**Code Changes**

Question: How many changes were made to the source code during a specified period?

## Description

These are changes to the source code during a certain period. For "change" we consider what developers consider an atomic change to their code. In other words, a change is some change to the source code which usually is accepted and merged as a whole, and if needed, reverted as a whole too. For example, in the case of git, each "change" corresponds to a "commit", or to be more precise, "code change" corresponds to the part of a commit which touches files considered as source code.

## Triggers

By analyzing the code changes in a certain period of time, the user's update cycle can be obtained, and the health of the code can also be obtained. You can also restore the unstable code to get the previous stable version.

Volume of coding activity. Code changes are a proxy for the activity in a project. By counting the code changes in the set of repositories corresponding to a project, you can have an idea of the overall coding activity in that project. Of course, this metric is not the only one that should be used to track volume of coding activity.

## Actors

User:

Users deal with the front-end, so they will see the volume of coding activity, see the code changes in the period of time.

Contributor:

Contributor will work on the code, and change the code, and update the code.

## Preconditions

There are certain amount of codes in the repo, and they are all having the time tag for each changes, which means, they are having a time line for their changes.

## Main Success Scenario (Goals)

Users can learn about the types and frequency of activities involved in developing code. By counting the code changes in the set of repositories corresponding to a project, you can have an idea of the overall coding activity in that project.

## Alternate Success Scenarios

Users can view the code changes during the period of time. And they can decide what going next, and what process they need to do for the next step.

## Failed End Condition

Users cannot tell the difference of the code changes during a period of time. It can not provide ang information of the volume of the code, or the volume of the code changes.

## Extensions

## Steps of Execution (Requirements)

## A use case diagram

图示

描述已自动生成

## Dependent Use Cases

GrimoireLab provides this metric out of the box.

* View an example on the CHAOSS instance of Bitergia Analytics.
* Download and import a ready-to-go dashboard containing examples for this metric visualization from the GrimoireLab Sigils panel collection.

**Specific description: Git**

Mandatory parameters (for Git):

* Date type. Either author date or committer date. Default: author date.  
  For each git commit, two dates are kept: when the commit was authored, and when it was committed to the repository. For deciding on the period, one of them has to be selected.
* Include merge commits. Boolean. Default: True.  
  Merge commits are those which merge a branch, and in some cases are not considered as reflecting a coding activity.
* Include empty commits. Boolean. Default: True.  
  Empty commits are those which do not touch files, and in some cases are not considered as reflecting a coding activity.

## Reference

<https://www.odoo.com/documentation/13.0/reference/guidelines.html#tag-and-module-name>

<https://github.com/chaoss/wg-evolution/blob/master/metrics/Code_Changes.md>