REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1994	Technical Report

4. TITLE AND SUBTITLE

Software Estimation Program (SWEEP)
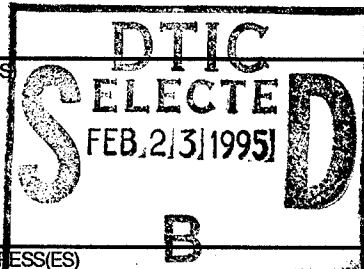
5. FUNDING NUMBERS

6. AUTHOR(S)
Produced by Software Productivity Consortium under contract
to Virginia Center of Excellence

G MDA972-92-J-1018

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Virginia Center of Excellence
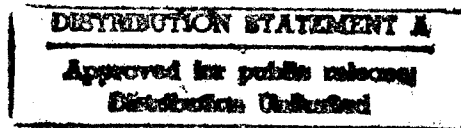SPC Building
2214 Rock Hill Road
Herndon, VA 22070

DTIC
SELECTED
FEB 23 1995
B

8. PERFORMING ORGANIZATION REPORT NUMBER

SPC-92017-CMC,
Version 02.01.00

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
ARPA/SISTO
Suite 400
801 N. Randolph Street
Arlington, VA 22203

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

This replaces SPC-92017-CMC, Version 02.00.10.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

No Restrictions

12b. DISTRIBUTION CODE

19950214 112   1

13. ABSTRACT (Maximum 200 words)

The Software Error Estimate Program (SWEEP) aids you in the management and prediction of errors and defects in software intensive systems. It will support you in establishing goals for error detection during software development and can help to track progress against those goals. You can also use SWEEP to monitor and help control the quality of software products by predicting the number of defects remaining in your software system, or latent error content. You use the latent error content to estimate the amount of testing and repair effort needed to reach different quality levels and to understand the quality of a delivered product. SWEEP can also help you project software outages due to software related defects in software intensive systems.

SWEEP supports a variety of different organizational approaches to monitoring defects during software development. Organizations that measure and track defects during the software implementation phase can iterate through development and test intervals, suing SWEEP to predict the latent error content. Organizations that measure and track in on e phase based upon the data collected in previous phases. SWEEP can also assist you in the planning of software projects. You enter the number of defects (per code unit) and the estimated latent error content to derive a profile pattern for error discovery that will project the number of remaining errors on a phase-by-phase basis.

DTIC QUALITY INSPECTED 4

14. SUBJECT TERMS			15. NUMBER OF PAGES
Error estimation, defect discovery, profile, software management			95
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

Software Error Estimation Program (SWEEP) User Manual

SPC-92017-CMC
Version 02.01.00

August 1994

Software Error Estimation Program (SWEEP) User Manual

SPC-92017-CMC

Version 02.01.00

August 1994

CHANGE HISTORY

Version Number	Date of Change	Change Description
Version 01.00.07	August 1992	Original document.
Version 02.00.10	December 1993	Complete rewrite. See Preface for a complete description of the changes since the previous version.
Version 02.01.00	August 1994	Adjusted terminology relating to the use of measurement in the management of software development.

CONTENTS

FIGURES

TABLES

This page intentionally left blank.

PREFACE

Finding and fixing software errors accounts for much of the cost of software development and maintenance. When one includes the cost of inspections, testing, and rework, as much as half or more of the typical development cost is spent in detecting and removing errors (Humphrey 1989).

The Software Error Estimate Program (SWEEP) aids you in the management and prediction of errors and defects in software-intensive systems. It supports you in establishing goals for error detection during software development and can help to track progress against those goals. You can also use SWEEP to monitor and help control the quality of software products by predicting the number of defects remaining in your software system—the latent error content. You use the latent error content to estimate the amount of testing and repair effort needed to reach different quality levels and to understand the quality of a delivered product. SWEEP can also help you project software outages due to software-related defects in software-intensive systems.

SWEEP supports a variety of organizational approaches to monitoring defects during software development. Organizations that measure and track defects during the software implementation phase can cycle through development and test intervals, using SWEEP to predict the latent error content. Organizations that measure and track defects throughout the life cycle can use SWEEP to predict the latent error content in one phase, based on the data collected in previous phases. SWEEP can also assist you in the planning of software projects. You enter into SWEEP the number of defects (per code unit) and the estimated latent error content to derive a profile pattern for error discovery that will project the number of remaining errors on a phase-by-phase basis.

SWEEP is applicable to a wide variety of development efforts, including software products, software-intensive systems, and systems engineering projects. SWEEP is independent of development language, methodology, or platform. You can use SWEEP to predict latent errors in virtually any software product including:

- Real-time embedded software systems

- Command, communication, and control information systems

- Data and information management systems

- Software development support tools

SWEEP is intended for use by a wide range of people and organizations involved in the development effort. Project managers can use SWEEP to aid in the scheduling and rescheduling of project tasks, based on latent error content and quality goals. Senior line engineers can use SWEEP to track the number of errors in code segments to determine when they have finished unit and integration testing efforts. Software quality managers and test and verification engineers can use SWEEP to estimate the

time needed to test code segments in order to help determine when they have completed sufficient testing.

Organizations can apply SWEEP during the design, development, and maintenance portions of the software development cycle, regardless of whether the process is proprietary, a classic waterfall approach, or an evolutionary spiral approach. SWEEP can benefit project and program offices by aiding in the planning, management, and development of their products. Centralized engineering organizations can use SWEEP to collect and monitor data, and subsequently to enact organizational process improvements based on that data. SWEEP can be an integral part of a systematic plan to improve continuously the quality, development, and management of deliverables.

Organizations can introduce SWEEP into a current development effort at various levels of involvement. At a minimal level, organizations can conduct parallel development efforts to collect data in a nonintrusive manner. You can use this shadow data to assess the impact that SWEEP and error prediction will have on your organization's process. At a more involved level, you record the number of defects found during test intervals and then use SWEEP to estimate the number of remaining errors. At the maximum or most involved level, you can use SWEEP as a planning aid to estimate latent defects in various phases of development.

ACKNOWLEDGMENTS

This page intentionally left blank.

1. INTRODUCTION

1.1 BACKGROUND

One of the difficult challenges in today's software development environment is controlling and managing software errors. An error is defined as a detected failure; a failure is defined as a manifestation of the software product's inability to conform to requirements. You can also refer to errors as defects; many people and projects use the terms interchangeably. SWEEP was developed to aid with the control and management of error data for software projects. It allows you to make software error estimates that are useful in setting project goals and monitoring the quality of the software as it is developed by your organization. SWEEP uses software error data available early in the development cycle to project both the number of errors that will be found in later stages of development and the errors contained in the software when it is delivered (latent error content). Latent error content is expressed as an error density; however, if the user knows the size of the software code, the latent error content can easily be converted to the number of errors at delivery.

1.2 BENEFITS

SWEEP is useful to anyone involved with metrics (data collection and monitoring) or verification activities within a software project's software life cycle. SWEEP can be used to make cost-effective decisions related to:

- Assessing the effectiveness of the verification processes based on the error discovery profile generated.

- Making the potential process adjustments to remove errors earlier, to minimize software rework, and to save time and money. This manual does not explicitly address cost and schedule impacts associated with software errors.

- Using the knowledge gained from inspections to predict the probable error content of the software both in later development phases and when the product ships.

- Estimating the rate at which defects will be discovered in delivered software products.

- Comparing various software products' error discovery histories that might not otherwise be possible if only time-based error data were available.

- Planning the software development process with respect to goal setting for error discovery.

- Estimating the number of outages to be expected for complete systems, not just for hardware components (Gaffney, Ferrando, and Jones 1993).

One of the main features of SWEEP is its ability to predict latent error content. This capability is useful in making decisions related to software maintenance and reliability. SWEEP offers a time-based model, a phase-based model, and a planning aid. The time-based model allows you to estimate and track errors during system test and integration cycles. The phase-based model allows you to predict and track defects for multiple phases and can provide defect information before you execute any code. This allows more time for a project to make software process adjustments and to enhance the cost-effectiveness of the projects' decisions. The planning aid enables an engineer to set error discovery objectives in a software project based on experience from previous projects. The planning-aid model generates an error discovery profile based on the corporate data to guide the software project toward those objectives.

The phase-based model approach evolved from work done at IBM (Gaffney 1984) and has been used there on large software projects. An aerospace company has successfully applied the SWEEP phase-based algorithm to project data. The SWEEP time-based model algorithm was validated by the Consortium on a large program.

1.3 AUDIENCE AND PURPOSE

SWEEP's intended audience includes senior line engineers, project managers, software quality managers, verification/test managers, and technologists who are interested in estimating, monitoring, and managing the quality of software under development. SWEEP may be useful for making two types of decisions:

- How to improve the software process, based on the data generated

- How that data measures up to quality goals for a given software project

SWEEP is a tool that enables organizations to:

- Establish appropriate goals for error discovery

- Estimate the error content of software earlier in the life cycle than is currently possible using existing techniques

1.4 SWEEP UPGRADE

The SWEEP tool has been enhanced to allow you to create and manipulate activity sets, organize data into groups, and customize as many as nine life-cycle phases. In addition, improvements have been made to the curve fitting algorithm and to the presentation of the incidence and cumulative curves.

1.5 MANUAL ORGANIZATION

This manual describes how to use SWEEP. The manual is organized as follows:

- Section 2, Using SWEEP within Your Software Process, describes how to introduce SWEEP into your development life cycle. It also identifies how software projects can use the three SWEEP modes both to evaluate better the effectiveness and efficiency of their current development process and to provide a means for initiating process improvements.

- Section 3, Invoking SWEEP, describes how to install and invoke SWEEP on both an IBM-compatible PC and an Apple Macintosh computer.

- Section 4, Operating SWEEP, describes the operation of Mode 1, the time-based mode 1; Mode 2, the phase-based mode 1; and Mode 3, the planning aid. The description covers the menu bar selections, dialog boxes, and windows presented during operation.

- Section 5, Case Study, demonstrates the use of SWEEP with real error data from a major software development project.

- Appendix A describes the mathematical models used within SWEEP.

1.6 TYPOGRAPHIC CONVENTIONS

This manual uses the following typographic conventions:

Serif font . . . . General presentation of information.

Boldfaced serif font . . . . Section headings and emphasis.

[ ] . . . . Screen buttons.

< > . . . . Keyboard key names, such as <RETURN> for the Return key.

This page intentionally left blank.

2. USING SWEEP WITHIN YOUR SOFTWARE PROCESS

2.1 MOTIVATION FOR USING SWEEP

Undetected defects in delivered software-intensive systems can cause disastrous failures. Since the cost of correcting software errors increases substantially in later development stages, it is important to be able to estimate how many defects exist in a software program and to correct them as early in the development process as possible.

SWEEP can help you manage your software development process better. You can use SWEEP as an aid to establish quality objectives and to determine if you are meeting them. You can employ SWEEP to assist you in meeting an availability or other defect-related quality objective established for the software product you are developing. SWEEP will provide you with some of the attributes required for processing software defect-related data that your software development organization must have for certification at the higher levels of the Software Engineering Institute (SEI) software process maturity model. For example, the ability to make software defect projections is one of the requirements that your organization must meet to attain higher software process maturity levels on the scale developed by the SEI (Paulk et al. 1993).

SWEEP will assist you in making software defect projections; it can estimate and predict latent defect content. According to the SEI Capability Maturity Model (CMM), a Level 1 organization typically detects and removes defects instead of imposing a process focused on preventing defects (Paulk et al. 1993). SWEEP can help you decide how to improve your software process by identifying where in your process defects will occur. As software organizations move towards higher maturity levels, they begin to practice defect prevention. The fundamental objective of software defect prevention is to detect defects as early as possible in the development life cycle.

On the basis of current SEI statistics (characterized by the number of organizations at Level 1 versus Level 4), most organizations concentrate on defect detection rather than defect prevention (Kitson et al. 1993). Using SWEEP will help you to manage software defects throughout the software development process and to create a product whose characteristics are more predictable. You will be able to use systematically data obtained during inspections to predict better the quality of your software. This will provide you with the necessary information to take corrective action earlier (and more cost-effectively) than if you were to wait until the system was operational. SWEEP enables you to achieve the availability objectives established for the software you are building.

2.1.1 SWEEP CAPABILITIES

The SWEEP tool consists of three modes to enable you to predict and track the discovery of defects and to predict the latent defect content of software products. SWEEP enables organizations or

development efforts that rely on system and integration testing for identification and repair of defects, to predict the number of defects that remain in the software system after each successive test interval. SWEEP helps organizations that use design reviews and verification activities to predict the number of defects for each phase in a development life cycle. Finally, SWEEP provides to organizations that build similar software systems and maintain a history of their defect discoveries, help in the planning and establishment of software quality goals.

SWEEP's Modes 1 and 2 enable you to use data obtained during inspections and other verification activities to determine if the defect discovery goals you have established for each stage of the development process are being satisfied. If you discover significantly more defects than you had projected, you can correct your software earlier in the process. SWEEP Mode 3 includes a capability to help you establish goals for defect discovery during the software development process. You can use SWEEP to compare your product's defect history with that of other products, to decide how to improve your software process based on how defects are being detected, and to evaluate software quality as the product is being developed.

In order for SWEEP to be effective, a project must apply SWEEP consistently over the development cycle—at the level of computer software units (CSU), computer software components (CSC), or computer software configuration items (CSCI). For example, if you collect and enter defect data at the CSC level during preliminary design review, you must also collect and enter the defect data at the CSC level during detailed design review.

Your organization may use a number of inspection or verification activities during each of the phases of the development process. You may have defect discovery data from several inspections for detailed design, for example. If you assume that defects are evenly distributed across your design, then you can average the defect densities found for each design unit into one defect density. You can then enter this average defect density number into SWEEP.

You can gather defect data from several sources, including software problem reports, system error logs, code inspections, verifications, etc. Defects are usually tracked as errors per thousand source lines of code KSLOC, but your project may use any measurement (e.g., modules, function points).

2.1.1.1 Assumptions Related to the Use of the SWEEP Tool

The following assumptions should be considered when using the tool:

- All detected defects should be recorded when they are detected.

- It is assumed that defects are fixed when they are discovered and that no new defects are injected during the repair effort.

- Defects should be tracked consistently. When using Mode 2, if you track a certain type of defect in the design phase, then you should track the same type of defect throughout the life cycle (i.e., if you choose to track typing errors, then every phase should track typing errors).

- Defects in software documentation should not be tracked with software defects. Documentation defects might inflate the latent defect content of the software product.

- The data input into SWEEP should be validated and updated on a regular basis.

2.1.1.2 Mode 1—Time-Based Defect Predictions

Mode 1 in the SWEEP tool allows you to estimate the defect discovery profile for your software system using **time-based** data. The time-based model is most effective for projects that rely heavily on testing to detect software defects and that lack static verification data obtained through inspections, peer reviews, etc. Mode 1 can help you both to develop schedule estimates to meet your quality goals and determine whether you will complete testing within a given schedule. You can use Mode 1 to help you to answer questions such as:

- What are the estimated remaining defects after N more test intervals?

- How many more test intervals do we need to remove X percentage of projected defects?

- When will we finish testing the system?

- When can we ship our software product?

You collect time-based data at regular intervals during the integration test and system test phases of the development process. The collected data represents the number of defects per code unit you found during a test interval. You enter the collected data into SWEEP to predict the number of latent defects for future test intervals. After each successive test interval, you enter actual defect data recorded during that interval and use SWEEP to refine the estimates.

SWEEP uses a Rayleigh distribution (Figure 2-1) to estimate the rate of discovery of remaining defects in a software system. The ideal graph of your test data would be a curve representing a Rayleigh distribution. You can use Mode 1 to produce a graph of your estimated defects that will have the same basic shape as the Rayleigh distribution. In addition, you can use Mode 1 to plot simultaneously the actual and estimated number of defects. These graphs offer an easy way to compare the estimates of your defect projections with your actual data. You can use the comparison as a management indicator to determine if you need more testing intervals, more test personnel, or if you need to make adjustments to your development schedule. For example, if the actual defect rates are lower than the estimates, project management may decide that adding additional test personnel will achieve the goals more quickly. On the other hand, if the actual defect rates are higher than the estimates, you may need to review your testing process. It is possible that you are injecting new defects into the system while the identified defects are being repaired.

You can also use Mode 1 to generate a cumulative curve that indicates the total number of defects found in test intervals. Figure 2-2 is an example of a cumulative curve generated by Mode 1. Experience has shown that when a software component is stable, the rate of defect discovery decreases with time (Davis and Gaffney 1988). The cumulative curve shows the point at which your rate of defect discovery is near zero, thus indicating that your software has reached a stable state. This type of curve provides a very valuable representation of when a project is nearing the end of testing.

2.1.1.3 Mode 2—Phase-Based Defect Predictions

Mode 2 of the SWEEP tool allows you to use a **phase-based** model to predict the number of defects you will find in each phase of the development life cycle. Mode 2 is most effective for organizations that perform static verification (e.g., inspections, reviews) during the entire life cycle. Mode 2 allows you to predict the number of defects that will be present in the software system before it enters the implementation, testing, and integration phases. Mode 2 has the following advantages over Mode 1:

Number
of Defects
Discovered
per Time
Interval
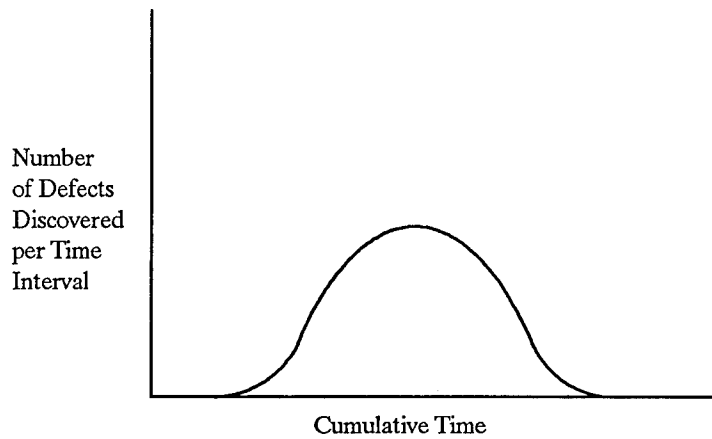
Cumulative Time

Figure 2-1. Typical Rayleigh Distribution Curve

Number
of
Cumulative
Defects

Time
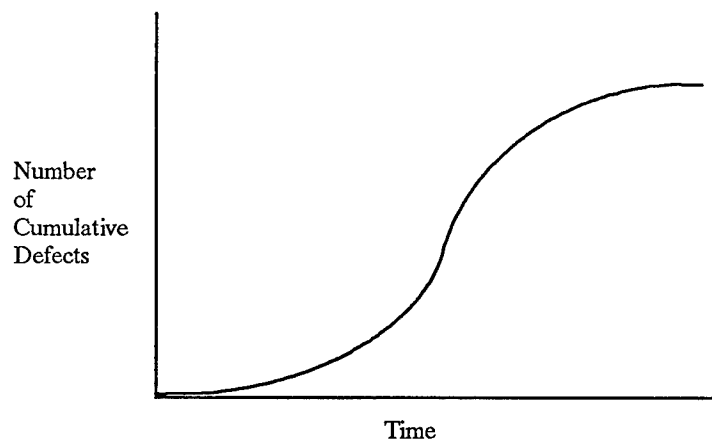
Figure 2-2. Cumulative Form of Rayleigh Distribution

- You can plan and schedule testing activities more accurately for the latter phases of development due to earlier prediction of defects.

- You can use your defect data to predict the discovery of defects in latter phases of the development life cycle.

- You can predict defects before your software is in an executable state.

- Your organization can compare the defect discovery histories for various software products.

SWEEP provides six default phases of a development process: preliminary design, detailed design, coding, unit test, integration test, and system test. You can customize the names of these phases, and you can add up to three additional phases to represent more accurately your development process. During the preliminary and detailed design phases, you collect defect data from static verification activities. Estimate an expansion ratio from the size of the design (such as number of data flow diagram bubbles or number of design pseudo-code statements) to the size of the code, based on the experience of your organization. For example, you may decide that a bubble in your design representation maps to about 500 source lines of code. Use the expansion ratio to calculate defect densities, and enter those numbers into SWEEP to produce an estimate of the defect latency for latter phases. As you progress

toward the latter development phases, input the new defect data to increase the accuracy of your projections.

The accuracy of defect density projection in Mode 2 is a function of the quality of the defect counts that you provide and a two-parameter Rayleigh distribution (see Appendix for details). The phase-based model requires defect counts from at least two different phases to develop a prediction. The phases need not be contiguous. The accuracy of the SWEEP estimation algorithm will improve proportionally as you enter actual defect counts from the other phases of your development process. The accuracy in Mode 2 will also be greater if you use the average of the densities found in several inspections, rather than just in one inspection.

Mode 2 covers the basic phases of the software development life cycle where verification activities are common practice. Use of Mode 2 with defect data obtained from the requirements activities of the software development process requires special consideration. As verification methods for the requirements phase mature, it may be feasible in future releases of SWEEP to add phases for requirements analysis and other systems engineering-related activities. For the present, defects detected during requirements analysis should be considered preliminary design defects.

2.1.1.4 Mode 3—SWEEP Planning Aid

Mode 3 of the SWEEP tool is a planning aid to help you develop estimated defect discovery and tolerance profiles. Mode 3 is most effective with, and designed for, organizations that maintain a history of latent defect counts over entire life cycles. You can use this historical data to estimate the total number of defects for each of the phases during a life cycle and the peak location of the distribution.

The planning aid allows you to set objectives for defect discovery ranges, based on user-selected criteria. You can enter data representing:

- Total number of defects injected per code unit and an estimate of your desired latent defect content

- Total number of defects injected per code unit and the estimated peak of the Rayleigh distribution

- Defect tolerances, based on either a percentage of the defects per phase or a signal-to-noise ratio

SWEEP computes:

- Distribution of defects per phase for each of the phases that you have defined

- Upper and lower defect tolerances, either in terms of upper and lower bounds or a signal-to-noise ratio

- Either the peak location of errors or the latent defect content

Using Mode 3, you can establish a desired goal for the defect discovery profile, and upper and lower tolerance profiles related to the projections. You define an acceptable defect discovery range by using the upper and lower tolerances profiles. This range, called the defect tolerance range, relates to the

control limits of your software quality control process. You can use the defect tolerance range to identify and anticipate the point(s) at which a software project deviates from this quality range. This will enable you to take earlier, more cost-effective actions to rectify problems in the development process. Both the establishment of defect discovery goals and the monitoring of progress against those goals are fundamental aspects of measurement-driven software management, equal in importance to establishing and monitoring cost, schedule, and size objectives.

You can establish goals for the total defect injection and latent defect content values in a number of ways. One way is to base these values on previous experience and take the values observed by earlier projects. You might determine that you have a better development process than previous projects and are willing to establish more ambitious quality goals. Your current development process may include the use of a more effective design methodology and computer-aided software engineering (CASE) tools. This new process could reduce the number of defects introduced in the system and might allow you to choose a smaller injected defect content. Your process may also include improved verification methods that would allow the distribution peak to occur earlier in the development process.

Alternatively, you might establish the parameters of the goal defect discovery profile based on the reliability requirements that the product must satisfy. For example, a software product might be required to have fewer than a specified number of latent defects for it to meet a reliability objective. The tolerance band about the goal defect discovery profile could correspond either to its ability to satisfy your customer's reliability requirement or the degree to which you believe that your process can control the level of defects introduced into your software. Another alternative for the selection of the defect discovery goal profile is a combination of customer requirements and general process improvement objectives (including verification) that have been established for your organization.

2.1.2 BENEFITS OF USING SWEEP

SWEEP provides support in two key process improvements activities, in addition to the benefit of improving your management of defects within a software product. First, SWEEP can provide improved management insight into both your development process and how well you are applying that process to developing products. Management personnel can have better access to more information earlier in the development process and be able to make better management decisions. Second, SWEEP generates valuable data for project planning and scheduling, thus improving your organization's ability to perform and meet milestones.

2.1.2.1 Program or Project Office

SWEEP has the potential to provide many benefits to organizations, such as program or project offices that are responsible for the development of software products and subsystems. The major capabilities of SWEEP that provide direct benefits to its users include the following:

- The **time-based model** (Mode 1) benefits project managers, quality managers, and test personnel by enabling them to predict more accurately how many test cycles will be needed to meet quality standards. This model also provides valuable project status information to project managers, thus enabling them to affect changes (i.e., adding new resources, revising schedules). This model can also be used to predict outages for complete systems.

- The **phase-based model** (Mode 2) benefits project members and related personnel by enabling them to predict the level of effort needed in each life-cycle phase to meet established

quality goals. This model extends the time-based model to include as many as nine phases. The phase-based model allows project managers to leverage their corporate history by using the average defect discovery rates from other projects as a starting point for their estimates.

- The **planning aid** (Mode 3) benefits project managers by helping them to establish defect discovery goals and defect-related quality goals. This aid allows users to leverage their historical data to produce accurate estimates of the latent defect content for various phases of the life cycle.

- **Cumulative curves** benefit the user by providing clear graphical representation of the number of defects discovered and a projection of the remaining defects. The cumulative curve is useful in determining when you have met your quality goals and helps you project how long it will take to reach those goals.

- **Incidence curves** benefit the user by providing side-by-side comparisons of actual and projected defect data. Incidence curves also provide a means to track your actual data against your projected data. This type of comparison provides an indication of the percentage of estimated defects removed from the system.

2.1.2.2 Centralized Engineering Organization

Centralized engineering or process metrics groups have a very different charter from that of project office and development teams. Still, these types of organizations can benefit from SWEEP. SWEEP provides a capability for establishing quality metrics, in terms of defect discovery rates, and for tracking progress against those goals. Organizations can use SWEEP to monitor progress against corporate goals and to increase management insight into development processes. The data can be used as part of a corporate or divisional process improvement plan. In addition, SWEEP can be beneficial to organizations seeking to improve the maturity of their software engineering capability.

Your organization can use the SWEEP tool to help you meet some of the goals of the key practices described in SEI's CMM. The CMM provides a model for determining the level of maturity of software engineering practices within organizations. The model defines five level, as shown in Table 2-1.

Table 2-1. Capability Maturity Model

Level	Description of Organization
Level 1: Initial Level	Organizations lack effective project management and do not maintain a solid, stable environment for developing and maintaining software systems.
Level 2: Repeatable Level	Organizations maintain policies and procedures for managing and developing software systems. Project planning is based upon experience with previously developed systems.
Level 3: Defined Level	Organizations have developed and documented a standard process for managing and developing software systems. This process embodies the policies and procedures developed at Level 2.

Table 2-1, continued

Level	Description of Organization
Level 4: Managed Level	Organizations set quantitative goals for both software products and development processes. Software processes use measurement instruments to collect important process and product metrics.
Level 5: Optimizing Level	Organizations are focused on continuous process improvement. Through the use of documented software processes and collected metrics, organizations can determine their weaknesses and develop strategies to correct them.

Each CMM maturity level prescribes a number of key practices that your organization must be able to demonstrate in order to be certified at that level. You can use SWEEP to develop and mature some of the capabilities you will need to meet the goals outlined in these key practices. The following sections briefly describe how you can use SWEEP within your organization to meet several of the goals of these key practices. Refer to Paulk et al. (1993) for a complete description of the CMM and its goals and activities for each level.

- **Level 2: Repeatable:** One goal of the CMM's software project planning practice is to document software estimates for use in planning and tracking a software project. Using Mode 3, project teams can develop an error discovery profile that indicates the number of defects for each phase of development. Project managers can use this data to help plan and schedule development and test resources.

  Two goals of the software project tracking and oversight practice are:

  - Actual results and performances are tracked against software plans.

  - Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.

  Project teams can use either Mode 1 or Mode 2 to develop error discovery profiles that indicate the number of defects discovered to date and that estimate the number of defects at latter stages of development. Using the error discovery profile, project managers can take corrective action by adjusting project plans and schedules.

  One goal for the software quality assurance practice is to verify objectively that software products and activities adhere to the applicable standards, procedures, and requirements. Project teams can use either Mode 1 or Mode 2 to track the project's progress against the quality goals of the project. Using both the incidence and cumulative curves, project managers can verify that their software products meet the defect removal goals set for the project.

- **Level 3: Defined:** One goal of the software product engineering practice is to ensure that software work products are kept consistent with each other. SWEEP provides a consistent means to estimate defect discovery profiles for software products. The profiles developed for each software unit will help to ensure that each unit meets the quality goals for the project. SWEEP can also be used in the measurement and analysis activities for this practice. These activities include tracking the number and severity of defects, both cumulatively and by stage.

- **Level 4: Managed:** One goal of the quantitative process management practice is to establish organization-wide quality goals and procedures. Project teams can use Mode 2 or Mode 3 to

utilize the defect discovery goals established by their organization. Projects can use the profiles generated by SWEEP (using the organization's quality goals) during project planning activities to estimate schedules and resource needs. SWEEP can help projects to estimate the amount of time and resources that testing will need in each phase. Also, SWEEP can help projects estimate if the current testing and development schedules will enable them to meet their quality goals. If it seems likely that current schedules will not enable a project to meet the organization's quality goals, project teams can take steps to correct their project plans.

Attaining the objectives of the process management practice includes the various operations to which SWEEP can contribute, such as establishing goals and measuring their degree of realization, and establishing tolerances on quality objectives and ensuring that they are not breached.

One of the goals of the software quality management practice is to quantify and manage the actual progress towards achieving the quality goals for the software product. Once an organization has established quality goals, project teams can use SWEEP to monitor their progress in removing defects to meet those goals. SWEEP provides both incidence and cumulative curves to help projects track their actual progress.

- *Level 5: Optimizing:* One of the goals of the defect prevention practice is to plan defect prevention activities. At this level, defect prevention activities, in addition to the normal testing and removal activities, are included in the software development plan. Project teams can use SWEEP to predict the discovery of defects at various stages of the development process. These predictions are used to develop strategies to prevent the injection of defects.

2.2 APPLYING SWEEP TO SOFTWARE DEVELOPMENT PROCESSES

SWEEP is applicable to any development process used by the project. Mode 1 is typically used during system test and integration activities, while Mode 2 is used throughout the life cycle. The following sections describe how you might use SWEEP on a project following a classic waterfall process and on a project using an Evolutionary Spiral Process (ESP).

2.2.1 CLASSIC WATERFALL PROCESS

The waterfall process model is an extension of a linear sequence of tasks. The standard waterfall permits a project to back track through the various phases until it returns to the phase in which a defect was introduced (Figure 2-3). The historical definition of the waterfall as a linear process restricts this backup to corrective actions only. The model does not allow iteration for any other reason.

You can apply Mode 1 to the testing and integration phases of the waterfall model. During the implementation phase, unit testing has identified and eliminated many of the defects in the system. You can then apply Mode 1 to project the number of defects remaining in the system, and you can estimate the number of test intervals it will take to reach the quality goals for the system.

You can apply Mode 2 after you have completed the preliminary and detailed design activities in the design phase. During these activities, you record the number of defects found during inspection and verification tasks. SWEEP calculates the estimated number of defects remaining in the system for each of the successive phases. As you progress through the phases, you replace SWEEP's projected
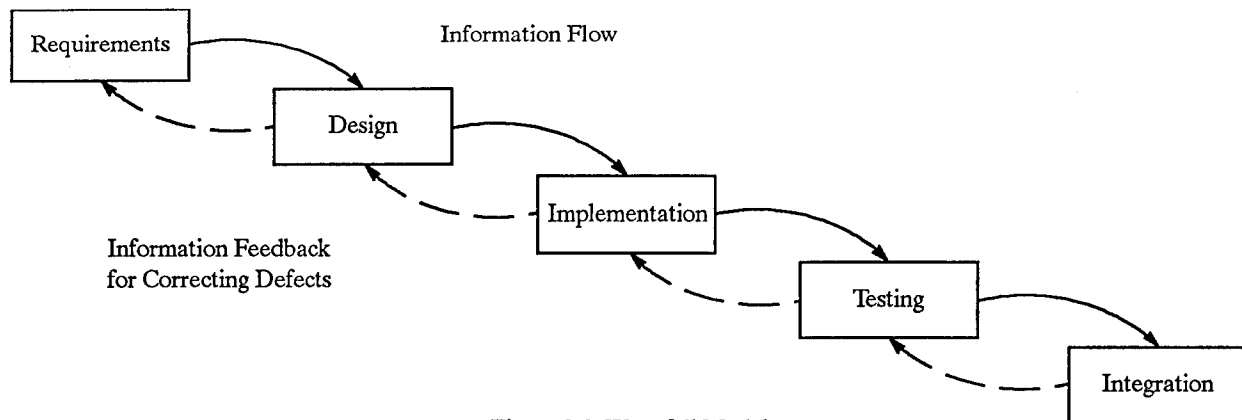
Figure 2-3. Waterfall Model

defect counts with your own actual counts for each phase. As you enter the actual defect counts, you increase the accuracy of SWEEP's projections for future phases.

Project managers can use Mode 3 to establish quality goals during the initial planning activities of the project. As the project progresses, management can use the planning aid during replanning activities to reestablish or reaffirm quality goals.

2.2.2 EVOLUTIONARY SPIRAL PROCESS

The ESP model (Software Productivity Consortium 1993) consists of five main steps. A cycle traverses all five steps, represented by a 360° rotation in the graph, denoting that some aspect of the product has matured by a specified amount (see Figure 2-4). A cycle starts in the upper left of the spiral with the step to determine objectives, alternatives, and constraints, and then moves clockwise to risk analysis and aversion, product development, and planning and management. The transition from the fifth step back to the first step of the following cycle is controlled by committing resources explicitly to continue the project.

The conceptual ESP model is essentially a management model. The model describes five main steps and several specific product and process management activities that can be used in conjunction with any life-cycle model. In addition, you can follow the ESP model activities to determine a life-cycle alternative that will adequately address your objectives and constraints, incorporate the life cycle into the model, and subsequently evolve the life cycle as product development proceeds and objectives and constraints change.

The ESP model is a repeating process that uses the knowledge and experience gained from the previous cycles to guide and improve future cycles that is, you traverse all five steps of the ESP model one cycle at a time. As noted above a cycle is a complete traversal of all five steps that completely mature the product by the amount defined in cycle-specific objectives and success criteria. A spiral is one or more cycles that combine to accomplish a specific objective, such as complete a project, product, work product, or other major milestone. A spiral may represent the complete life cycle or may include only those activities necessary to meet one or more of the life-cycle states.

Organizations can apply the Mode 1 to projects during the Develop Product step of the ESP model. You apply Mode 1 during the verification activities related to system testing and integration of this step. If the spiral for your project includes multiple cycles involving systems test and integration activities, you apply Mode 1 during the Develop Product step of **each** cycle. For example, if your spiral

1. Understand Context

2. Analyze Risks

Review Context

Perform Risk Analysis

Develop/Update Estimate of the Situation

Review Risk Analysis

Define Approach
- Stakeholders
- Objectives
- Alternatives
- Constraints

Plan Risk Aversion

Commit to Risk Aversion Strategy

Execute Risk Aversion

3. Plan Development

Review Alternatives

Commit to Proceed

Update Spiral Plan

Plan and Schedule

Commit to Plan

Review Progress

Develop and Verify Product

Product Change Control

Monitor and Review

Review Technical Product

4. Develop Product

5. Manage and Plan

Figure 2-4. Conceptual Evolutionary Spiral Process Model: A Management Focus

encompasses developing an alpha-test prototype, a beta-test prototype, and a final product for a software system, you apply Mode 1 during the Develop Product step for all three of these system releases.

Mode 2 could also be applied during the Develop Product step. The Develop and Verify Product activities within this step include efforts that relate to requirements definition, design creation, and product implementation. You collect and record defect data during these efforts and use the resulting SWEEP projections to influence your management decisions. As you progress through the cycles that define your spiral, you update your previous data values with your current defect counts or estimates. With each new update, you use SWEEP to produce more refined estimates of the latent defect content.

You may also be able to use your SWEEP projections from both Modes 1 and 2 to do the following:

- You can apply the information during the Risk Analysis activities of step 2, Analyze Risks.

- You can distinguish among multiple approaches to resolving a known risk. For example, if you are investigating two approaches to determine which is the better solution to a problem, you could use SWEEP to help determine which approach is better and which is more problematic.

- You can improve and refine your Plan and Schedule activities in step 3, Plan Development.

- You can develop better insight into actual status during the Review Progress activities in step 5, Manage and Plan.

The planning aid is useful during the Plan and Schedule activities in step 3, Plan Development. Project managers can use the estimates derived during this step to guide you when you establish the quality goals that you will use during the Review Progress activities of the Manage and Plan step.

2.3 SUMMARY

SWEEP is designed to assist you in establishing and managing software and system quality objectives. You can use SWEEP to make error projection using either time-based or development phase-based data. You can also use SWEEP in statistical process control activities, establishing and monitoring goals and tolerance ranges for error discovery at each stage of the development process.

SWEEP provides you with three modes of operation:

- Time-based Model

  You can use the time-based model to project the rate of error discovery during operation of the software or system based on data obtained in testing to help you decide upon the cost-effectiveness of continuing the test process.

- Phase-based Model

  You can use the phase-based model to predict the number defects in later development phases based on error data found during static verifications in earlier phases, before actual code execution. It also predicts the number of errors remaining in the software, the latent error content, when it is shipped.

- Planning aid

  You can use the planning aid to establish and monitor the error discovery objectives, on a phase by phase basis during the development process. You can also use planning aid in statistical process control activities to help you develop higher quality software.

You can apply SWEEP in many organization functions such as project or program offices and centralized engineering organizations that are interested in software quality assessment and measurement, and software process improvement. In addition, you can apply SWEEP in many software development processes, ranging from the classic waterfall process to the Evolutionary Spiral Process.

3. INSTALLING AND STARTING SWEEP

SWEEP is provided as a set of Microsoft Excel files on a diskette. The distribution diskette contains files with approximately 1 megabyte of disk space. SWEEP was developed using Microsoft Windows 3.1 and Microsoft Excel 4.0. SWEEP operates on an IBM-compatible PC or a Macintosh. Version 4.0 (or later) of Microsoft Excel must be installed on your computer before installing SWEEP.

The installation and operation instructions for SWEEP assume that you have a working knowledge of the computer system you will be using. For the PC implementation, you should be familiar with Microsoft DOS and Windows. For the Macintosh implementation, you should be familiar with using the Finder and other basic Macintosh tools. You do not need to be familiar with Microsoft Excel to use the SWEEP tool, but you may find some experience with Excel useful when using SWEEP.

3.1 SWEEP UPGRADE

Enhancements for SWEEP 2.0 upgrade SWEEP from Excel 3.0 to Excel 4.0.

The following characterize the enhancements made to the time-based model, Mode 1:

- The ability to partition input data into groups has been added.

- Improvements to the basic Rayleigh curve fitting algorithm have been implemented.

- The graphical presentation of the resultant cumulative and incidence curves have been improved.

The following characterize the enhancements made to the phase-based model, Mode 2, and the planning aid, Mode 3:

- The ability to create, change, delete, show, and select activity sets has been added.

- Modes 2 and 3 have been extended to handle between six and nine phases.

- The graphical presentation to handle a variable number of phases (between six and nine) has been improved.

3.2 USING SWEEP ON THE PC

3.2.1 RECOMMENDED MINIMUM CONFIGURATION

The recommended minimum configuration for running the PC version of SWEEP is:

- 80386 processor (tested on an 80486 processor)

- VGA monitor (tested on Super VGA)

- 4 megabytes of random access memory (RAM)

- MS-DOS, version 5.0 (tested on version 5.0)

- Microsoft Windows, version 3.1 or later (tested on version 3.1)

- Microsoft Excel, version 4.0 or later (tested on version 4.0)

- Enough space on the hard disk for the SWEEP files (1 megabyte)

3.2.2 UPGRADING SWEEP 1.0

The data files created under SWEEP 1.0 are not compatible with SWEEP 2.0. The data should be reentered into SWEEP 2.0.

3.2.3 INSTALLING SWEEP ON A HARD DISK

To install the SWEEP software on a hard disk, perform the following steps:

- Start Windows.

- Create a subdirectory named sweep under the excel directory on the hard disk (i.e., C:\excel\sweep).

- Insert the disk containing SWEEP into the disk drive.

- Copy all files from the diskette to this subdirectory.

- Open the Microsoft Excel program group.

- Select File from the menu bar and then select New from the choice of menu options. The New Program Object property sheet appears.

- Click the Program Item line and click [OK]. The Program Item Properties dialog box appears.

- In the Description field, type SWEEP, then press the <TAB> key or move the mouse to the Command line field.

- In the Command line field, type C:\excel\sweep\GO_SWEEP.XLM, then press the <TAB> key or move the mouse to the Working Directory field.

- In the Working Directory field, type C:\excel\sweep, then press <RETURN> or click [OK].

Installation of SWEEP on your hard disk is now complete.

3.2.4 INVOKING SWEEP ON THE PC

3.2.4.1 Invoking SWEEP From a Diskette

To invoke SWEEP from a diskette, perform the following steps:

- Start Windows.

- Make sure the diskette is not write protected.

- Place the diskette containing SWEEP into the disk drive.

- Double-click on the Microsoft Excel icon to start Microsoft Excel.

- Select the File and Open options.

- In the Directories list box, select the drive (a: or b:) into which you have inserted the diskette and click [OK]. The files contained on the diskette are now displayed in the Files list box.

- Select GO_SWEEP.XLM and click the [OK] button. The software loads and displays the Software Productivity Consortium logo with [Begin] and [End] buttons.

- Click the [Begin] button to dismiss the logo. The File and Modes pulldown menus appear.

You are now ready to use SWEEP. Its operation is described in Section 4.

3.2.4.2 Invoking SWEEP From a Hard Disk

To invoke SWEEP from a hard disk, perform the following steps:

- Start Windows.

- Open the Microsoft Excel program group.

- Double-click on the SWEEP icon. The software loads and displays the Software Productivity Consortium logo with [Begin] and [End] buttons.

- Click the [Begin] button to dismiss the logo. The File and Modes pulldown menus appear.

You are now ready to use SWEEP. Its operation is described in Section 4.

3.3 USING SWEEP ON THE MACINTOSH

3.3.1 RECOMMENDED MINIMUM CONFIGURATION

The recommended minimum configuration for running the Macintosh version of SWEEP is:

- A Macintosh capable of running Microsoft Excel, version 4.0

- 4 megabytes of RAM with default memory setting (tested on 24 megabyte)

- Finder, version 7.01 or later (tested on version 7.1.1)

- Microsoft Excel, version 4.0 or later (tested on version 4.0)

- Enough space on the hard disk for the SWEEP files (1 megabyte)

3.3.2 UPGRADING SWEEP 1.0

The data files created under SWEEP 1.0 are not compatible with SWEEP 2.0. The data should be reentered into SWEEP 2.0.

3.3.3 INSTALLING SWEEP ON A HARD DISK

To install SWEEP on your hard disk, perform the following steps:

- Double-click on the hard disk icon. A window displaying the hard disk contents appears.

- Place the SWEEP diskette in the disk drive.

- Double-click on the diskette icon. A window displaying the diskette contents appears.

- Copy the SWEEP folder by clicking once on the SWEEP icon and dragging it to the hard disk icon. Release the mouse button to start the copy procedure. The contents of the folder are placed in a folder named SWEEP.

Installation of SWEEP on your hard disk is now complete.

3.3.4 INVOKING SWEEP ON THE MACINTOSH

3.3.4.1 Invoking SWEEP From a Diskette

To invoke SWEEP from a diskette, perform the following steps:

- Make sure the diskette is not write protected.

- Place the diskette containing SWEEP in the disk drive.

- Open the diskette by double-clicking on the diskette icon. A window displaying the diskette contents appears.

- Open the SWEEP folder by double-clicking on the folder icon. A window displaying the folder contents appears.

- Start the SWEEP application by double-clicking on the file GO_SWEEP.XLM. The software loads and displays the Software Productivity Consortium logo with [Begin] and [End] buttons.

- Click the [Begin] button to dismiss the logo. The File and Modes pulldown menus appear.

You are now ready to use SWEEP. Its operation is described in Section 4.

3.3.4.2 Invoking SWEEP From the Hard Disk

To invoke SWEEP from a hard disk, perform the following steps:

- Open the hard disk by double-clicking on the hard disk icon. A window displaying the hard disk contents appears.

- Open the SWEEP folder by double-clicking on the folder icon. A window displaying the folder contents appears.

- Start the SWEEP application by double-clicking on the file GO_SWEEP.XLM. The software loads and displays the Software Productivity Consortium logo with [Begin] and [End] buttons.

- Click the [Begin] button to dismiss the logo. The File and Modes pulldown menus appear.

You are now ready to use SWEEP. Its operation is described in Section 4.

3.4 LIST OF FILES INCLUDED IN THE PC VERSION OF SWEEP

The PC version of SWEEP provides the following files:

MODE1A.XLM	MODE1A1.XLC	MODE1A2.XLC
MODE1AIN.XLS	MODE1A_G.XLS	MODE1A_D.XLS
MODE2A.XLM	MODE2A1.XLC	MODE2A2.XLC
MODE3A.XLM	MODE3A1.XLC	ACTIVITY.XLM
MENUS_AF.XLM	GO_SWEEP.XLM	SPC_LOGO.XLS
SWEEP.XLW	MODE1A.C	SWEEP.DLL
BC30RTL.DLL		

3.5 LIST OF FILES INCLUDED IN THE MACINTOSH VERSION OF SWEEP

The Macintosh version of SWEEP provides the following files:

MODE1A.XLM	MODE1A1.XLC	MODE1A2.XLC
MODE1AIN.XLS	MODE1A_G.XLS	MODE1A_D.XLS
MODE2A.XLM	MODE2A1.XLC	MODE2A2.XLC
MODE3A.XLM	MODE3A1.XLC	ACTIVITY.XLM
MENUS_AF.XLM	GO_SWEEP.XLM	SPC_LOGO.XLS
SWEEP.XLW	SWEEP	

This page intentionally left blank.

4. OPERATING SWEEP

4.1 USER INTERFACE CONVENTIONS

This section provides a description of the SWEEP user interface and a detailed example for each of the three SWEEP modes. Section 5 provides a case study describing the operation of the time-based model, Mode 1. Whenever possible, SWEEP adheres to the user interface convention for Microsoft Excel. These conventions help to protect you from accidently modifying the macros and software used to implement SWEEP. For more information about Microsoft Excel and its interface conventions, please refer to the *Microsoft Excel User's Guide* (Microsoft 1991, ix-xii).

This section also includes notes on specific aspects of using the SWEEP tool. Notes include warnings and explanations. Some notes are specific to the PC environment, some are for the Macintosh environment, and some notes are for both systems.

Notes: The graphics included in this section are from the Macintosh implementation of SWEEP. The dialog boxes and windows for the PC implementation have a similar appearance.

You may not have enough data points for SWEEP to properly fit a Rayleigh curve to your data set. In these cases, your results will not converge. You will observe that the incidence curve will continue to rise over the data set that you entered.

4.1.1 INVOKING SWEEP

To invoke SWEEP for a PC system, follow the instructions provided in Section 3.2.4. To invoke SWEEP for a Macintosh system, follow the instructions in Section 3.3.4. SWEEP first presents you with the Consortium's copyright notice (see Figure 4-1). This window consists of two parts: a static text area and a custom menu bar. The static text area displays the Consortium logo and the copyright notice. The menu bar features [Begin] and [End] buttons. Click on the [Begin] button to continue with the SWEEP tool. Click the [End] button to quit the tool.

4.1.2 SELECTING FROM MENUS

Make a menu selection by holding down the mouse button and dragging the mouse over the menu selections until you highlight your desired choice. Release the mouse button on the selected option to execute your choice.

4.1.3 THE FILE MENU

Figure 4-2 shows the menu hierarchy for SWEEP's functions. The File menu presents the following four options: Quit, Close, Save, and Print Preview. Initially, all the File menu options are disabled and
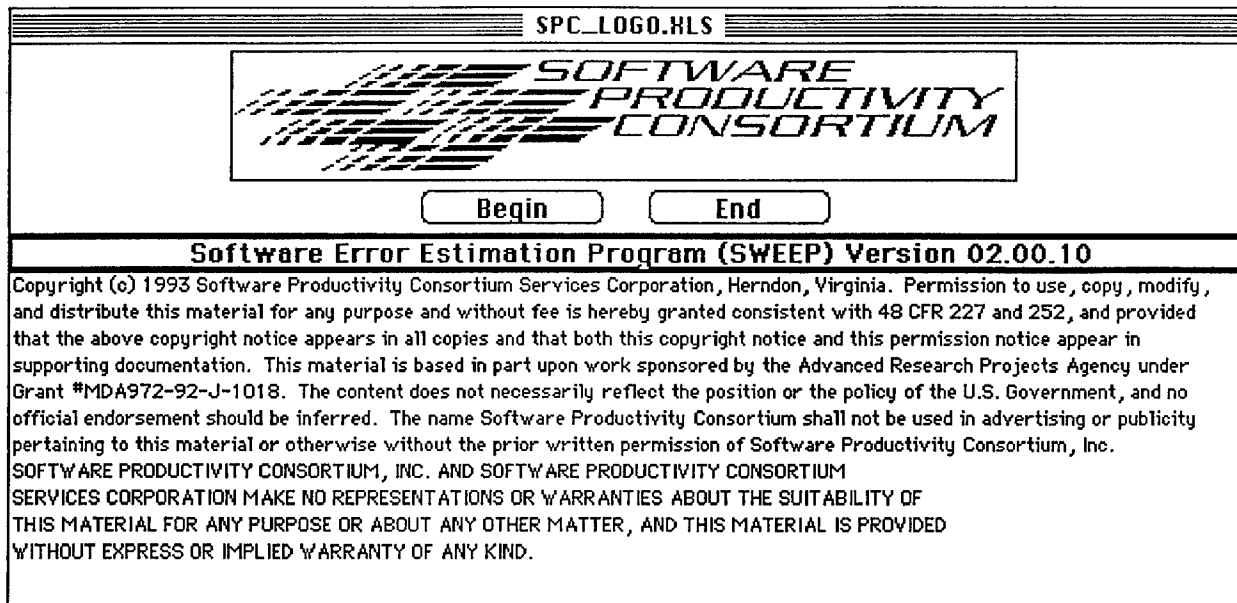
Figure 4-1. Copyright Notice for the SWEEP Tool
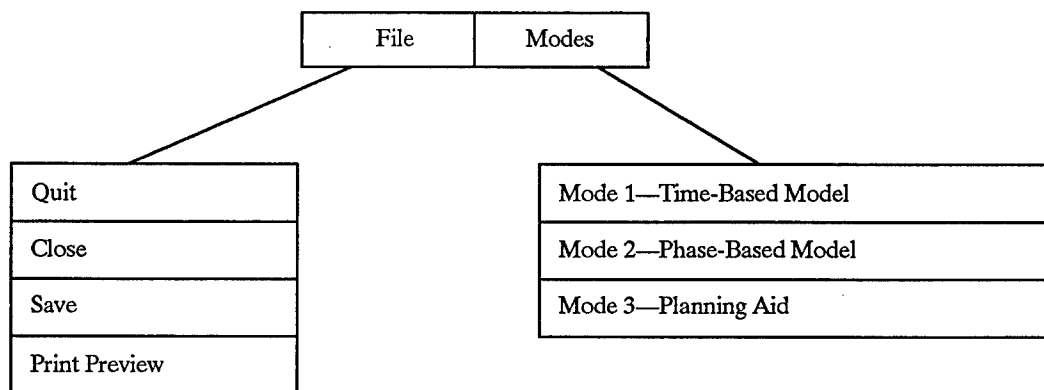
Figure 4-2. SWEEP Menu Hierarchy

dimmed except for the Quit option. The File menu options are active only while you are working in one of the modes. The File menu options and related caveats are described in Sections 4.2.8 (Mode 1), 4.3.7 (Mode 2), and 4.4.7 (Mode 3).

4.1.4 THE MODES MENU

The Modes menu allows you to choose one of SWEEP's three modes of operation:

- Mode 1—Time-Based Model

- Mode 2—Phase-Based Model

- Mode 3—Planning Aid

To choose a SWEEP mode, select one of the modes listed from the Modes menu. After a selection is made, all three modes are dimmed until you complete the selected mode. Following the selection

of any mode, you are presented with a series of dialog boxes requesting you to enter data. Dialog boxes also are used to display the calculated output data.

4.2 MODE 1—TIME-BASED MODEL OPERATION

To use Mode 1, specify the number of time intervals (n) and enter the number of errors discovered during each interval. Time intervals may be weeks, days, or other user-specified time period. SWEEP fits your data to a two-parameter curve. One parameter is E, the total number of errors expected to be made throughout software development and discovered over the total life cycle. The other variable is the (T_p) the peak rate of defect incidence occurs (T_p). To ensure that SWEEP produces a valid estimate in Mode 1 operation, e.g., be careful when entering error counts. The error discovery data should be for the **same** unit of code size (e.g., KSLOC, function points) in all test intervals used. If the size of the code changes after the test period, normalize the error discovery data by the size of the code at that time (e.g., if you measure code size by KSLOC, use errors/KSLOC instead of errors). SWEEP then estimates E as lifetime errors/KSLOC. Mode 1 is typically used for periodic error input and tracking; therefore, SWEEP maintains your data from session to session.

To start the time-based model, select Mode 1—Time-Based Model from the Modes menu. The Number of Intervals dialog box (see Figure 4-3) appears.
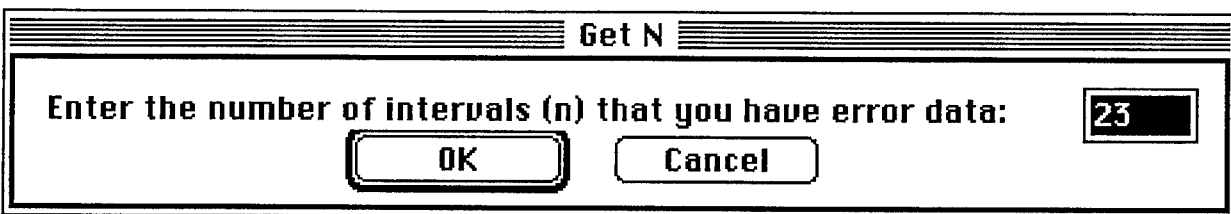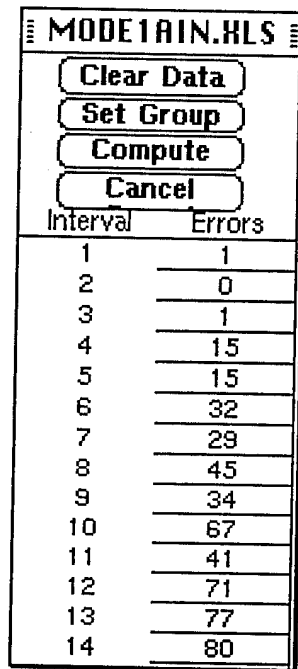
Figure 4-3. Number of Intervals Dialog Box

4.2.1 NUMBER OF INTERVALS DIALOG BOX

The dialog box presents one input field, an [OK] button, and a [Cancel] button. Enter in the input field the number of weeks or intervals over which error data has been recorded. The input value must be greater than or equal to 2 and less than or equal to 100. After you enter the number of intervals, click [OK] or press the <RETURN> key. If you click the [OK] button or press the <RETURN> key without entering any data, an error message will be displayed indicating that data must be entered. Once you have entered the number of intervals, SWEEP will close the dialog box and open a Data input box (see Figure 4-4).

The Data input dialog box allows you to enter the error data for each of the completed test intervals. It offers a [Clear Data] button, a [Set Group] button, a [Compute] button, a [Cancel] button, and two columns of data. The left column is a reference column listing the number of intervals. The left column extends down to the maximum number of intervals (100). The right column is the number of defects recorded for that interval. Input a value for the number of errors that were discovered for a particular time interval. If you do not have data for a particular interval, do not enter a value in the errors column. You must enter a value, however, for at least two intervals. If you enter a space or some other non-numeric character, SWEEP will display an error message after you click the [Compute] button. After

WARNING: Do not enter a slash (e.g., 1/2) in the Intervals Dialog Box. Excel will interpret it as a date and change it into a date cell. To correct this, type 0 1/2 to return the cell to a numeric cell.

MODE1AIN.XLS	
Clear Data	
Set Group	
Compute	
Cancel	
Interval	Errors
1	1
2	0
3	1
4	15
5	15
6	32
7	29
8	45
9	34
10	67
11	41
12	71
13	77
14	80

Figure 4-4. Example of a Data Input Box

data entry is completed, click the [Compute] button to compute the error projections. If you click the [Cancel] button while this dialog box is displayed, you will return to the Main window.

You can erase all previously entered data by clicking the [Clear Data] button. This erases all input data from the errors column (the right column). SWEEP will ask you to confirm that you wish to clear the data before any data is deleted (see Figure 4-5).

⚠ **Selecting OK will erase all data entered in the Error Column. Are you sure?**   OK   Cancel
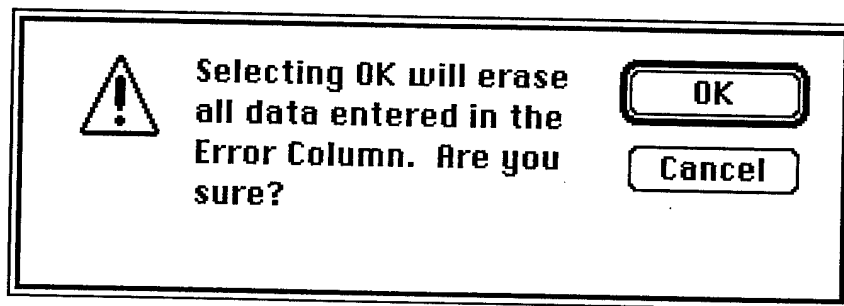
Figure 4-5. Clear Data Confirmation Dialog Box

EXAMPLE: You have error data for 13 weeks. Enter "13" into the Number of Intervals dialog box, and click [OK]. Click the [Clear Data] button in the Data input dialog box, click the [OK] button in the Clear Data Confirmation dialog box, and then enter your error data, one value per cell:

1 2 1 3 4 8 5 1 10 8 9 6 4

After your error data has been entered, you can click either the [Set Group] or the [Compute] buttons. If you click the [Set Group] button, the Grouping input box appears (see Figure 4-6). This box allows

Figure 4-6. Grouping Input Box

you to group the error data for analysis. The grouping function organizes consecutive intervals into groups. For example, if you select two intervals per group, then the values for intervals 1 and 2 will be added together as group 1; intervals 3 and 4 will be added together for group 2; etc. If the error data is sparse (many intervals have only a few or no errors), you will improve the accuracy of your estimates by grouping interval error data into larger groups. Grouping can also smooth your data and may improve the accuracy with which SWEEP plots your data. Input the number of intervals to be included in each group. If the number you select for your intervals per group does not divide evenly into the number of intervals in your data set, then you must decide whether to discard extra intervals from the beginning of your data or the end of your data. Select whether to trim data from the front or the back of the entered error data.

NOTE: Set Group will treat a grouping of null values as 0. To obtain a result that reflects null values, reenter your grouped data in the Intervals Dialog Box and replace the 0 with a null value.

EXAMPLE: If you have 23 intervals and you select 2 intervals per group, you will have an extra interval that cannot be put in a group. You must decide if you wish the first interval or the last to be trimmed. (The value will be ignored only by the grouping function; it will not be deleted from your data set.)

Click the [Show Group] button to display the Grouping display box, which shows the number of errors in each group (see Figure 4-7). Several minutes may elapse before this box is displayed, depending on the number of intervals you have entered and the speed of your computer.

EXAMPLE: You want to combine successive error intervals, discarding the last interval. In the Grouping input box, enter 2 into the Number of Intervals Per Group input field, select Trim Data from Back, and then click the [Show Group] button. The Grouping display box then shows the grouped data (a portion of the original error data is displayed in Figure 4-4).

```
┌─────────────────────────────┐
│ ☰ MODE1A_G.HLS ☰            │
│    ┌──────────────┐          │
│    │   Compute    │          │
│    └──────────────┘          │
│    ┌──────────────┐          │
│    │    Cancel    │          │
│    └──────────────┘          │
│                              │
│                              │
│     Group      Errors        │
│       1    ┌──────┐          │
│            │   1  │          │
│       2    └──16──┘          │
│       3       47             │
│       4       74             │
│       5      101             │
│       6      112             │
│       7      157             │
│       8      122             │
│       9      152             │
│      10       99             │
│      11      102             │
│      12                      │
│      13                      │
│      14                      │
└─────────────────────────────┘
```

Figure 4-7. Grouping Display Box

4.2.2 ESTIMATED ERRORS DIALOG BOX

When you click the [Compute] button in either the Data Input or Grouping display boxes, error projections are computed and the Estimated Errors dialog box (see Figure 4-8) appears.

```
┌──────────────────────────────────────────────────────────────┐
│ ≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡≡ Estimated Errors ≡≡≡≡≡≡≡≡≡≡≡≡≡≡               │
│                                                                │
│   Errors Discovered to Date: 1013                              │
│   Total Errors Projected: 1455.04                              │
│   Percentage of Projected Errors Found to Date: 69.62          │
│   Estimated Location of Peak: 14.90                            │
│  ┌─Error Estimate Selection─────────────────────────────────┐ │
│  │ ● Estimate Number of Errors in Next (m) Intervals        │ │
│  │ ○ Determine Number of Intervals Required to Achieve      │ │
│  │     the Following % (p), based on the Total Number of    │ │
│  │     Errors Projected                                     │ │
│  └──────────────────────────────────────────────────────────┘ │
│   Enter Data (m or p):  ┌────────┐  ┌────────────┐  ┌────────┐ │
│                         │ 10     │  │  Compute   │  │ Cancel │ │
│                         └────────┘  └────────────┘  └────────┘ │
│  ┌──────────────────┐  ┌──────────────────┐  ┌──────────────┐ │
│  │ Cumulative Curve │  │  Incidence Curve │  │  Data Sheet  │ │
│  └──────────────────┘  └──────────────────┘  └──────────────┘ │
└──────────────────────────────────────────────────────────────┘
```

Figure 4-8. Using (m) in the Estimated Errors Dialog Box

The dialog box displays the following four outputs based on the user inputs:

- **Errors Discovered to Date.** The sum of the number of errors you have entered.

- **Total Errors Projected.** The estimated number of errors injected in the software.

- **Percentage of Projected Errors Found to Date.** The percentage of total errors estimated to be discovered through interval (n).

- **Estimated Location of Peak.** The peak of the Rayleigh curve that SWEEP fits to your data.

The dialog box also features five buttons:

- **Compute.** Either estimate the number of errors in the next intervals or determine the number of intervals required to remove the given percentage of errors.

- **Cumulative Curve.** Draw the cumulative curve for this data set.

- **Incidence Curve.** Draw the incidence curve for this data set.

- **Cancel.** Close the Estimated Errors dialog box and return to the Data Input or Grouping dialog box.

- **Data Sheet.** Display the underlying data sheet.

The two radio buttons in the Error Estimate Selection segment of the dialog box enable you to select one of the two types of estimate that SWEEP calculates. These buttons are used in conjunction with the [Compute] button.

The first radio button option can help you determine if testing should be performed for a given interval, based on the number of errors expected to be discovered. This option allows you to estimate the number of errors projected in the next (m) intervals. If you select this option, you must enter an integer value greater than zero. SWEEP computes the total number of errors expected to be discovered in the next (m) intervals and provides an estimate of the percentage of total errors that (m) and the initial (n) intervals entered represent.

Select the first radio button option, enter a positive integer for (m), and click the [Compute] button. This opens the Calculated Values Based on (m) dialog box (discussed in Section 4.2.3 and shown in Figure 4-9). The dialog box displays outputs computed as a result of your input for (m). Clicking the [OK] button returns you to the Estimated Errors dialog box.

EXAMPLE: You want to estimate how many errors will occur in the next 10 test intervals. Select Estimate Number of Errors, then enter 10 in the Enter Data input field, and click the [Compute] button.

The second radio button option can help you determine how many more test intervals are required to detect a given percentage of the injected errors. This option allows you to enter a percentage (p), where (p) is less than 100 and greater than the percentage of errors discovered to date. SWEEP indicates both the number of intervals required to achieve that percentage and how many intervals beyond the (n) intervals are left to achieve (p).

```
┌─────────────────────────────────────────────────────────────────┐
│ ═══════════════════════════ Calculate M ═══════════════════════ │
│ ┌───────────────────────────────────────────────────────────┐   │
│ │ ┌─Estimated Errors based on Intervals Entered (n)────────┐ │   │
│ │ │   Number of Intervals Entered:  23                     │ │   │
│ │ │   Total Errors Discovered to Date:  1013               │ │   │
│ │ │   Total Errors Projected:  1455.04                     │ │   │
│ │ │   Intervals Needed to Achieve 99.99% of Total Errors: 63.95 │ │
│ │ └────────────────────────────────────────────────────────┘ │   │
│ │ ┌─Estimated Total  Errors based on Intervals through m (n+m)─┐ │
│ │ │   Number of Intervals Estimated (m):  10               │ │   │
│ │ │   Total Errors Discovered through Interval (m):  1329.80 │ │ │
│ │ │   Estimated Errors in (m) Intervals:  316.80           │ │   │
│ │ │   Percentage of Total Errors:  91.39                   │ │   │
│ │ └────────────────────────────────────────────────────────┘ │   │
│ │                  ┌──────────────────┐                       │   │
│ │                  │       OK         │                       │   │
│ │                  └──────────────────┘                       │   │
│ └───────────────────────────────────────────────────────────┘   │
└─────────────────────────────────────────────────────────────────┘
```

Figure 4-9. Calculated Values Based on (m) Dialog Box

Select the second radio button option, enter a value for (p), and click the [Compute] button. This opens the Calculated Values Based on (p) dialog box (discussed in Section 4.2.4), which displays outputs computed from your input for (p). Clicking the [OK] button returns you to the Estimated Errors dialog box.

4.2.3 CALCULATED VALUES BASED ON (m) DIALOG BOX

The Calculated Values Based on (m) dialog box, shown in Figure 4-9, features eight output fields and an [OK] button.

The output fields provide you with the following information:

- *Number of Intervals Entered (n).* The number of test intervals for which you have data.

- *Total Errors Discovered to Date.* The sum of the error counts that you have entered through interval (n).

- *Total Errors Projected.* The estimated total number of errors injected into the software.

- *Intervals Needed to Achieve 99.99 Percent of Total Errors.* The number of test intervals estimated to be required to discover 99.99 percent of the total errors projected.

- *Number of Intervals Estimated (m).* The number of intervals beyond the number of intervals entered (n) for which you would like to estimate the number of errors to be discovered.

- *Total Errors Discovered Through Interval (m).* The estimated number of errors expected to be discovered through (n+m) intervals.

- *Estimated Errors in (m) Intervals.* The number of errors estimated to be discovered during the next (m) intervals.

- *Percentage of Total Errors.* The percentage of total errors estimated to be discovered through (n+m) intervals.

Clicking the [OK] button returns you to the Estimated Errors dialog box.

4.2.4 CALCULATED VALUES BASED ON (p) DIALOG BOX

You can also estimate the number of intervals required to achieve a given percentage of defects discovered in the system. In the Estimated Errors dialog box, select the second radio button option in the Error Estimate Selection area (see Figure 4-10). Click the [Compute] button. The Calculated

```
╔══════════════ Estimated Errors ══════════════╗
║                                                ║
║  Errors Discovered to Date: 1013               ║
║  Total Errors Projected:  1455.04              ║
║  Percentage of Projected Errors Found to Date:  69.62  ║
║  Estimated Location of Peak:  14.90            ║
║ ┌Error Estimate Selection─────────────────────┐║
║ │ ○ Estimate Number of Errors in Next (m) Intervals │║
║ │ ◉ Determine Number of Intervals Required to Achieve │║
║ │   the Following % (p), based on the Total Number of │║
║ │   Errors Projected                           │║
║ └──────────────────────────────────────────────┘║
║  Enter Data (m or p):   ┌──────┐  ┌──────────┐ ┌────────┐║
║                         │99.97 │  │ Compute  │ │ Cancel │║
║                         └──────┘  └──────────┘ └────────┘║
║ ┌─────────────────┐ ┌─────────────────┐ ┌────────────┐ ║
║ │ Cumulative Curve│ │ Incidence Curve │ │ Data Sheet │ ║
║ └─────────────────┘ └─────────────────┘ └────────────┘ ║
╚════════════════════════════════════════════════╝
```

Figure 4-10. Using % (p) in the Estimated Errors Dialog Box

Values Based on (p) dialog box, depicted in Figure 4-11, appears. This dialog box features seven output fields and an [OK] button. The output fields provide the following information:

- *Number of Intervals Entered.* The number of test intervals for which you have data.

- *Total Errors Discovered to Date.* The sum of the number of errors for which you have entered data.

- *Total Errors Projected.* The estimated total number of errors to be injected into the software.

- *Percentage of Estimated Errors Discovered.* The percentage of total errors discovered through interval (n).

- *Percentage (p) Entered for Estimate.* The value is input to determine how many more intervals are required to achieve a specified percentage (p) of the total errors projected.

```
┌─────────────────────────────────────────────────────────────┐
│                     ≡ Calculate P ≡                          │
│ ┌───────────────────────────────────────────────────────┐   │
│ │ ┌Estimated Errors based on Intervals Entered (n)─────  │   │
│ │ │   Number of Intervals Entered: 23                    │   │
│ │ │   Total Errors Discovered to Date: 1013              │   │
│ │ │   Total Errors Projected: 1455.04                    │   │
│ │ │   Percentage of Projected Errors Discovered: 69.62   │   │
│ │ └──────────────────────────────────────────────────    │   │
│ │ ┌Estimated Errors based on Percentage (p)───────────   │   │
│ │ │   Percentage (p) Entered for Estimate: 99.97         │   │
│ │ │   Intervals Needed to Achieve (p): 60.01             │   │
│ │ │   Intervals Remaining After (n) to Achieve (p): 37.01│   │
│ │ └──────────────────────────────────────────────────    │   │
│ │              ╔═══════════════════╗                     │   │
│ │              ║        OK         ║                     │   │
│ │              ╚═══════════════════╝                     │   │
│ └───────────────────────────────────────────────────────┘   │
└─────────────────────────────────────────────────────────────┘
```

Figure 4-11. Calculated Values Based on (p) Dialog Box

- *Intervals Needed to Achieve (p).* The estimated total number of test intervals required to achieve (p).

- *Intervals Remaining After (n) to Achieve (p).* The number of intervals beyond (n) to achieve (p).

Clicking the [OK] button returns you to the Estimated Errors dialog box.

4.2.5 CUMULATIVE CURVE CHART

From the Estimated Errors dialog box (see Figure 4-10), you can generate two different graphs of error data: the cumulative curve and incidence curve charts. Click the [Cumulative Curve] button to display a chart showing both the cumulative number of actual errors per input interval and the estimated cumulative errors (see Figure 4-12). When the chart is displayed, use the File menu options to close, save, or print the chart (see Section 4.2.8).

4.2.6 INCIDENCE CURVE CHART

In the Estimated Errors dialog box, click the [Incidence Curve] button to display a chart showing the number of actual errors per input interval and estimated errors (see Figure 4-13). When the chart is displayed, use the File menu options to close, save, or print the chart (see Section 4.2.8).

4.2.7 THE DATA SHEET

The Data Sheet option on the Estimated Errors dialog box provides you access to the data used to produce both the incidence and the cumulative curves (see Figure 4-14). You may use the File menu options to close, save, or print the data sheet (see Section 4.2.8). The data sheet presents the following data values:

- *Errors Discovered to Date.* The sum of the number of errors you have entered.

- *Total Errors Projected.* The estimated number of errors injected in the software.
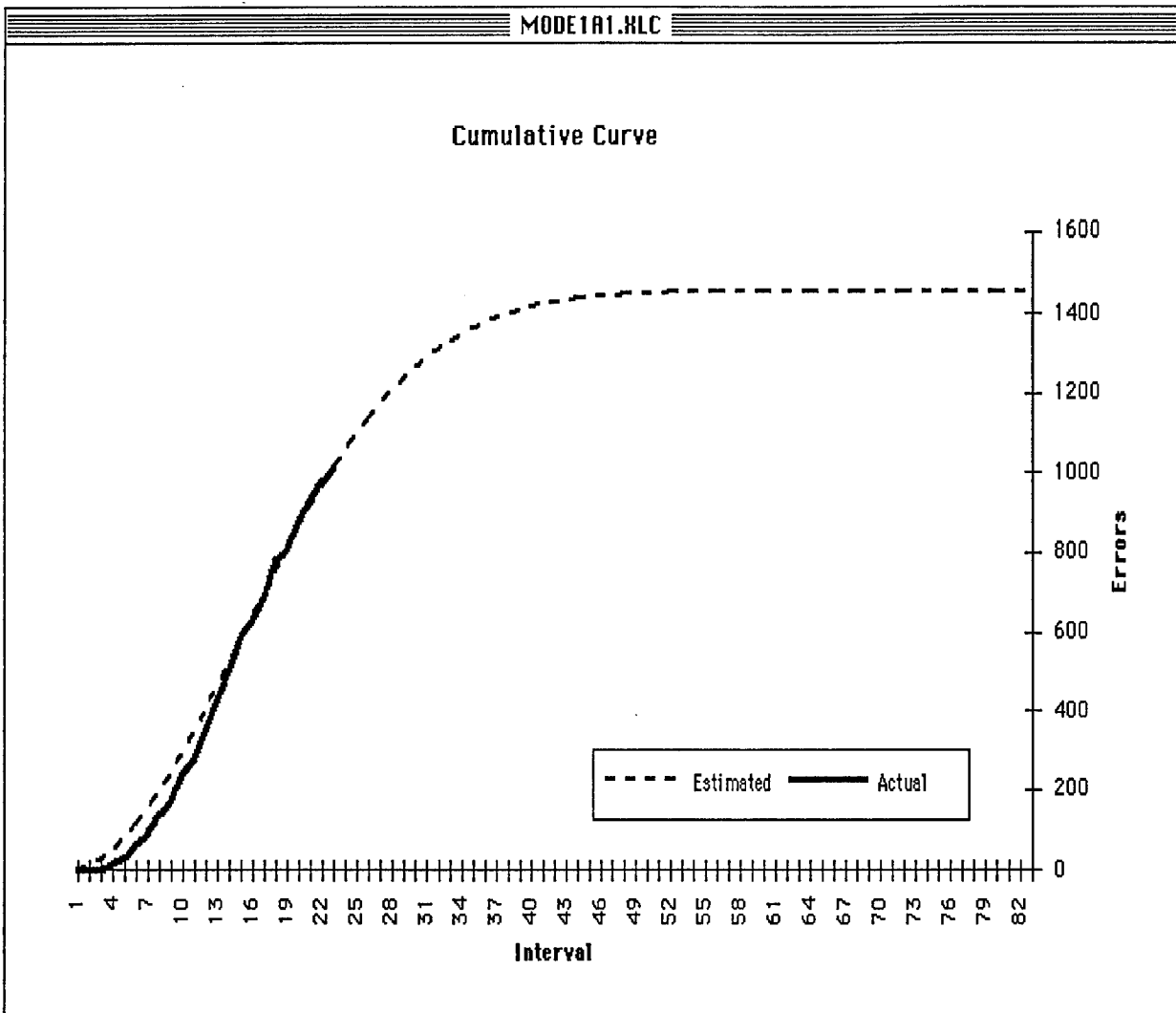
Figure 4-12. Cumulative Curve Chart

- *Percentage of Projected Errors Found to Date.* The percentage of total errors estimated to be discovered through interval (n).

- *Estimated Location of Peak.* The peak of the Rayleigh curve that SWEEP fits to your data.

In addition, the SWEEP data sheet lists the following values for each test interval:

- *Interval.* Test interval number.

- *Actual Error.* Your input data.

- *Estimated Error.* The number of errors estimated by SWEEP.

- *Error Delta.* The difference between the Actual Error and Estimated Error.

- *Relative Delta (percentage).* The difference between the Error Delta and the Actual Error, divided by the Actual Error.
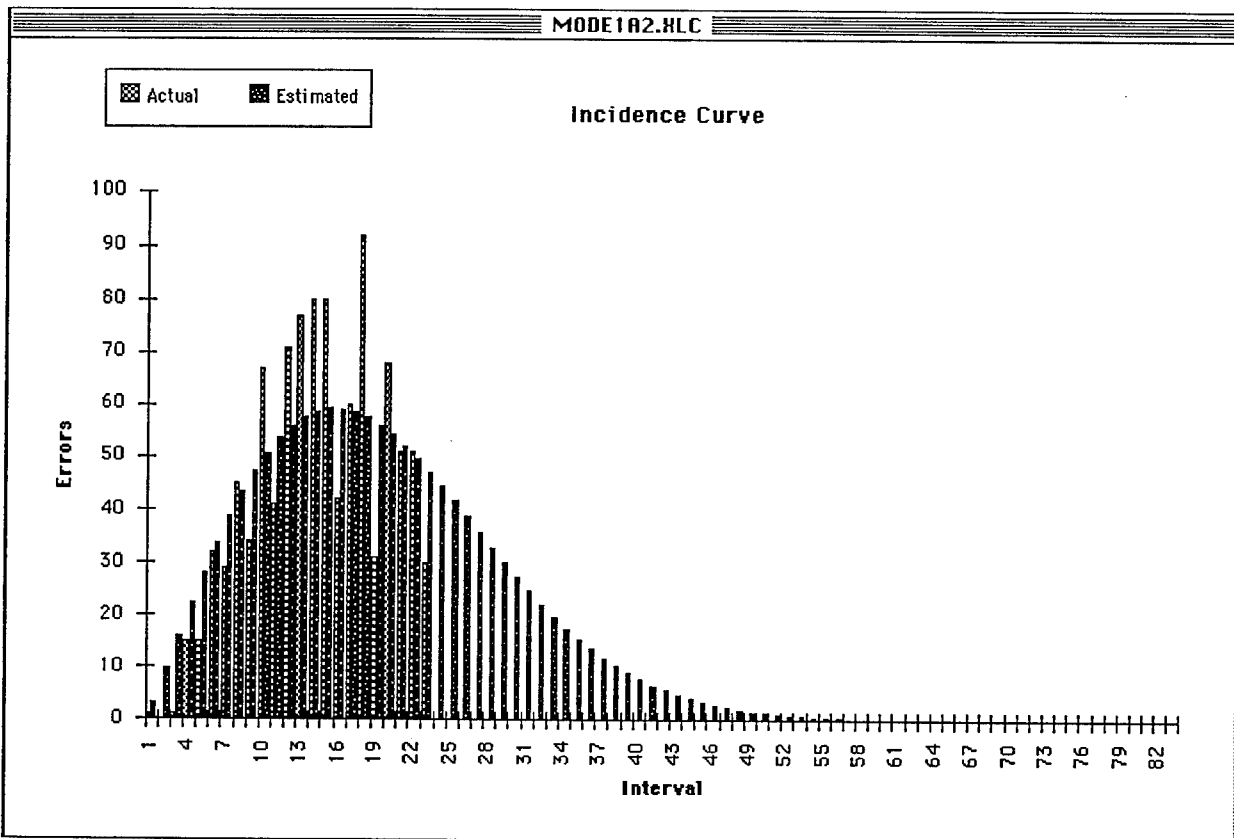
Figure 4-13. Incidence Curve Chart

- *Cumulative Percentage of E.* The estimated percentage of Total Errors Projected through (n) intervals.

- *Actual Cumulation.* The sum of your input data of actual errors.

- *Estimated Cumulation.* The number of errors through the interval indicated, as calculated by SWEEP.

- *Cumulation Delta.* The difference between the Actual Cumulation and Estimated Cumulation.

- *Relative Delta (percentage).* The difference between the Cumulation Delta and the Actual Cumulation, divided by the Actual Cumulation.

4.2.8 THE FILE MENU

The File menu offers the following options:

- *Quit.* Exits both the application and Microsoft Excel.

- *Close.* Closes the active document and returns to the previous screen.

- *Save.* Saves the displayed document to a file. A dialog box appears requesting both the name of the directory to be saved to and name of the file to be saved. Change the file name from the current file name, otherwise SWEEP will overwrite this file the next time it is run.

MODE1A_D.HLS

Time-based Model Output Data Sheet

Errors Discovered to Date: 1013
Total Errors Projected: 1455.04
Percentage of Projected Errors Found to Date: 69.62
Estimated Location of Peak: 14.90

Interval	Actual Error	Estimated Error	Error Delta	Relative Delta (%)	Cumulative % of E	Actual Cumulation	Estimated Cumulation	Cumulation Delta	Relative Delta (%)
1	1	3.2732829	-2.2732829	-227.33	0.22	1	3.2732829	-2.2732829	-227.33
2	0	9.77573314	-9.7757331	Zero Value	0.90	1	13.049016	-12.049016	-1204.90
3	1	16.1468261	-15.146826	-1514.68	2.01	2	29.1958422	-27.195842	-1359.79
4	15	22.3022576	-7.3022576	-48.68	3.54	17	51.4980998	-34.4981	-202.93
5	15	28.1625166	-13.162517	-87.75	5.47	32	79.6606164	-47.660616	-148.94
6	32	33.6545929	-1.6545929	-5.17	7.79	64	113.315209	-49.315209	-77.06
7	29	38.7134837	-9.7134837	-33.49	10.45	93	152.028693	-59.028693	-63.47
8	45	43.2834577	1.7165423	3.81	13.42	138	195.312151	-57.312151	-41.53
9	34	47.3190413	-13.319041	-39.17	16.68	172	242.631192	-70.631192	-41.06
10	67	50.7857049	16.2142951	24.20	20.17	239	293.416897	-54.416897	-22.77
11	41	53.6602371	-12.660237	-30.88	23.85	280	347.077134	-67.077134	-23.96
12	71	55.9308061	15.0691939	21.22	27.70	351	403.00794	-52.00794	-14.82
13	77	57.5967187	19.4032813	25.20	31.66	428	460.604659	-32.604659	-7.62
14	80	58.6678982	21.3321018	26.67	35.69	508	519.272557	-11.272557	-2.22
15	80	59.16411	20.83589	26.04	39.75	588	578.436667	9.56333296	1.63
16	42	59.1139725	-17.113972	-40.75	43.82	630	637.55064	-7.5506395	-1.20
17	60	58.5537946	1.44620537	2.41	47.84	690	696.104434	-6.1044341	-0.88
18	92	57.5262871	34.4737129	37.47	51.79	782	753.630721	28.3692788	3.63
19	31	56.0791935	-25.079194	-80.90	55.65	813	809.709915	3.29008521	0.40
20	68	54.2638894	13.7361106	20.20	59.38	881	863.973804	17.0261958	1.93
21	51	52.1339925	-1.1339925	-2.22	62.96	932	916.107797	15.8922033	1.71
22	51	49.7440271	1.25597287	2.46	66.38	983	965.851824	17.1481762	1.74
23	30	47.1481762	-17.148176	-57.16	69.62	1013	1013	0	0.00
24		44.3991525	No Value	No Value	72.67	1013	1057.39915	-44.399153	-4.38
25		41.5472116	No Value	No Value	75.53	1013	1098.94636	-85.946364	-8.48

Figure 4-14. Time-Based Model Output Data Sheet

NOTE: If you attempt to save a data sheet in a directory other than the working directory, a macro error may occur. Just rename the default file name and click the [OK] button to save the file properly in the current directory.

- *Print Preview.* To print the displayed data sheet, select Print Preview from the File menu. The standard Excel Print Preview window appears.

NOTE: Attempting to print when no printer is connected may result in a macro error. If this happens, click the [Halt] button in the Macro Error message box, select Quit from the File menu, and restart SWEEP.

4.2.9 THE MODES MENU

The Modes menu is dimmed until you close the displayed document.

4.3 MODE 2—PHASE-BASED MODEL OPERATION

When you select Mode 2—Phase-Based Model from the Modes menu, the Mode 2 Input dialog box (see Figure 4-15), which requests user input, is displayed.

4.3.1 MODE 2 INPUT DIALOG BOX

The Mode 2 Input dialog box allows you to select and manipulate activity sets and to enter data for phases in an activity set. The default defect activity set has six phases (see Figure 4-15). You can use

the [Create] button to design an activity set that more closely matches your development process than those given in SWEEP (see Section 4.3.3). You can enter up to three additional phases and can customize the names of all the phases in an activity set. Alter the number and the names of the phases in an activity set by clicking on the [Change] button (see Section 4.3.4). You may also remove activity sets with the [Delete] button (see Section 4.3.5). The Activity Set Selection menu provides you access to any stored activity sets.

```
╒══════════════ Mode 2 - Input ══════════════╕
│  ┌Activity Set Options─────────────────────┐ │
│  │  Activity Set Selection:  ┌Default────┬⬇┐│ │
│  │  ┌─────Create─────┐ ┌────Change────┐ ┌───Delete───┐ │
│  └─────────────────────────────────────────┘ │
│  ┌Data Input Area──────────────────────────┐ │
│  │  Phase 1  ┌3.9─┐    Preliminary Design   │ │
│  │  Phase 2  │7.8 │    Detailed Design       │ │
│  │  Phase 3  │    │    Coding                │ │
│  │  Phase 4  │    │    Unit Test             │ │
│  │  Phase 5  │    │    Integration Test      │ │
│  │  Phase 6  └────┘    System Test           │ │
│  └─────────────────────────────────────────┘ │
│         ┌────OK────┐  ┌───Cancel───┐          │
╘═════════════════════════════════════════════╛
```

Figure 4-15. Mode 2 Input Dialog Box

You enter your data in the fields in the Data Input Area. Enter a minimum of two values, representing error densities estimated during particular phases of the life cycle. If you enter fewer than two inputs and click the [OK] button, an error message is displayed. The input values must be greater than zero. After you enter your data, click the [OK] button. SWEEP will compute the best fit for the data you entered. It provides a plot (see Figure 4-16) of the fit as a function of the location of the peak. You can alter the peak location to override SWEEP's selection of the best fit. (If you click the [Cancel] button while the dialog box is displayed, you will return to the Main window displaying the File and Modes menu bar, which allows you to exit SWEEP or initiate any of the three modes of SWEEP.)

When the computation is complete, SWEEP displays a graph that indicates where the peak location of the distribution will occur, based on the input data. In the upper right corner of the graph, the values
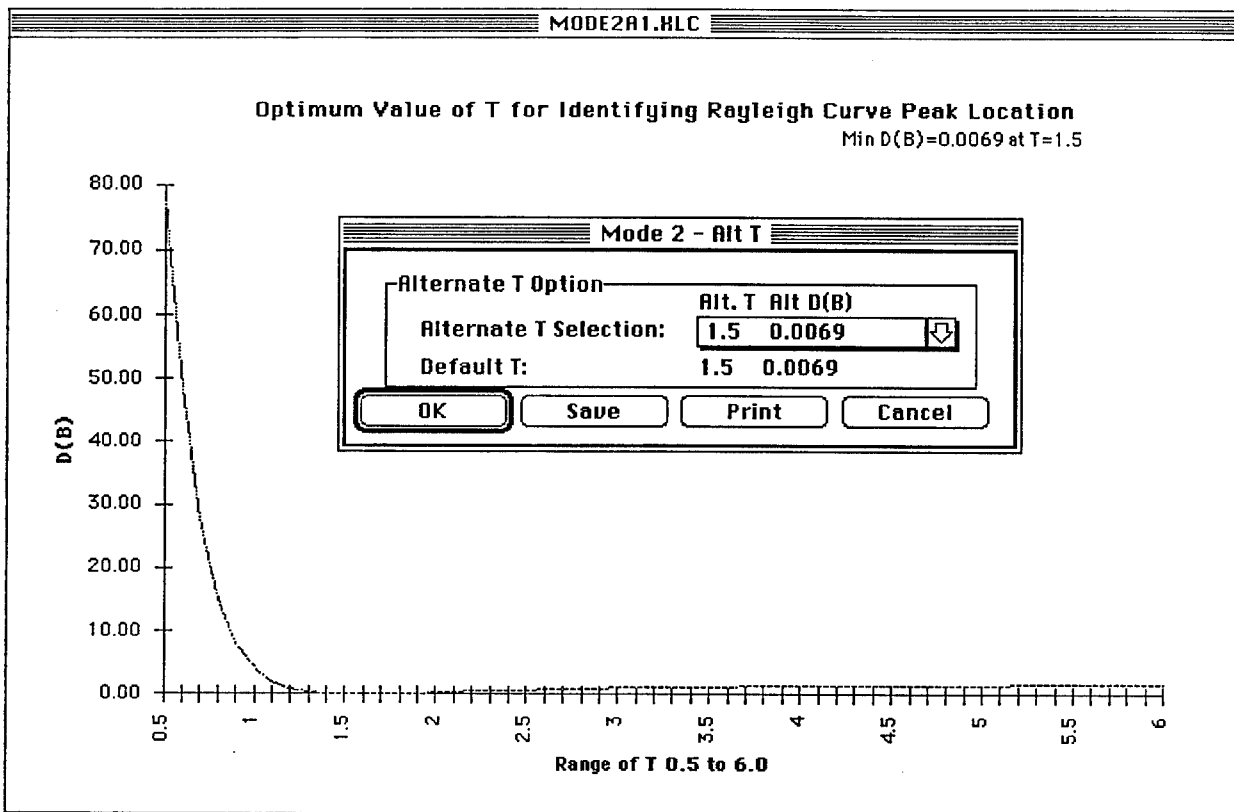
Figure 4-16. Example of a Line Graph Dialog Box

for D(B) and peak location (T) are displayed. The Line Graph dialog box and the corresponding line graph (shown in Figure 4-16) appear in the center of the screen.

4.3.2 LINE GRAPH DIALOG BOX

The Line Graph dialog box offers four options:

- OK

- Print

- Save

- Cancel

Click the [OK] button assumes that you accept the default values calculated by SWEEP for D(B) and T, prompting SWEEP to generate a Rayleigh distribution curve. The value of D(B) directly correlates to a value of T (peak location of Rayleigh distribution curve). Section 4.3.6 describes this graph in greater detail.

Click the [Print] button to print the line graph to a printer using the standard Excel Print Preview functions. Click the [Print] option to open the Excel Print Preview window. Refer to the *Microsoft Excel User's Guide* (Microsoft 1991) for details on the Print Preview options.

Click the [Save] button to save the displayed graph. A dialog box appears requesting the file name to be saved and the directory where the file is to be stored. IMPORTANT: Change the file name to a

name other than the listed default name. If you do not change the file name, the file is saved using the current file name. Be aware that the next time SWEEP is run, it will overwrite the default file name and the previously saved data will be lost. Therefore, be sure to use a unique file name when you save a graph.

NOTE: If you attempt to save a chart in another directory, a macro error may occur. Just rename the default chart name and click the [OK] button. This saves the chart in the current directory under the new chart name.

4.3.3 CREATE ACTIVITY SET

Creating activity sets allows you to customize both the number of phases and the name for each phase. Click the [Create] button in the Mode 2 Input dialog box (see Figure 4-15) to display the Create Activity Set dialog box, shown in Figure 4-17. This box provides you with the input fields to name the activity

```
╔══════════════ Create Activity Set ══════════════╗

    Activity Set Name:  │ My Activity Set       │
    Number of Phases:   │ 9                   ⬇ │
    ┌─Phase Names──────────────────────────────┐
    │   Phase 1      │ Phase One           │   │
    │   Phase 2      │ Phase Two           │   │
    │   Phase 3      │ Phase Three         │   │
    │   Phase 4      │ Phase Four          │   │
    │   Phase 5      │ Phase Five          │   │
    │   Phase 6      │ Phase Six           │   │
    │   Phase 7      │ Phase Seven         │   │
    │   Phase 8      │ Phase Eight         │   │
    │   Phase 9      │ Phase Nine          │   │
    └──────────────────────────────────────────┘
        (     OK     )      (   Cancel   )
╚══════════════════════════════════════════════════╝
```
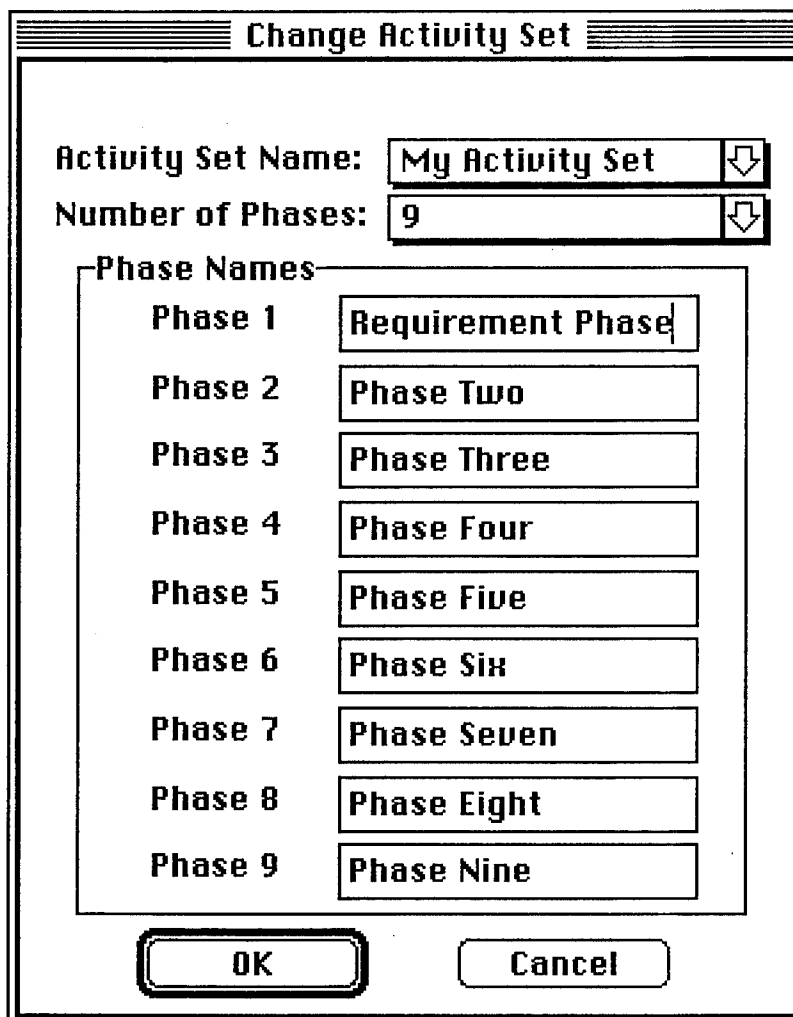
Figure 4-17. Create Activity Set Dialog Box

set and each of the phases in the activity set. A pulldown menu allows you to select the number of phases. You may designate from six to nine phases. Click the [OK] button to verify the data you have entered and to create the new activity set.

⋮

EXAMPLE: You want to create an activity set with six phases. Click the [Create] button and enter My Activity Set as the activity set name. Select 6 from the Number of Phases pulldown menu and then enter Preliminary Design, Detailed Design, Coding, Unit Test, Integration Test, and System Test into the first six phase-name fields, using the <TAB> or <DOWN ARROW> keys to move the input fields. Click the [OK] button to create this activity set.

4.3.4 CHANGE ACTIVITY SET

To change the phase names or number of phases in an activity set you have already created, click the [Change] button on the Input dialog box (see Figure 4-15). This displays the Change Activity Set dialog box (see Figure 4-18). From this box, you can select the activity set to be modified, change the number

```
═══════════ Change Activity Set ═══════════

    Activity Set Name:  [My Activity Set  ⇩]
    Number of Phases:   [9                ⇩]
    ┌Phase Names──────────────────────────┐
    │   Phase 1   [Requirement Phase]      │
    │                                      │
    │   Phase 2   [Phase Two    ]          │
    │                                      │
    │   Phase 3   [Phase Three  ]          │
    │                                      │
    │   Phase 4   [Phase Four   ]          │
    │                                      │
    │   Phase 5   [Phase Five   ]          │
    │                                      │
    │   Phase 6   [Phase Six    ]          │
    │                                      │
    │   Phase 7   [Phase Seven  ]          │
    │                                      │
    │   Phase 8   [Phase Eight  ]          │
    │                                      │
    │   Phase 9   [Phase Nine   ]          │
    └──────────────────────────────────────┘

        [   OK   ]      [  Cancel  ]
```

Figure 4-18. Change Activity Set Dialog Box

of phases, and modify the phase names. You cannot change the activity set name after you have created the activity.

EXAMPLE: In the activity set you created, you want to change a Phase 2 name to Design and Prototyping. Click the [Change] button on the Input dialog box. On the Change Activity Set dialog box, select the activity set you want to modify (My Activity Set) from the Activity Set Name pulldown menu. Click the Phase 2 input field and press the Delete key on the keyboard to erase the old phase name. Then enter Design and Prototyping and click the [OK] button.

4.3.5 DELETE ACTIVITY SET

To delete an activity set, click the [Delete] button on the Input dialog box (see Figure 4-15). This displays the Delete Activity Set dialog box (see Figure 4-19). Select the activity you want from the Activity Set Name pulldown menu and click the [OK] button.
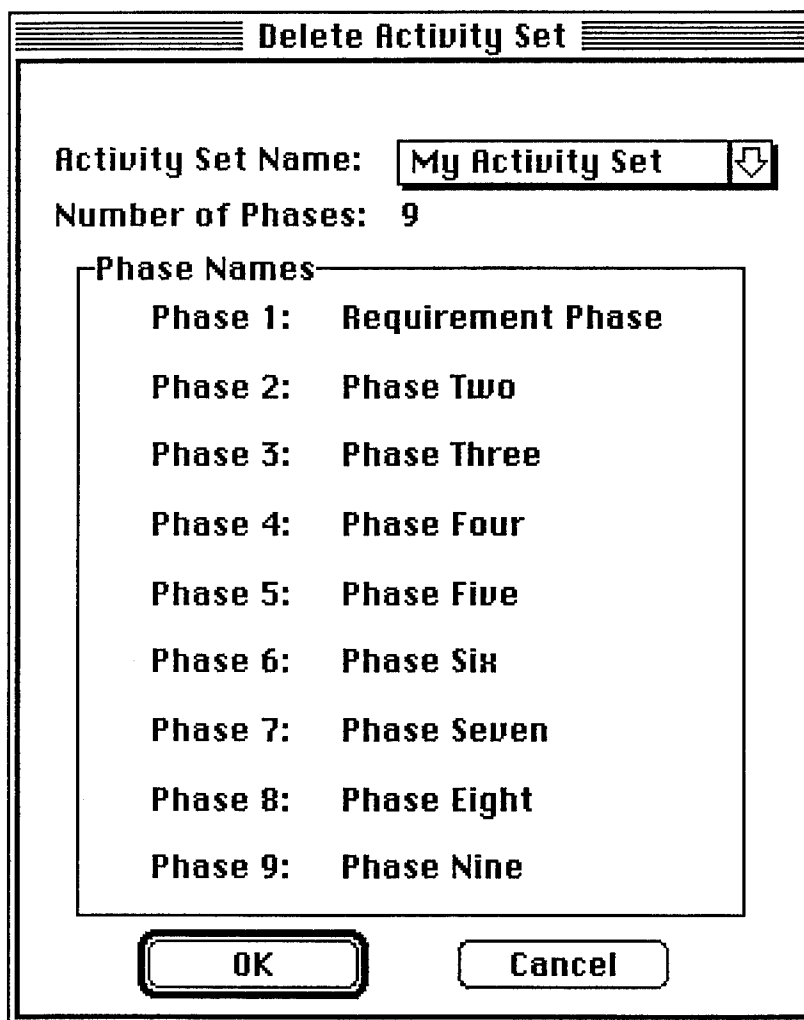
Figure 4-19. Delete Activity Set Dialog Box

4.3.6 RAYLEIGH DISTRIBUTION GRAPH

An example of a Rayleigh distribution graph is shown in Figure 4-20.
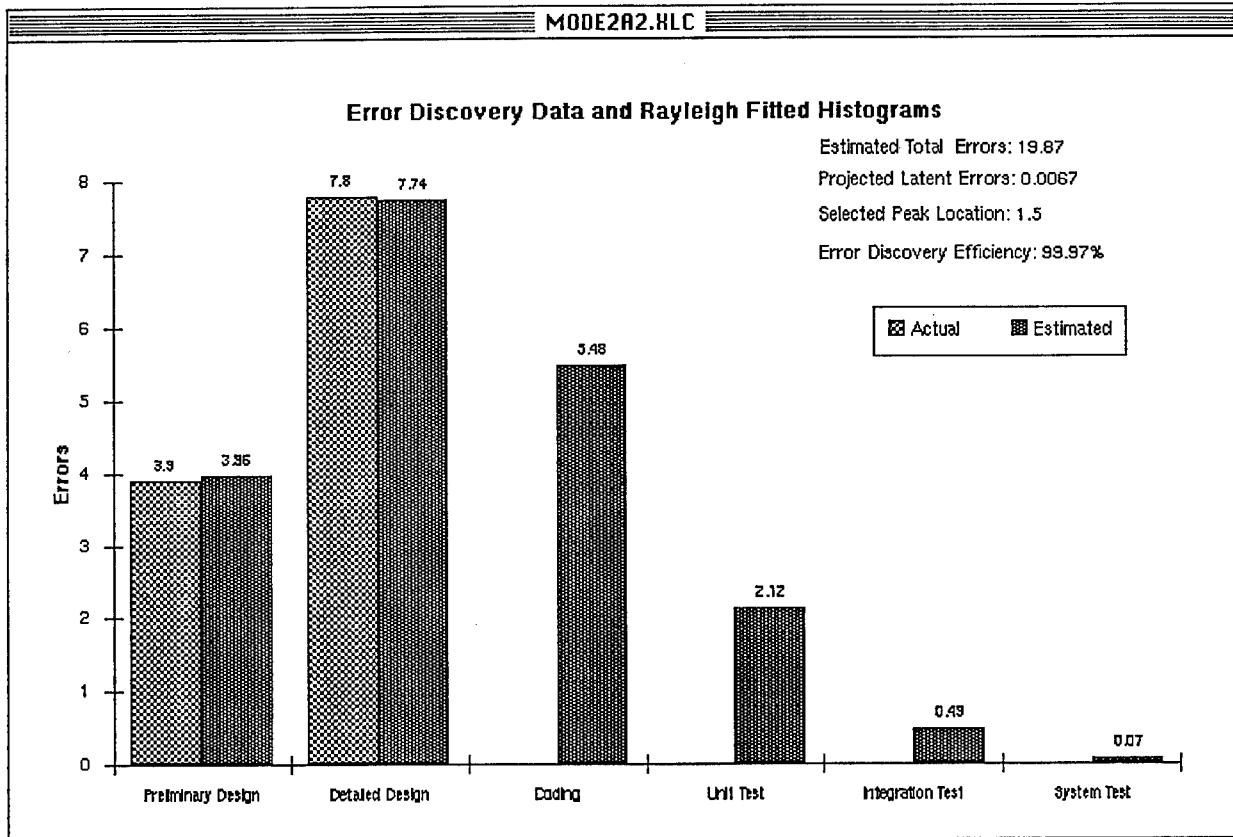
Figure 4-20. Example of a Rayleigh Distribution Graph

The graph provides the following information:

- Vertical axis (errors)

- Horizontal axis (phases)

- Blue bars

  The data you input is represented as the bar on the left hand side for each phase. If no input data exists for a phase, the bar is not displayed.

- Red bars

  The fitted data is represented as the bar on the right hand side for each phase. If no input data exists for a phase, only this bar is displayed.

- Selected peak location

- Total errors injected

- Latent errors

- Error discovery efficiency

After the graph has been displayed, you can select an option from the File menu.

NOTE: If errors are peaking toward the end phases, the error bars may conflict with the information in the top right corner of the graph.

4.3.7 THE FILE MENU

The File menu offers the following options:

- *Quit.* Exits the application and Microsoft Excel.

- *Close.* Closes the Rayleigh distribution graph and returns to the Mode 2 Input dialog box.

- *Save.* Saves the displayed graph to a file. A dialog box appears requesting both the name of the file to be saved and the directory where the file is to be stored. IMPORTANT: Change the file name to a name other than the current file name. Be aware that the next time SWEEP is run, it will overwrite the default file name. Therefore, be sure to use a unique file name when you save a graph.

  NOTES: If you attempt to save a graph in another directory, a macro error may occur. Just rename the default graph name and click the [OK] button. This saves the graph in the current directory using the new graph name.

- *Print Preview.* To print the displayed graph, select Print Preview from the File menu. The standard Microsoft Excel Print Preview window appears.

  NOTE: Attempting to print when no printer is connected may result in a macro error. If this happens, click the [Halt] button in the Macro Error message box, select Quit from the File menu, and restart SWEEP.

4.3.8 THE MODES MENU

The Modes menu is dimmed until you exit Mode 2.

4.4 MODE 3—PLANNING AID OPERATION

Mode 3—Planning Aid helps you to perform a statistical quality control of the software product under development and provides you with the capability to set goals for error discovery. SWEEP provides you with a corresponding error discovery profile on a phase-by-phase basis. You either enter an estimate for total errors injected and the estimated peak location of the distribution over the phases, or enter an estimate for total errors injected and the latent error content. You can enter tolerances for the goals you set. You can choose either a percentage of the errors/phase or a signal-to-noise approach to total errors (see Appendix A).

4.4.1 MODE 3 INPUT DIALOG BOX

When you select Mode 3—Planning Aid from the Modes menu, the Mode 3 Input dialog box (see Figure 4-21) is displayed. The dialog box provides the following:

- Activity Set Selection pulldown menu

- Activity set manipulation buttons

- Input Data Options radio buttons

- Tolerance Options data options

- Input fields for Estimated Total Injected Errors

- [OK] button

- [Cancel] button

Figure 4-21. Mode 3 Input Dialog Box

The *Activity Set Selection pulldown menu* lets you select an activity set to use in the estimate. The *four activity set buttons*—[Show], [Create], [Change], and [Delete]—let you create and manipulate new activity sets.

The *Input Data Options radio buttons* let you select whether to input an estimate either for peak location of errors or for latent error contents. (SWEEP computes an estimate for whichever value you do not

input. If you input peak location, SWEEP computes latent error content, and if you input latent error content, SWEEP computes peak location.) The *Input fields* (in the Data Entry Area) allow you to enter values for Estimated Total Injected Errors. Depending on which Input data option you select, enter either Peak Location (T) or Latent Error (L).

You must select an activity set name from the pulldown menu. To create a new activity set, click the [Create] button. Entering a value in the Estimated Total Injected Errors field is mandatory; the value must be greater than zero. Enter a value for either the peak location or the latent error content, whichever is displayed. Select the appropriate input field and enter a value. Values for peak location must be between 0.5 and the number of phases (m). Values for latent error content must be greater than zero.

Entering the tolerance input values is optional. You have the choice of either selecting a tolerance by inputting an upper and lower tolerance, based on a percentage of the nominal value calculated, or using a signal-to-noise ratio approach. You must select either Upper/Lower Tolerances or Signal to Noise Ratio using the Tolerance Options radio buttons.

If you select Upper/Lower Tolerances, you must enter values for the Upper Tolerance (%) and Lower Tolerance (%) input fields. If you enter values for these fields, they must be between 0 and 100.

If you select Signal to Noise Ratio, enter a value greater than or equal to 1.0 in the Signal to Noise Ratio input field.

NOTE: If you do not enter any tolerance values, the resultant graph will show the upper and lower tolerances equal zero in the Rayleigh distribution curve.

Click the [OK] button to compute the Rayleigh distribution curve and to display a Rayleigh distribution graph based on the inputs.

Click the [Cancel] button at any time to close the dialog box and return to the Main window displaying the File and Modes menu bar.

4.4.2   CREATE ACTIVITY SET

Creating activity sets allows you to customize both the number of phases and the name for each phase. Click the [Create] button in the Mode 3 Input dialog box (see Figure 4-21) to display the Create Activity Set dialog box, shown in Figure 4-17. This box provides you with the input fields to name the activity set and each phase, and a pulldown menu to select the number of phases. You may designate from six to nine phases. Click the [OK] button to verify the data you have entered and to create the new activity set.

EXAMPLE: You want to create an activity set with six phases. Click the [Create] button and enter My Activity Set as the activity set name. Select 6 from the Number of Phases pulldown menu and then enter Preliminary Design, Detailed Design, Coding, Unit Test, Integration Test, and System Test into the name fields for the first six phases, using the <TAB> or <DOWN ARROW> keys to move the input fields. Click the [OK] button to create this activity set.

4.4.3   SHOW ACTIVITY SET

You can view the phases of an activity set by selecting the activity set from the Activity Set Selection pulldown menu in the Mode 3 Input dialog box (see Figure 4-21)and clicking the [Show] button. This displays the phases of the selected activity (see Figure 4-22). Click the [Done] button to return to the Input dialog box (see Figure 4-22).

```
╔══════════════════════════════════════════════════╗
║  ▤▤▤▤▤▤▤▤▤▤ Show Activity Set ▤▤▤▤▤▤▤▤▤▤  ║
╟──────────────────────────────────────────────────╢
║                                                    ║
║   Activity Set Name:    Default                    ║
║   Number of Phases:     6                          ║
║  ┌Phase Names──────────────────────────┐          ║
║  │   Phase 1:  Preliminary Design       │          ║
║  │   Phase 2:  Detailed Design          │          ║
║  │   Phase 3:  Coding                   │          ║
║  │   Phase 4:  Unit Test                │          ║
║  │   Phase 5:  Integration Test         │          ║
║  │   Phase 6:  System Test              │          ║
║  │                                      │          ║
║  └──────────────────────────────────────┘          ║
║                                                    ║
║                 ┌──────────────┐                   ║
║                 │     Done     │                   ║
║                 └──────────────┘                   ║
╚══════════════════════════════════════════════════╝
```
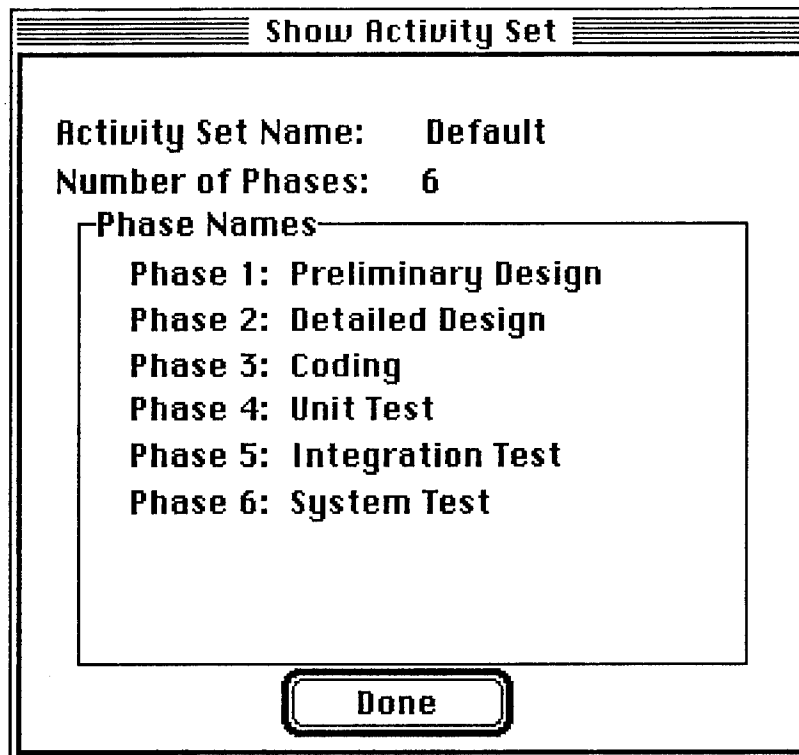
Figure 4-22. Show Activity Set Dialog Box

4.4.4 CHANGE ACTIVITY SET

To change the phase names or number of phases in an activity set you have already created, click the [Change] button on the Input dialog box (see Figure 4-21). This displays the Change Activity Set dialog box (see Figure 4-18). From this box, you can select the activity set to be modified, change the number of phases, and modify the phase names. You cannot change the activity set name after you have created the activity.

EXAMPLE: In the activity set you created, you want to change a Phase 2 name to Design and Prototyping. Click the [Change] button in the Input dialog box. On the Change Activity dialog box, select the activity set you want to modify (My Activity Set) from the Activity Set Name pulldown menu. Click the Phase 2 input field and press the Delete key on the keyboard to erase the old phase name. Then enter Design and Prototyping and click the [OK] button.

4.4.5 DELETE ACTIVITY SET

To delete an activity set, click the [Delete] button on the Input dialog box (see Figure 4-21). This displays the Delete Activity Set dialog box (see Figure 4-19). Select the activity you want from the Activity Set Name pulldown menu and click the [OK] button.

4.4.6 RAYLEIGH DISTRIBUTION GRAPH

When you select either an Input Data Option or Tolerance Option from the Mode 3 Input dialog box (see Figure 4-21), SWEEP generates a graph entitled Error Statistical Quality Control Chart. The

graph can display two types of data. A sample Mode 3 graph is shown in Figure 4-23. If you input the estimated peak location, the estimated latent error content is displayed; if you input an estimated latent error content, the graph displays the estimated peak location of the distribution. When the graph is displayed, use the File menu operations to close, save, or print the graph. (See Section 4.4.7.)
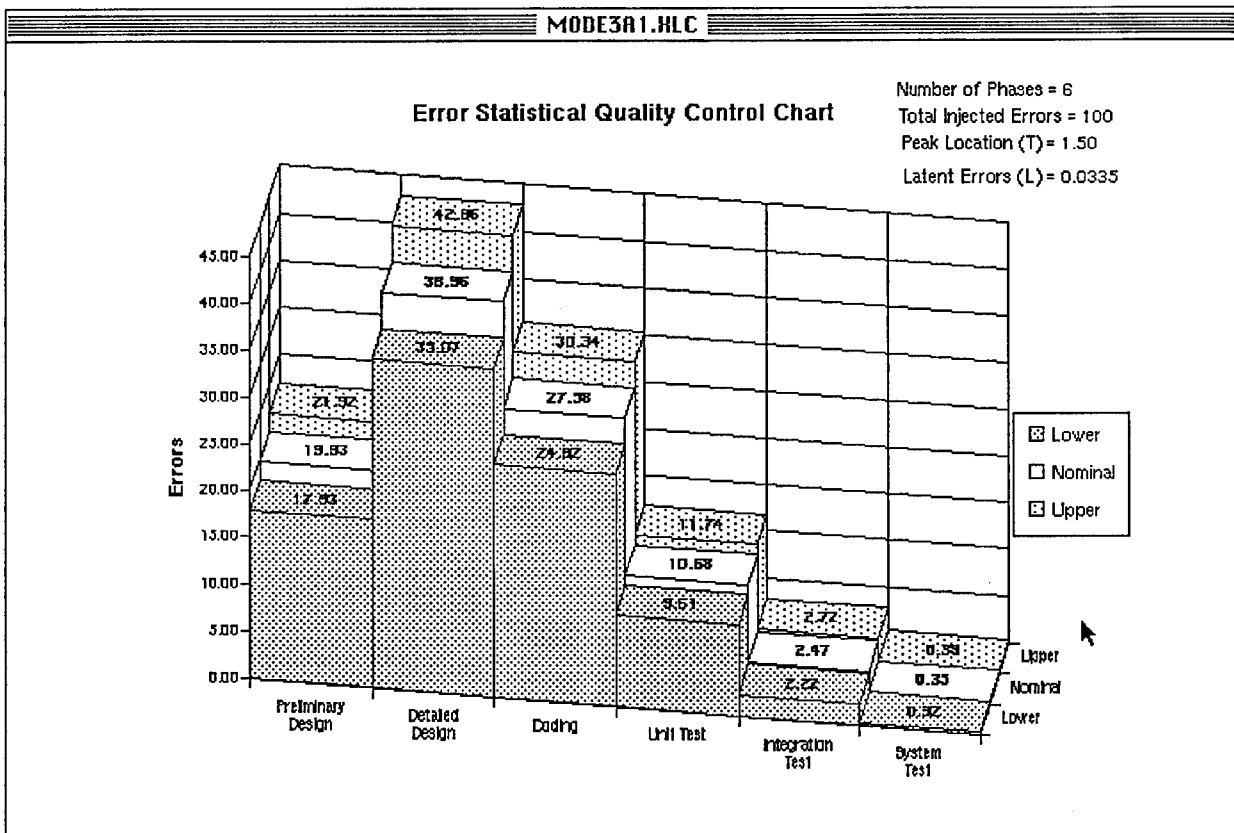
Figure 4-23. Sample Mode 3 Graph

NOTE: If you have chosen total errors and latent errors, and the number of latent errors is unacceptably high, SWEEP will indicate this by displaying a peak location (T) greater than the number of phases.

4.4.7 THE FILE MENU

The File menu offers the following options:

- *Quit.* Exits both the application and Microsoft Excel.

- *Close.* Closes the active document and returns to the previous screen.

- *Save.* Saves the displayed document to a file. A dialog box appears requesting both the name of the directory to be saved to and the name of the file to be saved. Change the file name from the current file name, otherwise SWEEP will overwrite this file the next time it is run.

   NOTE: If you attempt to save a data sheet in another directory, a macro error may occur. Just rename the default file name and click the [OK] button. This saves the file in the current directory.

- ***Print Preview.*** To print the displayed data sheet, select Print Preview from the File menu. The standard Excel Print Preview window appears.

    NOTE: Attempting to print when no printer is connected may result in a macro error. If this happens, click the [Halt] button in the Macro Error message box, select Quit from the File menu, and restart SWEEP.

4.4.8 THE MODES MENU

The Modes menu is dimmed until you exit the mode.

This page intentionally left blank.

5. CASE STUDY

5.1 OVERVIEW

This section shows how you can use SWEEP's time-based model (Mode 1) to help you to answer certain questions of practical interest, such as: How many problems should be expected to be found during the next five months of testing? The application of Mode 1 to the analysis of error (or defect or problem) detection occurrence data is illustrated here by applying it to data from a software development project conducted at the Raytheon Company (Rome Air Development Center 1977). You can use Mode 1 to set parameters for data related to the detection of problems during the testing process. Thus, you can make estimates of the number of problems expected to be observed later, over a period of time that you specify. You can also estimate the number of time intervals required to detect (and remove) a specified percentage of the total number of defects expected to be injected into your software product during its development. You can also use SWEEP to make projections of the outages of an overall system caused by software, hardware, or procedures (Gaffney, Ferrando, and Jones 1993).

5.2 DEFECT OR PROBLEM DATA ANALYSIS USING SWEEP

Mode 1 can help you to address issues related to the number of defects or problems you expect to find during the development of a software product. Refer to Section 4 as you read and think about the material in this subsection. Section 4 shows you what the dialog boxes for SWEEP look like, thus indicating where the values of various parameters that you enter, or that are produced by SWEEP, are available to you. This case study (see Sections 5.3 and 5.4) illustrates how you can obtain the values for some of these parameters. Mode 1 can help you to determine the answers to the following questions:

- How many defects should I expect to find during the next m intervals of testing?

  You may wish to know if it will be cost-effective to continue testing (and finding and removing bugs) for some period of time in addition to the time your organization has already invested for this purpose. An increasing amount of effort will probably be required to find and correct each problem found in the software product as the test process continues. This is a manifestation of the law of diminishing returns, the increasing marginal cost for identifying problems during the period of testing a software product. You can use SWEEP to estimate how many defects you should expect to find in testing over the next m time intervals. This information can help you to estimate the costs of discovering defects in the software system you are developing. If it costs $X per time increment to staff the testing function, then it will cost an average of $X/B per problem, if B problems are found during the next m time increments. Generally expect (and SWEEP can estimate for you) that the value of the parameter B will decrease over

time, after the peak rate of defect discovery has been reached (at time interval t_p). Thus, the average cost of discovering a problem, $X/B, will increase over that period of time.

- How long should I expect to test this software in order for it to be problem-free (to find essentially all of the errors in it)?

  You might find it useful to know how much additional test time is (estimated to be) required in order to find essentially all of the problems in the software product being tested. Clearly, this figure is likely to be excessively high, but it is an upper bound to the required test time, and you might find it of interest. SWEEP tells you how many time intervals would be expected to be required, in total, to find 99.99 percent of the (estimated) total number of errors injected during the development process (the total number that are potentially discoverable, the parameter E, the (estimated) total number of errors, as described in detail in Appendix A). E is one of the two parameters of the Rayleigh curve that SWEEP fits to the defect or problem discovery data that you provide to Mode 1. The other parameter is the location of the peak of this fitted curve, t_p; the peak location is the number of the time interval at which the highest rate of problem discovery is estimated to be experienced.

- How many defects should I expect to be injected into the software system during the integration process? What is the profile of defect or problem discovery that I should expect to be experienced during testing?

  Use experience on past projects to help you to develop your initial testing budget. Then, after the testing process has been underway for a while, review this budget and determine if it needs to be revised—increased or decreased—in accordance with the actual experience on this project. You use SWEEP to provide data you can use in developing your budget update. Use SWEEP to develop a projection for the number of problems or defects that you expect to be found during the remaining expected test time. If this figure is higher than your earlier estimate, it would probably be wise for you to revise your test budget upward, based on the increased number of defects that you now expect to find. Alternatively, if the number of problems expected is lower than your original estimate, you may wish to revise your test budget downward. SWEEP provides an estimate of the total number of problems that you should expect to find during the remainder of the scheduled testing period (i.e., during the remaining m time intervals). It provides an estimate of the number of problems that you should expect to find during each of those m intervals (the remainder of the defect or problem detection profile). An estimate based on the projection provided by SWEEP should be better than an estimate that was made before the project got underway, based on historical data. If you use Mode 1 to provide estimates of the values of the parameters of the profile of problem discovery during the integration process, then the value of the parameter E, the (estimated) total number of errors, would be the estimate of the total number of errors or defects expected to be injected into the software product during the integration process.

5.3 CASE STUDY PROJECT DESCRIPTION

The Raytheon Company developed project Q. The integration testing for project Q was performed over a period of approximately 33 months. During that time, there were about 40 releases of the product, which reflected error corrections, design changes, and improvements. During the 33-month time interval, 1,198 problems were reported, and their times of occurrence were recorded. Table 5-1

presents the number (out of the total of 1,198) of defects or problems discovered during each of the 33 months. This table also presents the cumulative number of problems or defects discovered from the beginning of testing through each month. Project Q personnel plotted the number of problems found each month. They noted that the incidence of defect discovery as a function of time, over the 33-month period, had peaks and valleys. Figure 5-1 is a plot of the incidence curve produced by
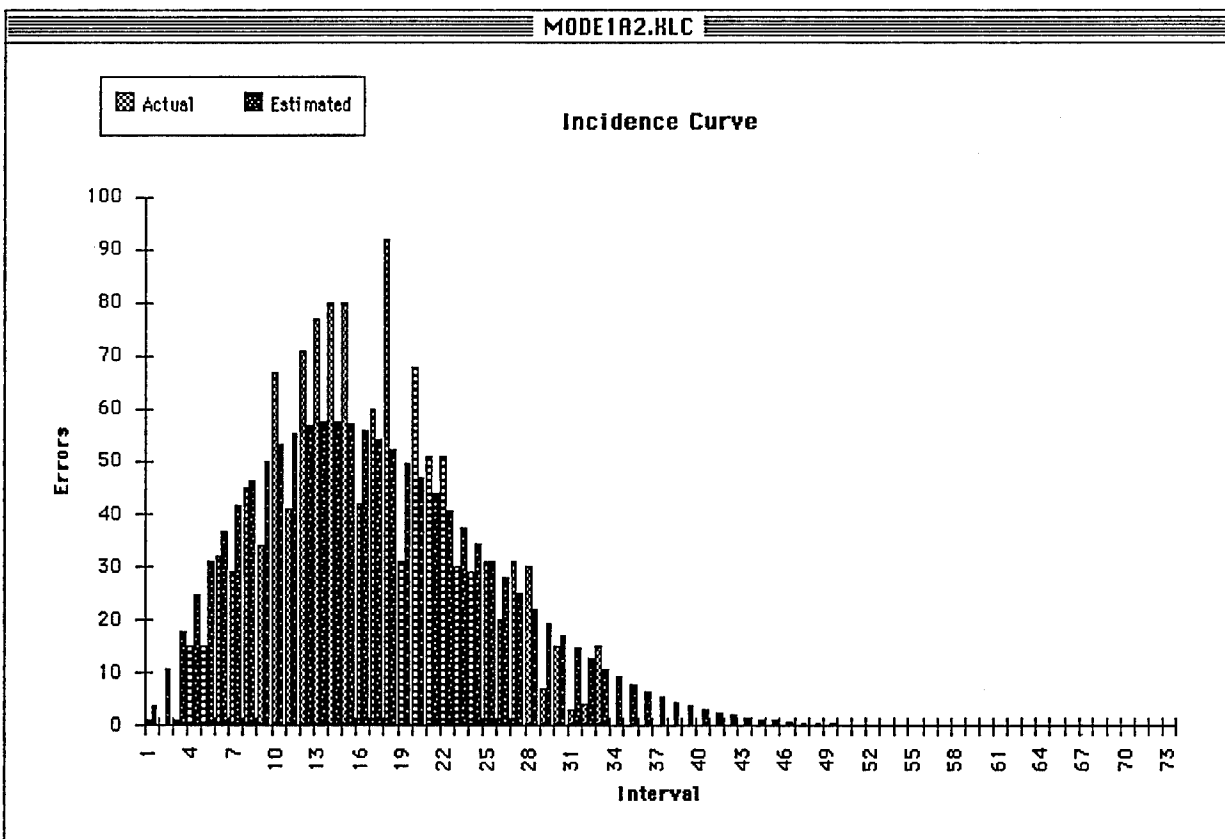
Figure 5-1. Error Incidence for 33 Months

SWEEP. This curve shows these peaks and valleys as illustrated by the number of defects discovered per month over the 33-month period. This data is also provided in Table 5-1. Project Q's analysis of the defect discovery, or problem incidence, data showed that the peaks occurred at the times of release of the builds when several problems had been resolved. The valleys occurred when the testing of particular functions took place, when the development team was preparing to work on the next release.

5.4 APPLICATION OF SWEEP TO THE RAYTHEON PROJECT DATA

This section describes the results of applying Mode 1 to the Raytheon Project Q data presented in Table 5-1. Various parameters that one could use to answer questions of interest to project development management, such as those given in Section 5.2, are described here.

The counts of defects, or problems found in each of the 33 months were provided to SWEEP, which fit a Rayleigh curve to them. The actual numbers of problems found per month for 33 months (see Table 5-1) and SWEEP's fit to these data are shown graphically in Figure 5-1 and in tabular form in Figures 5-3 and 5-4. The cumulative number of problems found through each month from 1 to 33 (see

Table 5-1) and SWEEP's fit to these data are shown graphically in Figure 5-2 and in tabular form in Figures 5-3 and 5-4.

Table 5-1. Defect Discovery History

Month Number	Number of Defects Discovered During This Month	Cumulative Number of Defects Discovered Through This Month
1	1	1
2	0	1
3	1	2
4	15	17
5	15	32
6	32	64
7	29	93
8	45	138
9	34	172
10	67	239
11	41	280
12	71	351
13	77	428
14	80	508
15	80	588
16	42	630
17	60	690
18	92	782
19	31	813
20	68	881
21	51	832
22	51	983
23	30	1,013
24	29	1,042
25	31	1,073
26	20	1,093
27	31	1,124
28	30	1,154
29	7	1,161
30	15	1,176
31	3	1,179
32	4	1,183
33	15	1,198
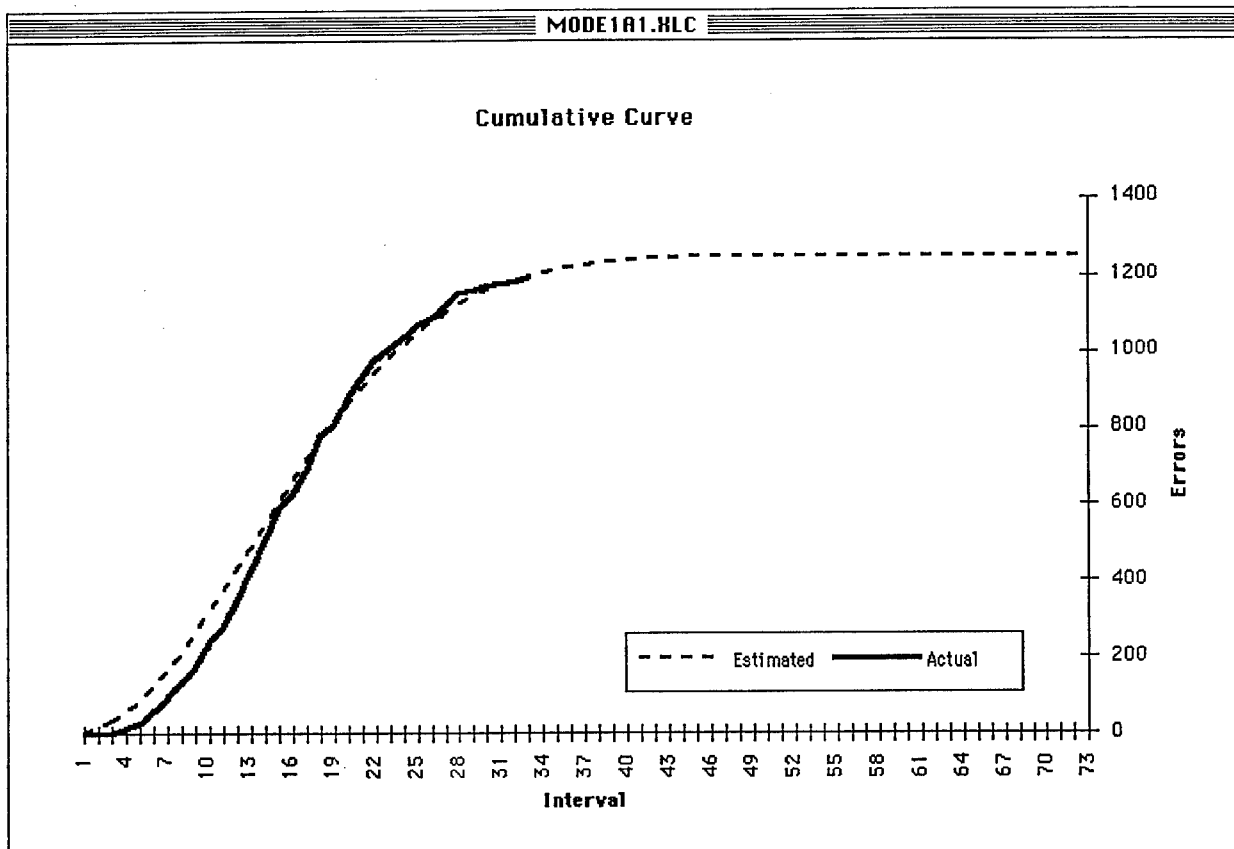
Figure 5-2. Cumulative Number of Errors for 33 Months

Using Figure 5-2 and Table 5-1, we can summarize results of applying SWEEP to this data. SWEEP estimated that:

- 99.9 percent of the total number of defects anticipated to be injected into the system would be discovered by month 49.

- The total number of estimated defects injected into the software was 1,250.

- There would be 52 defects remaining in the system at the end of month 33. The number of defects remaining is calculated by subtracting the actual total number of defects found (1,198) from the estimated total number of defects (1,250). Realistically, it is not very likely that every problem would be discovered.

- The value of the peak location was interval 13.

Figures 5-2 and 5-4 are images of the data sheet produced by SWEEP when it processed the data for 33 months of problem discovery. Shown are the actual and SWEEP-fit problem counts, as well as cumulative counts on a month-by-month basis. Additionally, the cumulative percentage of total problems injected on a month-by-month basis is indicated.

Time-based Model Output Data Sheet

MODE1A_D.HLS

Errors Discovered to Date: 1198
Total Errors Projected: 1250.37
Percentage of Projected Errors Found to Date: 95.81
Estimated Location of Peak: 13.10

Interval	Actual Error	Estimated Error	Error Delta	Relative Delta (%)	Cumulative % of E	Actual Cumulation	Estimated Cumulation	Cumulation Delta	Relative Delta (%)
1	1	3.63774995	-2.63775	-263.77	0.29	1	3.63774995	-2.63775	-263.77
2	0	10.8498722	-10.8498872	Zero Value	1.16	1	14.4876221	-13.487622	-1348.76
3	1	17.873698	-16.873698	-1687.37	2.59	2	32.3613202	-30.36132	-1518.07
4	15	24.5897447	-9.5897447	-63.93	4.55	17	56.9510649	-39.951065	-235.01
5	15	30.8873451	-15.887345	-105.92	7.03	32	87.83841	-55.83841	-174.50
6	32	36.6676747	-4.6676747	-14.59	9.96	64	124.506085	-60.506085	-94.54
7	29	41.8463106	-12.846311	-44.30	13.30	93	166.352395	-73.352395	-78.87
8	45	46.3552299	-1.3552299	-3.01	17.01	138	212.707625	-74.707625	-54.14
9	34	50.1441858	-16.144186	-47.48	21.02	172	262.851811	-90.851811	-52.82
10	67	53.1814247	13.8185753	20.62	25.28	239	316.033236	-77.033236	-32.23
11	41	55.4537459	-14.453746	-35.25	29.71	280	371.486982	-91.486982	-32.67
12	71	56.9659283	14.0340717	19.77	34.27	351	428.45291	-77.45291	-22.07
13	77	57.7395804	19.2604196	25.01	38.88	428	486.19249	-58.19249	-13.60
14	80	57.8114914	22.1885086	27.74	43.51	508	544.003982	-36.003982	-7.09
15	80	57.2315784	22.7684216	28.46	48.08	588	601.23556	-13.23556	-2.25
16	42	56.0605377	-14.060538	-33.48	52.57	630	657.296098	-27.296098	-4.33
17	60	54.3673126	5.63268743	9.39	56.92	690	711.66341	-21.66341	-3.14
18	92	52.2264918	39.77350082	43.23	61.09	782	763.889902	18.1100978	2.32
19	31	49.7157435	-18.715744	-60.37	65.07	813	813.605646	-0.6056457	-0.07
20	68	46.9133828	21.0866172	31.01	68.82	881	860.519028	20.4809715	2.32
21	51	43.8961518	7.10384821	13.93	72.33	932	904.41518	27.5848197	2.96
22	51	40.7372792	10.2627208	20.12	75.59	983	945.152459	37.8475406	3.85
23	30	37.5048619	-7.5048619	-25.02	78.59	1013	982.657321	30.3426787	3.00
24	29	34.2605973	-5.2605973	-18.14	81.33	1042	1016.91792	25.0820814	2.41
25	31	31.0588727	-0.0588727	-0.19	83.81	1073	1047.97679	25.0232087	2.33

Figure 5-3. Data Sheet for Fit to 33 Months of Data, Part 1

Time-based Model Output Data Sheet

MODE1A_D.XLS

Errors Discovered to Date: 1198
Total Errors Projected: 1250.37
Percentage of Projected Errors Found to Date: 95.81
Estimated Location of Peak: 13.10

Interval	Actual Error	Estimated Error	Error Delta	Relative Delta (%)	Cumulative % of E	Actual Cumulation	Estimated Cumulation	Cumulation Delta	Relative Delta (%)
26	20	27.9462051	-7.9462051	-39.73	86.05	1093	1075.923	17.0770036	1.56
27	31	24.9610078	6.03899223	19.48	88.04	1124	1100.884	23.1159958	2.06
28	30	22.1336499	7.8663501	26.22	89.81	1154	1123.01765	30.9823459	2.68
29	7	19.486768	-12.486768	-178.38	91.37	1161	1142.50442	18.4955775	1.59
30	15	17.0357839	-2.0357839	-13.57	92.74	1176	1159.54021	16.4597936	1.40
31	3	14.7895722	-11.789572	-392.99	93.92	1179	1174.32978	4.67022142	0.40
32	4	12.7512426	-8.7512426	-218.78	94.94	1183	1187.08102	-4.0810212	-0.34
33	15	10.9189788	4.08102121	27.21	95.81	1198	1198	0	0.00
34		9.2868984	No Value	No Value	96.55	1198	1207.2869	-9.2868984	-0.78
35		7.84589928	No Value	No Value	97.18	1198	1215.1328	-17.132798	-1.43
36		6.58446136	No Value	No Value	97.71	1198	1221.71726	-23.717259	-1.98
37		5.48938324	No Value	No Value	98.15	1198	1227.20664	-29.206642	-2.44
38		4.54643788	No Value	No Value	98.51	1198	1231.75308	-33.75308	-2.82
39		3.74093834	No Value	No Value	98.81	1198	1235.49402	-37.494019	-3.13
40		3.05820971	No Value	No Value	99.05	1198	1238.55223	-40.552228	-3.38
41		2.48396834	No Value	No Value	99.25	1198	1241.0362	-43.036197	-3.59
42		2.00461287	No Value	No Value	99.41	1198	1243.04081	-45.040809	-3.76
43		1.60743438	No Value	No Value	99.54	1198	1244.64824	-46.648244	-3.89
44		1.28075483	No Value	No Value	99.64	1198	1245.929	-47.928999	-4.00
45		1.01400407	No Value	No Value	99.73	1198	1246.943	-48.943003	-4.09
46		0.79774583	No Value	No Value	99.79	1198	1247.74075	-49.740749	-4.15
47		0.62366315	No Value	No Value	99.84	1198	1248.36441	-50.364412	-4.20
48		0.48451278	No Value	No Value	99.88	1198	1248.84892	-50.848924	-4.24
49		0.37405751	No Value	No Value	99.91	1198	1249.22298	-51.222982	-4.28
50		0.28698379	No Value	No Value	99.93	1198	1249.50997	-51.509966	-4.30

Figure 5-4. Data Sheet for Fit to 33 Months of Data, Part 2

Next, the ability of Mode 1 to make defect discovery projections was investigated. A situation in which only 23 months of testing had been done was simulated. SWEEP was used to project the defect, or problem, discovery for 10 more months (i.e., through month 33), based on the counts of problems found during months 1 through 23. SWEEP estimated that:

- The total number number of defects, E, injected into the software would be 1,455 (as compared with the 1,250 estimated, based on 33 months of data—just 16.4 percent more than the 1,250 based on the larger data set of 33 points).

- The peak location was interval 15 (rather than 13, based on 33 months of data).

- A total of 1,329 problems, or defects, would be found in the 33-month period. The actual figure was 1,198. Thus, SWEEP over-estimated the number of defects to be found over the 33-month period, based on 23 months of data, by 131—or 10.9 percent.

The actual numbers of problems found per month for 23 months (see Table 5-1) and SWEEP's fit to these data are shown in Figure 5-5. The cumulative number of problems found through each month from 1 to 23 (see Table 5-1) and SWEEP's fit to these data are shown in Figure 5-6.
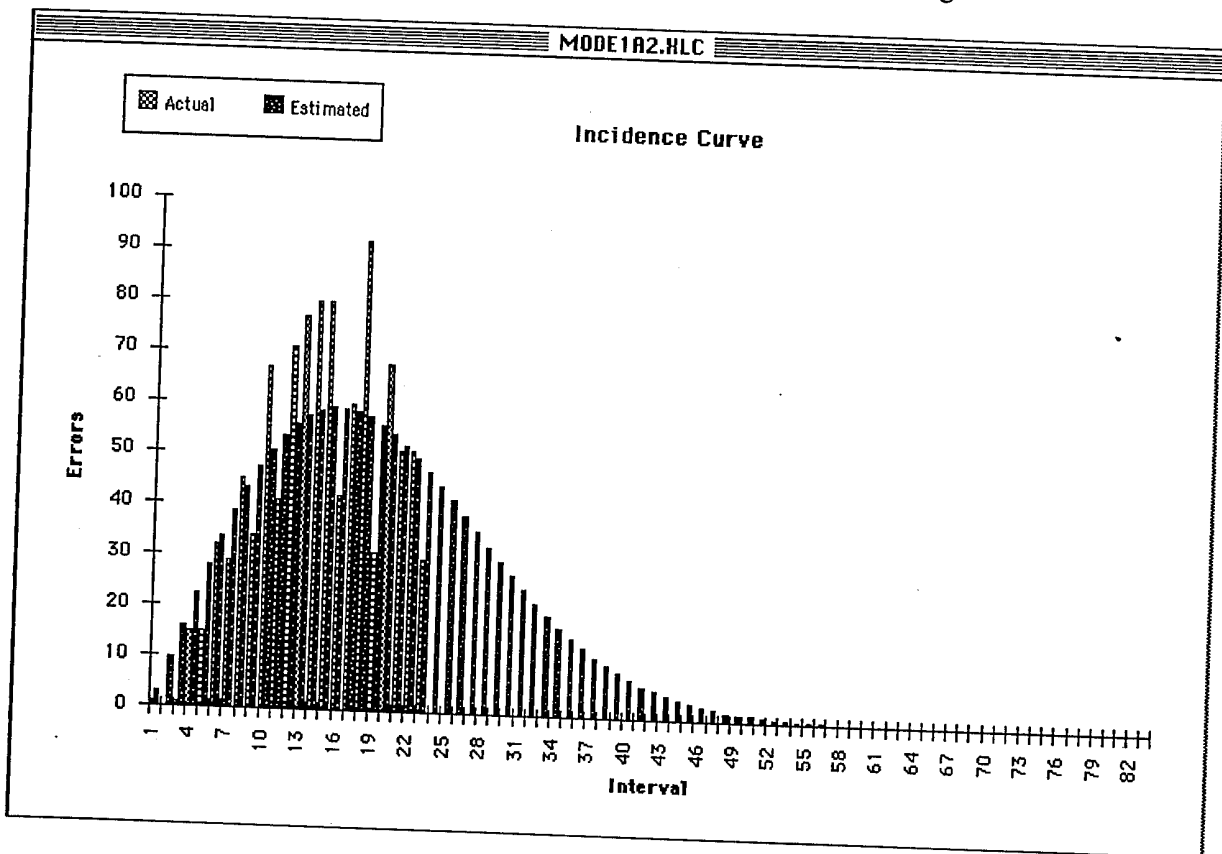
Figure 5-5. Error Incidence for 23 Months

Figures 5-7 and 5-8 are images of the data sheet produced by SWEEP when it processed the data for 23 months of problem discovery. Shown are the actual and SWEEP-fit problem counts, as well as cumulative counts on a month-by-month basis. Additionally, the cumulative percentage of total problems injected on a month-by-month basis is provided on the data sheet. Note from the data presented in Figure 5-8 that 99.6 percent of the defects injected (estimated to be 1,455) would be found by the end of month 49, based on data from months 1 through 23.
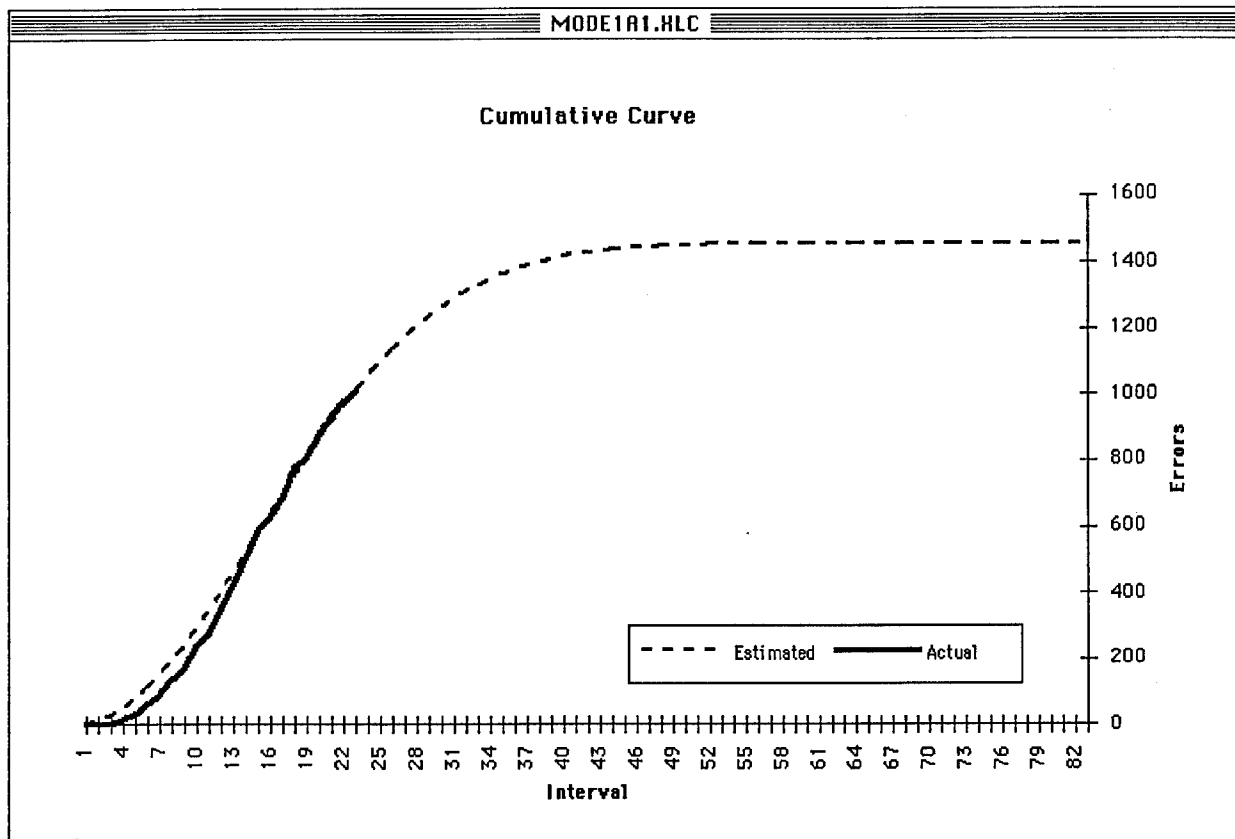
Figure 5-6. Cumulative Number of Errors for 23 Months

5.5 CONCLUSION

This case study demonstrates the types of information that can be made available to software development projects using SWEEP. Project managers can use the estimated number of defects when planning and scheduling testing cycles and resource levels. The total number of defects can be used to monitor progress against a project's quality goals. The study also shows that SWEEP can be used to refine estimates by entering actual project data.

After 23 months of testing and repairing defects, SWEEP predicted the number of defects discovered and repaired after 33 months within 11 percent of the actual number. Also at 23 months, SWEEP estimated within about 16 percent the total number of injected errors to be found after 33 months. Finally, SWEEP estimated that after 23 months of testing and development, 69 percent of the defects had been removed, and that after 49 months, 99.6 percent of the defects would be removed.

After 33 months of testing and repairing the software, SWEEP predicted that 95 percent of the errors had been removed and that after 49 months, 99.9 percent of the defects would be removed.

▐▐▐ MODE1A_D.HLS ▐▐▐

Time-based Model Output Data Sheet

Errors Discovered to Date: 1013
Total Errors Projected: 1455.04
Percentage of Projected Errors Found to Date: 69.62
Estimated Location of Peak: 14.90

Interval	Actual Error	Estimated Error	Error Delta	Relative Delta (%)	Cumulative % of E	Actual Cumulation	Estimated Cumulation	Cumulation Delta	Relative Delta (%)
1	1	3.2732829	-2.2732829	-227.33	0.22	1	3.2732829	-2.2732829	-227.33
2	0	9.7757314	-9.7757331	Zero Value	0.90	1	13.049016	-12.049016	-1204.90
3	1	16.1468261	-15.146826	-1514.68	2.01	2	29.1958422	-27.195842	-1359.79
4	15	22.3022576	-7.3022576	-48.68	3.54	17	51.4980998	-34.4981	-202.93
5	15	28.1625166	-13.162517	-87.75	5.47	32	79.6606164	-47.660616	-148.94
6	32	33.6545929	-1.6545929	-5.17	7.79	64	113.315209	-49.315209	-77.06
7	29	38.7134837	-9.7134837	-33.49	10.45	93	152.028693	-59.028693	-63.47
8	45	43.2834577	1.7165423	3.81	13.42	138	195.312151	-57.312151	-41.53
9	34	47.3190413	-13.319041	-39.17	16.68	172	242.631192	-70.631192	-41.06
10	67	50.7857049	16.2142951	24.20	20.17	239	293.416897	-54.416897	-22.77
11	41	53.6602371	-12.660237	-30.88	23.85	280	347.077134	-67.077134	-23.96
12	71	55.9308061	15.0691939	21.22	27.70	351	403.00794	-52.00794	-14.82
13	77	57.5967187	19.4032813	25.20	31.66	428	460.604659	-32.604659	-7.62
14	80	58.6678982	21.3321018	26.67	35.69	508	519.272557	-11.272557	-2.22
15	80	59.16411	20.83589	26.04	39.75	588	578.436667	9.56333296	1.63
16	42	59.1139725	-17.113972	-40.75	43.82	630	637.55064	-7.5506395	-1.20
17	60	58.5537946	1.44620537	2.41	47.84	690	696.104434	-6.1044341	-0.88
18	92	57.5262871	34.4737129	37.47	51.79	782	753.630721	28.3692788	3.63
19	31	56.0791935	-25.079194	-80.90	55.65	813	809.709915	3.29008521	0.40
20	68	54.2638894	13.7361106	20.20	59.38	881	863.973804	17.0261958	1.93
21	51	52.1339925	-1.1339925	-2.22	62.96	932	916.107797	15.8922033	1.71
22	51	49.7440271	1.25597287	2.46	66.38	983	965.851824	17.1481762	1.74
23	30	47.1481762	-17.148176	-57.16	69.62	1013	1013	0	0.00
24		44.3991525	No Value	No Value	72.67	1013	1057.39915	-44.399153	-4.38
25		41.5472116	No Value	No Value	75.53	1013	1098.94636	-85.946364	-8.48

Figure 5-7. Data Sheet for 23 Months of Data, Part 1

MODE1A_D.HLS

Time-based Model Output Data Sheet

Errors Discovered to Date: 1013
Total Errors Projected: 1455.04
Percentage of Projected Errors Found to Date: 69.62
Estimated Location of Peak: 14.90

Interval	Actual Error	Estimated Error	Error Delta	Relative Delta (%)	Cumulative % of E	Actual Cumulation	Estimated Cumulation	Cumulation Delta	Relative Delta (%)
26		38.6639322	No Value	No Value	78.18	1013	1137.58569	-124.58569	-12.30
27		35.7185032	No Value	No Value	80.64	1013	1173.30419	-160.30419	-15.82
28		32.8233324	No Value	No Value	82.89	1013	1206.12752	-193.12752	-19.06
29		29.9876186	No Value	No Value	84.95	1013	1236.11514	-223.11514	-22.03
30		27.2402329	No Value	No Value	86.83	1013	1263.35537	-250.35537	-24.71
31		24.6050845	No Value	No Value	88.52	1013	1287.96046	-274.96046	-27.14
32		22.1012245	No Value	No Value	90.04	1013	1310.06168	-297.06168	-29.32
33		19.7430591	No Value	No Value	91.39	1013	1329.80474	-316.80474	-31.27
34		17.5406521	No Value	No Value	92.60	1013	1347.34539	-334.34539	-33.01
35		15.5000966	No Value	No Value	93.66	1013	1362.84549	-349.84549	-34.54
36		13.6239357	No Value	No Value	94.60	1013	1376.46943	-363.46943	-35.88
37		11.9116125	No Value	No Value	95.42	1013	1388.38104	-375.38104	-37.06
38		10.3599334	No Value	No Value	96.13	1013	1398.74097	-385.74097	-38.08
39		8.96352776	No Value	No Value	96.75	1013	1407.7045	-394.7045	-38.96
40		7.71529286	No Value	No Value	97.28	1013	1415.41979	-402.41979	-39.73
41		6.60681225	No Value	No Value	97.73	1013	1422.0266	-409.0266	-40.38
42		5.62874015	No Value	No Value	98.12	1013	1427.65534	-414.65534	-40.93
43		4.77114631	No Value	No Value	98.45	1013	1432.42649	-419.42649	-41.40
44		4.023817775	No Value	No Value	98.72	1013	1436.45031	-423.45031	-41.80
45		3.37651619	No Value	No Value	98.95	1013	1439.82682	-426.82682	-42.13
46		2.81919127	No Value	No Value	99.15	1013	1442.64602	-429.64602	-42.41
47		2.34215136	No Value	No Value	99.31	1013	1444.98817	-431.98817	-42.64
48		1.93619464	No Value	No Value	99.44	1013	1446.92436	-433.92436	-42.84
49		1.59270403	No Value	No Value	99.55	1013	1448.51707	-435.51707	-42.99
50		1.30371006	No Value	No Value	99.64	1013	1449.82078	-436.82078	-43.12

Figure 5-8. Data Sheet for 23 Months of Data, Part 2

This page intentionally left blank.

APPENDIX: SOFTWARE ERROR ESTIMATION MATHEMATICAL MODELS

A.1 OVERVIEW

This section describes the mathematical models implemented by Modes 1, 2, and 3 of SWEEP. Mode 1 provides time-based error estimation, using error discovery data available on a time basis. It is preferred that the time variable should be the system run time during which error data was collected, rather than calendar time, and should be normalized so that each interval covers the same duration. SWEEP fits this data to a Rayleigh curve. Mode 2 provides a phase-based approach to error estimation, fitting available data on a phase-by-phase basis to a Rayleigh curve. This mode is based on the user providing a minimum of two successive (but not necessarily contiguous) development phases. SWEEP estimates errors for the remaining phases and the latent error content. Mode 3 lets the user set objectives for error discovery ranges based on user-selected criteria. It serves as a planning aid for forecasting the number of errors you expect to find during the development process.

A.2 MODE 1—TIME-BASED MODEL APPROACH TO ERROR ESTIMATION

Mode 1 fits a Rayleigh curve to the counts of errors discovered over time. The user works with currently available test data. The data used is the number of errors discovered and the number of test intervals executed to date. The user must normalize the error data so that each interval covers approximately the same amount of test time. For example, if the time interval for week one was 24 hours/day, but the time intervals for weeks two, three, and four were each 8 hours/day, the data for weeks two, three, and four could be combined into a virtual week two. The error discovery data must be for the same unit of code during all of the test intervals used.

Let $U(I)$ be the number of errors that you have found for the interval I ($I = 1, ...n, n \leq 100$). You must enter data for at least two intervals. The algorithm calculates the parameter $A(I)$ for each $U(I)$ according to whether you have entered a value for $U(I)$. $A(I)=1$ if you have entered a value for $U(I)$; $A(I)=0$ otherwise.

The following is a description of the nature of the iterative algorithm that determines the two parameters of the Rayleigh curve (E = the total area, and t_p = the peak) that best fit the data (the $U(I)$ that you have entered). First SWEEP iterates through T (the trial values of t_p), ranging from interval number 0.5 in increments of 0.1 to 300, or 2,996 iterations, to find the smallest value of $D(B)$, the squared error, (the square of the difference between the data points that you enter and the values generated by SWEEP). SWEEP uses the following formulas:

$$B = \frac{1}{2T^2} \quad \text{and} \quad T = \frac{1}{\sqrt{2B}}$$

$$E(B) = \frac{\sum\limits_{I=1}^{n} A(I) \cdot U(I)}{\sum\limits_{I=1}^{n} A(I) \cdot \left(e^{-B(I-1)^2} - e^{-B \cdot I^2}\right)}$$

$$DV(I,B) = E(B) \cdot (e^{-B(I-1)^2} - e^{-BI^2})$$

$$D(B) = \sum\limits_{I=1}^{n} A(I) \cdot (U(I) - DV(I,B))^2$$

Let BMIN be the value of B that produced D(BMIN), the smallest value of D(B) over the range of B. Then, the best estimate of t_p, denoted as TMIN, is given by:

$$TMIN = \frac{1}{\sqrt{2BMIN}}$$

Also, E(B) is the value of Rayleigh E—the total errors associated with the trial value of B. Thus, the best estimate of E, the total lifetime errors, symbolized as E(BMIN), is given by:

$$E(BMIN) = \frac{\sum\limits_{I=1}^{n} A(I) \cdot U(I)}{\sum\limits_{I=1}^{n} A(I) \cdot \left(e^{-BMIN(I-1)^2} - e^{-BMIN \cdot I^2}\right)}$$

SWEEP allows you to repetitively calculate S_m—the estimated number of errors to be discovered during the next (m) time intervals. Do this if you want to know how many errors are estimated to be discovered during the next m intervals. You must provide this (m) value for SWEEP to perform these calculations. S_m is calculated as follows:

$$S_m = \left(E(BMIN) \cdot \left(1 - e^{-BMIN \cdot (n+m)^2}\right)\right) - \left(E(BMIN) \cdot \left(1 - e^{-BMIN \cdot n^2}\right)\right)$$

or, simplifying:

$$S_m = E(BMIN) \cdot \left(e^{-BMIN \cdot n^2} - e^{-BMIN \cdot (n+m)^2}\right)$$

SWEEP also allows you to repetitively calculate k, the total number of time intervals required to yield a particular percentage (p) of the total number of errors (E). You must provide this (p) value for SWEEP to perform these calculations, and the percentage (p) must be less than or equal to 99.99.

SWEEP will also calculate r, where r=k−n, the number of additional intervals beyond n, required to reach p. Remember that n is the number of intervals for which error data is available.

To calculate r, you must first solve k using the following equation:

$$\frac{p}{100} = \left(1 - e^{-\text{BMIN} \cdot k^2}\right)$$

Rewriting the equation, you obtain:

$$e^{-\text{BMIN} \cdot k^2} = 1 - \frac{p}{100}$$

and:

$$-\text{BMIN} \cdot k^2 = \log_e\left(1 - \frac{p}{100}\right)$$

Therefore:

$$k^2 = \frac{\left(\log_e(1 - \frac{p}{100})\right)}{-\text{BMIN}}$$

Finally:

$$k = \sqrt{\left[\frac{\left(\log_e(1 - \frac{p}{100})\right)}{-\text{BMIN}}\right]}$$

The value of r is given by $r = k - n$, where r is the additional number of intervals over which testing must be done to reach (p) percentage of (E), the total number of errors injected during development. Having solved for k, SWEEP can now calculate S_k, the estimated total number of errors that will be discovered through time interval k.

S_k is determined using:

$$S_k = E(\text{BMIN}) \cdot \left(1 - e^{-\text{BMIN} \cdot k^2}\right)$$

S_r, the estimated number of errors that will be identified during the next r intervals in excess of S_n, the estimated errors already found, is determined by:

$$S_r = S_k - S_n$$

Where S_n is determined using:

$$S_n = E(\text{BMIN}) \cdot \left(1 - e^{-\text{BMIN} \cdot n^2}\right)$$

Note that S_n is an estimated value that may differ from the actual number of errors you enter, especially if you have null data in one or more intervals.

Thus, S_r becomes:

$$S_r = E(\text{BMIN}) \cdot \left(e^{-\text{BMIN} \cdot n^2} - e^{-\text{BMIN} \cdot (n+r)^2}\right)$$

A.3 MODE 2—PHASE-BASED MODEL APPROACH TO ERROR ESTIMATION

A.3.1 OVERVIEW

Mode 2 fits a Rayleigh curve to error discovery count data obtained on a phase-by-phase basis. It fits a Rayleigh curve to the numbers of errors/code unit discovered during each phase of the software development process. The model requires data from at least two different development phases; however, they need not be from contiguous phases. The model can estimate the errors expected to be found in later phases based on number of errors discovered in earlier phases. It can also be employed to estimate the latent error content—the expected number of errors in the post-delivery software. For the pre-test phases, the error counts are typically found during inspections. Using this mode enables you to use valuable data obtained before the code executes. This type of data is often ignored by software analysts and quality assurance personnel when estimating errors.

A.3.2 RAYLEIGH CURVE

An analysis of the number of errors discovered at successive stages of the software development process indicates that the normalized error discovery profile, when taken on a phase-by-phase basis, typically increases and then decreases as a function of phase as you move through the process (e.g., high-level design inspection to low-level design inspection). It is monomodal (has a single peak) in form. Thus, errors per code unit may be plotted as a function of each error discovery phase. The use of errors per code unit allows you to use error discovery data normalized with respect to the actual code size. Hence, you can employ data about different parts of a software product in developing an error profile, assuming a reasonable degree of homogeneity across its parts (modules or other elements of program structure).

One representation of a monomodal model is the Rayleigh curve (actually a discrete or histogram form since the data is discrete) depicted in Figure A-1. The Rayleigh curve was selected as the model of software error discovery used in SWEEP because of its high level of familiarity within the software community. It is widely used as a model of the "proper" application of labor to develop products in general (Norden 1958; Norden 1963) and more particularly for the software development process (Gaffney 1982) as well as the entire software development life cycle (Putnam 1978). The cumulative form of the Rayleigh curve, as applied to the error discovery process model used in SWEEP, is:

$$V_t = E\left(1 - e^{-Bt^2}\right)$$
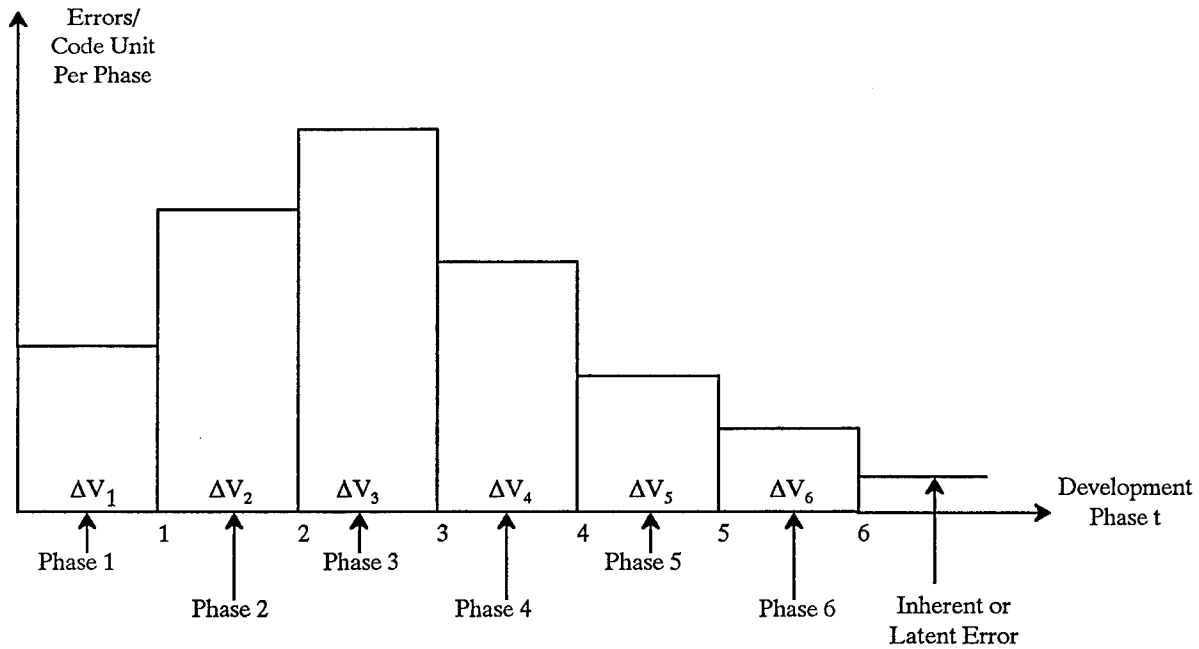
Rayleigh curve fit: $\Delta V_t = E\left[e^{B(t-1)^2} - e^{-Bt^2}\right]$

E = Total lifetime error rate per code unit

$B = \dfrac{1}{2t_p^2}$; t_p = Defect discovery phase constant, the location of the peak in a continuous fit to the data

Rayleigh curve, cumulative form:

$V_t = E\left(1 - e^{-Bt^2}\right)$

Figure A-1. Typical Rayleigh Distribution Curve for a Six-Phase Development Process

where:

V_t = Total number of errors (or errors per KSLOC) discovered through development phase (or activity number t)

E = Total lifetime error content or errors injected during development—the area under the Rayleigh curve between t=0 and t=∞

B = Shape parameter of the distribution curve

$$B = \frac{1}{\left(2t_p^2\right)}$$

where t_p= error discovery phase constant, the point at which 39% of E errors have been discovered.

The independent variable t represents the error discovery activity indexes, as illustrated in Figure A-1 for the case of six error discovery activities. SWEEP allows you to have between six and nine phases.

The incremental (phase-by-phase) form of this model is:

$$\Delta V_t = E\left[e^{B(t-1)^2} - e^{-Bt^2}\right]$$

where ΔV_t = the number of errors (or errors per KSLOC) discovered during development phase t, such as high-level design.

The number of latent errors L, the amount of errors remaining at the conclusion of development (including testing) process, is given by:

$$L = Ee^{-BM^2}$$

where M is the number of error discovery phases in the development process. If M= 6, then:

$$L = Ee^{-36B}$$

The efficiency of the error discovery process, EFF, is defined as:

$$\left(1-\frac{L}{E}\right) \times 100 \quad \text{or} \quad \frac{(E-L)}{E} \times 100$$

Thus, for the case of six cases:

$$EFF = 1 - e^{-36B}$$

Higher efficiency processes have larger values of B or smaller values of t_p; the earlier the peak is reached, the higher the efficiency of the error discovery process.

It is interesting that the two parameters of the Rayleigh curve, the location of the peak (t_p) and the area under the curve (E), correspond to mutually exclusive aspects of software error discovery. E corresponds to the quality of the development process, which relates directly to the area under the curve. Poorer processes produce (inject) more errors and have higher values of E. Poor verification methods let more of the injected errors leak to latter phases, resulting in higher values of t_p, which is the location of the peak.

A.3.3 Nature of Fitting the Data to the Rayleigh Curve

SWEEP employs an iterative procedure to fit the errors/code unit data for each phase of available data. It produces estimates of the values of error/code unit for each phase and compares them with the actual values that you enter. The algorithm estimates the values of t_p ranging from phase variable t=0.5 to t=M with increments of 0.1, where M ($6 \leq M \leq 9$) is the number of phases in the discovery profile you select. For example, the ranges of values of phase variable t for the six phases predefined in the SWEEP tool are:

- Values of 0 to 1:   Phase 1—Preliminary design

- Values of 1 to 2:   Phase 2—Detailed design

- Values of 2 to 3:   Phase 3—Coding

- Values of 3 to 4:   Phase 4—Unit test (CSU testing in DOD-STD-2167A)
  (U.S. Department of Defense 1988)

- Values of 4 to 5:   Phase 5—Integration test (CSC testing in DOD-STD-2167A)

- Values of 5 to 6:   Phase 6—System test (CSCI and system testing in DOD-STD-2167A)

The value of 0.5 indicates that it is half-way through phase one, and a value of 6 indicates the end of phase six. The Rayleigh distribution curve shown in Figure A-1 depicts these six phases. The algorithm used in SWEEP calculates the square of the difference between the actual and the estimate for each of the data points you enter. The minimum value for this estimate corresponds to the best fit for the Rayleigh approximation, which will be used to fit the data you provide to SWEEP.

There is no phase dedicated to requirements in the predefined set of SWEEP phases.

Let $U(I)$ be the number of errors that you have found for phase I ($I = 1, ...M, 6 \leq M \leq 9$). You must enter data for at least two phases (not necessarily contiguous). The algorithm calculates the parameter $A(I)$ for each $U(I)$ according to whether you have entered a value for $U(I)$. $A(I)=1$ if you have entered a value for $U(I)$; $A(I)=0$ otherwise.

The following is a description of the nature of the iterative algorithm that determines the two parameters of the Rayleigh curve (E = the total area, and t_p = the peak) that best fit the data (the $U(I)$ that you have entered). First, SWEEP iteratives through phase variable t (the trial values of t_p), ranging 0.5, in increments of 0.1, to M, where M ($6 \leq M \leq 9$) is the number of phases in the discovery profile you select, to find the smallest value of $D(B)$, the squared error, using the following formulas:

$$B = \frac{1}{2t^2} \quad \text{and} \quad t = \frac{1}{\sqrt{2B}}$$

$$E(B) = \frac{\displaystyle\sum_{I=1}^{M} A(I) \cdot U(I)}{\displaystyle\sum_{I=1}^{M} A(I) \cdot \left(e^{-B(I-1)^2} - e^{-B \cdot I^2}\right)}$$

$$DV(I, B) = E(B) \cdot (e^{-B(I-1)^2} - e^{-BI^2})$$

$$D(B) = \sum_{I=1}^{M} A(I) \cdot (U(I) - DV(I, B))^2$$

Let BMIN be the value of B that produced D(BMIN), the smallest value of $D(B)$. Then, the best estimate of t_p, denoted as TMIN, is given by:

$$TMIN = \frac{1}{\sqrt{2BMIN}}$$

Also, $E(B)$ is the value of Rayleigh E—the total errors associated with the trial value of B. Thus, the best estimate of E, the total lifetime errors, symbolized as E(BMIN), is given by:

$$E(BMIN) = \frac{\sum\limits_{I=1}^{M} A(I) \cdot U(I)}{\sum\limits_{I=1}^{M} A(I) \cdot \left(e^{-BMIN(I-1)^2} - e^{-BMIN \cdot I^2} \right)}$$

A.4 MODE 3—PLANNING AID

A.4.1 OVERVIEW

Mode 3 allows you to set objectives for error discovery ranges based on user-specified criteria. Mode 3 uses a phase-based model based on a Rayleigh curve to estimate latent error content and the peak location of errors.

A.4.2 SOFTWARE STATISTICAL QUALITY CONTROL

Some of the basic elements of software statistical quality control that are directly related to Mode 3 are:

- Using graphs to present goals and actual performance at each phase of the software development process. Note that the software process activities, such as preliminary design, are analogous to the stations of a manufacturing process.

- Plotting error discovery versus time or equivalent (development phase number).

- Defining a control (tolerance) band that establishes a tolerance range for each error discovery phase; the tolerance range relates to the control limits of your quality control process.

- Using the graphs to indicate departures from satisfactory (quality-level) behavior and to anticipate and help forecast departures from it.

By using the planning mode with the aid of the graphics it provides, you can employ software product data to make inferences about the development process used to create the product, both during development and after it has been completed. By monitoring the product, you can improve your control of the software development process. Such data can be used in your decision process and action strategy to support modifying the product under development while you are creating it, the process you employ to produce it, or, if post facto analysis is employed, the process after you have created it. The Mode 3 planning aid provides developers with data in a timely manner to aid them in discovering departures from the development process and supports returning it to control. You can use the information displayed in the planning mode graphs to prompt corrective action.

The planning aid provides an alternative to establishing tolerance profiles and is based on the work done by Genichi Taguchi (Barker 1990). Taguchi has focused on improving the quality of measuring systems and measuring procedures by introducing the concept of sensitivity (Cherng 1989). Taguchi suggests using the concept of signal-to-noise ratio to evaluate the sensitivity of a measurement system. The concept of signal-to-noise ratio is widely applied by electrical engineers. The basic concept is that there is a desired signal value, G, and it is embedded in noise whose power is σ^2, corresponding to statistical variance. Then S, the signal-to-noise ratio, is equal to G/σ. The inverse of this value is known

to statisticians as the coefficient of variation, when G is taken as the expected value of a random variable. Larger values of signal-to-noise ratio correspond to smaller values of the error tolerance range. It does not appear that these ideas have been applied to software before the application in SWEEP as described here.

A.4.3 EQUATIONS FOR THE ERROR DISCOVERY PROFILES DISPLAYED

The primary equation in the planning aid is the same one used in the phase-based model. Listed below are the primary inputs required for using Mode 3.

There are two ways to establish the goal error discovery profile pattern. You can enter:

- E, the desired total error content (total injected), and L, the desired latent error content

- E, the desired total error content (total injected), and t_p, the desired location of peak error discovery

There are two ways to establish the (optional) tolerance profile patterns. You can enter:

- p_u, the desired percentage upper tolerance level, and p_l, the desired percentage lower tolerance level

- S, the desired value of the signal-to-noise ratio

The equation for the goal error profile (G_t) value for phase t is:

$$G_t = E\left[e^{-B(t-1)^2} - e^{-Bt^2}\right]$$

The equations for the upper (U_t) and lower (L_t) error profile values for phase t are:

$$U_t = E \cdot (1 + p_u \cdot 0.01) \cdot \left[e^{-B(t-1)^2} - e^{-Bt^2}\right]$$

and

$$L_t = E \cdot (1 - p_l \cdot 0.01) \cdot \left[e^{-B(t-1)^2} - e^{-Bt^2}\right]$$

When you enter the desired signal-to-noise value S, the program sets $p_u = p_l = p$, and $p = 100/S$.

Then,

$$U_t = \frac{E(S+1)}{S} \cdot \left[e^{-B(t-1)^2} - e^{-Bt^2}\right]$$

and

$$L_t = \frac{E(S-1)}{S} \cdot \left[e^{-B(t-1)^2} - e^{-Bt^2}\right]$$

Note that S must be greater than 1 for these equations to hold.

This page intentionally left blank.

GLOSSARY

Code execution	When code has the ability to demonstrate the functionality of the particular unit of code.
Code inspection	Peer review of a programmer's code to find problems and improve quality.
Cumulative curve	A graphical representation of a Rayleigh distribution curve. The Y, or vertical, axis represents the total number of errors discovered, and the X, or horizontal, axis represents successive test intervals.
Defect	*See* Error.
Defect density	The ratio of the number of defects to the amount of code in which they were discovered.
Defect discovery	The finding of defects in a software element during a verification activity.
Defect discovery profile	A plot of the number of defects (errors) found in each development phase or in each time interval, over all of the development phases or test time intervals, respectively.
Error	The term error is summarized and defined using the following four definitions:
	Failure: A manifestation of an inability of the software product to conform to requirements.
	Fault: An error in software that could cause it to produce an incorrect result or an invalid output.
	Defect: The evidence of the existence of a fault.
	Error: The human action of commission or omission that results in the software containing a fault. *See also*: Injected error.
Error detection	The method or practice used to discover an error during software development. Same as error removal.

Error discovery profile	A plot of the number of errors (defects) found in each development phase or in each time interval, over all of the development phases or test time intervals, respectively.
Error discovery rate	The rate at which errors are discovered during a given time interval or development phase.
Error prevention	The process of detecting, fixing, and preventing errors from reoccurring.
Expansion ratio	The ratio of the number of statements of code to the number of statements of design from which they were derived.
Fault	*See* Error.
Failure	*See* Error.
Incidence curve	A graphical representation of a Rayleigh distribution curve for a particular interval. The Y, or vertical, axis represents the number of errors discovered in the interval, and the X, or horizontal, axis represents one or more data groupings for the interval.
Injected error	An error introduced into the software during any phase of software development.
Latent error content	The number of errors remaining in the software at the time the software is delivered to the customer.
Macro error	When running SWEEP an EXCEL macro instruction fails, and an indication of this is provided to the user.
Metrics	Quantities indicative of some aspect of a software product, such as the number of defects found during the preliminary design activity or during the period of testing.
Peak location	The number of the time interval or phase of the peak of the Rayleigh fit to the error or defect discovery.
Radio buttons	User selectable buttons on SWEEP dialog boxes that have the form of a small circle with a dot in the middle.
Rayleigh distribution curve	The Rayleigh curve fit to error or defect discovery data produced by the SWEEP models.

Signal-to-noise ratio

The ratio of the amplitudes of a signal and the noise in which in which it is imbedded. Used in SWEEP to indicate the ratio of the desired value of error discovery in some particular phase and the size of the tolerance about it.

Squared error

The sum of the squares of the differences between the error count data entered into SWEEP by a user and the counts that SWEEP generates as part of the algorithm SWEEP employs to fit a Rayleigh curve to the user-entered data.

Tolerance band

The user selected range about a desired value or goal for defect discovery in each phase of the discovery profile.

Verification

The process of determining whether a representation of a software system, such as detailed design, is a correct elaboration of the requirements.

This page intentionally left blank.

LIST OF ABBREVIATIONS AND ACRONYMS

CMM	Capability Maturity Model
CSC	computer software component
CSCI	computer software configuration item
CSU	computer software unit
ESP	Evolutionary Spiral Process
KSLOC	thousand source lines of code
PC	personal computer
RAM	random access memory
SEI	Software Engineering Institute
SWEEP	Software Error Estimation Program

Abb-1

This page intentionally left blank.

Abb-2

REFERENCES

Barker, Thomas B.
1990

"Engineering Quality by Design." In *Interpreting the Taguchi Approach.* New York, New York: ASQC Quality Press.

Cherng, John G.
1989

"Improvement of Sound Measurement Procedures Using the Taguchi Method." In *Seventh Symposium on Taguchi Methods,* American Supplier Institute, October 1989.

Davis, Charles F., and
John E. Gaffney
1988

An Approach to Estimating Software Errors and Availability, SPC-TR-88-007. Herndon, Virginia: Software Productivity Consortium.

Gaffney, J.E., Jr.
1982

"A Macroanalysis Methodology for Assessment of Software Development Costs." In *The Economics of Information Processing,* 2, 177. Edited by R. Goldberg and H. Loring. New York, New York: John Wiley and Sons.

1984

"On Predicting Software-Related Performance of Large-Scale Systems." In *Proceedings of the International Conference of the Computer Measurement Group.* December 1984.

Gaffney, J.E., Jr. , V. Ferrando,
and S. Jones
1993

A Method of Outage Estimation: Application to the Voice Switching and Control System. Presented at the IEEE Communications Society, Quality Assurance Management Committee 1993 Workshop, May 1993.

Humphrey, Watts S.
1989

Managing the Software Process. Reading, Massachusetts: Addison-Wesley.

Kitson Masters, Zubrow
1993

Software Engineering Symposium, Software Engineering Institute, Pittsburgh, Pennsylvania.

Microsoft
1991

Microsoft Excel User's Guide. Redmond, Washington: Microsoft Corporation.

Norden, P.V.
1958

"Curve Fitting for a Model of Applied Research and Development Scheduling." *IBM Journal of Research and Development,* July 1958, 232.

1963

"Useful Tools for Project Management." In *Operations Research in Research and Development.* Edited by B.B. Dean. New York, New York: Wiley.

Paulk, M.,
C. Weber,
S. Garcia,
M. Chrissis, and
M. Bush
1993

Key Practices of the Capability Maturity Model, Version 1.1, CMU/SEI-93-Tr-25. Pittsburgh, Pennsylvania: Software Engineering Institute.

Putnam, L.H.
1978

"A General Empirical Solution to the Macro Software Sizing and Estimating Problem." In *IEEE Transactions on Software Engineering,* July 1978, 345.

Rome Air Development
Center
1977

Software Systems Reliability: A Raytheon Project History, RADC-TR-77-188, June 1977, Rome Air Development Center, Griffiss Air Force Base, New York.

Software Productivity
Consortium
1993

Process Engineering With the Evolutionary Spiral Process Model, SPC-93098-CMC. Herndon, Virginia: Software Productivity Consortium.

U.S. Department of Defense
1988

Defense Systems Software Development, DOD-STD-2167A. Washington, D.C.: U.S. Department of Defense.