The Software Measurement Tool That Software Developers Use

Introduction

Function point analysis is a widely used software measurement technique that helps organizations measure the size and complexity of their software projects, estimate the effort and cost involved in developing the software, and track the progress of software development projects. Function point counting is a standardized method for measuring the functional size of software, and it is based on the number of user-visible functions that the software provides.

Function point analysis is a valuable tool for software project managers, software developers, and other

stakeholders involved in software development. It can help them to:

- Estimate the size and complexity of software projects more accurately
- Estimate the effort and cost involved in developing the software more accurately
- Track the progress of software development projects more effectively
- Identify potential risks and challenges in software development projects early on
- Make better decisions about software development projects

Function point analysis is a relatively simple and straightforward technique to learn and use. It can be applied to software projects of all sizes and types. Function point analysis is also a very flexible technique, and it can be tailored to meet the specific needs of different organizations.

In this book, we will provide a comprehensive overview of function point analysis. We will discuss the basics of function point counting, as well as more advanced topics such as function point analysis for object-oriented software, component-based software, service-oriented architectures, and multi-tiered architectures. We will also discuss how function point analysis can be used to improve software quality, estimate software cost, measure software productivity, and improve software process.

By the end of this book, you will have a thorough understanding of function point analysis and how it can be used to improve the efficiency and effectiveness of your software development projects.

Book Description

Function point analysis is a powerful software measurement technique that can help organizations improve the efficiency and effectiveness of their software development projects. This book provides a comprehensive overview of function point analysis, covering everything from the basics of function point counting to more advanced topics such as function point analysis for object-oriented software, component-based software, service-oriented architectures, and multi-tiered architectures.

In this book, you will learn how to:

- Use function point analysis to estimate the size and complexity of software projects
- Use function point analysis to estimate the effort and cost involved in developing software
- Use function point analysis to track the progress of software development projects

- Use function point analysis to identify potential risks and challenges in software development projects
- Use function point analysis to make better decisions about software development projects

Function point analysis is a valuable tool for software project managers, software developers, and other stakeholders involved in software development. It can help organizations to deliver software projects on time, within budget, and with the desired quality.

This book is written in a clear and concise style, and it is packed with practical advice and examples. It is the perfect resource for anyone who wants to learn more about function point analysis and how it can be used to improve software development projects.

Whether you are a software project manager, a software developer, or another stakeholder involved in software development, this book will provide you with the knowledge and skills you need to use function point

analysis to improve the efficiency and effectiveness of your software development projects.

Chapter 1: The Art of Software Measurement

The Importance of Software Measurement

Software measurement is the process of collecting, analyzing, and interpreting data about software products and processes. It is an essential part of software engineering, and it can be used to improve the quality, productivity, and efficiency of software development.

There are many different reasons why software measurement is important. Some of the most important reasons include:

- **To improve software quality:** Software measurement can be used to identify defects in software products and processes. This information can then be used to improve the quality of the software and to prevent defects from occurring in the future.

- **To improve productivity:** Software measurement can be used to identify bottlenecks in software development processes. This information can then be used to improve the efficiency of the software development process and to increase productivity.
- **To improve efficiency:** Software measurement can be used to track the progress of software development projects. This information can then be used to identify areas where the project is falling behind schedule or over budget. This information can then be used to take corrective action and to improve the efficiency of the project.
- **To make better decisions:** Software measurement can be used to provide decision-makers with the information they need to make better decisions about software development projects. For example, software measurement can be used to help decision-makers decide

which software development tools and technologies to use, how to allocate resources, and how to schedule software development projects.

Software measurement is an essential part of software engineering. It can be used to improve the quality, productivity, and efficiency of software development. By measuring software products and processes, organizations can identify problems and take steps to improve them. This can lead to better software, faster development times, and lower costs.

Chapter 1: The Art of Software Measurement

Different Types of Software Metrics

Software metrics are a set of quantifiable measures that are used to assess the quality, progress, and productivity of software development projects. They provide valuable insights into the health and performance of software projects, and they can be used to identify areas for improvement.

There are many different types of software metrics, each with its own purpose and application. Some of the most common types of software metrics include:

- **Size metrics:** These metrics measure the size of a software project, such as the number of lines of code, the number of function points, or the number of modules.
- **Complexity metrics:** These metrics measure the complexity of a software project, such as the

number of control flow paths, the number of decision points, or the number of cyclomatic complexity.

- **Quality metrics:** These metrics measure the quality of a software project, such as the number of defects, the number of change requests, or the mean time between failures.
- **Progress metrics:** These metrics measure the progress of a software project, such as the percentage of tasks completed, the number of milestones reached, or the number of days since the project started.
- **Productivity metrics:** These metrics measure the productivity of a software project, such as the number of lines of code produced per day, the number of function points produced per day, or the number of defects fixed per day.

Software metrics can be used to:

- **Estimate the effort and cost of software development projects:** Software metrics can be used to estimate the amount of time and resources that will be required to develop a software project.
- **Track the progress of software development projects:** Software metrics can be used to track the progress of a software development project and to identify any potential problems.
- **Identify areas for improvement in software development projects:** Software metrics can be used to identify areas in a software development project that can be improved, such as areas where the code is too complex or where there are too many defects.
- **Make better decisions about software development projects:** Software metrics can be used to make better decisions about software development projects, such as decisions about

which features to implement or which development methodologies to use.

Software metrics are an essential tool for software project managers, software developers, and other stakeholders involved in software development. They can help organizations to deliver software projects on time, within budget, and with the desired quality.

Chapter 1: The Art of Software Measurement

Function Points: An Overview

Function point analysis is a standardized method for measuring the functional size of software. It is based on the number of user-visible functions that the software provides. Function point analysis is a valuable tool for software project managers, software developers, and other stakeholders involved in software development. It can help them to:

- Estimate the size and complexity of software projects more accurately
- Estimate the effort and cost involved in developing the software more accurately
- Track the progress of software development projects more effectively
- Identify potential risks and challenges in software development projects early on

- Make better decisions about software development projects

Function point analysis is a relatively simple and straightforward technique to learn and use. It can be applied to software projects of all sizes and types. Function point analysis is also a very flexible technique, and it can be tailored to meet the specific needs of different organizations.

Function point analysis is based on the following key concepts:

- **User-visible functions:** These are the functions that the software provides to the user. They are typically identified by interviewing users and reviewing software requirements.
- **Transaction:** A transaction is a unit of work that is performed by the software. It is typically characterized by a single input and a single output.

- **Data element:** A data element is a piece of information that is used by the software. It can be a single character, a number, or a complex data structure.

To calculate the function point count for a software project, the following steps are typically followed:

1. Identify the user-visible functions in the software.
2. Count the number of transactions for each user-visible function.
3. Count the number of data elements for each user-visible function.
4. Apply a weighting factor to each user-visible function based on its complexity.
5. Sum the weighted transaction counts and data element counts to calculate the function point count.

The function point count can then be used to estimate the effort and cost involved in developing the software.

It can also be used to track the progress of software development projects and to identify potential risks and challenges.

Function point analysis is a valuable tool for software project managers, software developers, and other stakeholders involved in software development. It can help them to improve the efficiency and effectiveness of their software development projects.

This extract presents the opening three sections of the first chapter.

Discover the complete 10 chapters and 50 sections by purchasing the book, now available in various formats.

Table of Contents

Chapter 4: Advanced Function Point Counting Techniques * Function Point Counting for Object-Oriented Software * Function Point Counting for Component-Based Software * Function Point Counting for Service-Oriented Architectures * Function Point Counting for Multi-Tiered Architectures * Function Point Counting for Distributed Systems

Chapter 5: Function Point Analysis and Reporting * Function Point Analysis Reports * Function Point Analysis Tools * Function Point Analysis Best Practices * Function Point Analysis Case Studies * Function Point Analysis Trends

Chapter 6: Function Points and Software Quality * The Relationship Between Function Points and Software Quality * Using Function Points to Improve Software Quality * Function Points as a Measure of Software Maintainability * Function Points as a Measure of Software Reliability * Function Points as a Measure of Software Testability

This extract presents the opening three sections of the first chapter.

Discover the complete 10 chapters and 50 sections by purchasing the book, now available in various formats.