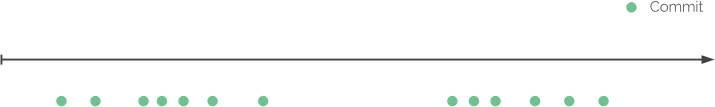
#### value-measure using automated Tool

# Sample Git Commit

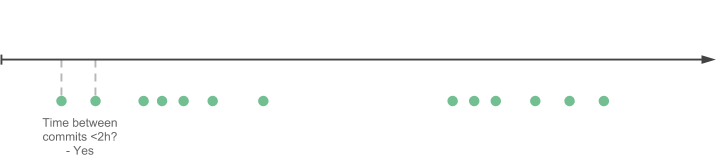


## **Algorithm for estimating hours**

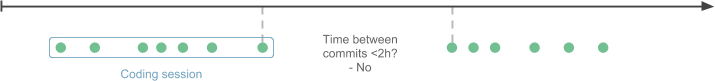
The algorithm for estimating hours is quite simple. For each author in the commit history, do the following:



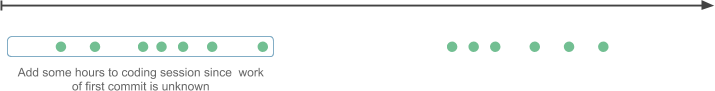
Go through all commits and compare the difference between them in time.



If the difference is smaller or equal then a given threshold, group the commits to a same coding session.



If the difference is bigger than a given threshold, the coding session is finished.



To compensate the first commit whose work is unknown, we add extra hours to the coding session.



Continue until we have determined all coding sessions and sum the hours made by individual authors.

# Anomaly Detection

1. Large no of files added in single commit (first time or bootstrap code committed) (100)

2. Ignore file types (auto generated, protobuff, package.lock.json)

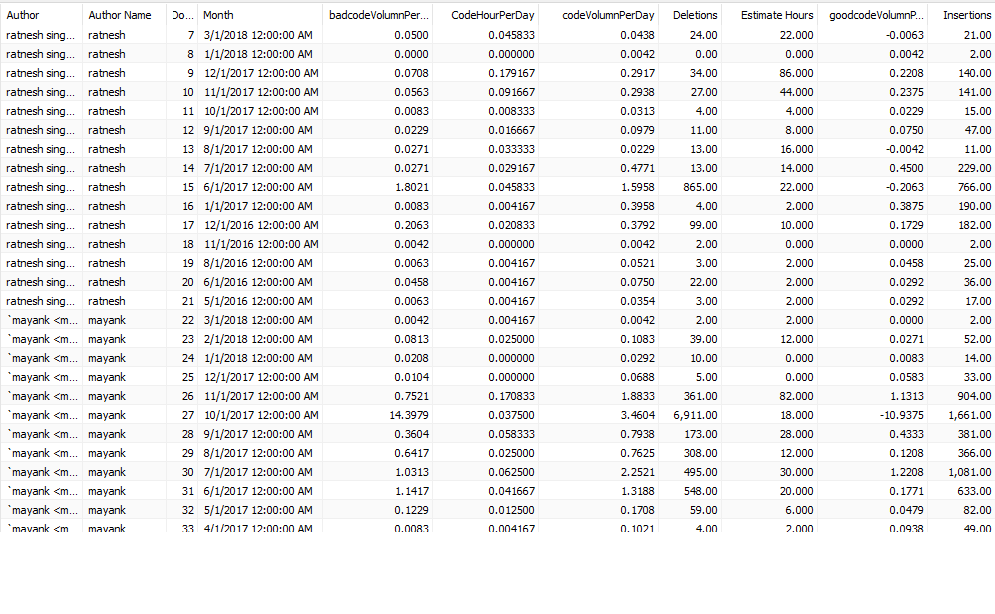
3. Large no of lines added in a file in single commit (1k)

4. Large no of lines added in single commit (all files)(5k)

5. Difference between lines added and lines deleted (<5 %) (Reformatting, copy-paste)

6. Large no of commits with in a duration (1 Hr , 100 commits )

# Format of output CSV



# Code Quality

Code quality is output from sonar-qube .

Essentially Its should run beginning of the term and end of the term.

Support for (JS , TS, JAVA , scala , python ,php , html, jsx ,etc)

|  |
| --- |
| **EntityQualityDetail** |
| entity-name |
| lineno |
| noOfLines |
| IssueSeverity(major,minor,crtical , blocker |
| IssueDebt(5min, 1day) |
| IssueType(code-smell, complexity ,design) |

The above detailed data can be summarize into following entity wise data

|  |
| --- |
| **EntityQualityScore** |
| entity-name |
| ScoreBegining |
| ScoreCurrent |
| ScoreImprovement  Loc  Score-per-line |

For a team Overall quality score could be avg(Score-per-line)

For an individual if we know the contribution percentage of author we can provide the score

# Project Cicero

Project Cicero has been created to figure out following Dimensions

Design Score, leadership score, Ownership score, collaboration score . Details can be found in

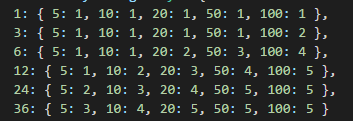
Combine with productivity, utilization, quality final Data mining output will le look like this.

Color blue represents output of project Cicero

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **EntityAge** |  | **AuthorProductivity** |  |
|  | entity-name |  | author-name |  |
|  | Age-months-start |  | month |  |
|  | Age-months-last |  | year |  |
|  | AgeScore | **EntityAuthorEffort** | development-hours |  |
|  |  | entity-name | productivecode-volume |  |
|  | **EntityChanges** | author-name | unproductive-code-volume |  |
|  | entity-name | author-revs | no-of-days | **AuthorCommunication** |
|  | no-of-authors | total-revs | code-hour-per-day | author-name |
| **EnityCoupling** | no-of-changes |  | code-volume-per-day | peer-name |
| enity-name |  | **AuthorEntityOwnerShip** |  | shared |
| coupled-enity-name | **EntityChurn** | entity-name | **AuthorLeadership** | average |
| degree | entity-name | author-name | author-name | strength |
| avg\_revs | no-of-additions | no-of-additions | no-of-peers |  |
|  | no-of-deletions | no-of-deletions | no-of-subordantes |  |
|  | no-of-commits | no-of-commits | shared |  |
|  |  |  | average | **MainDevelopersByContributing** |
|  |  |  | strength | entity-name |
|  |  |  | LeaderShipScore | main-author-name |
| **EntityQualityDetail** | **EntityQualityScore** |  |  | added |
| entity-name | entity-name |  |  | total-added |
| lineno | ScoreBegining |  | **AuthorOwnerShip** | ownership |
| noOfLines | ScoreCurrent |  | author-name | byrevs |
| IssueSeverity | ScoreImprovement |  | no-of-components-owned | byloc |
| IssueDebt |  |  | total\_ownership\_size |  |
| IssueType |  |  |  | **MainDevelopersByRefactoring** |
|  |  |  |  | entity-name |
|  | **EntityFragmentation** |  | **AuthorChurn** | main-author-name |
|  | entity-name |  | author-name | removed |
|  | fractal-value |  | Added | total-removed |
|  | total-revs |  | Deleted | ownership |
|  |  |  | commits |  |

# Explanation of Parameters

**Design Score: - Stability (Entity Age, AuthorEntityOwnerShip)**

****

**If entity is less than 1 month and change percentage is lest than 50% score will be 1**

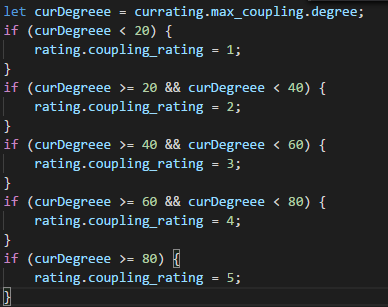
**If entity is less than 3 month and change percentage is more than 50% score will be 2**

**If entity is less than 6 month and change percentage is between 20% and 50% score will be 2**

**Then final score will be weightage avg based on AuthorEntityOwnerShip**

**Churn(EntityChurn , AuthorEntityOwnerShip, EntityChanges)**

**Churn is percentage of lines of code deleted as compare to added, It is calculated using below matrix**

****

**Then final score will be weightage avg based on AuthorEntityOwnerShip**

**Coupling(EnityCoupling , AuthorEntityOwnerShip)**

Logical coupling refers to modules that tend to change together.

Calculates the degree of logical coupling.

Returns a seq sorted in descending order (default) or an optional, custom sorting criterion.

The calulcation is based on the given coupling statistics.

The coupling is calculated as a percentage value based on

the number of shared commits between coupled entities divided

by the average number of total commits for the coupled entities.

# Value Measures

With proper scoring model above data can be crunch into following matrices

Explanation of Higher Level Value Measures for Individual contributors in Last Six months

1. Development Hours
2. Coding Volume
3. Quality Score
4. Design Score
5. Ownership Score
6. Leadership Score

|  |  |  |  |
| --- | --- | --- | --- |
| **Measure** | **What it is** | **How It is measure** | **Expected Values** |
| Development Hours | A Heuristic measurement Indicating no of Hours Spent during durations Per Day. Unit is No Of Hours. Higher the better. | We identify coding sessions (continuous hours of coding) and start time to end time we calculate as Development Hours. For first commit of Session we give extra values. | <1 Bad  1-2 ok  2-3 good  >3 excellent |
| Coding Volumn | A statistical measurement about no of lines of code committed Per Day. Higher the better. | Source is git /svn which provided details like no of lines added/deleted/Modified in each commit. | Bootstrap (<100 bad ,100-200 good, >200 excellent ),Scale(<50 bad , 50-100 good , >100 excellent)Mature(<10 bad , 10-50 good, >50 excellent) |
| Quality Score | An aggregated value (avg) of all 5 point quality (Maintainability, security, reliability, complexity, cognitive complexity) measurements, lesser the better. | Sonar Generated 5 point quality for all code commits. We know the commit author and take industry standard average. | <=1 Excellent , 1-2 good/ok, 2-3 bad , >3 very bad |
| Design Score | An aggregated value (avg) of all 3 point Design (Coupling, Stability, Churn) measurements, lesser the better. | Our Code Generated 3 point Designs for all code commits. We know the commit author and take standard average. | <=1 Excellent , 1-2 good/ok, 2-3 bad , >3 very bad |

### Guidelines to improve Scores

1. Make Sure your have WIGs(Widely Important goals).
2. Decide on initiatives with team .
3. Both 1 and 2 will give you a good back log
4. Before Doing any development and testing try to do action plan and Impact analysis
5. Regular commits
6. Try not to commits in out times(make better plans)
7. Give Tasks to Team members with lower scores
8. Build team member capabilities to improve
9. For Quality check [http://172.19.101.120:9393](http://172.19.101.120:9393/)
10. For Design read the reports in details and figure out components where you can have potential problem,

Create a list of component with high scores and for every change in these files consider refactoring

#### Value Measure 2.0 Productivity Score

## Definition:-

**For individual**

Sum of Labelled scores( Code Volumn and Code hours)(on scale of 1-10)

**For Team**

Median Score of all team members

## Companywide current score and expected improvements

|  |  |  |  |
| --- | --- | --- | --- |
| Project Type | Current score(min) | Target improvement | Stretch improvement |
| Mature | 1.5 – 2.0 | 30% | 40% |
| Scale | 1.8-3.0 | 40% | 50% |
| Bootstrap | 2.5-4.0 | 50% | 60% |

### Clustering

Companywide all developers are cluster into four brackets (L1, L2, L3, L4)

## How the Target is calculated

Target is avg difference between median of all (L1, L2, L3,L4) to median of (L2,L3)

## How the Stretch is calculated

Strech is avg difference between median of all (L1, L2, L3,L4) to median of (L3,L4)

## Value Measure 2.0 Quality Score

Quality score is average score of all team members which come out of 8 factors(5 sonar quality scores and 3 design scores).

Scale is 1-5 (1 is best and 5 is worst).

It is independent of type of project.

Current avg score of Talentica is around 2.3.

|  |  |  |
| --- | --- | --- |
| Project Type | Target improvement | Stretch improvement |
| Mature | 15% | 20% |
| Scale | 20% | 30% |
| Bootstrap | 30% | 50% |

## How the Target is calculated

Target is avg difference between avg of all (L1, L2, L3,L4) to avg of (L2,L3)

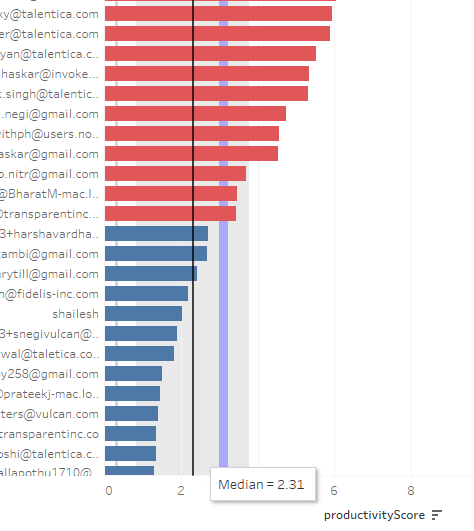
## How the Stretch is calculated

Stretch is avg difference between avg of all (L1, L2, L3,L4) to avg of (L3,L4)

**Point to note:- These percentages are from Talentica wide and they can be changed for a particular team**

## Guidelines For Setting KRA

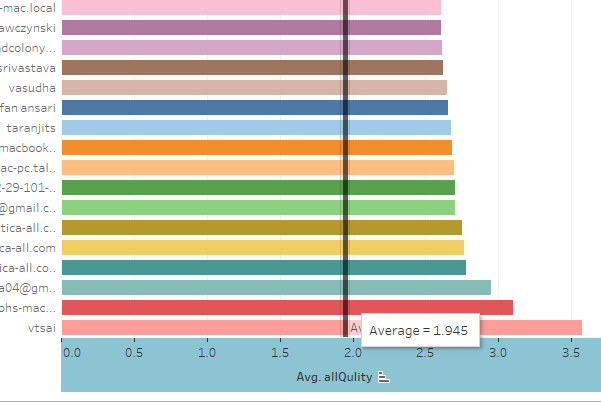
1. Go to Your project in portal
2. Complete the VM for h2
3. Go to New Reports (New Productivity , New Quality)
4. In new Productivity you will see a Median score
5. Using filters remove client side developers , devops , qa, etc.
6. Now you will get a median score like below



1. Now use following matrix to calculate your KRA for productivity

|  |  |  |
| --- | --- | --- |
| Project Type | Target improvement | Stretch improvement |
| Mature | 30% | 40% |
| Scale | 40% | 50% |
| Bootstrap | 50% | 60% |

1. For quality score open new quality score sheet
2. Again use the filtering and check your current average of quality



10 .Now use following matrix to calculate your KRA for Quality

|  |  |  |
| --- | --- | --- |
| Project Type | Target improvement | Stretch improvement |
| Mature | 15% | 20% |
| Scale | 20% | 30% |
| Bootstrap | 30% | 50% |

**Point to note:- These percentages are from Talentica wide and they can be changed for a particular team**

1. You can create two KPIs(one for productivity 15% , one for quality 15%) , or you can create single score as well.
2. We need to have a teamwise discussion (preferred) or all M1 for transition.

I will publish a document explaining all calculations.