HARVARD | BUSINESS | SCHOOL

9-503-064

REV: FEBRUARY 2, 2006

DAVID GODES

Precise Software Solutions

As Shimon Alon pulled out of his driveway early on the morning of July 22, 1999, he could not help but think about the tremendous opportunity that lay before him and his company, Precise Software Solutions. Located in Westwood, Massachusetts, Precise developed software that helped its clients to monitor, evaluate and manage the performance of their information technology (IT) systems. In 2000, the market for "performance management and availability" software was a small but growing one. Precise had thus far focused on a specific niche of this market. The bulk of its revenues had come from products managing the performance of Oracle database applications. As President and CEO, Alon felt it was clear that Precise now had the concept for an important new product with the potential to establish Precise as a leader in the market on a much broader level.

What was less clear, however, was *when* and *how* to introduce this new product, Insight, to the market. Alon and his team—CFO Ben Nye, General Manager Steve Campbell, and Aki Ratner, VP of Research & Development—had spent considerable time the previous evening considering and discussing the possible options. The "when" problem was essentially a tradeoff between time to market and functionality. More functionality in the initial version meant more development time and more time meant at least two things. First, they might not have the product ready for Oracle OpenWorld in September 2000. This annual conference gathered together thousands of interested and motivated prospects who could hear about, and see demonstrations of, the new product. Second, any delay might give Precise's competitors time to develop a similar product. While they left the previous evening's meeting without an agreement, the team did reach a consensus on one thing: they had to get this decision right. Going to market too early could be just as damaging as going too late.

Regardless of when they introduced the product, they also had to decide what their selling strategy would be. Precise had focused most of its efforts on a small set of products since its founding in November 1990. Thus, it had made a lot of sense to maintain a single sales force. However, with this new product and all of its potential, they faced for the first time the question of whether they should have a separate sales force dedicated to the new product.

The time had come for a decision on these issues. Putting it off any longer would be detrimental to the future success of the product. They had all agreed that today would be the day and they planned to meet for an early breakfast away from the office to have one more chance to sort it all out.

This case was prepared by Professor David Godes solely as the basis for class discussion. Professor Robert Dolan played a significant role in the original conceptualization and analysis of this case and his assistance is gratefully acknowledged. Cases are not intended to serve as endorsements, sources of primary data, or illustrations of effective or ineffective management.

Copyright © 2003 President and Fellows of Harvard College. To order copies or request permission to reproduce materials, call 1-800-545-7685, write Harvard Business School Publishing, Boston, MA 02163, or go to http://www.hbsp.harvard.edu. No part of this publication may be reproduced, stored in a retrieval system, used in a spreadsheet, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without the permission of Harvard Business School.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Industry Background

A company's information technology (IT) resources were a complex network of hardware and software components. The specific structure these resources took was determined by a combination of factors including the firm's data processing needs, the legacy systems in place, available financial resources and the personal preferences of the management team. Generally speaking, in 2000 IT had a significant impact on nearly all aspects of most firms' businesses. In particular, IT played an essential role in the processing of large amounts of data. In most cases, the vast majority of all employees in a firm were connected to, and were users of, the system. The data were stored physically in some storage device such as a disk drive or an optical drive. The uses of these data and the processing performed on them varied greatly across users and contexts. A "typical" IT configuration is shown in **Figure A**.

It was common to think about the IT structure in terms of the layers or *tiers* in which it was built:

Tier 1: Web or Network

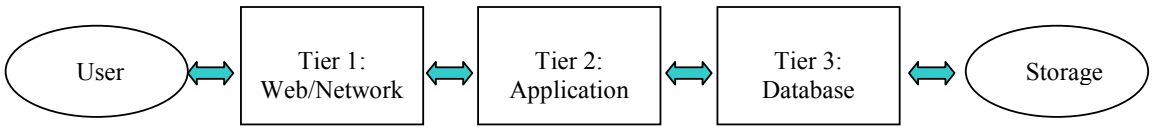
Tier 2: Application

Tier 3: Database

Users typically accessed the system either through the Web or over a network. In either case, Tier 1 provided the user with access to a centralized repository of IT resources. The specific functionality that the user sought was located in Tier 2, the application. Finally, most applications required data to provide the users what they wanted. Tier 3, the database, was a software component providing the application with access to the information it needed to perform the required function. The information itself was located physically in storage hardware. The average company had databases ranging in size from a hundred to millions of entries. The standard means of accessing this information was via a language called SQL ("Structured Query Language"). This standard had been adopted by all major database providers.

As shown in **Figure A**, the tiers were dependent on each other to accomplish their objectives. However, the tiers rarely worked as a single, dedicated and integrated unit. It was common for users to access many different applications over the Web. As well, it was common for an application to be accessed by many different users simultaneously. Many databases were accessed by several different applications and a given database often provided the input for multiple applications.

Figure A: 3-Tier IT Infrastructure[1]



Most firms ran hundreds of different applications. These included applications for searching the database, for managing email, for entering data into the database, for performing billing, etc. Essentially, any software-driven activity that a user performed was represented by an application or a group of applications. These applications varied in complexity from simple 100-line pieces of code

[1] Note that each "tier" is a software component of the IT systems and thus neither "Storage" nor the "Customer" is included as such.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

that were written by the company's IT staff to extremely complex pre-packaged applications suites purchased from commercial software developers.

As was often the case in IT, the hardware and software components of the database were typically purchased separately. The leading seller of database management software was Oracle. Founded in 1977, Oracle's revenues in 2000 were $10.1 Billion. The leading storage device vendor was EMC, which was founded in 1979 and in fiscal year 2000 had revenues of $8.9 billion.

Overwhelmingly, the starting point for a user in accessing the IT system was a personal computer. In 2000, over 95% of such system requests were made via a PC. While in the past the mainframe terminal had been the other key access point, it was expected that other devices—PDA's and cell phones, for example—would grow in popularity as the starting point of choice. The person *using* the PC might have been a customer, an employee, or a business partner such as a supplier or a distributor.

Example: Customer Relationship Management Software

A common and popular example of an application was a Customer Relationship Management (CRM) system. These systems were commonly used by companies with large sales forces to manage information about customers and prospects. The core data in a CRM system included information about each customer, names and contact information for key decision makers and influencers, and information about the budgeting cycle and purchase process employed by the firm. Moreover, ideally, each time sales rep made a call on a customer, she created a new record associated with this call to document the outcome and any new information obtained. Often, CRM systems allowed for prospects to be categorized into various "buckets" associated with different likelihoods of purchase over a certain horizon. In this way, the system helped senior managers to forecast sales.

While conceptually straightforward, CRM systems were used and deployed in such a manner that a complex set of demands was placed on each of the tiers of the IT system. For example, a sales rep calling on the local office of a major multinational customer may have wanted to check to see whether any of her colleagues in other market areas had had any success in penetrating the account. To access the system remotely, she would have typically entered the URL for the corporate CRM site into her Web browser. She then selected the relevant link on the CRM system homepage. After entering the name of the prospect firm in the appropriate field, she would be presented with the information she sought. Thus, this relatively simple objective—to find data about other reps' experience at this firm—required the IT system to utilize every one of the system's tiers. The Web provided the entry point to the system. A range of applications helped the rep to navigate the CRM website and to communicate her information needs. The database software then translated these needs into a form that the physical storage hardware could "understand" in order to relay the desired information to another application for formatting and display.

The IT Organization

See **Exhibit 1** for a typical organizational chart for a multi-business unit firm. Often, the IT group was established along functional lines. So, there were separate people dedicated to the management of each of the tiers displayed in **Figure A**. The database was the responsibility of a database administrator (DBA). Similarly, network administrators handled networks. Thus, much of the software for each of these areas was purchased by the associated IT "experts."

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Ownership of, and responsibility for, the firm's applications was a little more complicated. This was true because many applications had significant impact on the operations, and thus the profitability, of the business unit. Thus, it was hard to define a single functional area as being "responsible" for an application. The business unit for which the application would perform, and IT typically made purchase decisions for applications jointly. The former was most concerned with whether or not the functionality of the application was worth the price of the software. The IT organization, on the other hand, was most concerned with whether the claimed functionality was truly there (did the software "do" what it was supposed to do?) and what the cost would be to support the application. Moreover, IT tended to be relied upon to evaluate the software vendor itself: Could they be trusted? What was their reputation? Would they be around in five years if there were a problem?

Of course, all firms were different. So, for example, while **Exhibit 1** shows a case in which the head of IT reported to the COO, it was not uncommon for him or her to report to the CFO instead. Moreover, many large business units had considerable IT resources of their own and relied less on central IT departments for support or advice.

Software Performance

All applications, whether developed in-house or purchased in a pre-packaged form, were thoroughly tested prior to being deployed. For applications that accessed a database, this included testing the set of SQL statements to ensure that they were *effective*: they did what they were supposed to do. In addition, the developer was concerned about the *efficiency* of the application. Efficiency could be measured in a number of ways including the time it took to fill the user's request and/or the resources that the application drew on to complete its task. Two different sets of SQL statements could be functionally equivalent—i.e. always return the same, correct information—but differ by orders of magnitude on how quickly they provided that information. The application's use of critical system resources—CPU and memory, especially—was also important to the company because there were generally many applications running at a single time. Thus, a "greedy" application—one that took up a lot of space and computing power—degraded the performance of other applications running concurrently.

The efficiency with which an application ran was dependent not only on how well it was written but also on the environment in which it ran. Because this was difficult to predict beforehand, building an efficient application was a challenging task. The size of the database could change (for example, as a bank acquired new customers); the number of times the application ran in a day could increase (as an on-line bookseller increased its account base); and other applications may have been added that competed for system resources. Ideally, the application would have been robust to such environmental changes. It was, however, impossible to predict all of the changes that might occur and to build in the ability to handle all such contingencies. As a result, it was very difficult for the buyer of a software application to evaluate how efficiently it would run *ex ante*.

The Market for Performance Management and Availability Software

Broadly speaking, the category of software referred to as "performance management and availability" included products that measured and/or managed the efficiency and effectiveness of the various components of the IT system. Dataquest estimated the market to be $2.4 billion in 1999, with a doubling expected by 2003.[2] The market was fragmented, however. Products were differentiated

[2] Precise S-1.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

principally in two ways. On one hand, products differed widely in terms of the underlying resources and platforms with which they were associated. Some products were targeted at the performance of databases while others focused on the network and others focused on the application. All available products were specifically designed for a single resource within the IT infrastructure. Moreover, each was designed to manage the performance of a particular product within each resource category. So, a company needed one piece of software to manage the performance of its Oracle database and another to manage the performance of its SAP application suite. No vendor offered a product that managed the performance of the entire IT system.

The available products also differed in terms of the functionality they offered. This ranged greatly and could be roughly categorized into four groups, as follows.[3]

Monitoring: The database[4] was monitored on a regular basis and detailed performance statistics captured. At a minimum, information was provided as to whether or not it was "available" and for how long. More sophisticated products recorded the total time that the system spent on each calling application. Further, this time was often decomposed into (i) the time spent *using* the system resources (CPU, memory, and input/output, for example) and (ii) the time spent *waiting for* these resources. These performance data were sometimes stored to create a historical record for benchmarking purposes.

Detection: The data were analyzed for possible evidence of problems. This evidence may have been a value that crossed some pre-determined threshold or deviated substantially from some historical average. For example, if it was common for a database to take 10 seconds on average to complete a task but on one occasion it took 20 seconds, this may have represented an underlying problem. The four most common underlying causes of application performance problems were:

1) Poorly-designed SQL statements

2) Input-Output (I/O) waits[5]

3) Suboptimal database configuration (for example, the way a data table was organized)

4) Inadequate resources (for example, storage shortages)

Improvement: The system suggested a remedy for the specific problem. For example, when a poorly designed SQL statement was detected, the system may have recommended replacement code. Note that the replacement always provided the same outcome as the original (i.e., have the same effectiveness) but did so more efficiently. In other cases, the performance management software made recommendations regarding the database itself (for example, a change in its structure). In the case of a pre-packaged application, this type of remedy was more likely than a recommended change to the SQL statements.

Analysis: The software analyzed the expected change in performance that would result were the suggested changes implemented. This was useful when the recommended change required a significant investment to implement.

---

[3] Note that these groups are ordered from least to most informative and sophisticated. Thus, it was typically the case that products that provided functionality that fell into a given group also provided all of the functionality of the "lower" groups. So, a product that did "Improvement" would also have done "Monitoring" and "Detection" as well.

[4] This describes the functionality of *database* performance availability and management software. The same description would apply to software designed to manage the other tiers as well.

[5] An I/O wait occurs when a resource is in use by another application.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

One of the largest segments of the market was for software to manage the performance of Oracle databases. The three biggest players in this market were Oracle itself, BMC Software, and Quest Software. Oracle offered the "Oracle Optimizer" as part of its basic database package, which built some performance monitoring functionality into SQL itself. This helped to execute SQL statements it received in the most efficient fashion. In addition, Oracle also offered an add-on package to provide some measures to show how efficiently an application ran.

BMC Software was a large independent software developer with a wide range of products and revenues of $1.7 billion in its fiscal year 2000. BMC's "Patrol" product line was targeted at the performance management and availability market. The product was positioned as providing superior monitoring functionality. Thus, it provided timely and accurate data about the availability of the database.

BMC's revenues from this line had grown in recent years:

Year	Total BMC License Revenue	Patrol License Revenue	% of Total
1998	$ 658	$ 99	15%
1999	907	127	14%
2000	1,719	201	17%

(dollars in millions)

Quest Software, founded in 1987, was booking revenues at an approximate $150 million per year rate in mid–2000. It offered around 25 different products for use in the availability and performance management space across a range of platforms. Most relevant to Precise was a set of five products—Foglight, I-watch, Spotlight, StorageXpert, and QuestCentral—which each offered the Oracle database user some functions ranging from monitoring to detection. Like all of Quest's products, these applications were essentially stand-alone products and were not integrated with each other in any meaningful way.

Sales

In 1999, nearly every company in the industrialized world spent a significant amount of money every year on software purchases. These purchases ranged from $99 for an application running on a personal computer to print mailing labels to more than $10 million for integrated enterprise-wide applications running on both mainframe and distributed computing environments. As a result of this broad range, the process of buying software differed greatly, even within a given company, across applications.

Sales Channels: The three most common channels for distributing commercial software applications were direct sales, value-added resellers (VARs) and systems integrators. While the latter two were similar in the sense that they represented a third-party approach to selling, they differed in significant respects. VARs tended to be characterized by a "vertical" focus, specializing in sales to specific industries. Their strategy was often based on a deep understanding of customer's needs and an ability to deliver the right solution for a given problem. They tended to sell not only software but also services such as consulting, support and custom software development. Systems integrators, on the other hand, purchased software and hardware from multiple sources, combined the elements, and sold the integrated systems to their customers. On average, VARs and systems integrators earned a margin of about 30-35% on the sale of the software they purchased from other developers. Both VARs and systems integrators tended to focus on mature products for which demand had

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

already been established and value was known and understood. It was very difficult to launch a new product through these channels.

Besides VARs and systems integrators, a third approach to indirect software sales was to enter into Original Equipment Manufacturer (OEM) agreements. In these cases, the software was sold to another company, either a hardware or a software company usually, which then included it as an integrated part of its end product. This was less common than the use of VARs and integrators but was popular in some small niche areas.

While most software developers employed selling strategies that incorporated elements of each of these channels, direct selling was generally the preferred method for higher-priced applications and those for which the value proposition and sales process were more complex. BMC utilized primarily a direct sales force for most of its products, including the Patrol product line. Similarly, Quest sold its products through a proprietary sales force of over 300 sales reps.

Sales Cycle: The sale of large software systems tended to take a long time from initial contact to closing. On average, such systems took six to 12 months to sell but there was a great deal of variance. More complicated and expensive systems could take over a year to sell while simple packages could be sold in a single meeting or phone call. These long sales cycles resulted primarily from two factors: First, the products tended to be complicated and it therefore took significant time for the customer to fully understand the benefits of a product. Also, there were often many people involved in the purchase decision. It was quite common for software systems to benefit multiple business units or departments within a single company. Moreover, in addition to these "application owners," the client's IT department often played an important role in the purchase process. This added significant time to the selling process since these parties sometimes had different objectives.

Performance management software was average in this regard. On one hand, their benefits were fairly complicated and difficult to explain. On the other, the decision-makers were fairly concentrated in the organization.

Pricing Structure: A buyer typically paid a one-time license fee for software. This granted them the right to use the software in perpetuity. Recurring revenue was generated from most customers via annual maintenance and service contracts. These annual contracts, priced typically at 15% to 20% of the one-time licensing fee, entitled the user to any upgrades of the software as well as to ongoing technical support. Additional revenue from existing customers was also available via product upgrades and cross-selling other products to satisfied customers.

Precise Software Solutions

In 1996, Precise introduced its first product into the performance management market called Precise/SQL. This software product was designed to manage the performance of applications utilizing Oracle databases. By 1999, this single offering had expanded into a suite of performance management tools for Oracle databases and other software applications. With these products, Precise had built a strong reputation as a developer of high quality products that delivered what they promised. Precise/SQL remained the company's core offering. In 1999, revenues from Precise/SQL accounted for 86% of all of Precise's software licensing fees. See **Exhibit 2** for income statements and **Exhibit 3** for a graphical depiction of the functionality delivered by Precise's core products.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Precise/SQL

Between 1995 and 1997, while Precise/SQL was being developed, most of the Oracle-based applications at which the product was targeted were developed either by in-house IT groups or by contract developers building proprietary applications for a firm's own use. As a result, many of the performance problems that users experienced were due to poorly written code. Precise/SQL was targeted at the users of these applications with the promise of finding and fixing underlying problems. The results were at times dramatic. One client, a bank running a pre-packaged application, typically experienced an 18-hour run time. When the application was optimized using Precise/SQL, the run time decreased to 2 hours. Similarly, a health insurer's system was able to reduce its 18 seconds average health claim processing time to less than ½ second.[6]

Precise conducted a survey of ten Precise/SQL clients drawn from a range of different businesses in order to assess the product's impact. The three main benefits to these clients was found to be savings associated with DBA productivity, savings due to the deferral of hardware upgrades and savings in end user productivity. See **Exhibit 4** for specific quantitative findings. Based on these data, the firm was able to interview a prospective client and generate an expected ROI for the purchase of Precise/SQL.

The average price paid for a Precise/SQL license had historically been between $15,000 and $25,000. The price varied greatly across customers and depended on many factors. In 1997, Precise instituted a pricing policy that charged more to install these products on machines that were more powerful. They based this decision on the idea that higher-powered computing environments benefited more from performance management. So, for example, a Precise/SQL license to run on a Sun Ultra Enterprise 10000 might have listed at $50,000 while the same software to run on a Sun SPARC 20, a much less-powerful machine, might have listed at $12,500. Also, since workstations very often had multiple CPUs, Precise charged a "CPU uplift" which was a 15% increase in list for every additional CPU over a designated number. Many customers did not, however, pay the "list" price. As was typical of software developers, Precise often offered discounts. On average, product sales were closed at prices approximately 25% lower than the list price.

Other Products

In late 1997, Shimon Alon joined the company as President and CEO. It was apparent to him that there was a trend toward more pre-packaged applications, particularly those from SAP and Oracle, and away from the proprietary applications towards which Precise/SQL had been targeted. This was an important development for Precise because users were generally not enthusiastic about altering the code in pre-packaged applications they had just paid millions of dollars to buy and deploy. To target this new segment, Precise developed Precise/Interpoint. Interpoint was built to monitor the database efficiency of large Enterprise Resource Planning (ERP) applications, particularly those offered by Oracle and SAP. The product was priced similarly to SQL.

In 1997, the company also introduced Precise/Pulse which continuously monitored the underlying database. Pulse compared real-time performance statistics to either historical benchmarks or user-specified thresholds and automatically generated alerts to system administrators if the performance statistics moved out-of-bounds. Precise further extended into the domain of storage devices with Presto which monitored the performance of EMC storage products. This product was sold on an OEM basis by EMC as an add-on to its own storage systems.

[6] Documented in Precise Software Case Studies "Crestar Bank" and "Alberta Blue Cross," 1999.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Precise's Sales Strategy

Sales Force

Precise sold its products, other than Presto, through a dual-channel distribution system. The North American sales force was comprised of 17 account executives. Each was paid a salary of $75,000 and a commission according to the following schedule:

Net License Sales[7]	Commission Rate[8]
0$-$400,000	5%
$399,999-$799,999	6%
$800,000-$1,199,999	7%
$1,200,000-$1,599,999	8%
>$1,599,999	9%

The average rep had sales of $800,000 annually and thus earned $120,000. The highest paid rep made about $300,000. Both of these numbers were somewhat below average for commercial software sales forces.

Each salesperson was trained to sell all of the company's products, other than Presto. Alon felt strongly that this approach ensured the rep would be able to spend the time necessary to understand the customer's needs and be in a position to offer the right solution. As a result, Precise's reps were often considered by many DBAs to be valuable partners in solving important database problems. The sales cycle typically ranged from one to four months.

Internationally, the company sold through distributors (both VARs and systems integrators) in most countries.[9] In 1999, 64% of the company's revenues were in North America and Latin America. About 55% of its revenues came through the direct channel and 45% came through resellers. At the end of 1999, Precise had a blue-chip client list of about 400 companies including Amazon, Boeing, Ford, General Electric, Southwestern Bell, and Staples. Nearly 100% of these customers used Precise's products in conjunction with an Oracle database.

Target: The DBA

Precise's account reps generally focused most of their time calling on the DBA. In fact, many of them had developed very strong relationships with their DBA clients. The DBA was the natural target for Precise/SQL for several reasons. First, the benefits of the product disproportionately accrued to the DBA. Precise/SQL was focused on making the DBA's life easier by helping her to monitor and improve the performance of the database. Due to the close interdependence between the application and the database, this meant that Precise/SQL had a significant impact on the performance of the application as well. Precise/SQL also gave the DBA an impartial evaluation of *her* performance. If, for example, there were complaints about system performance, the DBA could point to Precise/SQL as proof that the database was not to blame.

---

[7] "Net License Sales" reflects any discount off of list price.

[8] The rate is applied to the sales in the range on the left. So, sales of $500,000 would earn the rep a 5% commission on the first $400,000 and 6% on the last $100,000

[9] They sold through a direct sales force in the U.K., Germany, France and Holland.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Most DBA's were well trained and thus very capable of recognizing the value delivered by Precise's products. The savings in terms of time and potential lost business were fairly easy to quantify and the DBAs were able to evaluate how it would fit into their operation. It was also the case that many DBA's were authorized to purchase a product in the price range of Precise/SQL or Precise/Interpoint. For products priced significantly higher, however, the DBA was rarely able to make the decision alone. Very often, for more expensive products—those significantly above $25,000—a higher-level IT executive (the Chief Information Officer (CIO) or Chief Technology Officer (CTO), for example) would need to approve the purchase.

Sales Process

The process generally began with a phone call to the DBA. Precise purchased industry lists from several vendors that provided them with the names and phone numbers of specific key contacts inside target firms. Precise looked for companies that had sales of at least $100 million and had a business model that required database-intensive applications. In this phone call, the account rep attempted to secure an appointment with the DBA during which she would demonstrate the product.

At the appointment, the account rep showed the DBA a laptop-based demonstration of the types of information that he'd be able to get with the product. In the course of the meeting, the account rep looked for two important cues that would dictate her selling approach as well as her likelihood of success. First, she tried to ascertain whether the DBA had the budgetary authority to purchase the product. If not, she sometimes asked the DBA for an introduction to his manager. However, appointments with CIOs and VPs were far more difficult to set. Moreover, even when they were able to set a meeting, the conversion ratio in these cases was far lower than those in which the DBA could make the decision. Most reps found it tough to capture the attention of senior IT managers with a product that was so narrowly focused on the database. As a result, most account reps had developed few relationships with managers at this level and many had little experience selling to them.

The other important cue that the account rep sought was the extent to which the examples she offered resonated with the DBA. Did the prospect identify with a scenario in which a CIO is screaming about the database being too slow? Did he have examples of his own in which a business unit head was looking for someone to blame for an under performing application? When the answers to these were "yes," the account rep knew that she had a sale.

Sometimes the product was purchased as a result of this initial meeting. More often, after qualifying the prospect as a serious potential buyer, the rep offered to put the product on the client's system and demonstrate with real data in real time what the product would do. This was often a very effective sales tool. About one out of every four of these "qualified trials" resulted in a sale. The most common reason for a client not purchasing the product was budgetary: they did not have the money to purchase the product. When faced with this, the account rep attempted to determine the budget cycle to ensure that she would make contact again when the budget was up for discussion.

The Insight Opportunity

In February 1999, Precise conducted a survey of its customers. According to General Manager Steve Campbell,

> "We did this survey pretty regularly, asking a different set of questions each time, just to keep in touch with our growing customer base. That time, though, I was skimming the responses and one really stood out. We asked 'Which metrics are important indicators of your

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

system's effectiveness?' and nearly every respondent, I think it was about 95%, said *end-to-end response time*."

Referring to **Figure A**, "end-to-end response time" meant the *total* time that elapsed from the point at which a query left the user's computer to the receipt of an "answer." Returning to the CRM example, this was how long it took from the time the salesperson entered the name of the company in which she was interested until she saw the data on her screen. In practice, this could have taken anywhere from a few seconds to minutes. The actual timing depended on the performance of each of the system's tiers. A longer-than-expected response time could have been due to problems in any (and all) of the tiers. Most importantly, any delay was certain to alienate end-users of the system.

To ensure the system ran effectively and efficiently—in this example, to be able to deliver critical information to salespeople quickly—the IT group needed to know at least two things. First, it needed to know if and when a problem *existed*. Were there a substantial number of users who were consistently waiting a long time for information? Second, it needed to know the *source* of the problem. Was it the communication network, the web server (which received the request from the field), the CRM application or the database in which the information was stored? No product on the market, including Precise's products, delivered on either of these needs. All of the available products focused on the performance of *each* of the components of the system but nothing told the IT manager the *whole system* was under performing, much less *why* it might have been doing so.

The best the IT group could do was to work from anecdotal information. For example, they might have responded to enough complaints from salespeople about slow response times associated with the CRM system or they might have heard from customers about excessive delays in "checking out" from an online store. They may have heard complaints from the accounting department that their end-of-quarter data needs took far too long to process. These data might have *suggested* that there was a problem. However, the IT staff was effectively powerless to identify the specific source of the problems. Very often, this resulted in a phenomenon that Precise referred to as "blamestorming" in which the various IT constituencies denied any responsibility for the supposed problem and, instead, pointed fingers at others. The DBAs, for example, might argue that the Web server was simply too slow in processing the requests from the field while the webmaster would argue that the application was simply too slow for the volume of business they were handling. As a result, slow end-to-end response times were often undetected and, when detected, often went unaddressed.

Alon felt that the feedback in this survey was too important to ignore.

"This reinforced my belief that Precise should be positioned as a provider of *solutions*, not products. At that point, we had a line of products—very good ones, at that—that provided a great deal of value but never solved the customer's real problem, which was reducing the time it took for the system to do what it was supposed to do."

A true end-to-end product held a great deal of appeal for Alon for several reasons: the market seemed to be quite large, there was no current direct competition, it would allow Precise to diversify its customer base, and compared with Precise's current offerings it would appear to have a substantial value proposition. This latter point was of particular interest to Alon as his immediate goal was to build Precise into a $100 million company.

The Market for an End-to-End Performance Management Solution

Precise believed the principal demand for an end-to-end performance management tool would come from firms using enterprise-wide applications. These offered an attractive opportunity for

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

several reasons. First, the investment into enterprise applications like ERP[10], CRM or supply chain management was often very large, $10 million and above. Moreover, these were just the "hard costs" associated with the application's purchase. Years of person-hours were also invested into their deployment as well as the training necessary to run them efficiently and effectively. After an investment of this magnitude, any sort of subpar performance was both disappointing and embarrassing to the internal champions of the purchase. Thus, it was thought that it would be somewhat easier to make the benefits of an end-to-end performance management product tangible and understandable. As well, the hallmark of most enterprise applications was the fact that users were often distributed across many locations, often separated by many thousands of miles. Moreover, the applications tended to be highly data intensive. All of these factors seem to match well with the core of the Insight value proposition.

Online retailers whose sales depended critically on the ability to deliver information quickly to the user's browser represented another segment that seemed to offer promise for this new product. To companies engaging in e-commerce, Insight offered the promise of never violating the sacred "eight-second rule." While this "rule" was of unknown origin and questionable veracity, according to Zona Research, "it is now widely believed that if users cannot boot up a Web page within a mere eight seconds, they may be at some risk of taking their business to another Internet destination."[11]

Precise/SQL and Precise/Interpoint faced direct competition from Quest, BMC and, to a lesser extent, Oracle itself. However, as of 1999, none of these companies offered an end-to-end solution. By moving quickly and being first to market, Alon felt that Precise could establish itself as *the* company in the end-to-end performance management space.

Decision Time: July 22, 1999 7:30 AM

As agreed, Alon, Nye, Ratner and Campbell met for an early breakfast the next morning. As the waiter walked away from the table with their orders, Alon spoke first, "My friends, the clock is ticking. OpenWorld is next September and I think we'd be *meshuganah*, crazy, to miss it. What do you think? Can we do it?"

Alon felt strongly that, in order to make a splash, Precise needed to be able to introduce a GA ("generally available") version of Insight at Oracle's annual OpenWorld conference in San Francisco in September 2000, just 14 months away. Otherwise, they would have to wait another full year until OpenWorld 2001 in order to have that many motivated and qualified prospects in the room at one time.

Campbell spoke first. "Look, I think we all know that this opportunity is massive, just enormous, so there's no question that we should pursue it." The other three nodded their heads in assent. "My concern is just that we get it right. To be honest, I think we should take our time. This thing has so many moving parts. If we have to wait until OpenWorld 2001, then so be it."

Nye stepped in, "Let's walk through the issues first and *then* decide. As I see it, there are really three key decisions to make. First, what should the product look like? Or, maybe, what *can* it look

[10] Enterprise Resource Planning applications were actually suites of applications that integrated the various functions in the firm. So, for example, most ERP systems had components dedicated to finance, accounting, human resources, manufacturing, supply chain and marketing.

[11] Zona Research White Paper, "The Economic Impacts of Unacceptable Web Site Download Speeds." 1999.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

like by OpenWorld 2000? Then, how are we going to sell it and how are do price it? Let's just take them one by one and see where we land."

As a dyed-in-the-wool optimist, Alon always believed in shooting high. "What do you mean, what *can* it look like?" He turned to Ratner. "Aki, we'll have no problem delivering a full-functionality end-to-end product by September 2000, right?"

Ratner hated being the bearer of bad news, "Shimon, we've been through this a couple of times already. If, by 'full-functionality,' you mean from Monitor to Analyze—like we have with SQL —it's very unlikely, in fact it's impossible. If you mean a GA end-to-end product that does Monitoring, now *perhaps* that's a possibility."

Both Precise/SQL and Precise/Interpoint offered the full range of functionality from Monitor to Analyze (see **Exhibit 3**). However, the complex multi-tier environment in which Insight would exist made the development process far more complicated. They could not simply "port" the existing system over to the end-to-end environment. This was a new product from the ground up. Moreover, developing an end-to-end product required a skill set that was very different from Precise's current capabilities. They had built their software development team around expertise in database environments. Yet, at least for their initial approach at the end-to-end product, they believed that experience in a networking environment would be crucial. They had already hired an entirely new team of developers, led by Shay Naes who was well known in the industry as one of the developers behind the RMON standard.[12]

Ratner went on. "My best guess is that it's going to take us somewhere between six and nine months to get a really basic product out there. Purely monitoring, nothing else. No nice interface. No application-specific bells and whistles. And remember, this is just an alpha version."

In order to estimate how long it would take to produce a GA version, they'd need to add the fact that an alpha-test and a beta-test would each take between 1 and 3 months to complete and assess. These didn't include the time it would take to recruit participants in each of the test phases. Nor, of course, did these estimates account for the common situation in which major problems arose in one or both phases that could require significant development effort to resolve.

Campbell knew that this wasn't even the worst of it, "Aki, what if we want 'Detect' capability?"

Ratner thought for a moment. "A year, maybe 18 months."

Campbell pressed further. "And a full product, something that we'd be proud to put next to SQL. Nice user interface, etcetera?"

"Two years easy."

At Ratner's answer, all eyes turned back to Alon who appeared unswayed by this information.

"This is what I'm talking about," said Campbell. "If we shoot for a fully-functional product and can't deliver by 2000, then we've got nothing. If the product can only Monitor, then we go to market with a lightweight product and maybe give Quest a chance to catch up. Let's keep it under wraps, talk about the latest enhancements to SQL at OpenWorld and really 'wow' them next year."

[12] RMON was short for "remote monitoring" and was a standard protocol for monitoring and analyzing a network.

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Alon didn't like where this was heading. "What was the second issue you wanted to talk about, Ben? Sales strategy? Aren't we just going to do what we've been doing? It's worked for all of our other products, why change?"

"I'm not saying we *should* change, I'm just raising the issue. Do we have the right people? Should we just give this product to our guys and tell them to sell it in addition to what they already have? They're running in ten directions at once. It's not clear."

Alon looked down at his watch. It was already 9:00. "OK, we're running out of time. What do you think, I pay you guys to sit around and drink coffee?" Alon joked. "I'm sure you guys can solve these issues. Ben, tell us what the final issue is and then let's head back. We can think this stuff over one more time and meet for lunch in my office. We will not leave until we come up with a decision."

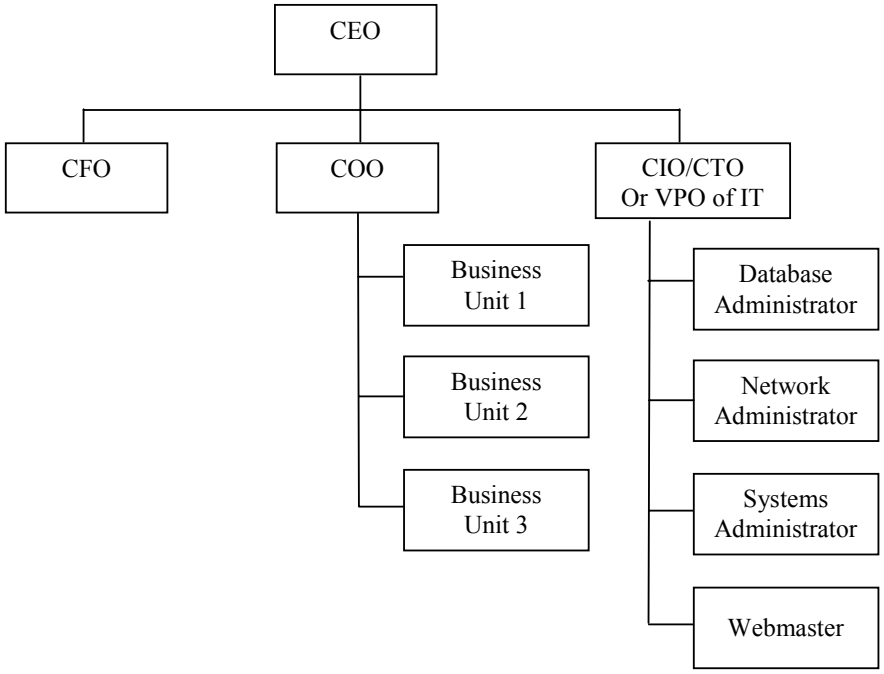
Nye finished, "OK, the final thing is the pricing issue. But I don't want to minimize the complexity of the sales issues. We need to think about incentives as well. What should the commissions look like? As for pricing, there are two questions, 'how much' and 'how?' Also, how much authority should we give the reps in setting the price?"

Alon had not hidden the fact that he saw Insight as a key factor in his drive to make Precise a $100 million company. He had in mind the number $250,000. There was little question that a fully functional end-to-end product would deliver this kind of value. He was well aware, however, that it might take some time and several iterations for the product to warrant such a price tag. Certainly, a limited-functionality version that might be announced at OpenWorld 2000 would not come close to doing so.

"If anything," said Alon, as he put his hands on the table and stood up, signaling the end of the meeting, "we should price it a little too *high*."

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Exhibit 1: Typical IT Organization



Source: Author

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

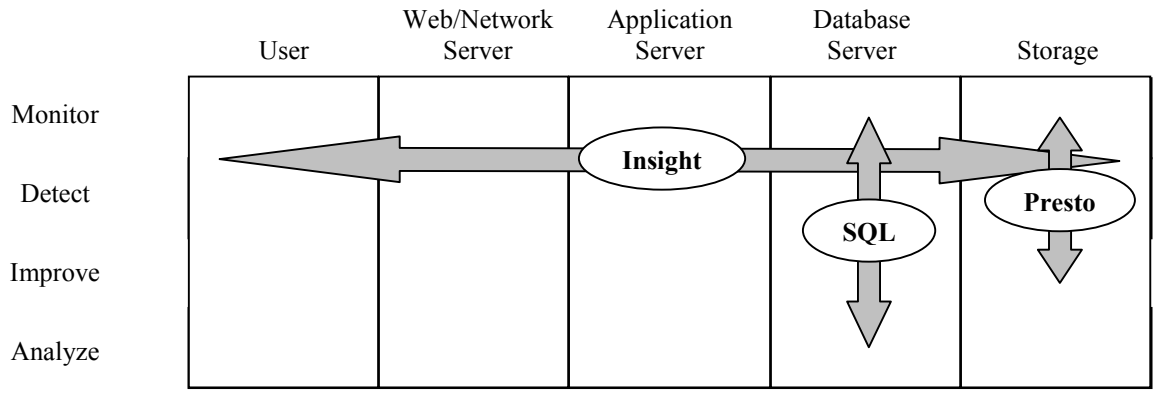
Exhibit 2: Income Statements

	1996	1997	1998	1999
Revenues				
Software License	954	2,382	5,331	9,770
Services	272	406	858	1,844
Total Revenue	1,226	2,788	6,189	11,614
Cost of Revenues				
Software licenses	130	349	522	741
Services, net	54	86	198	906
Total cost of revenues	184	435	720	1,647
Gross Profit	1,042	2,353	5,469	9,967
Operating Expenses, net				
R&D	602	1,737	2,214	2,891
Sales and marketing	2,498	3,278	5,739	7,913
General and administrative	1,264	1,341	1,272	1,598
Other	0	0	300	234
Total Operating Expenses	4,364	6,356	9,525	12,636
Operating Loss	(3,322)	(4,003)	(4,056)	(2,669)

Source: Annual Report

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Exhibit 3: Some of Precise's Product Offerings



Source: Precise Software

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.

Exhibit 4 Highlights from the Precise/SQL Savings Study

Hours saved per DBA per week	9.4
Average DBA salary	$60,000
Average end user salary	$30,000
Average # of DBAs	10
Average annual hardware budget affected by Precise/SQL	$1,430,000
Hardware cost savings if postpone purchase by one year	30%
Probability of postponing hardware purchases by three months	60%
Average end-user response time per transaction prior to using Precise/SQL	15 sec
Improvement in end-user response time per transaction	25%
Average employee burden rate (as percent of salary)	33%
Number of simultaneous users	215
Average daily transactions processed by all users combined	194,000

Note: These data were collected from a set of telephone surveys of ten purchasers of Precise/SQL. The survey lasted approximately 30 minutes, had 23 questions and was implemented by a professional market research firm skilled in this area.

Source: Company documents

This document is authorized for use only by Clayton Rosenberg in Entrepreneurial Marketing - SHA1 at Hult International Business School, 2017.