```php
<?php

namespace Helper;

use Models\TestSerie;

class TestHelper
{
    private static $series = [];
    public static $tests = [];

    /*
     * Loads all Tests into local array
     */
    public static function loadTests()
    {
        foreach(glob(__DIR__.'/../Tests/*.php') as $testClass)
        {
            require_once($testClass);
            $className = "Tests\\".substr(basename($testClass),0,-4);
            array_push(Self::$tests, $className);
        }
    }

    /*
     * Runs one Test Serie
     */
    public static function runTestSerie()
    {
        $startTime = microtime(true);
        $serie = new TestSerie;
        $serie->runTests();
        $results = $serie->result();
        array_push(Self::$series, ['startTime' => $startTime,'results' => $results]);
    }

    /*
     * Calculates averages and returns as a php array
     *
     * @return array Results
     */
    public static function result()
    {
        $results = [];


        for($i = 0; $i < count(Self::$series); $i++)
        {
            $serieResult = Self::$series[$i];
            for($j = 0; $j < count($serieResult['results']); $j++)
            {
                if(isset($results[$serieResult['results'][$j]['name']]))
                    $results[$serieResult['results'][$j]['name']] = (($i * $results[$serieResult['results'][$j]['name']]) + $serieResult['results'][$j]['value']) / ($i + 1);
                else
                    $results[$serieResult['results'][$j]['name']] = $serieResult['results'][$j]['value'];

            }
        }
        return ['average' => $results, 'series' => Self::$series];
    }

}
```