

**SIG EVALUATION CRITERIA
PERFORMANCE EFFICIENCY:
GUIDANCE FOR PRODUCERS**
Version 1.2

Author

Marco di Biase
Consultant
+31 6 1561 9102
m.dibiase@softwareimprovementgroup.com

Version 1.2 – March, 2021

TABLE OF CONTENTS

1.	Introduction	3
2.	Guidance for producers	4
2.1	Terminology.....	4
2.2	Internal communication.....	5
2.3	External communication.....	5
2.4	Single Transaction Optimization	6
2.5	Transaction Scalability.....	6
2.6	Data Scalability.....	6
2.7	Isolation.....	6
2.8	Resource Elasticity	7
2.9	Observability.....	7

1. INTRODUCTION

This document describes the SIG evaluation criteria for the performance efficiency of software systems. These criteria are intended for the standardized evaluation of aspects that influence performance efficiency: product, process, and operational support. The purpose of the evaluation is to provide an instrument to developers for guiding improvement of the products they create and enhance, and to acquirers for comparing, selecting, and accepting pre-developed software.

This guidance document provides explanation to software producers about the measurement method of SIG applied for evaluation. This document is not intended to be a guide to developing software with high performance attributes.

2. GUIDANCE FOR PRODUCERS

The performance model assesses the *design* of a system regarding its ability to reach its performance goals. The applied model has a high rating when the assessed system is designed, implemented and operated with performance in mind, e.g. when it is likely that the design allows a good performing system and when system engineers are able to detect and solve performance incidents in a timely manner. A low rating indicates that the performance of that system is unmanaged.

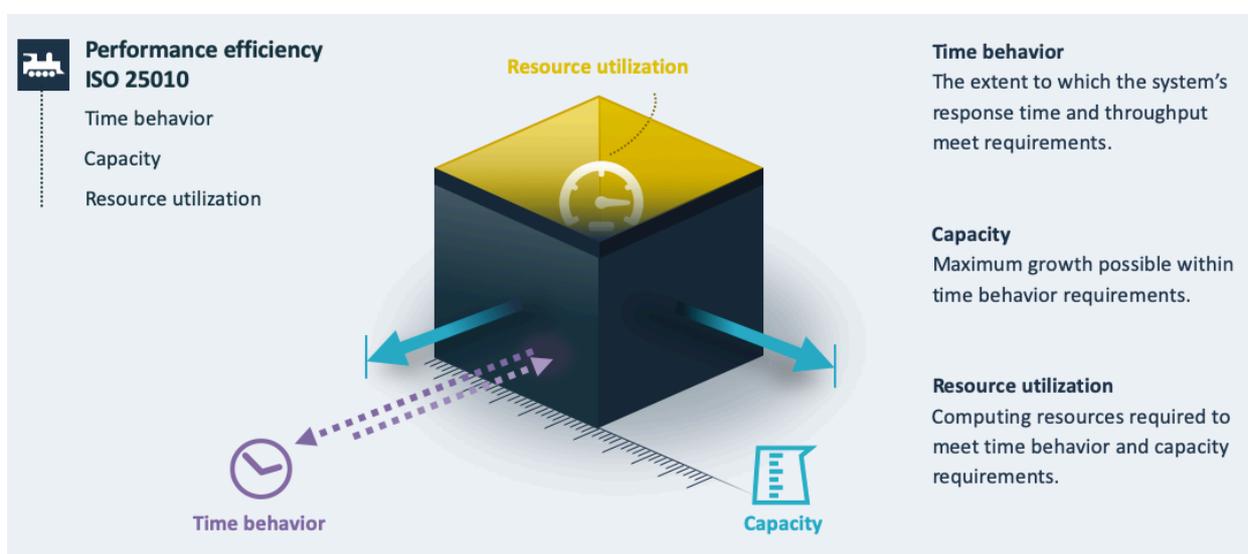
The model does not directly assess the performance *behavior of a system* in use, i.e. a slow system can have a high rating and a fast system a low rating. However, one of the key aspects in the model is *Observability*, a property which makes it transparent whether performance objectives can be observed. Thus, for a system with a good score it is easy to determine how well it performs in use.

The performance model is based on nine system properties. These systems properties do not only cover the computational efficiency of the application software, but also the runtime architecture and the ability to monitor performance in production. A view on the deployed system in addition to the source code is necessary to answer the underlying questions of the performance system properties.

2.1 TERMINOLOGY

The ISO 25010 describes performance efficiency as “Performance relative to the amount of resources used under stated conditions”.

Performance efficiency is divided into three sub-characteristics as follows:



In addition to the ISO 25010 terminology we use the following terms:

- **Transaction:** A collection of steps that are executed by a system in order to fulfil a specific functionality (e.g. a user request).
- **Step:** Isolated unit of functionality contained within a single transaction. Steps are typically isolated by process or machine boundaries.
- **Connection:** Communication channel between two steps, possibly capable of temporarily storing requests.
- **Workload:** A characterization of the work a system has to handle (e.g. the number of request that has to be handled at a certain time interval).

To determine the time behavior, capacity and resource utilization of a system SIG analyses a number of system properties that influence these characteristics. System properties are traits of a system or its processes that can objectively assessed by analyzing process artefacts, design or source code. The following figure shows how the characteristics are operationalized by system properties. In this table, a cross (X) indicates that a property has a major influence on a performance efficiency sub-characteristic.

	Internal comm.	External comm.	Single transaction optimization	Transaction scalability	Data scalability	Isolation	Resource elasticity	Observability
Time behavior	■	■	■				■	■
Capacity				■	■	■		■
Resource utilization			■				■	■

In the following sections the properties that are investigated by SIG are explained in more detail.

2.2 INTERNAL COMMUNICATION

A business transaction in a system is normally composed of several steps. The overall transaction time is influenced by the amount of communication between (sub)processes of the transaction. Moreover, the amount of data and these inter-process communication messages can vary largely. It might also be needed to transform data when sending it to another process, for example to intermediate message formats like XML or JSON. In general, having more steps and data transfer within a transaction makes the transaction time less predictable. *Internal communication* expresses the degree to which the internal communication between steps of a transaction influences the total transaction time. Less interaction, small messages, and few transformations will have a positive effect.

A low rating will typically result from evidence of unnecessary transformations, chattiness and excessive amounts of data in the communications.

2.3 EXTERNAL COMMUNICATION

Applications sometimes invoke external services. These external services influence the transaction time, which introduces a factor of uncertainty since the system has no control over the external service's time behavior. *External communication* expresses the degree of influence of external processes on the total transaction time.

A low rating will result for example from the lack of guaranteed response times and redundant calls to external services.

2.4 SINGLE TRANSACTION OPTIMIZATION

Every step needs an amount of time or resources to handle a part of a complete transaction. *Single Transaction Optimization* indicates to what extent common, relatively easy to implement strategies for optimization have been applied. The optimization can be directed towards single steps or towards the number of steps typically executed for a single transaction.

A low rating will result if there are obvious optimizations that would apply but have not been implemented or investigated. Another reason would be if single transaction times differ largely from what is common in the industry or if the root cause of slow transactions has not been investigated.

2.5 TRANSACTION SCALABILITY

Every system has a capacity up to which the response time and other performance parameters will satisfy requirements. When the workload increases (e.g. in terms of requests or number of concurrent users) it may be necessary to extend the capacity. The *Transaction scalability* of the system captures how easy it is to increase this capacity, both for individual components and for the system as a whole. Because if one of the component cannot be scaled it will eventually become a bottleneck, even if all the other components can scale without problems. The rating for transaction scalability also reflects the level of automation at which the system can scale up to handle more transactions. If scaling is not automated the manual activities that need to be performed will eventually become a bottleneck for handling more transactions.

A low rating results if there is evidence that the systems current or future transaction load cannot be accommodated or only with a large manual effort.

2.6 DATA SCALABILITY

A transaction operates on data. The volume and characteristics of the data influence the response time of a transaction. Often there will be multiple data-related parameters that have an impact on performance. For example, the approach used to display a list of 10 items can typically not be used to display a list of 100 million items, as this would cause significant performance problems in multiple areas (e.g. database queries take too long, the user interface is no longer responsive).

Data Scalability expresses how easy it is to adjust the system when the amount of data the system has to work on is increased. The rating does investigate if there are valid scaling strategies for each component of the system that is affected by the increased data and the system as a whole.

A low rating results if there is evidence that the amount of data the system has to handle currently or in the future cannot be accommodated or only with a large manual effort.

2.7 ISOLATION

Isolation determines the degree to which a system in operation excludes external influences that may impact the performance and especially the consistency of response times.

A low rating results for example from a system that shares hardware or software resources with other applications. Another case is a system where maintenance activities or batch jobs are executed during service hours or when a substantial workload is put on the application by requests from external systems.

2.8 RESOURCE ELASTICITY

Most systems have strong variations in workload. For instance, during office hours the load may be significant, while at night, in weekends, and during holidays the load may dwindle. Ideally the consumed computer resources and associated costs of a service rise and fall proportionally to the workload.

A low rating for *Resource Elasticity* results when the consumed computer resources are not scaling up and down according to the workload of the application.

2.9 OBSERVABILITY

In order to know the actual performance of the system and to support problem analysis and resolution it is key to be able to measure and monitor the system. The system design and operation should support this. There are three general areas of interest for performance monitoring:

- **Detection:** How is my system currently performing? Does it satisfy requirements? Is the performance consistent?
- **Prediction:** what are the capacity limits of the system. How do upcoming changes in workload and system design impact performance?
- **Problem analysis:** What caused the performance problems I have? In which external or internal component are they manifest? Can they be correlated with external influences?

A low rating results if essential data on the system behavior is not collected or not preserved over longer period for trend analysis. Also all relevant types of data should be observable: resource consumption, response times, throughput and workload fluctuations.



Fred. Roeskestraat 115
1076 EE Amsterdam
The Netherlands

www.softwareimprovementgroup.com
marketing@softwareimprovementgroup.com

 Getting software right for a healthier digital world