4.2.6.1 Effectiveness Metrics

Return to Usability

About

Return to Top

There are two ways to calculate effectiveness:

- Count the success of completing tasks
- Count the errors while completing tasks

Using Task Completion Rates

Return to Top

Effectiveness measures the ratio of tasks completed versus those that are not completed. It is usually reported as a simple percentage such as 80% of the tasks are completed. The data is collected using a simple binary (i.e., yes = '1' and no = '0') for each user for each task.

Mathematically:

$$Effectiveness = \frac{\sum_{user=1}^{numberOfTesters} Completed\,Tasks\,by\,user}{\sum_{user=1}^{numberOfTesters} Total\,Tasks\,assigned\,to\,user} \times 100\,\%$$

Here is an example where the system uses 10 tasks and all the tasks are completed by 5 users. The results are:

user	Completed Tasks	Incomplete Tasks	Total Tasks	Effectiveness
user 1	8	2	10	**80%**
user 2	9	1	10	**90%**
user 3	10	0	10	**100%**
user 4	5	5	10	**50%**
user 5	6	4	10	**60%**
total	**38**	**12**	**50**	**76%**

This efficiency ratio is referred to as the **Completion Rate** or **Fundamental Usability Metric**. As a result of its simplicity and ease of understanding, the **Completion Rate** is a popular way of reporting efficiency. However, in order to determine the efficiency, a working system with real users needs to be available. This is a major advantage of using the Agile Model since the efficiency can be calculated during each Sprint and used as a guidance for future sprints.

Using Error Rates

Return to Top

Another way to understand an application or system efficiency from a usability perspective is to count the errors instead of counting the success of completing tasks. Every error can have more consequences requiring more work to “undo” thus negatively impacting Effectiveness and efficiency. The following is a partial list that users can make while completing tasks[1] :

- **Unintended actions** - using a `next` operation without intending.
- **Slips entering information** - pressing a `return escape` key accidentally
- **Mistakes in data entry** - twiddling the digits in a phone number or social security number or having autocorrect inadvertently “correct” an entry
- **Omissions in data entry** - forgetting login as an existing user rather than creating a new user, skipping filling in age

Recording each instance of an error along with a description can help refine the system to prevent these errors in the future. For example, “user entered last name in the first name field”.

In many ways, this metric can be considered similar to the Task Complete Rate metric but instead of counting completed tasks the errors are counted. Note, many there can be many errors required to complete a single task. For example, a data entry form required for one task can have errors associated with each field. Each of these errors can have a negative impact on the overall efficiency.

Another enhancement to this metric wold be to assign a weight to the error. For example, data entry errors are less dramatic and important than unintended actions. Therefore, unintended actions would receive a higher weighting than data entry.

DIDO Specifics

Return to Top

☐ TBD

1)
Jeff Sauro, 10 Essential Usability Metrics, Measuring U, 30 November 2011, Accessed 19 November 2020, https://measuringu.com/essential-metrics/

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:2_nonfunc:30_usability:effectiveness&rev=1610747163

Last update: **2021/01/15 16:46**