

BI Tools Are Morphing into ADEs

The goal of a BI professional is to deliver an effective analytic application that makes it easy for a range of users to access, analyze, and act on information tailored to their business processes and domain. Unfortunately, we often get caught up in the details of delivering the technical and data architectures that underlie those applications instead of focusing on how well users can digest and use our output.

The best analytic applications contain navigational logic that steps users through the process of analyzing and acting on data. Building intuitive analytic applications is not easy. Organizations spend too much time customizing and extending commercial products to meet user requirements. On average, organizations customize 33 percent of the final analytic application using mostly SQL and other hand-written code. The total process takes 7.5 months on average, too much time to meet fast-changing user needs.

Fortunately, help is on the way. Most BI vendors are componentizing their BI tools and exposing them through a graphical interface. TDWI calls these tools analytic development environments (ADEs), and they promise to greatly accelerate development time and lower costs. IT will lay the technical and data groundwork for these ADEs, including the semantic mapping to back-end data sources. They will then turn them over to power users, who will tailor the analytic application to users in their workgroups.

The Future Is Clear. Many vendors have released their first-generation ADEs, and many are now tailoring them to dashboards and scorecards, where there is a heightened need to customize the interface, especially for executive-level scorecards. Next-generation BI tools will in fact become ADEs or will be embedded in packaged applications to facilitate customization. This next generation of BI tools will support users' desire to "buy and extend" existing tools and packages rather than start from scratch. We still have a long way to go, but the path to the future is now clear.

Appendix Outline

See the complete Appendix online: www.tdwi.org/adeapp

Evaluation Criteria: Analytic Development Environments

We're still in the early stages of the ADE market, but TDWI has high hopes for ADEs. We think they represent the future of BI tools. So what should you look for in an ADE? We've collected a number of requirements. Truth be told, no ADE on the market today possesses all of these characteristics, but many are making significant progress toward supporting most of them.

1. **Supports a Broad Range of Applications**
2. **Empowers Business Analysts**
3. **Supports Rapid Development**
   a. Graphical Workbench
   b. Point-and-Click Development
   c. Hidden Screens
   d. Flexible Object Placement
   e. Dynamically Linked Objects
   f. Flexible Page Navigation
   g. Automatic Data Connection
   h. Effective Debugging
   i. Easy Deployment
4. **Supports Robust Application Logic**
   a. Supports multiple types of components
      i. GUI Objects
      ii. Data Manipulation Objects
      iii. Query Objects
      iv. Event Objects
      v. Visualization Components
      vi. Administrative Objects
   b. Reusable Components
   c. Portable Components
   d. Extensible Components
5. **Accesses Multiple Sources**
   a. Support for any source
   b. Support for any schema
   c. Distributed Query
6. **Guided Analysis and Action**
7. **Read/Write Access**
8. **Flexible Output and Publishing**
9. **Team Development**
10. **Integration Architecture**
11. **Flexible Platform Support**
    a. Clients
    b. Servers
12. **Scalability**
13. **Robust Administration**

Evaluation Criteria: Dashboards and Scorecards

Many vendors now offer ADEs for building dashboards and scorecards. Some solutions are more canned than others, prescribing a certain look and feel from the get-go. However, others present developers with a flexible portal interface they can populate and paint with substantial flexibility.

In addition to the criteria listed, dashboard/scorecard-specific ADEs should support:

1. **Specialized Visualization Components**
2. **Rules Engine**
3. **Alerts and Agents**
4. **Time-Series Analysis**
5. **Drill to Detail**
6. **Collaboration**
7. **Strategy Mapping**
8. **Customization and Personalization**

ADVIZOR Solutions, Inc.
1333 Butterfield Road, Suite 280
Downers Grove, IL 60515
630.971.5250
Fax: 630.971.5289
info@advizorsolutions.com
www.advizorsolutions.com

ADVIZOR Solutions® business visualization software enables people to make better and faster fact-based decisions from their business data. Powered by patented Visual Discovery™ technology, ADVIZOR is the result of 15 years of product development including a decade of Lucent/Bell Labs research on how human beings perceive and react to displayed information. ADVIZOR is world-class in empowering people to display, interact with, and understand their business information.

arcplan, Inc.
West Valley Road, Suite 204
Wayne, PA 19087
610.902.0688
Fax: 610.902.0689
infoUS@arcplan.com
www.arcplan.com

arcplan is a globally successful software developer whose business intelligence solutions enable companies to align technology with business initiatives for optimal business performance. With arcplan's dynaSight analytic development platform, business and IT professionals can easily tailor analytic applications to unique business processes and deliver via the Web to all levels of the organization. arcplan supports more than 1,800 customers and 230,000 users worldwide. Headquartered in Düsseldorf, Germany, and Philadelphia, PA, arcplan delivers its solutions through a global direct sales force and a network of more than 130 service and implementation partners in over 30 countries. For more information, visit www.arcplan.com.

Business Objects
3030 Orchard Parkway
San Jose, CA 95134
866.681.3435
www.businessobjects.com

Business Objects is the world's leading business intelligence (BI) software company. With more than 29,000 customers worldwide, including over 80 percent of the *Fortune* 500, Business Objects helps organizations gain better insight into their business, improve decision making, and optimize enterprise performance. The company's business intelligence platform, BusinessObjects™ XI, offers the BI industry's most advanced and complete platform for reporting, query and analysis, performance management, and data integration. BusinessObjects XI includes Crystal Reports®, the industry standard for enterprise reporting. Business Objects has also built the industry's strongest and most diverse partner community, with more than 3,000 partners worldwide. In addition, the company offers consulting and education services to help customers effectively deploy their business intelligence projects.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
425.882.8080
www.microsoft.com/sql

Microsoft Corporation gives companies a competitive edge, by helping them turn information into action by using business intelligence solutions built with Microsoft technology. From executives to line operators, having the right information at the right time enables employees to make better decisions faster. Microsoft's vision is to bring "BI to the Masses" by providing a scalable, comprehensive solution that anyone can learn to use and can afford to own. See why Microsoft SQL Server Analysis Services is #1 in the market by visiting www.microsoft.com/sql today!

MicroStrategy
1861 International Drive
McLean, VA 22102
866.966.MSTR
Fax: 703.848.8610
info@microstrategy.com
www.microstrategy.com

Since 1989, MicroStrategy has helped corporations transform their operational data into actionable information. Our Business Intelligence platform, MicroStrategy 8, gives businesses solutions to all of their query, reporting, and advanced analytical needs, and distributes insight to users via Web, wireless, and voice. We have more than 2,600 satisfied customers, including Bank of Montreal, Lowe's Companies, and AT&T, and over 500 technology and integration partners, including IBM, PeopleSoft, Sun, HP, and Teradata, a division of NCR. As they can attest, MicroStrategy 8 is the best, most complete solution for any business intelligence need.

ProClarity Corporation
500 South 10th Street
Boise, ID 83702
208.344.1630
Fax: 208.343.6128
sales@proclarity.com
www.proclarity.com

ProClarity delivers analytic software and support services that recognize the high velocity of today's economy through the use of innovative real-time analytic technologies, patented visualizations, Web-like navigation, and powerful calculations that transform information into individual understanding. With ProClarity, real-time analytics meet historical context to promote better business decisions and optimal organization performance. Founded in 1995, ProClarity supports more than 2,000 customers globally including AT&T, Barnes & Noble, Ericsson, Hewlett-Packard, The Home Depot, Pennzoil QuakerState, Reckitt Benckiser, Roche, Siemens, USDA, Verizon, and Wells Fargo. To learn more, visit www.proclarity.com or call 208.344.1630.

SAP America Inc.
3999 West Chester Pike
Newtown Square, PA 19073
610.661.1000
info.america@sap.com
www.sap.com/analytics

SAP is the world's leading provider of business software solutions. Today, more than 26,150 customers in over 120 countries run more than 88,700 installations of SAP® software—from distinct solutions addressing the needs of small and midsize businesses to enterprise-scale suite solutions for global organizations. Powered by the SAP NetWeaver™ platform to drive innovation and enable business change, mySAP™ Business Suite solutions help enterprises worldwide improve customer relationships, enhance partner collaboration, and create efficiencies across supply chains and business operations. SAP industry solutions support the unique business processes of more than 25 industry segments, including high-tech, retail, public-sector, and financial services. With subsidiaries in more than 50 countries, SAP is listed on several exchanges, including the Frankfurt stock exchange and NYSE under the symbol "SAP."

5200 Southcenter Blvd.
Suite 250
Seattle, WA 98188
Phone: 206.246.5059
Fax: 206.246.5952
E-mail: info@tdwi.org
www.tdwi.org