APRIL 1988

ANALYSIS OF CASEPAC MARKET: POSITION, POTENTIAL, PROBLEMS, AND SOLUTIONS PHASE II: MARKET SEGMENTATION

Prepared for

ON-LINE SOFTWARE INTERNATIONAL
2 Executive Drive
Fort Lee, New Jersey 07024

Prepared by

INPUT
Suite 201
959 Route 46 East
Parsippany, New Jersey 07054

Table of Contents

Exhibits

A

Introduction

INPUT proposed to define, size, and project growth for appropriate segments of the CASE marketplace. After a fairly exhaustive search of the literature and after discussions with outside experts and within INPUT, very little agreement was found on what the appropriate and proper segmentation of the market is and on the definitions applicable to each. Even greater difficulty was encountered in attempting to find reliable indicators of the size and growth of these somewhat confusing and indistinct components of CASE.

Suffice it to say, the market is still in its infancy, very little agreement exists on any of the factors enumerated above, and a good deal of confusion exists among all categories of vendors, users, and consultants as to what rightfully belongs to CASE and how it should be logically compartmentalized.

Therefore, it is with a certain measure of trepidation that we step into this arena.

B

Definition

The term CASE is used very loosely within the industry in connection with a wide variety of application development tools, techniques, and approaches.

CASE is best defined by stating what it is not:

- It is *not* merely fourth-generation languages and code generators.

- It is *not* merely incremental improvements in retrieving information from data bases.

- It is *not* merely enhanced diagramming tools for systems analysts.

Software engineering is essentially a change in people's methods of working. The essence of CASE is the development and use of systematic strategies for the production of software that:

- Meets application requirements.
- Exploits new methods and supporting tools.
- Is available on time.
- Is available within budget.
- Allows for easy maintenance and support.

CASE is more than just the application of sophisticated tools. It implies an evolution from traditional manual methods of software production and inflexible, unprofessional approaches toward managing and organizing a software engineering function.

To define CASE in terms of application development tools (ADTs): it is a total development environment of RDBMS, a data dictionary or repository, fourth-generation languages (4GLs), two-dimensional graphical programming languages, and an integrated workbench covering all phases of the development life cycle.

CASE products ideally allow an engineer to build a graphical image of the desired application on a workstation and then automatically generate the code to produce it.

Ideally under CASE the process of building applications becomes a pure specification exercise, in which engineers never write code but simply use tools to bring together dictionary code and data structures.

An effectively implemented CASE environment is *the* most promising answer to the problems of the software industry and the organizations whose mission-critical applications it serves.

C

Evolution Toward CASE

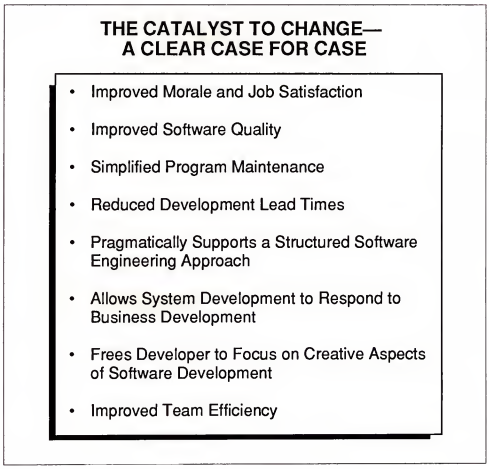
Recent INPUT user research revealed two strategic concerns for data processing that are accelerating the evolution toward CASE approaches to software development.

First, users face the challenge of systems integration, i.e., building complex on-line systems that seek to link disparate decentralized systems. Coupled with the growth of end-user computing, this is causing increasing backlogs in 45% of our research sample organizations.

Second, users are seeking productivity gains from all corporate assets—including capital investments, human resources, and information assets—in order to meet the challenges of complexity and competitiveness in the business environment.

An effectively implemented CASE approach offers a solution to these problems at a macro level that is impacting data processing developers. The benefits from a CASE approach are illustrated in Exhibit 1.

EXHIBIT 1



On the other hand, the momentum behind the effective implementation of CASE has been slowed by several factors:

- Cultural resistance to changing working practices.
- Lack of management commitment to training and structured methods.
- Difficulties in justifying investment owing to the lack of standards for productivity measurement and the initial unsuitability of a CASE approach to relatively immature data processing environments.

Nevertheless, INPUT sees a window of opportunity for vendors of advanced CASE products in the late 1980s. The market will evolve as CASE becomes the standard industry method of building and maintaining software.

D

Structure of CASE Market

The structure and evolution of the CASE market is depicted in Exhibit 2.

A complete CASE System is essentially an integrated software technology that provides an automated engineering discipline for software development, maintenance, and project management. It includes structured methodologies and integrated sets of tools that have a common user interface and automate all the phases of the software lifecycle in a common computer environment.

In terms of ADTs, complete CASE systems provide a total development environment of RDBMS, a central data dictionary or repository, 4GLs, and two-dimensional graphical programming languages. Complete CASE systems allow the software engineers to build a graphical image of the derived application on his/her workstation and then automatically generate the code to produce it.

EXHIBIT 2

TOTAL MARKET STRUCTURE AND EVOLUTION

Application Development Tools (ADTs)

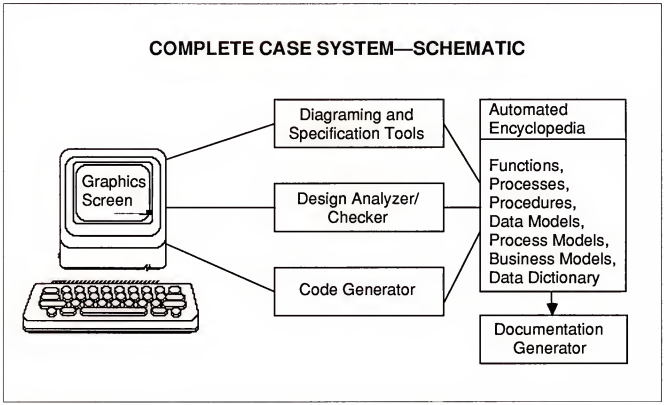
DBMS	Program Development Tools (PDT)	Languages	CASE Tools/Tool Kits
• DBMS	• Application Generators	• 4 GLs	• Methodology Companions
• Data Dictionaries	• Documentation Generators	• Assemblers/ Compilers	• Tool Kits

Complete CASE Systems

- Workbenches
- Full-Life-Cycle Environments

The characteristics of complete CASE systems are illustrated in a schematic given as Exhibit 3.

EXHIBIT 3



Absolutely central to any workable CASE system is the automated encyclopedia or repository, which holds all the information needed to create, modify, and evolve software systems. That is:

- Information on problem to be solved.
- Problem domain.
- Emerging solution.
- Software process being used.
- Project resources and history.
- Organizational context.

The central repository then provides the basis for:

- CASE tools integration, including foreign tools.
- System specification consistency and integrity.
- System information sharing.

- Document standardization.
- System documentation generation.
- Code generation.
- Software reusability.
- Project management and control.

In that sense, On-Line Software International is directly on target with CasePac, and its central theme and unifying element is the data dictionary/repository. There are, however, other major components of the CASE marketplace that not only can be viewed somewhat independently but are actually being sold separately. For the purposes of this study we have adopted the taxonomy developed by *CASE Outlook*, a monthly newsletter published in Oregon with a subscriber base in excess of 1,000. *CASE Outlook* breaks the CASE marketplace into four components:

1. Analysis, design, and prototyping tools. All the methodologies, workbenches, and in the case of a mainframe-based product, on-line graphical specification, modeling, and planning tools fall into this category.

2. Project management, configuration management, repository/dictionary, and DBMS tools. This is a rather mixed category that we would have preferred to be more finely divided but, as will be seen later, with the help of some assumptions can be appropriately dealt with.

3. The third category is code generators and 4GLs. Once again we would have preferred *CASE Outlook* to have separated the two, and we will do so.

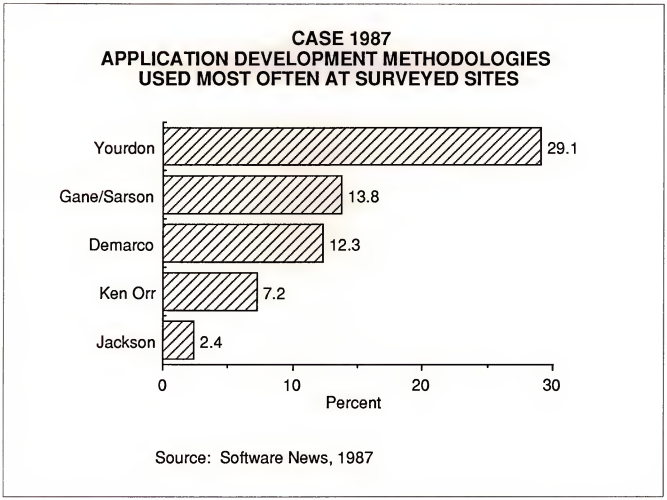
4. The last category is another catch-all defined as language-sensitive editors, compilers, debuggers, and testing tools. If we are successful in removing compilers, as such, we can create a useful CASE category here too.

E

Product Requirements

Product requirements for the first category are clearly a friendly and highly pictorial graphics front end, with both data and process diagraming and modeling facilities. Support of one, preferably several, of the popular methodologies is widely featured. Exhibit 4 shows the relative popularity of the five most common methodologies reported from a 1987 survey conducted by *Software News*.

EXHIBIT 4



The fundamental idea, obviously, is to make the front-end product as comprehensive and paperless as possible, so that the analyst can do virtually all of his/her work at the terminal or workstation without resorting to any unnecessary computer or paper and pencil activity. The net result is supposed to be a total system and data base design that has been checked for logical consistency and subjected to a high degree of simulation and prototyping sufficient to give the designer and the end user a very clear idea of how the system will work and how well it will work, and finally a design that has been deposited in a comprehensive and useable state suitable for entry into the next stage of the development process.

The second grouping requires a repository for all of the output from stage one and includes, or is intimately associated with, a data dictionary that controls the structure of the data base or data bases that will be used by the systems specified and documented in the repository. A companion to the dictionary is all of the customary data administration and data base administration tools that make the creation, maintenance, and interfacing of programs to the data base manageable. It seems to matter very little whether the main feature at this level is thought of as a dictionary, repository, or encyclopedia, so long as it provides the aforementioned information and functions and provides a point of common reference for the developers and their programs.

Clearly, then, the question of an "active" dictionary comes into play here. Most of the dictionaries in use today are not active in the sense that production programs reference them at run time or whose data references are altered by changes in file definitions in the dictionary automatically by some other means. However, INPUT perceives an industry trend toward active dictionaries and an upwelling of user demand for that capability. Current products such as MSP, TI's IEF, and a new mainframe-oriented product from Knowledgeware called Knowledge Coordinator and Encyclopedia have active dictionaries, and it is our belief that most of the users who forecast implementation of full-life-cycle CASE systems view them as having active rather than passive dictionary capabilities.

In the third category (eliminating 4GLs, which are outside the scope of this study) are code generators, which, to be successful, clearly have to generate solid, efficient structured COBOL code for the IBM environment and, to be broadly competitive across the whole firmament, for the DEC and Tandem platforms as well. Pansophic, for example, has telegraphed its intentions to target a variety of hardware platforms and popular data base management system environments. Configuration management, version control, good documentation, and intelligent or selective maintenance facilities are all very important in the applications generator environment.

The fourth category, excluding compilers themselves, consists of all of the editing, debugging, and testing tools required to create, test, and maintain large and complex programs. It is here that the "rubber meets the road," or shall we say leaves the road, in getting a system airborne, and ideally should tie back to the tools and controls inherent in all three of the prior stages to facilitate quick and efficient recycling through those stages for testing, bug correction, maintenance, and enhancement.

Every sign post in the industry points to the demand for full-life-cycle seamless integration throughout all four stages defined above. Many users demand it now. Others view it as essential in the future. Survey results in Phase I of this study made this abundantly clear. And, finally, logic simply demands that elaborate CASE systems and components that go to the lengths described cannot possibly be left dangling in mid air, unable to work with each other.

Most of the present CASE tools in the marketplace reside on PCs and are targeted to the IBM mainframe environment, with a minority residing on and targeting DEC VAX. In the future we are likely to see CASE systems running on and targeted for a wide variety of mainframes, minicomputers, workstations, and PCs. However, the "big money" will remain in the large mainframe sites, 70% or more of which are IBM: these are where the largest need is and where the big money to be spent is. It is a reasonable bet that upward of three-quarters of all of the dollars to be spent on CASE technology over the next few years will come out of the very large IBM mainframe-based accounts, the top two or three thousand of whom will be running much of their work under DB2 and, in all probability, the rest under SQL/DS.

If we look at a total CASE marketplace of $1.63 billion, as our projections in the next section would suggest (see Exhibit 8), and assume that 3/4 is IBM-based, about 1/2 of that DB-2 based in dollar terms, we winnow to a marketplace of over $600 million available to a product like CasePac, which confines itself to the very large IBM MVS/XA DB2 environment. There is a further question about CasePac's full integration and the presence or absence of a workstation-based front end, but assuming the very best for CasePac would put it in a market whose total worldwide potential is very substantial.

F

Factors Influencing the Acceptance of CASE

INPUT's recent and extensive European interviews revealed the following:

Shortage of qualified staff was mentioned as a key inhibitor, and each site interviewed typically had at least five technical staff vacancies.

A further major inhibitor and problem for data processing developers is the rapid increase in demand for on-line information support systems that reflect the trend toward end-user computing and the growing strategic importance of IS to corporate success.

Another inhibitor frequently mentioned was the increasing accountability of data processing to respond to changes in the business environment. IS is becoming a competitive weapon in many organizations, and INPUT's research reveals that data processing departments do not have sufficient resources and flexibility to adjust.

In many of the establishments interviewed by INPUT, expensive development resources were being tied up in the maintenance of existing systems (especially IBM sites) that had been developed using "seat of the pants" analysis and design techniques and hand coded using traditional technologies (e.g., COBOL). Documentation of these old, but strategically important systems, is also poor—a factor that exacerbates the problem.

In the current data processing environment as much as 70% of the cost associated with applications software is in maintenance, a key opportunity for proponents of the CASE approach.

Several users adopting a CASE solution pointed to significant difficulties with the practical implementation of the new breed of intelligent software tools.

CASE implementation effectively means fundamentally changing the way people work; implementors of software automation have found that, as in factory automation, there is a considerable amount of resistance to change in the data processing environment.

Users pointed to the high cost of training that, in the short term, is greater than the productivity and quality benefits of the software.

The "skills gap" in terms of availability of trained software engineers experienced in implementing systems using CASE techniques is so large that many organizations have resorted to traditional methods, i.e., what they know and understand. Data processing managers are notoriously conservative when it comes to changing working methods.

Inhibitions to the adoption of CASE are characterized in Exhibit 5 below:

Many managers have perceived CASE tools as a panacea for the problems of system development. Innovative users have found that this is not the case.

The major inhibitor of market development is the adaptation of personnel to change. Cultural attitudes that tend to regard training as an inappropriate activity are a major barrier to CASE implementation.

The key education problem is not in learning the techniques but in the application of techniques. Consequently, successful CASE tool suppliers are those that can differentiate on quality of service and offer thorough

EXHIBIT 5

USERS' PERCEPTION OF PROBLEMS ASSOCIATED WITH THE CASE APPROACH

(Total Western Europe-Synthesis of Most Frequently Mentioned/Valuable Comments)

Rank	Comment	No. of Mentions
1	Staff Education and Training	16
2	Adaptation of Personnel to Change	8
3	Excessive Rigidity	7
4	Increases Analysis Time	5
5	Increases Development Time—Short Term	4
6	Lack of Standards	3
7=	Cost	2
7=	Maintenance/Upgradeability	2

implementation support, not only in the application of CASE products but also in the management style and approach toward maintaining productivity and quality in systems development.

A further major stumbling block for the CASE approach is the implication of change in the traditional roles of analysts and programmers, coupled with change in the organization and structure of data processing development.

The software engineer of the future will be a business analyst who spends more time working with end users as a consultant and creative problem solver with experience and knowledge of the commercial environment. Current traditional approaches toward systems development demand highly specialized staff grouped into specialist teams with little interchange of staff between them.

The organizational challenge and opportunity faced by the CASE approach is to share staff and skills between development projects, thereby

reducing inflexibility, improving internal skills development, improving job satisfaction, and achieving the objective of developing high-quality, easily maintainable systems, on-time and within budget.

G

Revenue Forecasts

Exhibit 6 shows *CASE Outlook's* estimate of revenues in each of those aforementioned categories from 1985 through 1987 and its forecast for 1990. As can be seen, *CASE Outlook* is forecasting robust growth in all categories, with only the fourth, which includes compilers, in the same range as INPUT's overall Systems Software forecast for comparable periods. All the rest exceed INPUT's growth expectations for Systems Software and, indeed, for CASE in Europe, which is expected to grow in total as a category at 36% over the period 1987-1991.

EXHIBIT 6

CASE OUTLOOK'S WORLDWIDE FORECAST

($ Millions)

Product Category	1985	1986	1987	1990	AAGR (Percent)
1. Analysis, Design, and Prototyping Tools	52	82	130	436	53
2. Project Mgmt.,Config. Mgmt., Repository/Dictionary,and DBMS Tools	48	80	140	680	70
3. Code Generators and 4GLS	120	170	235	645	40
4. Language-Sensitive Editors, Debugging, Compilers,and Testing Tools	350	450	540	1,300	30

In Exhibit 7 INPUT attempts to analyze the 1986 numbers—the last year for which anything like "actual" figures might apply—into various ranges and component breakdowns from three sources: *CASE Outlook*, IDC, and INPUT, wherever appropriate or comparable numbers can be found. As can be seen from the Exhibit, all sources appear to be in close agreement on the first category. Therefore in the next Exhibit, Exhibit 8, INPUT felt comfortable in carrying forward an $80 million 1986 estimate for category one.

EXHIBIT 7

SYNTHESIS OF 1986 ESTIMATES

Product Category	CASE Outlook	IDC	INPUT
1. Analysis, Design, and Prototyping Tools	82	78	60-80
2. Project Mgmt.,Config. Mgmt., Repository/Dictionary,and DBMS Tools	80	-	
• Project Mgmt. Portion			40
• Net			40
3. Code Generators and 4GLS	170		
• Code Generators		114	75
• 4GLs		327	
4. Editors, Compliers, Debuggers, and Testing Tools	450		
• Compilers			297
• Net			153

In the case of category two, if one accepts *CASE Outlook's* total of $80 million, and the INPUT estimate of $40 million in 1986 for software project management tool revenues, the net remainder of $40 million can be ascribed to configuration management, repository/dictionary, and other DBMS tools—and, therefore, is carried over into the second category in Exhibit 8.

EXHIBIT 8

BEST EFFORTS 5-YEAR PROJECTION*

Product Category	1986	1991	AAGR (Percent)
1. Analysis, Design, and Prototyping Tools	80	512	45
2. Config. Mgmt., Repository/Dictionary, and DBMS Tools	40	215	40
3. Code Generators	75	336	35
4. Editors, Debugging, and Testing Tools	153	568	30
TOTALS	348	1,631	37

* Pending INPUT's full market survey as part of its MAPS Program, scheduled for later in 1988.

Similarly, the data is spotty on category three, code generators, and 4GLs. *CASE Outlook* says the "total" category accounted for $170 million in revenues in 1986, whereas IDC separately projected code generators at $114 million and 4GLs of all varieties at $327 million. At this point INPUT feels that the *CASE Outlook* numbers are far too low if they, indeed, include all categories of 4GLs, and that IDC's number for code generators is probably too high. INPUT picked $75 million for an appropriate baseline for code generators of all types in 1986. This number was therefore carried over into the next Exhibit.

Category four is actually the most difficult to estimate. If we accept *CASE Outlook's* $450 million for the category and assume that two-thirds of that is compiler revenue, this yields a net of $153 million for the other components of the category, including editors, debuggers, and testing tools. Given the amount of business being done by CA, On-Line Software International, VM software, Candle, and many others with such tools, one would feel the $153 million figure is reasonable. Therefore that figure is carried forward into Exhibit 8.

In terms of growth rates, INPUT believes that *CASE Outlook's* rates are exaggerated and that more-conservative projections are in order, although the rates chosen are still considerably above the rate at which total Systems Software business is growing. Applying the chosen rates to the 1986 base yields 1991 figures as shown in the middle column of Exhibit 8.

Overall we would evaluate these projections as "best estimate" and probably quite conservative, but even in their conservative form representative of substantial markets worthy of pursuit in all four categories.

On the question of total-life-cycle products, clearly the "slice" through all four of these categories occupied by full-life-cycle products will be substantial and will absorb a considerable part of the projected revenues in each category. It is even conceivable that by 1991 virtually all of this revenue will be in context with the whole, although if one is to believe what the users in our Phase I survey told us, the components do not necessarily have to come from the same vendor, although they must be fully and "seamlessly" integrated.

As to an estimate of how much of this business will come from single-vendor, multifunction full-life-cycle products, one can only conjecture. Perhaps a reasonable guess would be 50% by 1991, perhaps more if IBM "gets its act together" with DB2 and a full supporting development environment. Still, IBM's fleeter, smaller competitors will probably always be able to field a product suite with better price performance characteristics in certain key respects, and will be able to rifle-shoot specific components within the evolving industry-standard environment.

H Important Trends

To quote from INPUT's recent study of CASE in Europe:

"Essentially, there are four major trends impacting CASE application development tools summarized in Exhibit 9.

"The trend toward distributed processing and demand for end-user computing has led to a link between RDBMs and 4GLs. All the major vendors of DBMs now offer compatible 4GLs—i.e., Software AG, ADR, IBM, Cullinet, and CINCOM.

"In the distributed processing environment 4GLs are now positioned as two-way interfaces between different systems and machines, effectively becoming a programming common denominator for prototyping new applications and filtering existing data.

EXHIBIT 9

"Traditional 4GLs are being linked with graphics front ends, code generators, data dictionaries, and integrated tool sets.

"Vendors of integrated 4GL/RDBMs products are slowly emerging as CASE companies in response to customer demand for full development and maintenance tools, such as Information Builders' plans for a full-life-cycle applications generator, and Cortex's launching of Corvision, a graphics front end, to the Application Factory 4GL, fronting DEC data bases. Mature products such as Pansophics' application generator, Telon, are being recycled via increasing life-cycle coverage, (e.g., the link with Excelerator, a design and analysis tool from INDEX Technologies).

"There is a major opportunity for independent software houses to develop CASE products. IBM's DB2 and SQL do not have an adequate data dictionary or integrated set of tools for comprehensive project support. James Martin Associates with Information Engineering Facility (IEF) and Arthur Andersen with Foundation are already attacking this gap in the IBM marketplace."

I

The Coming Shakeout

INPUT forecasts a market shakeout based on the mindboggling profusion of ADT and CASE tools in the market today and the rapid rate of new introductions.

- Players

There are at least 50 vendor hats in the U.S. CASE ring already and, according to Business Technology Research, over 40,000 "users" of CASE, which includes each workstation currently hosting a CASE package.

Exhibit 10 attempts to gather the presently fragmentary information about CASE vendors and products to illustrate the level and diversity of activity in the field.

Several vendors—such as CADRE, Index Technologies, Knowledgeware, M. Bryce & Co., McDonnell Douglas, Nastec, Polyton, and Techtronix—have sold PC-based products numbering in the thousands. CGI, Pansophic, and Sage Software are making significant headway on the "big ticket" side among large mainframe users. We believe Arthur Andersen is doing likewise, but no census numbers are available at this time, and much of AA's value package is in the form of Professional Services. TI and Manager SW have been modestly successful with their mainframe products.

INPUT strongly believes the vendors that will survive are those that can offer full-life-cycle, maintenance, project management, prototyping, and automatic code generation support linked to automated design-and-analysis methodology and documentation support.

To quote again from the aforementioned report:

"New products appearing on the CASE marketplace offer full-life-cycle coverage and address the development of commercial applications—for example, Computer Associates Programmers' Workbench or Arthur Andersen's Foundation products.

"Code generators have evolved to provide for automatic documentation and analysis capabilities for error checking and include a data dictionary or central repository. For example, CGI's PACBASE has evolved to include a COBOL generator, schematic design and analysis tools, testing

EXHIBIT 10

IMPORTANT PLAYERS IN IBM CASE MARKET

Company	Product	Type*	Typical Price $	Estimated # Installed
AGS	Multicam		90K	N/A
AMS	Lifecycle	FLC	PC 6,500-16,000 MF 100-225K	N/A N/A
Arthur Andersen	Foundation	FLC	200-250K	N/A
CADRE	Teamwork	A&D	5,000	1,000
CGI	PacBase	REP & CG	140K	500
Index Tech.	Excelerator	A&D	8,400	8,000
Knowledgeware	IEW	A&D Rep	PC 7,500 MF 100K	2,500 30
Manager SW	MSP	FLC	PC 3,000 MF 14-100K	30
M. Bryce & Assoc.	Pride	A&D	N/A	1,200
McDonnell Doug.	Prokit	A&D	7,000	2,400
Nastec	Design Aid	A&D	6,900	2,800
Promod Inc.	Promod	A&D	500-4,000	500
Pansophic	Telon	CG	200K	300
Polyton	PVCS	A&D	N/A	1,500
Sage	APS	FLC	200K	115
TI	IEF	FLC	225-250K	25
Tektronix	Integral		3.6-54K	5,000

* Types: FLC = Full-Life-Cycle
A&D = Analysis and Design
CG = Code Generator
REP = Dictionary/Repository

tools, a data dictionary, and graphics front ends. Consequently, CASE products are evolving and converging toward complete systems as the need for structured techniques and systems development complexity grows within the software industry.

"Many users have decided that CASE is not a bona-fide strategy for Systems Development until complete systems are available. It is difficult for users to evaluate products, and quite understandably they cannot justify the ad-hoc investment in individual tools.

"It is possible to isolate five criteria that categorize a product that provides a full CASE system:

- Front-end software that can pictorially represent the applications cycle.
- Prototyping software that makes use of graphics capabilities, i.e., automatic screen generation, data, and system modeling.
- A central, commonly accessible data dictionary and project data base.
- An efficient code generator.
- A methodology or engineering approach that needs to be flexible in the number of logical design approaches supported, e.g., Yourdon, Gane/Sarson, Chen, Orr, Jackson, etc.

"INPUT has been able to identify over fifty U.S.-owned players in the CASE tool market. The vast majority of their products address automation in the design-and-analysis or coding phases of the life cycle.

"The challenge of the future is to offer full-life-cycle support —a truly integrated CASE approach."

J

Summary

The CASE market is diverse, fragmented, and confused in its terminology and approaches, but is addressing a very real problem and a very large opportunity.

One way of classifying CASE tools is to sort them into four basic categories:

- Analysis, design, and prototyping tools, which are mainly PC "workbench"-based and appear to be gaining rapidly in sales.

- Configuration management, repository/dictionary, and DBMS tools. The repository segment of this group, especially, appears to be growing rapidly and benefits from a more substantial price tag than the prior category.

- Code Generators, which are hefty in price and appear to be doing nicely in the marketplace.

- Editors, debuggers, and testing tools: possibly the largest CASE submarket at present, but one that is destined to be surpassed in the early 1990s by its faster-growing counterparts, above.

One very important fact derives from user interviews in Phase I of the current study, from other closely related INPUT user interviews, and from a review of current literature: the full-life-cycle approach is the one most favored by users in the long run. This means that the entire process of creation, modeling, testing, installation, and maintenance of applications must fall under CASE purview. Interfaces among the various stages and functions are expected to be "seamless."

Nevertheless, the present submarkets are each individually large and growing at projected rates of from 30-45% per annum. By 1991 our preliminary projection shows the total worldwide CASE market at over $1.6 billion and still growing rapidly. Other consultants have projected even larger numbers, and ours are subject to considerable refinement and revision based on a large-scale market analysis planned by INPUT later this year.

Certain important questions remain to be addressed in the third and final phase of this study: What is holding back CasePac? Is an MVS/XA/DB2-only approach viable? What will true full-life-cycle status do for CasePac? Will CasePac have to support a PC-Workbench front end in order to be successful?

APRIL 1988

ANALYSIS OF CASEPAC MARKET: POSITION, POTENTIAL, PROBLEMS, AND SOLUTIONS PHASE III: CONCLUSIONS

Prepared for

ON-LINE SOFTWARE INTERNATIONAL
2 Executive Drive
Fort Lee, New Jersey 07024

Prepared by

INPUT
Suite 201
959 Route 46 East
Parsippany, New Jersey 07054

INPUT®
1280 Villa Street, Mountain View, California 94041-1194 (415) 961-3300

Table of Contents

Exhibits

A

Introduction

In Phase I of the present investigation INPUT presented the results of a survey of IBM DB2 mainframe computer users chosen from a list provided by On-Line Software International of those who have had some exposure to CasePac. Thirty-five user interviews were conducted and results exhaustively analyzed in the aforementioned report.

In Phase II, INPUT defined the size and projected growth of various segments of the CASE marketplace, positioned CasePac within that framework, and indicated problems in CasePac's "fit" with market requirements.

In the third and final phase, the present report, INPUT will endeavor to integrate the findings of Phase I and II and draw conclusions from them with respect to CasePac adequacy, positioning, and future strategy. This report is in context with those of the previous phases, so the reader is encouraged to review both in detail before embarking upon the present effort.

B

Executive Overview

CasePac faces a steep climb up a narrow path since it is limited to the IBM MVS/XA/DB2 environment; absent a "workbench" front-end, active dictionary, code generation, and support for the Yourdon methodology; and selling into a new, complex, naive, and untutored market in which group decision-making will be the rule.

In the longer run, as DB2 continues to gain ascendancy and features are added to CasePac, the product could address an estimated $600-million-per year stratum of the CASE market, which we tentatively rate at $1.6 billion in total by 1991.

To reach its potential, INPUT believes CasePac will have to include an active dictionary facility, a code generator, and a front-end distributed workbench, preferably PC-based.

To remain in the marketplace, CasePac will have to improve its visibility and the receptivity of its audience. Principally, this is a marketing communications and buyer education problem. The problem could be eased by unbundling components of CasePac, particularly the dictionary and documentation repository features, and selling them separately to pinpoint special-interest audiences.

Total, massive, sweeping commitments to CASE are slow in developing in major companies. They will likely come eventually, but are now slowed by the involvement of multiple departments, levels, and interests in the decision; the inertia inherent in present development methods; and the mind-set of the people performing the work.

To the extent that the buying decision can be granularized and confined to a single, or more limited number of interests, the educating, coordinating, and selling process should proceed much more readily. This is an argument for unbundling and for adding a front-end product that can be sold as an experiment, learning tool, or a step on the way to full-scale CASE implementation.

C

The CASE Marketplace

CASE use is very much in its infancy among the population studied. Conversely, respondents forecast dramatic increases in CASE use two years from now, with a significant number projecting usage for 40% or more of their development work and 11 out of 35 projecting use of CASE for 100% of their development work. Other studies bear out an expected ground-swell of CASE adoption in the next few years.

Total CASE industry sales are extremely hard to define because of the wide differences in the way various analysts divide the marketplace. In its broadest context, as defined in Exhibit 1, INPUT sees the market at around $348 million per annum and growing to $1.6 billion per year in 1991 at a annual growth rate of 37% overall. Various other analysts with differing definitions of CASE and varying degrees of optimism show the current market at as little as $100 million per annum and as much as $450 million.

The CASE market is diverse, fragmented, and confused in its terminology and approach but is addressing a very real problem and represents a very large opportunity.

EXHIBIT 1

BEST EFFORTS 5-YEAR PROJECTION*

Product Category	1986	1991	AAGR (Percent)
1. Analysis, Design, and Prototyping Tools	80	512	45
2. Config. Mgmt., Repository/Dictionary, and DBMS Tools	40	215	40
3. Code Generators	75	336	35
4. Editors, Debugging, and Testing Tools	153	568	30
TOTALS	348	1,631	37

* Pending INPUT's full market survey as part of its MAPS Program, scheduled for later in 1988.

One way of classifying CASE tools is to sort them into four basic categories:

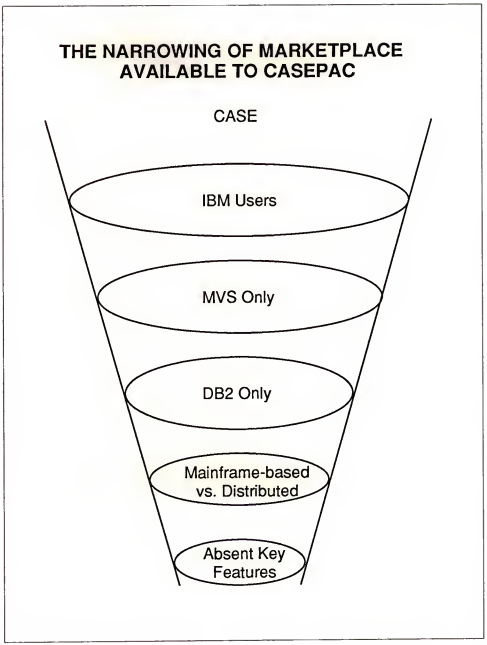
- Analysis, design, and prototyping tools, which are mainly PC "workbench"-based and appear to be gaining rapidly in sales.
- Configuration management, repository/dictionary, and DBMS tools. The repository segment of this group, especially, appears to be growing rapidly and benefits from a more substantial price tag than the prior category.
- Code Generators. These enjoy a substantial price point and appear to be doing nicely in the marketplace.
- Editors, debuggers, and testing tools: possibly the largest CASE submarket at present, but one that is destined to be surpassed in the early 1990s by its faster-growing counterparts, above.

If we look ahead at a total potential CASE marketplace of $1.63 billion and assume factoring to be 3/4 IBM-based, about 1/2 of that DB2-based in dollar terms, we winnow to a marketplace of over $600 million avail-

able to a product that confines itself to the IBM MVS/XA DB2 environment, as does CasePac.

At present, however, we believe less than three-quarters of the CASE activity is IBM-based, and only about 10% of IBM mainframe sites have DB2 (and many of those are opting for an off-line approach or want features not presently included with CasePac), so only a small proportion of the market is available to CasePac. The situation is presented pictorially in Exhibit 2.

EXHIBIT 2



Certain important questions remain to be addressed in this final phase of the study: What is holding back CasePac? Is an MVS/XA/DB2-only approach viable? What will true "full life cycle" status do for CasePac? Will CasePac have to support a PC-Workbench front-end in order to be successful?

User Survey Results:

As previously discussed in the Phase I report, among a rather large population of CasePac suspects there were no technological impediments to purchasing CasePac and no indication of significant price resistance.

Actual CASE usage among the sample population was quite low, with about half the population using CASE for 10% or less of their workload. These results accord quite closely with those from a Focus Research study reported recently in *The New York Times* of the general population of large users.

However, according to our respondents CASE will be in use by virtually 100% of this population within the next two years, and for a substantial part of their development workload. Thirty-three out of the thirty-five respondents said they will be increasing their use of CASE tools in the next two years, the majority of them very markedly.

Virtually everyone in the survey indicated that they will be using design tools, data dictionaries, and code generators within the next two years, with the majority citing full-life-cycle products as the anticipated solution.

Considerable confusion exists about definitions of terms central to an understanding of products like CasePac, and the population queried does not have an accurate idea of what is and is not contained in CasePac and its close competitors, IEF and MSP.

The marketplace does not know the difference between a data dictionary, a repository, and an encyclopedia, and as INPUT stated in the Phase I report, any vendor attempting to position a product in this arena must define its terms at every turn.

The mainframe environment is generally viewed as more suitable for dictionaries and code generation than for design and full-life-cycle CASE products. In answer to the question regarding the suitability of CASE components to the mainframe environment, twenty-nine out of thirty-five viewed data dictionaries as appropriate to the mainframe, twenty-two viewed code generators as appropriate, but only eleven cited full-life-cycle environments and nine cited design tools as appropriate to the mainframe.

Five out of the eight users who responded to detailed questions about CasePac functions indicated their belief that CasePac includes code generation, which it does not. Also, the absence of code generation was not specified by anyone as a reason for failing to purchase CasePac.

In terms of the products currently in use or under evaluation, the popular PC workbench products were runaway favorites. Among those who have made a decision in favor of a mainframe-based product, the tide is running with IEF and MSP, not with CasePac.

Price was not an issue, and when asked to apportion the $200,000 CasePac price tag among the major components of the product, users seemed to attach significant value to all aspects, not out of proportion to On-Line Software's own internal assessment of relative values.

Respondents are not concerned that CASE tools all come from the same vendor, but they want them "seamlessly integrated" and expect them eventually to "produce code from pictures," but this degree of automaticity is not very important to them in the short run.

It is likely that the CASE purchasing decision will be a group decision, with heavy participation from MIS and applications development as well as from data and data base administration. Our sample indicated a wide variety of decision influences and a prevalence of group decision making with respect to CASE. It will be impossible to target a single decision influence in a prospect organization, necessitating a very broad, complex, and time-consuming sales campaign to arrive at a successful decision for a comprehensive product like CasePac.

One very important fact derives from user interviews in Phase I of the current study, from other closely related INPUT user interviews, and from a review of current literature: the full-life-cycle approach is the one most favored by users in the long run. This means that the entire process of creation, modeling, testing, installation, documentation, and maintenance of applications must fall under CASE purview. Interfaces among these various stages and functions are expected to be "seamless."

In summary, CasePac is running against a powerful PC-Workbench tide, into a complex decision-making matrix of functions and people and, if this sample is any indication, is playing to a relatively naive and uninformed audience.

D

What's Right and Not Right about CasePac

What's right:

A very large and well-heeled population of large-scale IBM mainframe users running DB2 and MVS/XA.

A very strong desire on the part of virtually all of these users to employ CASE for some, and in many cases a very large, amount of their development work.

A fairly sizeable number already experimenting with CASE.

A growing tendency to employ DB2 for an increasing amount of the total workload, with a corresponding decrease in the use of IMS.

No apparent trauma over spending $200,000 for a CASE product.

A clear recognition of the need for a better dictionary function than that supplied by IBM and a fair degree of pessimism and skepticism about IBM's intentions in that regard.

What's wrong:

Not very many users appear to be buying or planning to buy mainframe-based CASE products at this moment.

Of the mainframe CASE products mentioned, CasePac is running a fairly distant third against IEF and MSP.

Users want a full-life-cycle CASE environment, including code generation.

Most users do not view the "front end" analysis, design, and prototyping tools as appropriate to the mainframe, but rather see them as appropriate to an attached workstation or PC.

Users are very naive and unknowledgeable about CASE and CASE terminology.

Users don't know which products have which features.

The above is particularly true with respect to CasePac, even among those who profess to have evaluated it.

E

Other Factors

There is no consistent pattern reported as to who the key decision makers are with respect to a product like CasePac.

Much of the decision making will be by committee or task force.

The composition of the task force will vary across many levels and functions in the organization.

Users say they want a fully integrated (seamless) full-life-cycle CASE environment.

Users who want a dictionary in front of DB2 want an **active** dictionary. They expect changes in data definitions to be propagated to programs automatically.

IBM has not yet made its move with respect to a comprehensive dictionary for DB2 nor with respect to CASE, overall.

This creates opportunity in the short run, but in a significant number of installations it may also have a paralyzing effect, postponing decision making for a protracted period and providing IBM "right of first refusal" on such important mainframe products as dictionaries and CASE systems. Probably half the target population will wait for signals, if not product, from IBM.

CASE is a very fragmented business. As *The New York Times* recently commented: "The potentially vast market looks like a tangled collection of niches."

If there is any structure and leadership in the CASE market at present, it belongs to the leading PC-based front-end vendors such as Index Technologies, Knowledgeware, Cadre, M. Bryce & Company, McDonnell Douglas, Nastec, Polyton, and Techtronix, who together have sold about 40,000 workstation packages to date and appear to be experiencing continued lively growth.

On the surface CasePac appears to be aimed at a very large market, namely the IBM mainframe DB2 MVS/XA segment. This excludes, of course, other markets including the IBM VM segment, the non-DB2 segment, and a host of submainframe marketplaces including the already considerable and fast-growing DEC VAX segment.

The funnel then begins to narrow even more dramatically as we eliminate users who want code generation, an active dictionary, a workstation front-end, and/or a full-life-cycle capability. Given these limitations, CasePac may actually apply to an audience too small to support the product.

There is also some feeling among users and observers alike that optimistic growth projections for CASE will not be achieved because of the one overriding requisite for implementing CASE in any development organization: the introduction of a new and tightly disciplined design and programming methdology. Many companies are finding it difficult, or are hesitating even to embark on such a course because of the conservatism and inertia inherent among developers, the extreme difficulty and cost of retraining them, and a feeling that changing methodology, even if of benefit to new development, still leaves the huge pile of existing software untouched.

At the outset, most of the development problem for most installations consists of the running base of "spaghetti" code. Introducing a CASE product, if anything, makes it even more difficult to focus programmers on the all-important maintenance function so necessary in keeping the using organization in business. And, in the realm of new development, much of the problem has to do with changing minds, introducing new methods, and establishing a kind of regimentation foreign to the psyche of most programmers. Therefore the problem is motivational and educational, not technological. Often CASE is much more a learning tool than a production or development tool.

Since the issue is learning, many users may conclude that this can go on more incrementally, privately, flexibly, and inexpensively on workstations than through a massive conversion to CASE on-line with the mainframe. Clearly the reason that the PC-based products have been relatively successful and the mainframe and full-life-cycle products less so has been the convenience with which an introductory or experimental product can be introduced into the environment as a learning tool and "camel's nose under the tent," as opposed to a much more significant and sweeping commitment to a full-scale product. A decision for CasePac is clearly not likely to be viewed as anything short of a major move.

F
Recommendations

If On-Line Software International intends to stay in the CASE marketplace, it should, among other things, undertake a tutorial approach to educating a broad audience within the buying organization to the exact meaning and implication of the methodology, features, functions, and terminology applicable to the CASE environment. That may well be the only hope of bringing the diversity of decision influences within the organization to the point of communicating with each other effectively and arriving at a buying decision (short of toying with CASE, as many are now doing by buying a few inexpensive PC packages). Confusion promotes delay and indecision, which can only reduce revenues, increase marketing costs, and lengthen the window of opportunity for IBM and other actual or potential competitors to move into the territory occupied by CasePac.

Given the absence of such key features as a workstation front-end, an active dictionary, and a code generator, and given the long decision cycle involved in selling a complete or nearly complete full-life-cycle product, it stands to reason that more CasePac dollars will cross the counter if some of the features can be sold separately to a pinpointed rather than a diverse audience within the buying organization.

INPUT feels strongly that CasePac needs to be unbundled, although there should be a significant incentive for purchasing the whole product in its current and its later, more developed form. One piece of CasePac sold and in use in an installation is far better than all of CasePac under endless consideration. As additional features are added, the total package becomes more and more attractive as a bundle, but still without detracting from the viability of offering certain feature sets either as a wedge into the organization or as a satisfactory solution for some piece of the problem applicable to a specific functional area.

With respect to what features and functions can and should be added, the answers are obvious: whether or not one believes that the majority of programmers belong on-line and will eventually migrate from detached workstations to a more tightly integrated and comprehensive on-line system, the facts are that most of them at present and for the immediate future are starting down the CASE road with distributed or detached workstations. To be effective in the marketplace, On-Line Software must add a workstation (preferably PC) front-end.

The latter can be accomplished in a number of different ways including in-house development, acquisition, or strategic alliance/interface. At this point an Excelerator interface might be advisable, although a preferable approach would be to find or develop a PC-compatible product that has a look and feel more like the present CasePac mainframe product and might be touted as having some competitive advantage over Index and all of its current partners. A PC product that, also, brought along the Yourdon methodology could be a plus given its popularity.

Almost as compelling, it seems to INPUT, is the "activation" of the CasePac dictionary. Much of the benefit of centralization and on-lineness of the current product is lost without an active dictionary. Clearly, a passive dictionary can only be viewed as an interim stopping place on the way to the ultimate CASE environment.

Code generation appears to be one of the lively and viable subsegments of the current CASE market and an essential participant in the ultimate full-life-cycle evolution. Once again, the same alternatives exist for OLSI: develop, acquire, or form a joint venture. Here, by carefully choosing or crafting code generation capability, the product could be adaptable to other target platforms such as VM and VAX. One can certainly visualize a significant number of organizations in which central

MIS will retain responsibility for applications development even if targeted to distributed 9370s and VAXs or other remote systems.

And finally, since so much of the acceptability and ultimate success of CASE depends upon educating and shifting the psychology of designers and programmers, it might make sense for On-Line Software to go heavily into CASE education and training in parallel with its efforts to further develop and promote CasePac. The training itself would have to be "nondenominational" to reach its broadest audience, but CasePac could serve as a vehicle for much of the practical hands-on and demonstration side of the curriculum, thus subtly biasing future users, buyers, and influencers. Well-publicized success stories in "retreading" old programmers into modern CASE devotees would help fill classrooms. If so, this would produce both immediate training revenues and, perhaps, have a long-term beneficial impact on CasePac sales.

G

A Word about RAMIS

No discussion of CASE would be totally complete without mention of 4GL facilities, such as RAMIS. Unquestionably, RAMIS is a CASE-like product, used mainly for development of end-user systems by end users, though not exclusively. To the extent that RAMIS, or some of its end-user features, can be incorporated under the overall CasePac umbrella, then the latter is strengthened and some of the stigma of production inefficiency of the former is neutralized. Whether, and how, to accomplish this feat is well beyond the scope of the present assignment, but it is certainly an issue that should be added to the list of topics for future consideration.

H

Summary

INPUT's recommendations are based on what users are actually buying and say they want to buy, INPUT's view of the complexity and diversity of the CASE decision, and the difficulty of implementing sweeping changes in development methodology in a large company. INPUT's recommendations are as follows:

- Unbundle components of CasePac, particularly the dictionary, and sell them separately.
- Add, or interface to, a front-end distributed workstation, preferably PC-based.
- To the extent possible, add methodological and other saleable features and refinements such as the Yourdon method, an active dictionary, code generation, and items from a list of other major and minor enhancements. Such a list could be easily developed and prioritized based on further user feedback and competitive analysis.

- Add a tutorial flavor to all CasePac-related communications with the marketplace to help bridge the confusion factor that now exists.

- Consider parlaying that tutorial thrust into a full-scale, fee-based CASE education and training effort, which could pay direct dividends and indirectly raise CasePac awareness and saleability.

INPUT®

Parsippany Place Corporate Center, Suite 201, 959 Route 46 East, Parsippany, NJ 07054 (201) 299-5999

FAX TRANSMITTAL FORM

COVER SHEET - Page 1

DESTINATION: INPUT, Mountain View

FAX NUMBER: _____

ATTENTION: Shiela Collins

Telephone Number/Location _____

_____

FROM: Orinda

DATE: 6/13/88

PAGES: 1 of 4

TYPE: CONFIDENTIAL CORRESPONDENCE YES ____ NO ____

URGENT YES XX NO ____

CHARGE CODE: YDB2

COMMENTS:

Don Shaw found three errors in the bound YDB2, Phase II and III. They have to be corrected and reprinted for the client. Please let us know when you can schedule this. Thanks.

Corrections attached.

Phase III
This should be same as Exhibit 8, Phase II

EXHIBIT 1

5-YEAR PROJECTION*

Product Category	1986	1991	AAGR (Percent)
1. Analysis, Design, and Prototyping Tools	80	512	45
2. Config. Mgmt., Repository/Dictionary, and DBMS Tools	40	215	40
3. Code Generators	75	336 ~~75~~	35
4. Editors, Debugging, and Testing Tools	153	568 ~~153~~	30
TOTALS	348	1,631 ~~348~~	37

* Pending INPUT's full market survey as part of its MAPS Program

One way of classifying CASE tools is to sort them into four basic categories:

- Analysis, design, and prototyping tools, which are mainly PC "workbench"-based and appear to be gaining rapidly in sales.
- Configuration management, repository/dictionary, and DBMS tools. The repository segment of this group, especially, appears to be growing rapidly and benefits from a more substantial price tag than the prior category.
- Code Generators. These enjoy a substantial price point and appear to be doing nicely in the marketplace.
- Editors, debuggers, and testing tools: possibly the largest CASE submarket at present, but one that is destined to be surpassed in the early 1990s by its faster-growing counterparts, above.

If we look ahead at a total potential CASE marketplace of $1.63 billion and assume factoring to be 3/4 IBM-based, about 1/2 of that DB2-based in dollar terms, we winnow to a marketplace of over $600 million avail-

Phase II

In Exhibit 7 INPUT attempts to analyze the 1986 numbers—the last year for which anything like "actual" figures might apply—into various ranges and component breakdowns from three sources: *CASE Outlook*, IDC, and INPUT, wherever appropriate or comparable numbers can be found. As can be seen from the Exhibit, all sources appear to be in close agreement on the first category. Therefore in the next Exhibit, Exhibit 8, INPUT felt comfortable in carrying forward an $80 million 1986 estimate for category one.

EXHIBIT 7

SYNTHESIS OF 1986 ESTIMATES

Product Category	CASE Outlook	IDC	INPUT
1. Analysis, Design, and Prototyping Tools	82	78	60-80
2. Project Mgmt.,Config. Mgmt., Repository/Dictionary,and DBMS Tools	80	-	680
• Project Mgmt. Portion			40
• Net			40
3. Code Generators and 4GLS	170		75
• Code Generators		114	
• 4GLs		327	
4. Editors, Compilers, Debuggers, and Testing Tools	450		
• Compilers			297
• Net			153

add line between -70's for next figure

In the case of category two, if one accepts *CASE Outlook's* total of $80 million, and the INPUT estimate of $40 million in 1986 for software project management tool revenues, the net remainder of $40 million can be ascribed to configuration management, repository/dictionary, and other DBMS tools—and, therefore, is carried over into the second category in Exhibit 8.

Phase II

reducing inflexibility, improving internal skills development, improving job satisfaction, and achieving the objective of developing high-quality, easily maintainable systems, on-time and within budget.

G

Revenue Forecasts

Exhibit 6 shows *CASE Outlook's* estimate of revenues in each of those aforementioned categories from 1985 through 1987 and its forecast for 1990. As can be seen, *CASE Outlook* is forecasting robust growth in all categories, with only the fourth, which includes compilers, in the same range as INPUT's overall Systems Software forecast for comparable periods. All the rest exceed INPUT's growth expectations for Systems Software and, indeed, for CASE in Europe, which is expected to grow in total as a category at 36% over the period 1987-1991.

EXHIBIT 6

CASE OUTLOOK'S WORLDWIDE FORECAST

($ Millions)

Product Category	1985	1986	1987	1990	AAGR (Percent)
1. Analysis, Design, and Prototyping Tools	52	~~80~~ 82	130	436	53
2. Project Mgmt.,Config. Mgmt., Repository/Dictionary,and DBMS Tools	48	80	140	680	70
3. Code Generators and 4GLS	120	170	235	645	40
4. Language-Sensitive Editors, Debugging, Compilers,and Testing Tools	350	450	540	1,300	30