



CORPORATE BACKGROUNDER

SOFTWARE EMANCIPATION TECHNOLOGY

Software Emancipation Technology (SET) develops, markets, and supports DISCOVER[®] a flexible and scalable software Development Information System (DIS). SET designed DISCOVER to help enterprises evolve their mission-critical software development processes and to more effectively manage their software systems as strategic assets. By using DISCOVER, enterprises can measurably improve software quality and software development as well as reduce maintenance costs and time-to-market.

Since its introduction in May 1995, DISCOVER has set the standard for an emerging class of software engineering products. The DISCOVER DIS offers capabilities that no other commercially available software application can provide. These unique, comprehensive solutions form an integrated set of applications and tools based on SET's proprietary information modeling and analytical technologies.

DISCOVER: Groundbreaking Development Information System

DISCOVER's unique family of DIS tools offers a wide-ranging Return on Investment. Customers gain improved productivity, quality assurance, customer satisfaction, and the competitive advantage of rapid time-to-market.

DISCOVER provides a highly graphical and accurate representation of an application's source code. Based on a database--SET's unique Information Model[™]-- the DISCOVER family of solutions addresses acute problems like improving software Quality, application maintainability, and time-to-market. The Information Model[™] captures all source code entities, their attributes and interrelationships. The resulting detailed view offers a high-level, "as-built", architectural perspective of the entire application

SET's unique Tree Pattern Matching[™] (TPM) yields descriptions of an abstract syntax tree that allows sophisticated representation of code and superior analysis of technology underlying applications, while helping to automatically transform source code

patterns. Data Flow Analysis™ (DFA) is a key remediation tool that traces through assignment statements to unmask all entities receiving the value of a variable. These and other DISCOVER features combine to offer powerful analysis and transformation capabilities as well as facilities to generate reports, automate software restructuring, and accelerate source code comprehension and analysis. With these technologies, enterprise-wide programmers, managers, and executives can more effectively organize, manage and execute all elements of the software development process.

The Benefits of DISCOVER

With Discover, users achieve engineering discipline in program management, applying unique tools and quantitative reports on internal source code structure and quality. DISCOVER's interoperability and ease-of-use bring rapid and significant results to developers, managers, and customers. DISCOVER users can:

- enhance developers' code comprehension with powerful queries, navigation, and graphical views;
- support evolutionary improvement of the development process by performing sophisticated analyses of the underlying source code and related software objects;
- intercept errors and reduce the formal code inspection by automating traditional manual code reviews;
- simplify and reduce the risk of large-scale changes and software enhancements with automated change propagation;
- use impact analysis to explore risk associated with making changes to source code;
- deliver project management data, including useful metrics, and statistics for both static and trend analysis of current programs;
- proactively monitor and elevate quality by analyzing, enforcing, controlling, and improving software quality early in development cycle, ultimately reducing the cost of correcting defects;
- facilitate knowledge capture and asset management with the Information Model by creating an inventory of all source code and software artifacts.

Customer-documented improvements in efficiency and cost are remarkable, affecting areas from code comprehension to migration to new hardware platforms or software languages.

<u>Development Area</u>	<u>Potential Savings</u>
Code Comprehension	80%
Code Review	50%
Testing Time	25%
Code Compilation time	10%
Defect Correction	\$10,000 per defect

Market Environment

Sophisticated software applications are a major part of almost every organization's operation. Elaborate systems promise enhanced value and increased productivity for business activities like financial trading, enterprise resource planning and healthcare reimbursement. Advances in miniaturization and computer processing power have made complex software a part of daily activities by engendering the rapid growth of embedded software applications. They are now essential parts of products as varied as telecommunications switches and anti-lock brakes. The industry estimates that embedded applications will continue to increase by 25%-30% in the near future (Hurwitz Group, "Taking Advantage of New Opportunities through Organizational Flexibility": October, 1999).

With each successive generation, a company's software becomes more intricate. Additional features increase size and profoundly complicate the process of developing and maintaining modern applications. Even well organized, highly integrated applications can require millions of lines of source code. The intensifying demand for software functionality has led to a "complexity crisis" with far-reaching implications (International Data Corporation #15523: March, 98).

When software applications provide benefits ranging from operating efficiencies to product differentiation, organizations that rely on them increasingly view their software systems as strategic assets and as important sources of competitive advantage. For these enterprises, the quality of their software systems has emerged as a critical determinant of their continued success or even their survival. However the "complexity crisis" has taken a toll on quality and management efficiency.

Rapid product cycles contribute to the crisis. To keep pace with demand for dramatic improvements in a competitive market, developers often sacrifice some degree of organizational integrity. The result is functional but poorly written code. With quick time-to-market as a priority, developers may neglect documentation and take other shortcuts that further compromise the reliability and efficacy of the software.

The development climate often aggravates the situation. Today's enterprises may distribute code execution among many teams, possibly geographically dispersed, and all working in a heterogeneous, networked environment. Developers may be using several different programming languages, databases, operating systems, and hardware platforms. Working independently of each other, programmers often produce changes that are incompatible either with parts of the existing system or with other revisions, or both. To exacerbate the problem, development teams suffer high turnover and subsequent loss of intellectual capital that leaves gaps in the process.

This combination of circumstances presents organizations with a high likelihood of code errors. Experience suggests that the cost of fixing defects increases ten-fold as the company moves from implementation to testing to the customer site. The \$100 error in development becomes a \$1000

error in testing, a \$10,000 error when the customer identifies it. (Marilyn Bush, "Improving Software Quality: The Use of Formal Inspections at the Jet Propulsion Laboratory": March, 1990). Where the software is mission critical or affects e-business, the error impact deepens, with potentially disastrous results. To make matters worse, business partners' increasing supply chain interdependence means that poor quality in an application has a ripple effect far beyond a single company.

Instead of adding value with new functionality, businesses dedicate an increasing percentage of scarce programming resources --as much as 70%-- to understanding and maintaining the integrity of existing systems. Line managers find it steadily more difficult to control costs, assess project status, monitor quality, and plan resource utilization because of the growing fragmentation of the development process. Senior IS executives find it nearly impossible to comprehend the software system or development process as a whole, so they cannot effectively manage strategic software assets, or integrate disparate systems from multiple sources like mergers.

In sum, the issues raised by the "complexity crisis" threaten software-dependent organizations. Failure to address these challenges effectively can result in:

- compromised software quality;
- lengthened development cycles;
- impaired ability to innovate;
- increased risks to project schedules and budgets;
- heightened vulnerability to loss of valuable intellectual capital;
- reduced Return on Opportunity (ROO).

Ultimately these problems jeopardize the competitive position of the entire enterprise.

Organizations seeking to surmount these challenges have adopted a variety of methods, most of them time-consuming, inefficient, and susceptible to human error. They try structured design and analysis techniques; attempts to impose programming and documentation standards; manual review of development submissions to determine compliance and project status; and automated software testing. All these efforts do not promise success. They may fail to effectively identify non-obvious errors, design flaws, or conflicts among widely distributed elements of a software system. With the increasing inadequacy of existing methods to address the "complexity crisis", organizations risk impairing the strategic value of their software assets.

Businesses need a solution that will enable them to understand their software. They need solutions that make it possible to effectively organize and manage systems-related processes, whether development, maintenance, or enhancement. It is these needs that led to the development of DISCOVER, the ground-breaking software Development Information System.

The DISCOVER Solution

DISCOVER was the brainchild of Dr. Vladimir P. Geisberg, founder of Software Emancipation, who envisioned a solution that would respond to his professional needs as a software executive supervising a thousand developers. Dr. Geisberg, a former corporate officer of Prime Computer, required a comprehensive development system that would organize, automate, and document the software engineering process. Such a system would also ensure the transfer of critical software knowledge from the individual to the organization, so that a company's valuable software assets would be less dependent on only a handful of people. With these goals in mind, in 1991 Geisberg founded Software Emancipation Technology and created a new software category, - Development Information Systems- the product that is known today as DISCOVER.

In a revolutionary step forward, DISCOVER allows developers to design high quality into the code early in the development cycle, without sacrificing productivity. DISCOVER finds and reduces errors and redundancies; supports global changes to even the most complex code, and enables programmers to verify quality before submitting work to the code base.

DISCOVER's flexibility and graphical views give it an ease-of-use that guarantees improved productivity. Developers can continue to use tools they know because of DISCOVER's interoperability with various configuration managers, editors and debuggers. Highly graphical and accurate representations promote rapid code comprehension for both the experienced and the new programmer. With little or no training, DISCOVER users can clearly see the source code structure. The net gain for DISCOVER customers is reduced errors, lower development costs, and improved time-to-market.

DISCOVER's Unique Information Model™

DISCOVER's power emerges from its complete and scalable database, the DISCOVER Information Model, which is the core of Software Emancipation's Development Information System (DIS). Software Emancipation's proprietary Information Modeling technology constructs a database that identifies every element in a software system. The omni-directional, associative nature of this database allows each element within the model to be connected with its definition and to all occurrences of the element in the software system, including software program references, documentation, test results, and other software objects. The Information Model then transforms that data into a corporate asset accessible throughout the enterprise. Developers and managers, armed with an analysis tool for both existing code and impact of proposed changes, can precisely measure compliance, progress, and quality. Developers update the Information Model on a regular basis to reflect source code modifications. Once built, the Information Model becomes the engine for improvements in quality throughout the development organization.

The Information Model provides several unique features to expedite high-quality development. Combined with Tree Pattern Matching™ (TPM), Data Flow Analysis™ (DFA), impact analysis,

and change propagation, DISCOVER tools offer developers an industry-leading solution to the “complexity crisis”.

DISCOVER is available in a set of seven basic modules, each providing a successively higher level of abstraction and increasingly sophisticated features, matching the needs and skill sets of particular groups of users. SET has also begun to prepackage a number of Solution SETs -- subsets of DISCOVER-- which, when combined and fully deployed, cover the entire information systems paradigm to address common development problems. Each Solution SET includes a combination of product modules to address particular development tasks, plus professional training and consulting services to complement those modules. The first in the series of new Solution SETs --the Quality Solution SET-- was announced in October of 1999, with others to follow.

SET also offers a broad range of professional consulting, training and support services as an integral component of its solutions to assure quality deployment and start up.

Software Emancipation's Management Team

Software Emancipation's senior management team has an impressive depth of experience in high-technology software companies.

David Champagne, President and CEO, is a twenty-year veteran of the software industry who brings extensive experience implementing, managing and growing software consulting and support services. Most recently, Mr. Champagne was Executive Vice President and COO of Point Information Systems, where he was responsible for sales, marketing, product development, professional services, enterprise customer relationships, and strategic alliances. Prior to that, he spent 14 years at Lotus Development Corporation in key management roles including Senior Vice President - Worldwide Services. Mr. Champagne holds an MBA from Bryant College and a BS in Industrial Engineering and Operations Research from the University of Massachusetts at Amherst.

Michael Brook, Senior Vice President of Engineering was previously Director of Advanced Development at Computervision. He has over 20 years of research and development experience in the conceptualization, design, and implementation of advanced products. Mr. Brook holds an MS degree in Electrical Engineering and Computer Science from the College of Radio and Electronics in Kharkov, Ukraine.

Bruce Boes, Vice of President of Marketing brings to Software Emancipation over fifteen years of experience in establishing marketing and business development programs for technology companies. Prior to joining Software Emancipation, he served as President of the Engineering, Manufacturing and Design Group at D.H. Brown Associates. Mr. Boes held different positions at Matra Datavision, Inc., including Vice President, North American Operations and then Corporate Vice President, Marketing Strategy. He holds an MBA in Management/Finance from Rensselaer Polytechnic Institute, Troy, New York.

Gerald R. Covello, Vice President of North American Sales, brings over 20 years of professional, information systems sales experience to his role at Software Emancipation Technology, including six years with the company. Prior to joining SET, he served as Vice President of Sales for PROCASE Corporation and Director of National Sales for Martin Marietta Data Systems. Mr. Covello holds a BS degree from the University of Maryland.

DISCOVER Products

DISCOVER is available in a set of seven basic software packages modules, each providing a successively higher level of abstraction and increasingly sophisticated features, matching the needs and skill sets of particular groups of users. SET also offers bundled Solution SETs and Centers that target specific development problems. The current modules of DISCOVER are *Developer*, *Developer Xpress*, *Engineer*, *Project Leader*, *QA Analyst*, *QA Cockpit* and *Architect*.

Developer

DISCOVER Developer is the baseline for all of the DISCOVER modules. *Developer* facilitates detailed and system-level understanding, automates labor-intensive tasks, and accelerates code knowledge and understanding. *Developer* fully integrates and cross-references source code text, graphical views (e.g., flowchart, class inheritance), documentation, and tests. In addition, the application performs a submission check, filtering any new code through management-defined, quality standards before integrating it into the main code body. With the module's impact analysis capability, users can determine what influence changes will have throughout the system.

DISCOVER Developer Xpress is a stand-alone alternative to DISCOVER *Developer*, providing instant navigation and querying of the Information Model. *Developer Xpress* is available to developers immediately after the Information Model is built. Used at the desktop level, *Developer Xpress* is ideal for fast implementation, product adoption, and in-house training.

Engineer

DISCOVER Engineer brings greater depth to the searches and analyses of *Developer* with semantic and syntactic source code analysis. *Engineer's* Tree Pattern Matching™ (TPM) and Data Flow Analysis™ (DFA) capabilities add structure and repeatability to developers' and senior engineers' tasks. With these new analytical technologies, software designers can search syntactic patterns to:

- uncover complex and error-prone constructions;
- implement automated change propagation or precise single changes;
- encapsulate a sequence of tasks to promote repeatability;
- customize associations from code to documentation and tests.

Engineer includes all of the features of *Developer*.

Project Leader+

Project Leader+ provides the full power of **DISCOVER** to drive software development from start to finish. With *Project Leader+*, senior engineers and managers can build models, establish performance criteria and enforce them throughout the life cycle of the software. Managers can customize filters and scripts, plus define process and quality standards.

Project Leader+ includes all of the features of *Developer* and *Engineer*.

Architect I and Architect II

Together *Architect I* and *Architect II* provide an automated and manageable strategic development environment essential for aligning reengineering efforts with business objectives.

Architect I makes it possible for developers to reorganize and optimize the entire application by cleaning up and simplifying the code. *Dormant* leaves only live code for the development team to maintain, while *Simplify.h* reduces header files to include only those necessary to the source code. The result is reduced code size and compilation time.

Architect II uses *Extract*, *Package*, and *Partition* to help componentalize and reengineer software. *Extract* uses mathematical algorithms, based on user-specified criteria, to organize the code into logical subsystems. *Package* divides the application into logical modules to help it better address dynamic market needs. *Partition* automates code division while preserving the functionality and dependencies of the relocated entities.

Architect I and *II* include all the features of *Developer*, *Engineer*, and *Project Leader+*.

QA Analyst

QA Analyst lets quality assurance teams track code defects and pinpoint weak structure. *DefectLink* helps eliminate chronic problems by using mapping that links defect to code. *Delta* tracks changes down to the entity level and *Modularity* enforces the architectural integrity by checking compliance of modified code against system design. *TestLink* identifies the minimum set of regression tests necessary as a result of source code changes. *TestCoverage* takes *TestLink* one step further, measuring test suite efficiency by identifying the application's untested areas. With the *QA Analyst*, testing engineers can take a proactive approach to quality control.

QA Cockpit offers software executives the premier quantitative reporting tool for software quality, equipping them with the means to define, measure, and control quality during the development process. DISCOVER's Quality Assessment Report (QAR) provides quantitative and graphical analysis of factors like programming constructs, portability, globalization, and structure. This x-ray view into the current quality of the entire system enables early problem detection and remediation, focused and informed business decisions, and improved risk management. With the QAR, organizations can resolve memory leak problems, manage partially undocumented code history, and build in filters to help find specific problems.

Solution SETs

DISCOVER Quality Solution SET

DISCOVER Quality Solution SET uses the unique DISCOVER tools and expertise of SET services to help organizations build quality into their software. The *Quality Solution SET* delivers quantifiable business value to customers, addressing the “complexity crisis” challenge. With the

Quality Solution SET, new code is efficient and uncomplicated, compilation time is reduced, and test time is decreased.

Quality Solution SET begins with a combination of software and services to build an Information Model and generate a baseline scorecard for the existing code. This Quality Assessment Report (QAR) provides graphical and textual information about the internal software quality. The QAR explores a variety of programming rules (based on ANSI and customer standards) including programming constructs, structure, portability, and globalization rules. By identifying instances of non-compliance, DISCOVER can investigate impact on code quality, and the required remediation. Armed with tangible and quantifiable results, management can proceed to the second stage.

With the baseline established, DISCOVER provides measures to prevent new software defects from entering the system, eliminates unnecessary code, and optimizes testing. Before any new code becomes part of the master pack, DISCOVER applies the established programming rules, enhanced with company specific standards. This automated code review combines with impact analysis, which analyzes the effect of any change on the code base. The result is defect prevention, elevating quality control to quality assurance.

Additional applications simplify header structures and eliminate dormant code to reduce complexity and decrease compile time.

As part of the complete *Quality Solution SET*, Software Emancipation offers industry-leading support in project planning, implementation, process adaptation, quality rule customization, and user training.

DISCOVER *Quality Solution SET's* ROI is multi-faceted and far-reaching. In mission-critical or e-business applications, the ramifications of a software error can be disastrous. Even in less urgent situations, a defect that reaches the customer can cost \$10,000. Correcting the flaw in the first stages of development can be one-hundredth the cost of a later fix. With DISCOVER tools for quality, businesses can build quality into applications early in the process, and derive significant benefits. Time required for code review may decrease by 50%, and testing time may drop as much as 25%. Key personnel, freed from repetitive testing, can apply their talents to improved and increased product output. All these savings in time mean an accelerated development cycle that provides the competitive edge of an improved time-to-market.

DISCOVER *Quality Solution SET* includes:

Stage 1 - Quality Assessment

- Project Leader - "Information Model" creation and administration
- QA Cockpit
- Quality Filter SETs - Choice of 2 from SET's premier list of filters to supplement QA Cockpit
- Developer Express - Instant navigation and model query with impact analysis for use in defect

remediation

- Project Management, Model Build (up to 1M LOC), Defect Identification and Repair Training
- Ongoing product support and maintenance

Stage 2 - Quality Assurance

- QA Analyst
- Developer Express with Submission Check
- Process Consulting, CM Integration and Filter Customization to adapt DISCOVER to company specific needs
- Ongoing product support and maintenance

Solution Centers

Xpress Center - Enhance code comprehension

DISCOVER *Xpress Center* provides instant access to the Information Model, expediting developer familiarity with the source-code base. Used at the desktop level, *Xpress Center* is ideal for fast implementation, product adoption, and in-house training. *Xpress* users need virtually no training and will see minimal disruption of the existing development environment as they use SET tools to improve productivity.

Ramp-Up Center - Shorten release cycle for large bodies of code

DISCOVER *Ramp-Up Center* provides quick navigation and comprehension of the code base. *Ramp-Up Center* integrates and fully cross-references source code, graphical representations of code, internal documentation, and tests, providing seamless transitions from one view to another. *Ramp-Up Center* also automatically verifies quality upon file check-in. These new perspectives on the code motivate developers, improving their efficiency and increasing the organization's overall productivity.

Process Improvement Center - Automate and accelerate development processes

DISCOVER *Process Improvement Center* is a vital aid to managing unpredictable processes, no matter what the cause. *Process Improvement Center* automates common development tasks and ensures that both experts and novices can employ a streamlined task sequence. Managers will find it easier to assess risks, monitor project progress, and keep all parts of the team in sync as the release date approaches. *Process Improvement Center* provides a full audit trail, from initial product specifications through testing.

Structure Center - Enable system architecture re-engineering

DISCOVER *Structure Center* makes reengineering manageable, and reduces maintenance costs - all without the risk associated with manual reengineering projects. With *Structure Center*, the system architect can reengineer an application to respond to the ever-growing demand for better performance. The developer can combine, break apart, or reorganize an application's components; account for code interdependencies; and assure that the resulting modules are compilable.

Ultimately *Structure Center* enables strategic software development by aligning development with business objectives

Y2K Center - Automate Year 2000 remediation

DISCOVER *Y2K Center* offers the only end-to-end solution for managing Year 2000 compliance projects in client/server applications. Leveraging DISCOVER technology, *Y2K Center* enables companies to plan, estimate, and complete their Year 2000 software conversion efforts for a fraction of the cost and time than otherwise would be possible. *Y2K Center* supports any conversion and addresses all four phases of the conversion: asset inventory, impact analysis, conversion, and testing. With its robust audit-trail capability and ability to offer rapid assessment and change, *Y2K Center* also provides a powerful insurance policy and back-up plan, post millennium, to minimize potential Year 2000 liability.

DISCOVER-Supported Languages, Platforms, and Tools

DISCOVER currently supports the following programming languages:

ANSI

C/C++

K&R C

Visual C++

Java

Oracle Embedded SQL and PL/SQL

Informix SQL

Sybase SQL

Platforms/Operating Systems

SunOS

Solaris

HP-UX

SGI IRIX

Windows NT

Tools

GNU

Gdb

Sun dbx

HPxdb

HPdde

DISCOVER integrates seamlessly with all industry-standard configuration management systems, and most popular browsers, editors, debuggers, and other software development applications.

Integrates with:

CMs for UNIX -	Clearcase, RCS, CVS, SCCS, Continuus
CMs on NT -	PVCS & SourceSafe, Clearcase, and Continuus
Applications -	Microsoft Word, Adobe FrameMaker. Quantify, and Pure Coverage
Editors -	Emacs, vi, Microsoft Developer Studio