



Brunel
University
London

Department of Computer Science
CS3003

Software Engineering

Dr. Giuseppe Destefanis


giuseppe.destefanis@brunel.ac.uk

Sentiment analysis

the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. *(from Google)*

Sentiment analysis and SE

On negative results when using sentiment analysis tools for software engineering research

Robbert Jongeling¹ · Proshanta Sarkar² ·
Subhajit Datta³ · Alexander Serebrenik¹ 

Published online: 10 January 2017

© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract Recent years have seen an increasing attention to social aspects of software engineering, including studies of emotions and sentiments experienced and expressed by the software developers. Most of these studies reuse existing sentiment analysis tools such as SENTISTRENGTH and NLTK. However, these tools have been trained on product reviews and movie reviews and, therefore, their results might not be applicable in the software engineering domain. In this paper we study whether the sentiment analysis tools agree with the sentiment recognized by human evaluators (as reported in an earlier study) as well as with each other. Furthermore, we evaluate the impact of the choice of a sentiment analysis tool on software engineering studies by conducting a simple study of differences in issue resolution times for positive, negative and neutral texts. We repeat the study for seven datasets (issue trackers and STACK OVERFLOW questions) and different sentiment analysis tools and observe that the disagreement between the tools can lead to diverging conclusions. Finally, we perform two replications of previously published studies and observe that the results of those studies cannot be confirmed when a different sentiment analysis tool is used.

Are

Marco C

Abstract-
person, pla
or break a t
developer
information
As a first m
between ser
than 560K
found that
as *JOY* an
fixing time
as *SADNES*
plays a mor
on develop
Index Ter
ware Engir

Team sp
productivit
talent in a
i.e., the wa
their collea
skills will
anything to

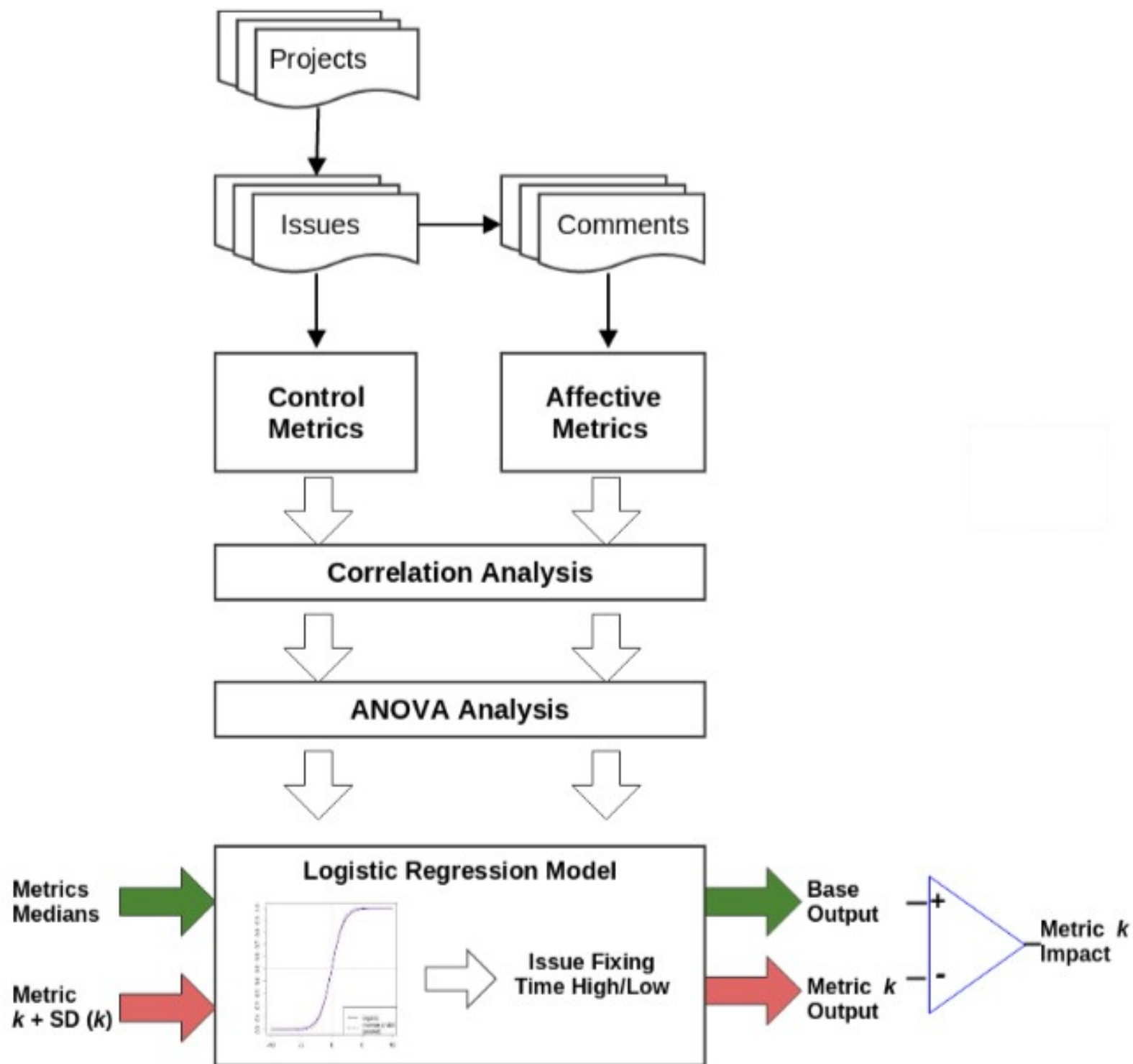


Fig. 2: Experiment Schema

y of

o Tonelli *

of the Linux

ally analyzed
contain emo-
gratitude, joy
presence of
sue resolution
analysis, the
emotions are
measures, and
sue resolution

e than 560K
acking system
ectiveness and
affectiveness
s, then build
e metrics can
address the

ess correlated

de correlation

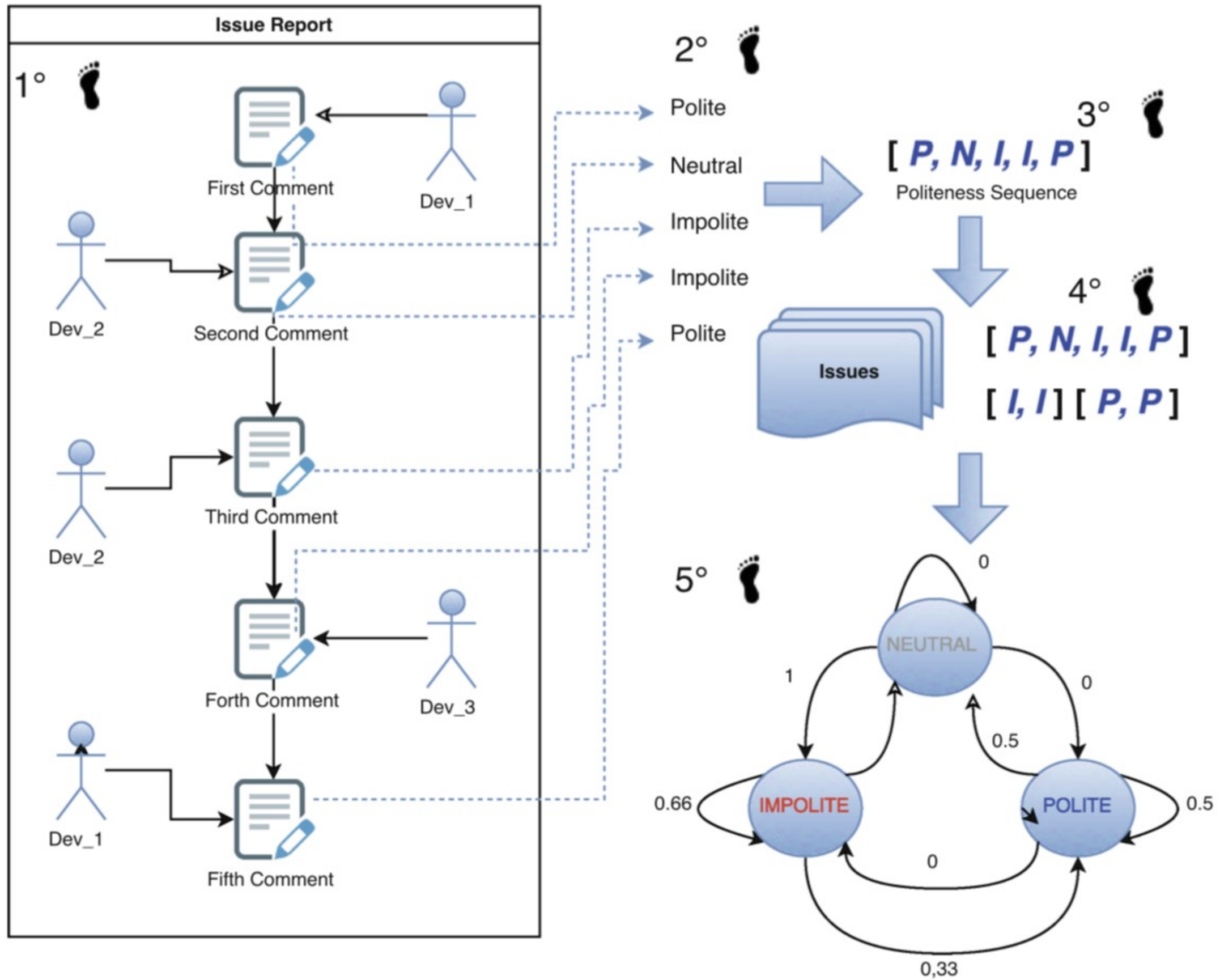
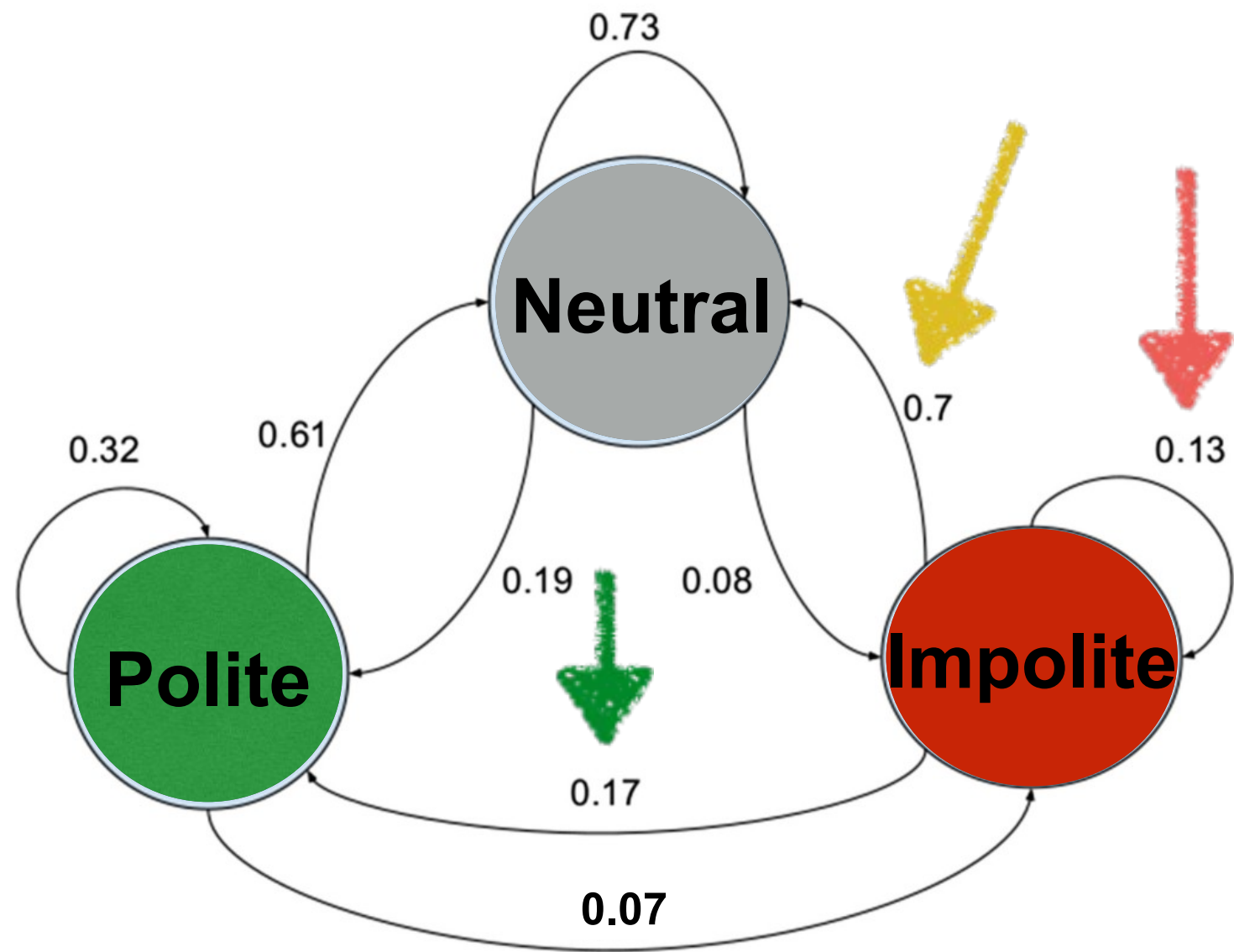


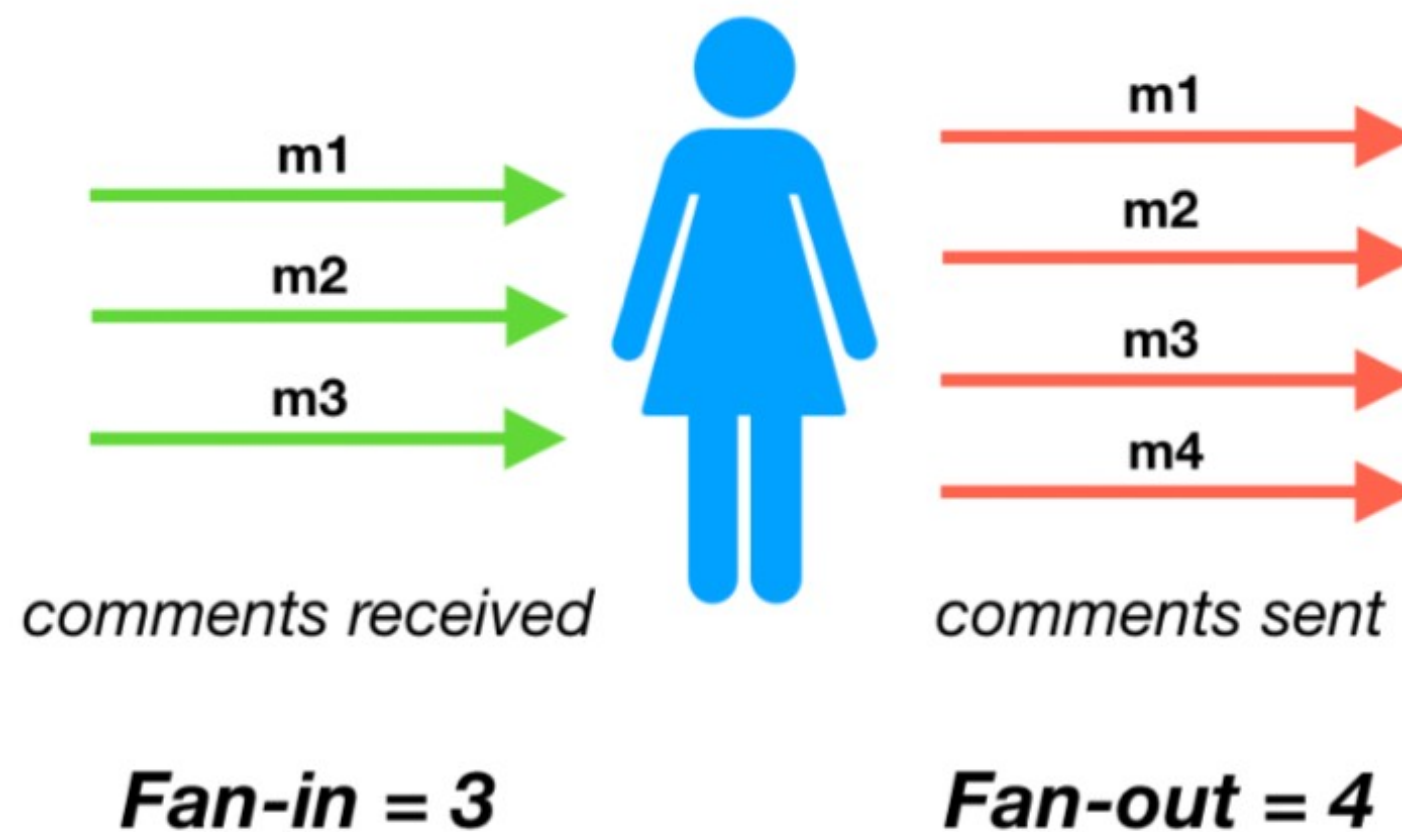
Fig. 1. Politeness' Markov's chain schema



Mining Communication Patterns in Software Development: A GitHub Analysis

Marco Ortu

Tracy Hall



RESEARCH

How diverse i gender and n teams

Marco Ortu¹, Giuseppe Destefa
and Michele Marchesi¹

*Correspondence:
 g.destefanis@herts.ac.uk
²University of Hertfordshire,
 Hatfield, UK
 Full list of author information is
 available at the end of the article

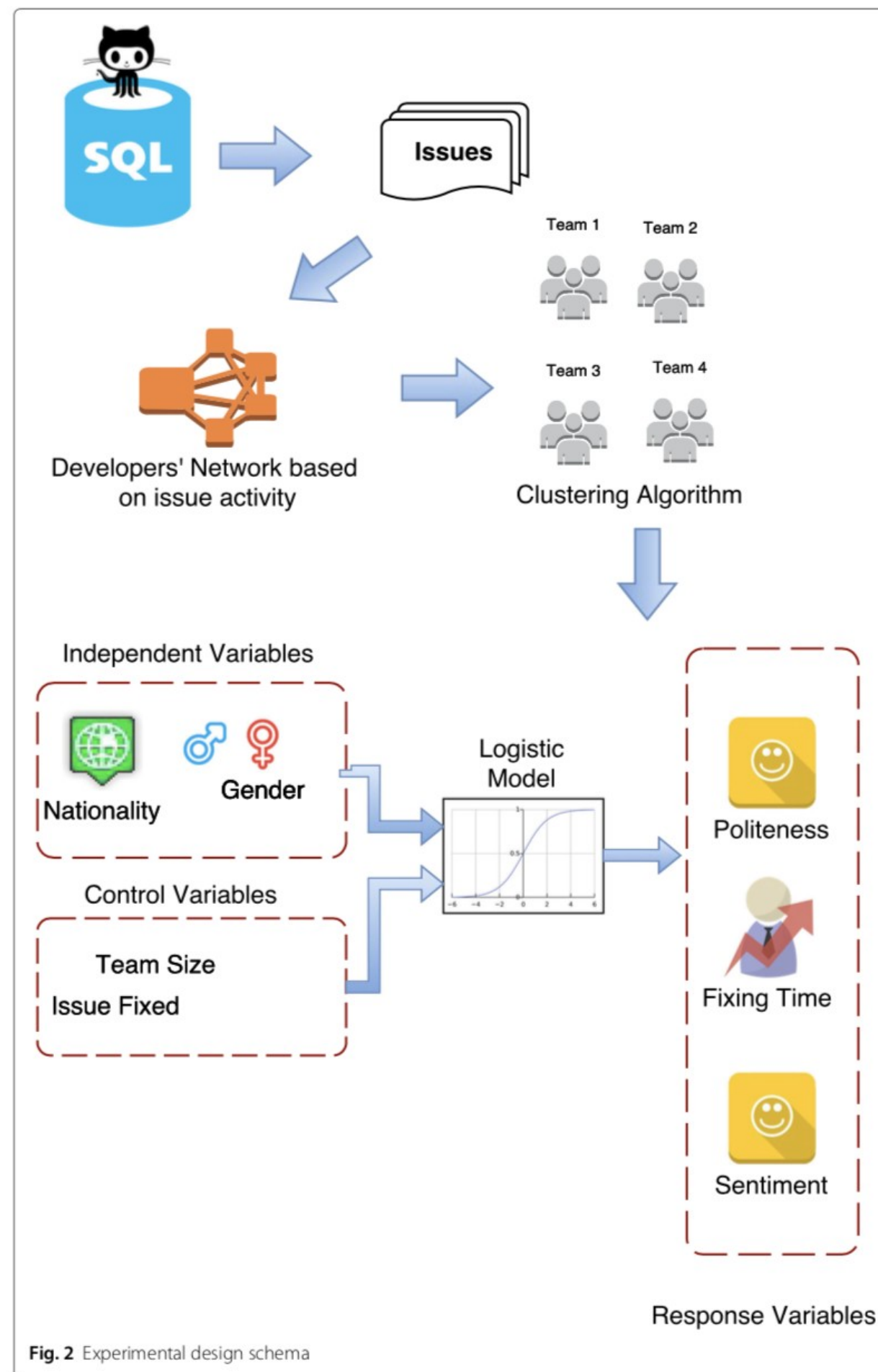


Fig. 2 Experimental design schema



A **successful** software project is the result of a **complex process** involving **people**

respectful and polite environment



Happy software developers solve problems better: psychological measurements in empirical software engineering

Daniel Graziotin, Xiaofeng Wang and Pekka Abrahamsson

Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy

ABSTRACT

For more than thirty years, it has been claimed that a way to improve software developers' productivity and software quality is to focus on people and to provide incentives to make developers satisfied and happy. This claim has rarely been verified in software engineering research, which faces an additional challenge in comparison to more traditional engineering fields: software development is an intellectual activity and is dominated by often-neglected human factors (called human aspects in software engineering research). Among the many skills required for software development, developers must possess high analytical problem-solving skills and creativity for the software construction process. According to psychology research, affective states—emotions and moods—deeply influence the cognitive processing abilities and performance of workers, including creativity and analytical problem solving. Nonetheless, little research has investigated the correlation between the affective states, creativity, and analytical problem-solving performance of program-

The JIRA board

Apache Mesos

Backlog Kanban board Reports Board

QUICK FILTERS: Assigned Unassigned Assigned To Me Missing Shepherd I'm Shepherding Newbie Recently Updated (1 day) Recently Updated (7 days) Epics

670 Backlog (Accepted)

93 In Progress

197 Reviewable

174 Done <4wks

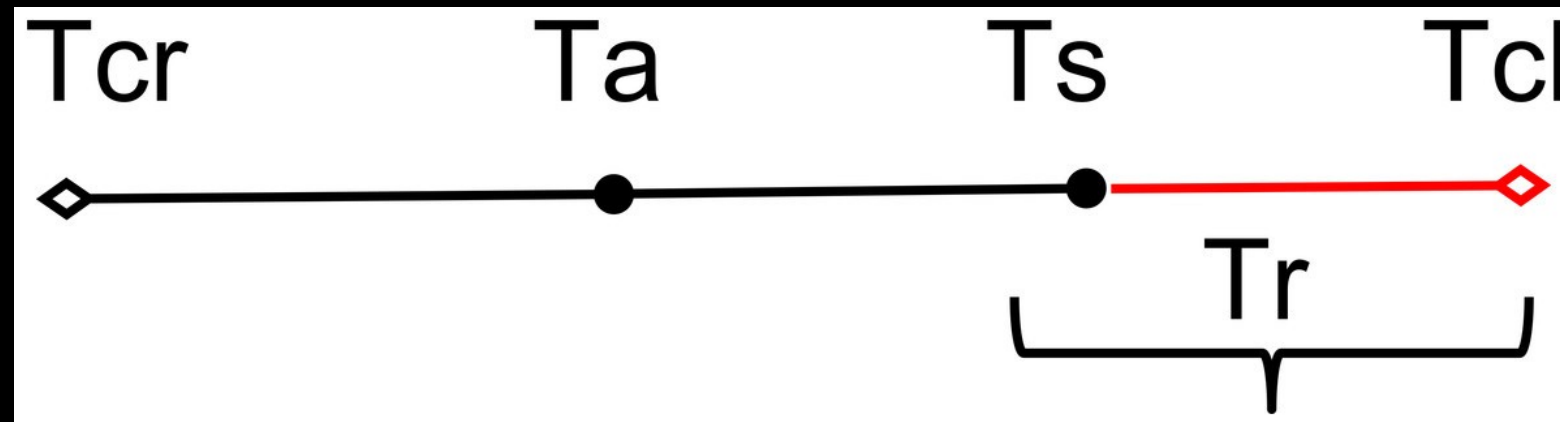
- [MESOS-4233](#)
↑ Logging is too verbose for sysadmins / syslog **K**
- [MESOS-4766](#)
↑ Improve allocator performance. **M**
- [MESOS-5401](#)
↑ Add ability to inject a Volume of Nvidia GPU-related libraries into a docker container. **K**
GPU
- [MESOS-5343](#)
↓ Behavior of custom HTTP authenticators with disabled HTTP authentication is inconsistent between master and agent
- [MESOS-5369](#)
↑ Coarse-grained authorization of endpoints is supported only for short url paths.
- [MESOS-5028](#)
↑ Copy provisioner cannot replace directory with symlink **G**
Unified Container
- [MESOS-4053](#)
↑ MemoryPressureMesosTest tests fail on CentOS 6.6 **G**
- [MESOS-4823](#)
↑ Implement port forwarding in 'network/cni' isolator **A**
CNI Support
- [MESOS-5027](#)
↑ Enable authenticated login in the webui **H**

- [MESOS-4690](#)
↑ Reorganize 3rdparty directory **K**
- [MESOS-4626](#)
↑ Support Nvidia GPUs with filesystem isolation enabled. **K**
GPU
- [MESOS-5392](#)
↑ Design doc for adding resource limits support for Mesos containerizer **J**
Container Security
- [MESOS-5570](#)
↑ Improve CHANGELOG and upgrades.md **J**
- [MESOS-5588](#)
↑ Improve error handling when parsing acls. **J**
- [MESOS-4544](#)
↑ Propose design doc for agent partitioning behavior **N**
- [MESOS-3235](#)
↑ FetcherCacheHttpTest.HttpCachedSerialized and FetcherCacheHttpTest.HttpCachedConcurrent are flaky **H**
- [MESOS-4882](#)
↑ Add support for command and arguments to mesos-execute. **G**
- [MESOS-4941](#)
↑ Support update existing quota. **Z**
Quota

- [MESOS-5221](#)
↓ Add Documentation for Nvidia GPU support **K**
GPU
- [MESOS-2043](#)
↑ Framework auth fail with timeout error and never get authenticated **B**
- [MESOS-4099](#)
↑ parallel make tests does not build all test targets **K**
Reorganize 3rdparty
- [MESOS-5445](#)
↑ Allow libprocess/stout to build without first doing 'make' in 3rdparty. **K**
Reorganize 3rdparty
- [MESOS-4749](#)
↓ Move HTB out of containers **C**
- [MESOS-5582](#)
↑ Create a 'cgroups/devices' isolator. **K**
- [MESOS-5228](#)
↑ Add tests for Capability API. **J**
Container Security
- [MESOS-5232](#)
↑ Add capability information to ContainerInfo protobuf message. **J**
Container Security
- [MESOS-5275](#)
↑ Add capabilities support for unified containerizer. **J**
Container Security

- [MESOS-5550](#)
↑ Remove Nvidia GPU Isolator's link-time dependence on 'libnvidia-ml' **K**
GPU
- [MESOS-5558](#)
↑ Update 'Containerizer::resources()' to use the 'NvidiaGpuAllocator' **K**
GPU
- [MESOS-5559](#)
↑ Integrate the 'NvidiaGpuAllocator' into the 'NvidiaGpuIsolator' **K**
GPU
- [MESOS-5562](#)
↑ Add class to share Nvidia-specific components between containerizers **K**
GPU
- [MESOS-5563](#)
↑ Rearrange Nvidia GPU files to cleanup semantics for header inclusion. **K**
GPU
- [MESOS-5699](#)
↑ Create new documentation for Mesos networking. **A**
Containerization Documentation
- [MESOS-5576](#)
↑ Masters may drop the first message they send between masters after a network partition **J**
- [MESOS-4938](#)
↑ Support docker registry authentication **G**
Unified Container
- [MESOS-5216](#)
↑ Document docker volume driver isolator. **G**

Issue fixing time



- T_{cr} - time when an issue is created
- T_a - time when an issue is assigned to a developer
- T_s - time when a developer subscribes to an issue that has been assigned to her
- T_{cl} - time when an issue is closed

Magnet or Sticky?

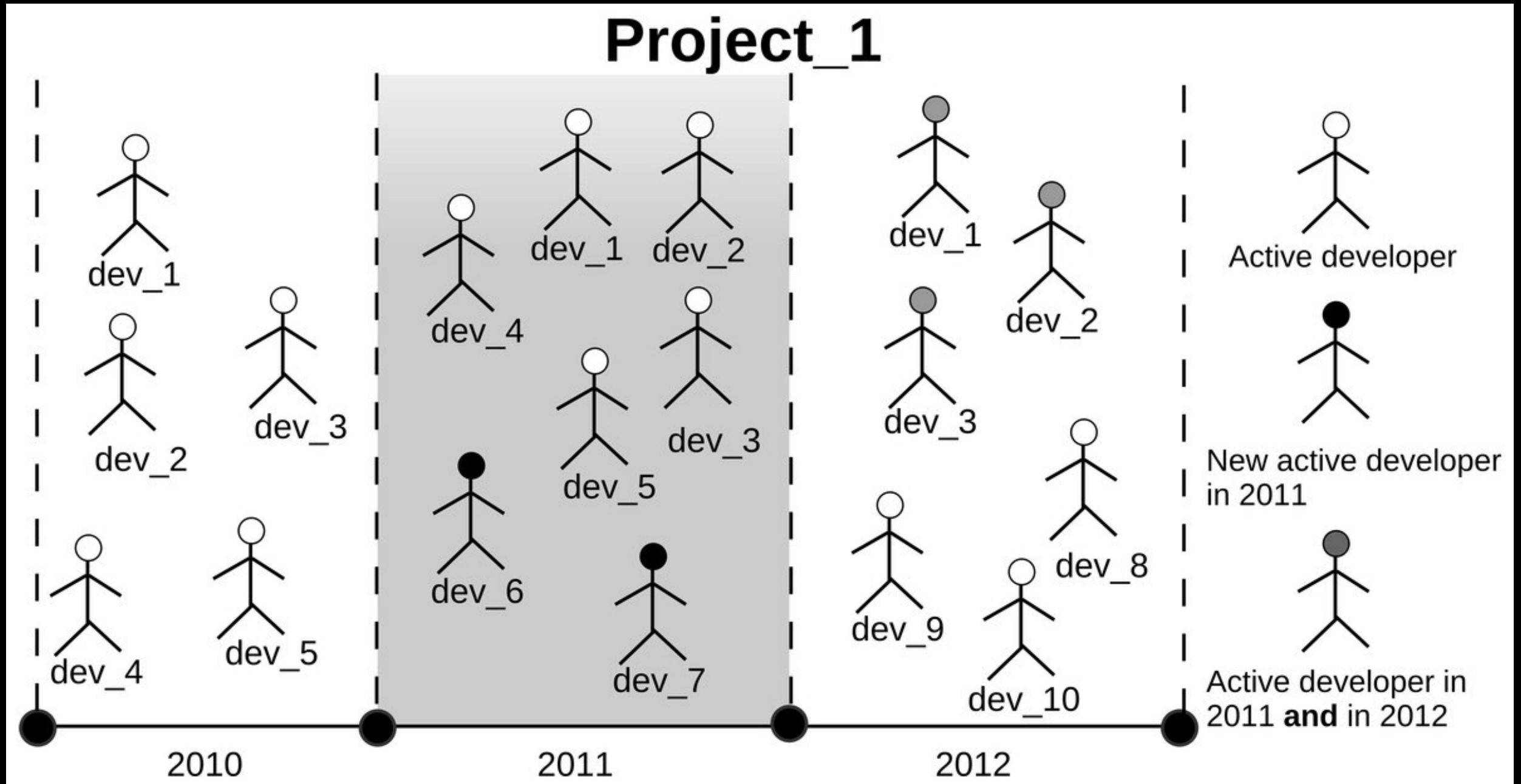


Magnetism - fraction of new active developers during
selected time interval

- **Stickiness** - fraction of active developers that
were also active during next time interval



Attractiveness



Magnet or Sticky?



n - in our example **2/10**, dev_6 and dev_7
in 2011 but not in 2010

- **Stickiness** - in our example **3/7**, dev_1, dev_2, dev_3 were active in 2011 and in 2012



Dataset

- From 2002 to 2013
- Significant amount of monthly activity
- Projects with more than 15K comments

22 projects

Project	# of comments	# of developers
HBase	91,016	951
Hadoop Common	61,958	1,243
Derby	52,668	675
Lucene Core	50,152	1,107
Hadoop HDFS	42,208	757
Cassandra	41,966	1,177
Solr	41,695	1,590
Hive	39,002	850
Hadoop Map/Reduce	34,793	875
Harmony	28,619	316
OFBiz	25,694	578
Infrastructure	25,439	1,362
Camel	24,109	908
ZooKeeper	16,672	495
GeoServer	17,424	705
Geronimo	18,017	499
Groovy	18,186	1,305
Hibernate ORM	23,575	4,037
JBoss	23,035	453
JRuby	22,233	1,523
Pig	21,662	549
Wicket	17,449	1,243
Tot	737,572	18,144

Detecting politeness

Comment	POLITE
Hey <dev_name_a>, Would you be interested in contributing a fix and a test case for this as well? Thanks, <dev_name_b>	YES
<dev_name>, can you open a new JIRA for those suggestions? I'll be happy to review.	YES
<dev_name>, the latest patch isn't applying cleanly to trunk - could you resubmit it please? Thanks.	YES
<dev_name>, Since you can reproduce, do you still want the logs? I think I still have them if needed.	YES

Why are you cloning tickets?
Don't do that.

NO

shouldnt it check for existence of
tarball even before it tries to allocate
and error out ???

NO

<dev_name_a>, why no unit test?
<dev_name_b>, why didn't you wait
for +1 from Hudson???

NO

> this isn't the forum to clarify
Why not? The question is whether
this is redundant with Cascading,
so comparisons are certainly relevant, no?

NO

Issue politeness

- **POLITE issues:** only polite comments
- **IMPOLITE issues:** only impolite comments
- **MIXED issues:** both polite and impolite comments

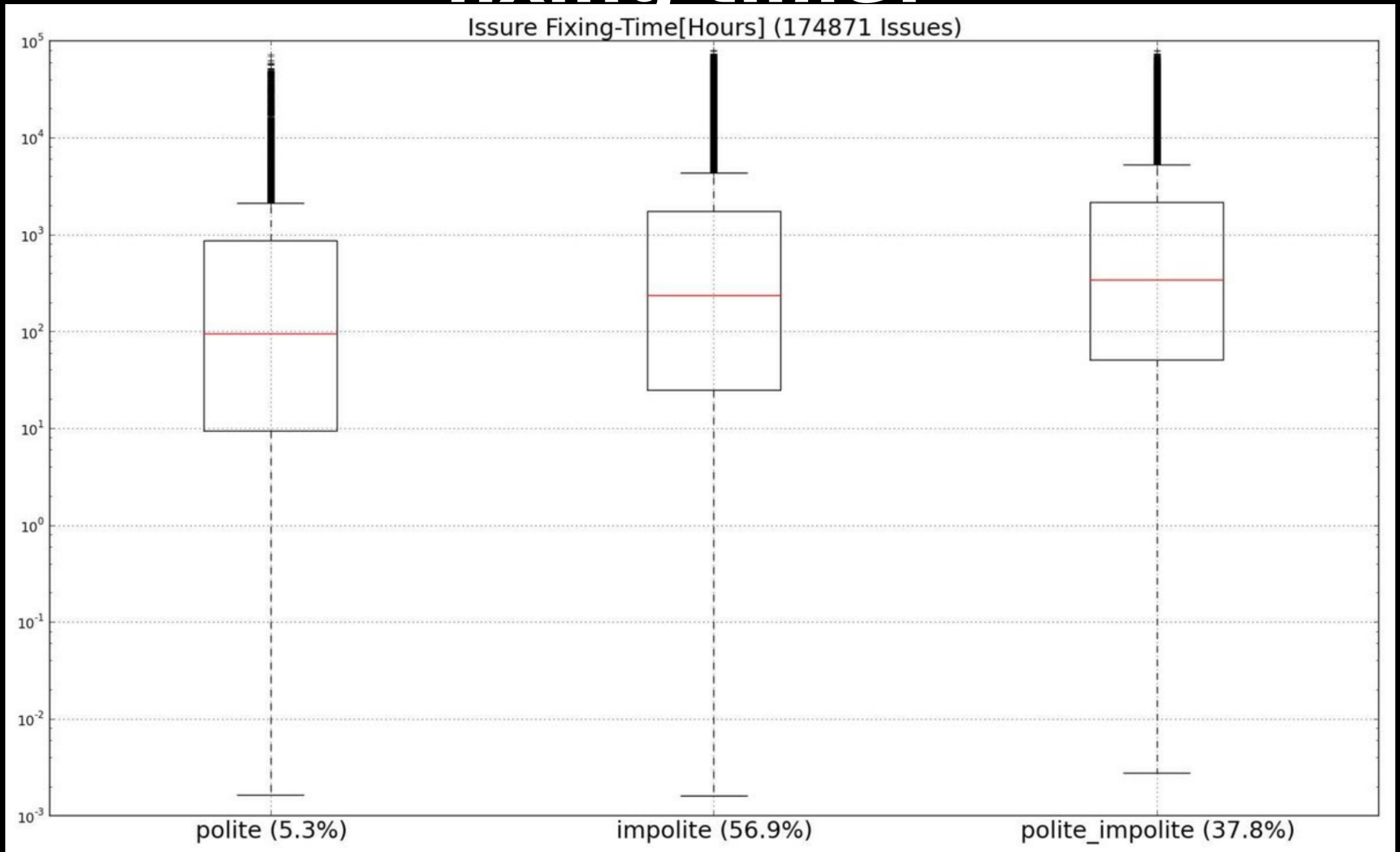
Research questions

1. Does a relationship exist between **politeness** and **issues fixing time**?
2. Does **politeness** among developers affect the **attractiveness** of a project?
3. How does **politeness** vary with respect to **JIRA maintenance types** and **issue priorities**?

RQ1: Does a relationship exist between politeness and issues fixing time?

- ***Kruskal-Wallis test***
- ***H0: the three distributions of issue fixing time are equal for the three typologies of considered issues (polite, impolite, mixed)***
- ***p-value $p < 2^{-16}$ —> the three distributions are statistically different***

RQ1: Does a relationship exist between politeness and issues fixing time?



Issue fixing time for polite
issues is **shorter** than issue
fixing time for impolite and
mixed issues

RQ2: Does politeness among developers affect the attractiveness of a project?

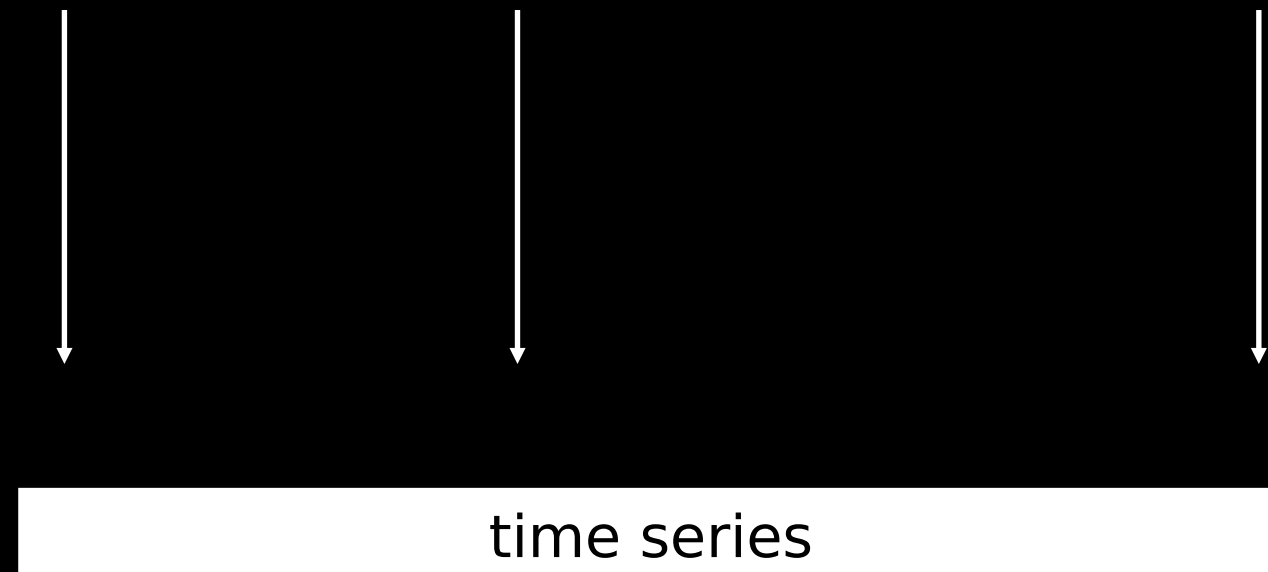


RQ2: Does politeness among developers affect the attractiveness of a project?

- Observation time of one month.
- We measured **stickiness, magnetism, % of polite comments.**

RQ2: Does politeness among developers affect the attractiveness of a project?

- Observation time of one month.
- We measured **stickiness**, **magnetism**, % of **polite comments**.



RQ2: Does politeness among developers affect the attractiveness of a project?



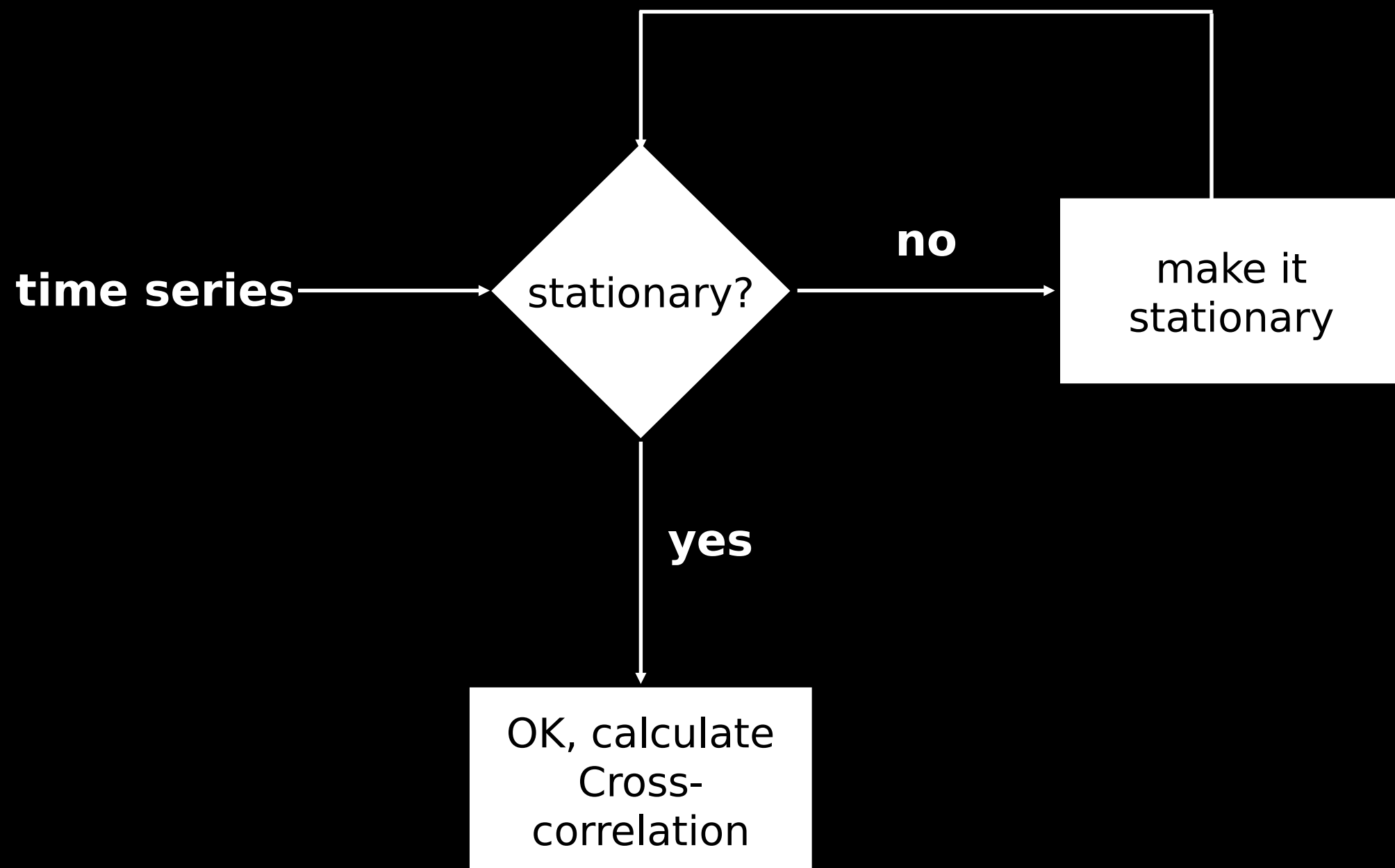
RQ2: Does politeness among developers affect the attractiveness of a project?



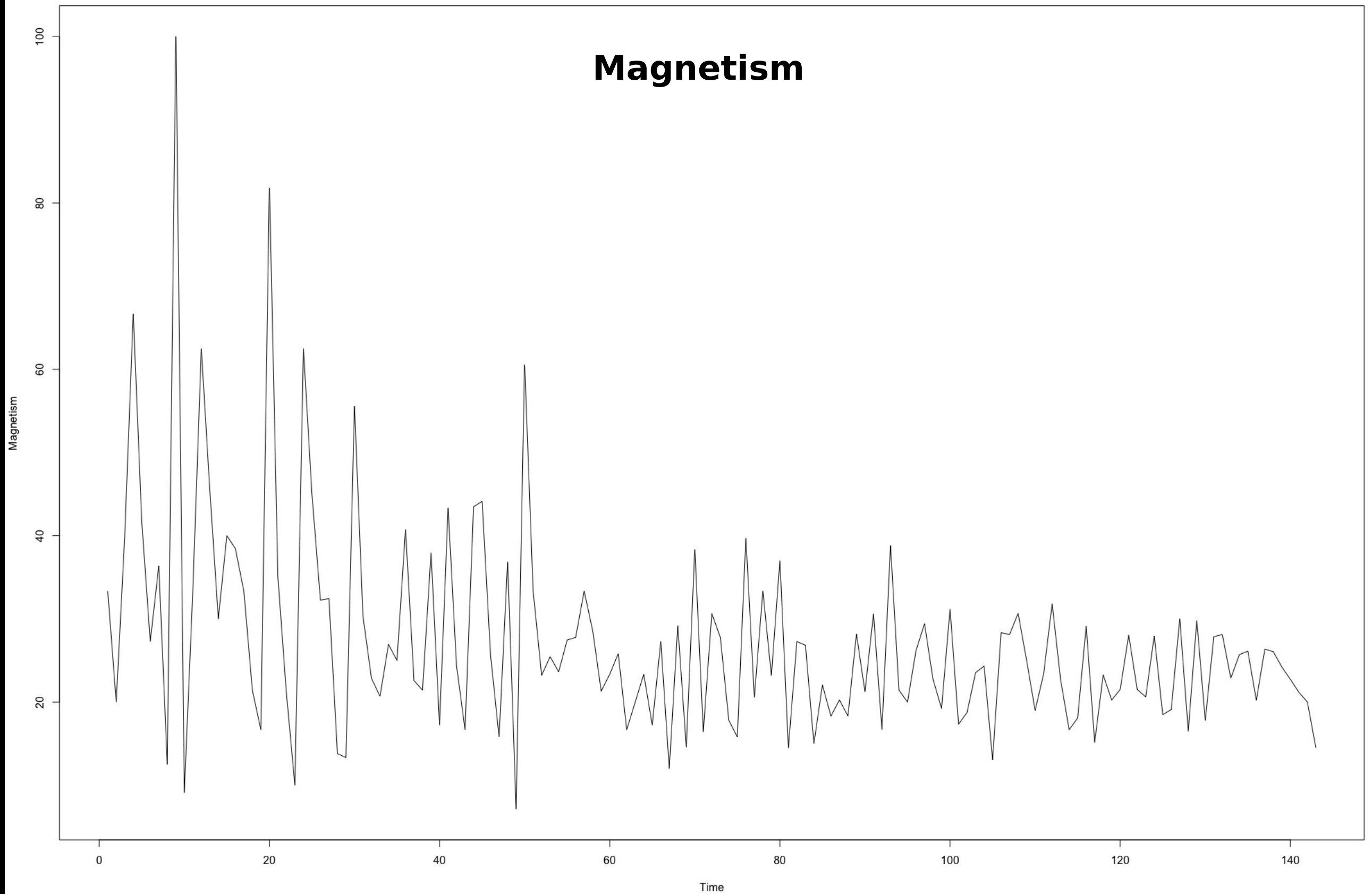
Tests for stationarity

- **Ljung-Box**: confirms independence of increments, **rejection** of the null hypothesis H_0 **indicates stationarity**.
- **Augmented Dickey-Fuller (ADF)**: **rejection** of the null hypothesis H_0 **indicates stationarity**.
- **Kwiatkowski-Phillips-Schmidt-Shin test (KPSS)**: **rejection** of the null-hypothesis H_0 indicates **no-stationarity**.

RQ2: Does politeness among developers affect the attractiveness of a project?

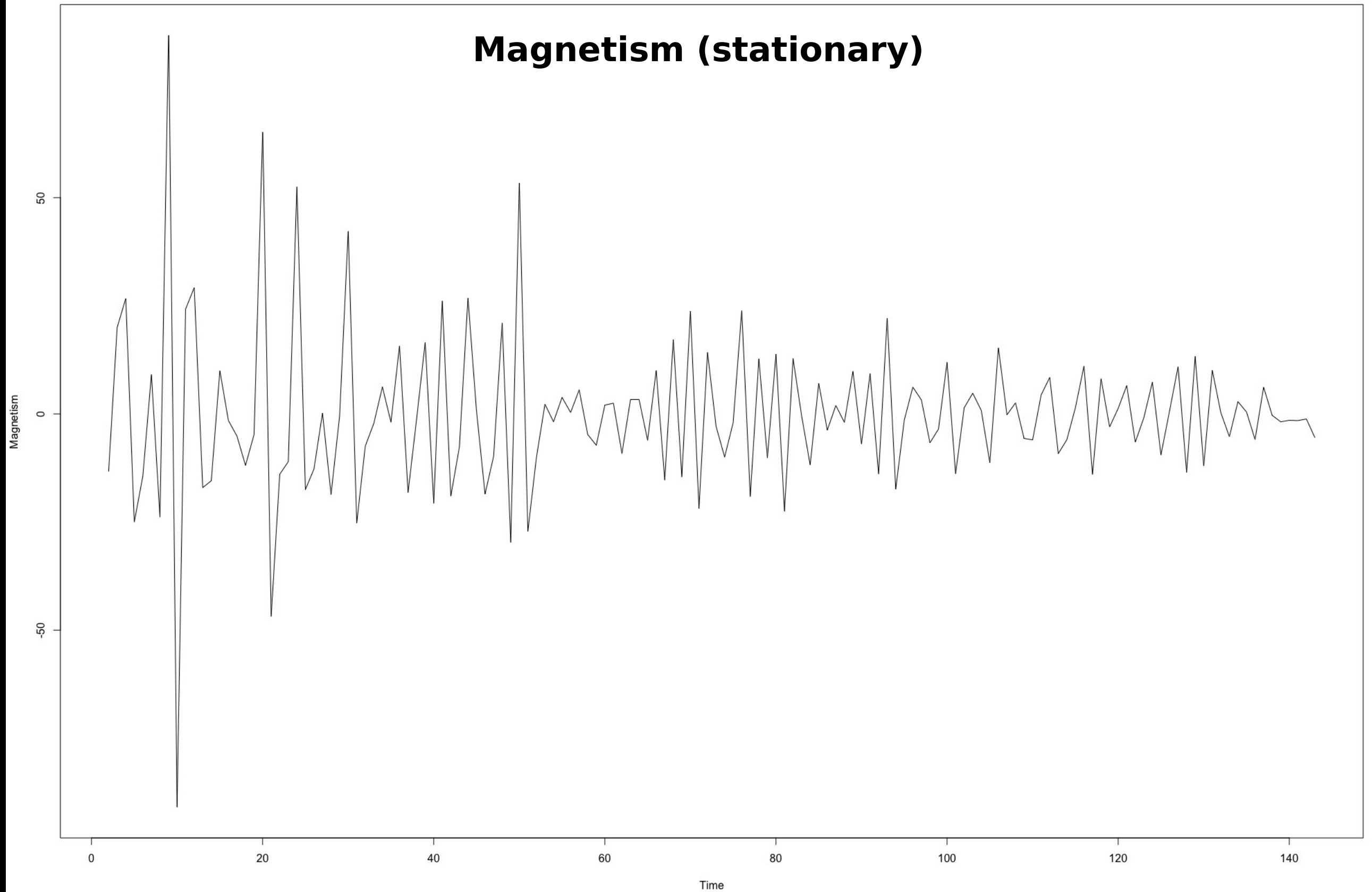


Magnetism



Lucene-Core (Stationary)

Magnetism (stationary)



For the majority of the cases there is a
positive correlation between
politeness and Magnetism (14
projects out 22) and **politeness and**
Stickiness (13 projects out 22)

RQ3: How does politeness vary with respect to JIRA maintenance types and issue priorities?

- Goal: understanding which **typology of issue** attracts more impolite comments.

RQ3: How does politeness vary with respect to JIRA maintenance types and issue priorities?

- Goal: understanding which **typology of issue** attracts more impolite comments.
- Great for helping both managers and developers in better understanding the development process.
- Take action to manage the distribution of issues within development teams.

Issue-maintenance

Type	IMPOLITE	POLITE
Bug	201,359	163,489
Sub-task	24,333	22,909
Task	19,379	14,442
Improvement	90,477	90,564
New Feature	33,640	39,538
Wish	2573	2681
Test	3788	3270
New JIRA Project	172	210
Brainstorming	98	172
Umbrella	87	144

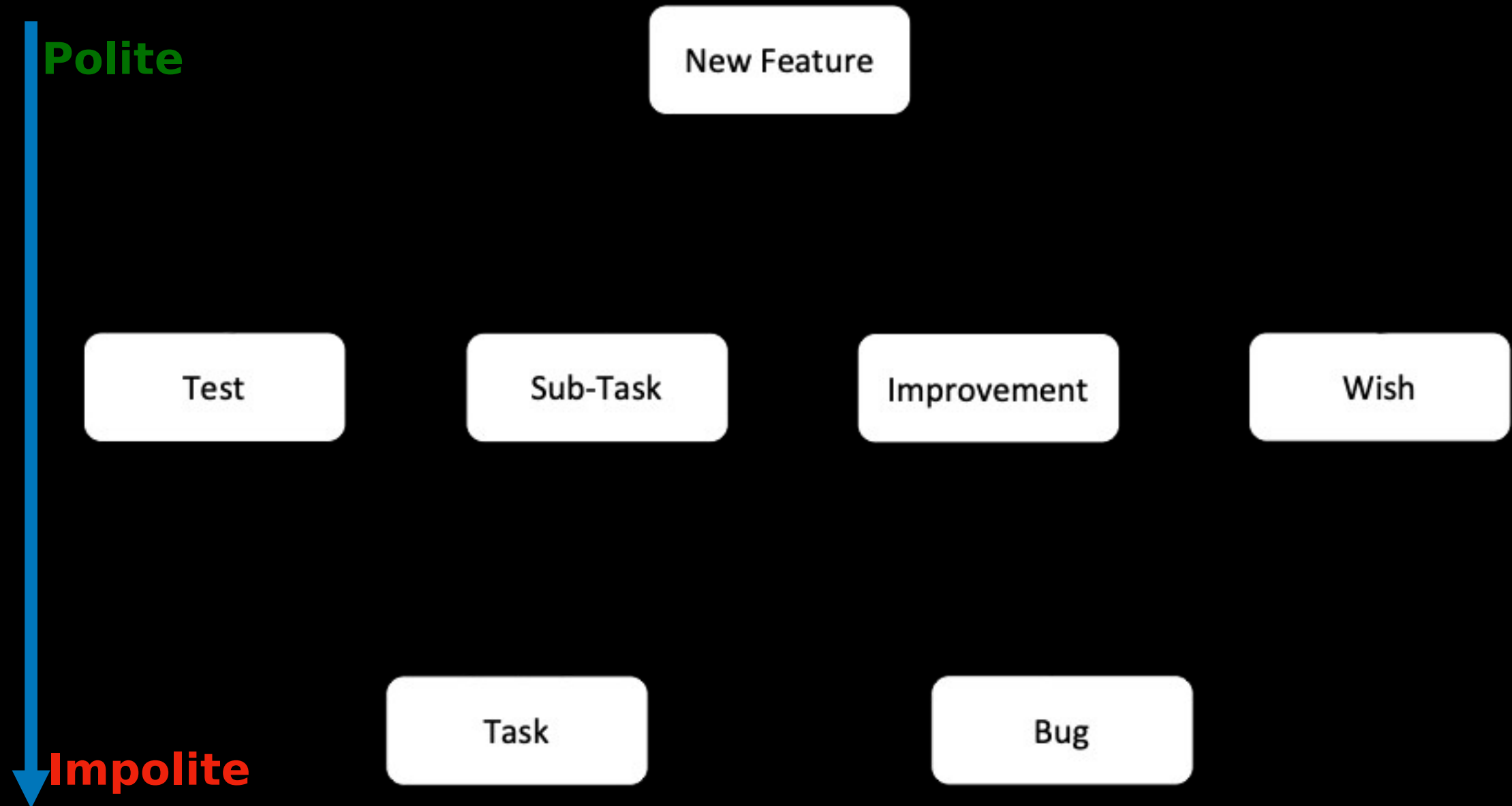
Issue-priority

Priotity	IMPOLITE	POLITE
Blocker	20,657	19,049
Critical	21,517	19,410
Major	241,012	219,841
Minor	82,892	71,905
Optional	105	52
Trivial	12,009	8,479

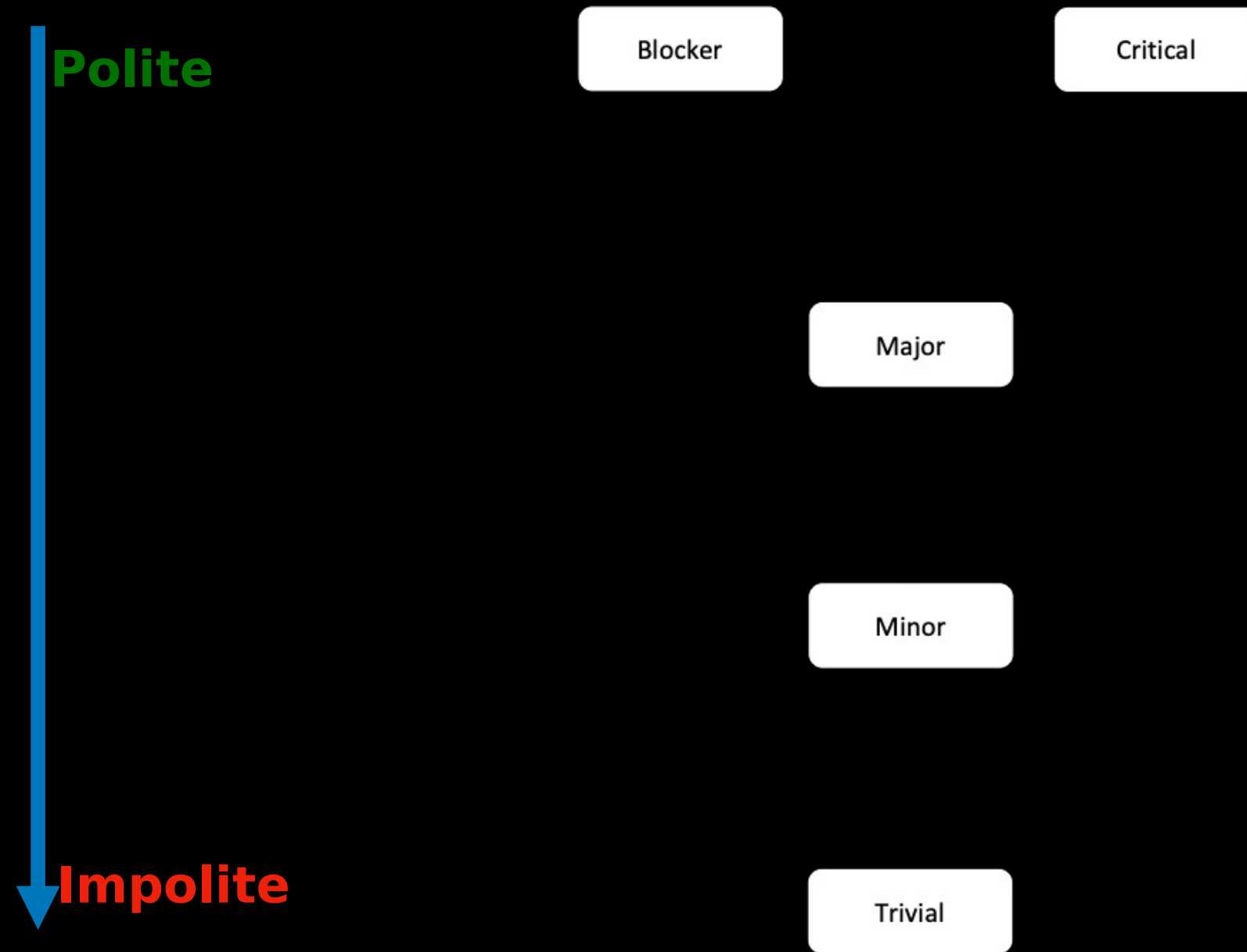
Methodology

- **Tukey's test:** single-step multiple comparison procedure and statistical test.
- **T~ -graph:** used to visualise the results obtained from the multiple contrast test procedure.

Issue-maintenance



Issue-priority





**KEEP
CALM
AND
BE
POLITE**

Appendix

- **Kruskal-Wallis test:** non parametric and unpaired.
- The test can be used with no restrictions or hypotheses on the statistical distribution of the sample populations.
- The test is suitable for comparing differences among the medians of two or more populations when their distributions are not gaussian.
- We grouped all the issues (by category) of all the projects contained in our corpus and then we tested the null hypothesis H_0 : the three distributions of issue fixing time are equal for the three typologies of considered issues (polite, impolite, mixed).