Function Points - background information

On this page:

A bit of history...

The use of Function Points for measuring software functional size was initially introduced in the mid-1970s and is used today by organizations worldwide. Allan Albrecht (IBM) was the first person to publicly release a method for functionally sizing software. This method was called Function Point Analysis (Albrecht, 1979, 1981). Since its formation in 1986 the International Function Point Users Group (IFPUG) has continuously maintained and enhanced Albrecht's original method.

Evaluating the software Functional Size

Function Point is now a normalized metric that can be used consistently with an acceptable degree of accuracy. The value is centered on the ability to measure the size of any software in terms of functionalities provided to end-users. Function Point counting is qualified as "technology agnostic" in that it does not depend on the technologies or programming languages.

Function Point Counting method evaluates a software deliverable and measures its size based on its functional characteristics. It takes in to consideration the following constituents of an application:

- **External Inputs**: Input data that is entering a system (logical transaction inputs, system feeds)
- **External Outputs and External Inquires**: data that is leaving the system (on-line displays, reports, feeds to other systems)
- **Internal Logical Files**: data that is processed and stored within the system (logical groups of user defined data)
- **External Interface Files**: data that is maintained outside the system but is necessary to satisfy a particular process requirement (interfaces to other systems)

Along with selected other measures, Function Points can be used by organizations for different purposes:

- Quality and productivity analysis
- Estimating the costs and resources required for software development, enhancement, and maintenance
- Normalizing data used in software comparisons
- Determining the size of a purchased application package (Commercial On-the-shelf or customized systems) by sizing all the functionalities included in the package
- Enabling users to determine the Return on Investment (RoI) of an application by sizing the functionalities that specifically match the requirements from their organization

Keep in mind

Function Point is a unit of measurement related to the functional size of software applications like meters and inches are about to measure distance, degrees to measure temperature...

There are several basic usages and characteristics: it measures the amount of functionalities in a software application, where the larger the number of function points, the more functionalities are implemented in the application. Function points are also used to size business requirements.

All systems have input, output, and storage components. These components are always considered in the estimation of the number Function Points in a system. An application will be composed of some interwoven elementary processes, many of which do not make sense until they are woven together. All these elementary processes make up the entire software application, even when they are not built by the same person. Therefore, a software application can be defined as an interwoven defined set of elementary processes, where the processes are woven together becoming interdependent and forming what is called a software application.

In Function Points we are looking for data that is at rest. Some elements will be viewed as persistent such as storage elements like customer and student enrollment data files. Other elements will be view as data in motion via some transaction like: Add customer, Inquire on an employee, or Print check…

Function Point Usage Scenarios

There are three frequent scenarios in using Function Points: measuring software size in order to estimate effort and cost, normalizing other measures, and benchmarking to help decision-making.

1. Function Points have been defined around components that can be identified in a well-written specification. Consequently Function Points can be counted from the specification before other sizing measures such as lines of code are available. This is the reason why many commercial cost estimation formulas integrate Function Points in their calculations as the preferred sizing measure to deduce effort and cost figures. These formulas can be adjusted to increase the result accuracy as soon as the number of Function Points is available for the delivered software.
2. Function Points are frequently used to normalize other measures. For instance, the total number of defects detected in a software system can be normalized by the number of Function Points to deduce the defect density per Function Point. This allows a better comparison between systems that differ by their size as well as the programming language used.
3. Because they are counted from constructs and are independent of the programming language, Function Points are a preferred method for comparing software systems when benchmarking the sub-contractor productivity. As a consequence, productivity KPI are often based on Function Points.

Keep in mind

Function Point is a unit of measurement that can be used on the following use cases:

1. **Normalization & Benchmarking**: Detect portfolio outliers, identify improvement opportunities and track evolution of size, risk, complexity and quality
2. **Productivity Measurement**: Monitor, track and compare ADM teams’ utilization, delivery efficiency, throughput and quality of outputs
3. **Quantify Effectiveness of Transformation Initiative**: Optimize operating costs while preserving throughput and de-risking business transformation initiative
4. **ADM Supplier Outcome Measurement**: Provide visibility to management; manage risk, quality and throughput through enhanced Service Level Agreement
5. **Optimizing ADM Estimation**: Provide visibility to estimation team to better understand challenges and fine tune effort estimation ratios.

How does CAST AIP count Function Points?

CAST AIP uses **OMG compliant Automated Function Point counting technology** as the basis for its measurements. In addition, CAST AIP uses two methods to count **Enhancement Function Points**:

- **OMG Automated Enhancement Points (AEP)** - the default
- **OMG Enhancement Function Points (EFP)** - legacy, not active by default

You can find out more information about these methods in the following documentation:

CAST OMG-compliant Automated Function Points

- CAST Enhancement Function Points Estimation - EFP
- CAST Automated Enhancement Points Estimation - AEP
- ISO declaration of conformance