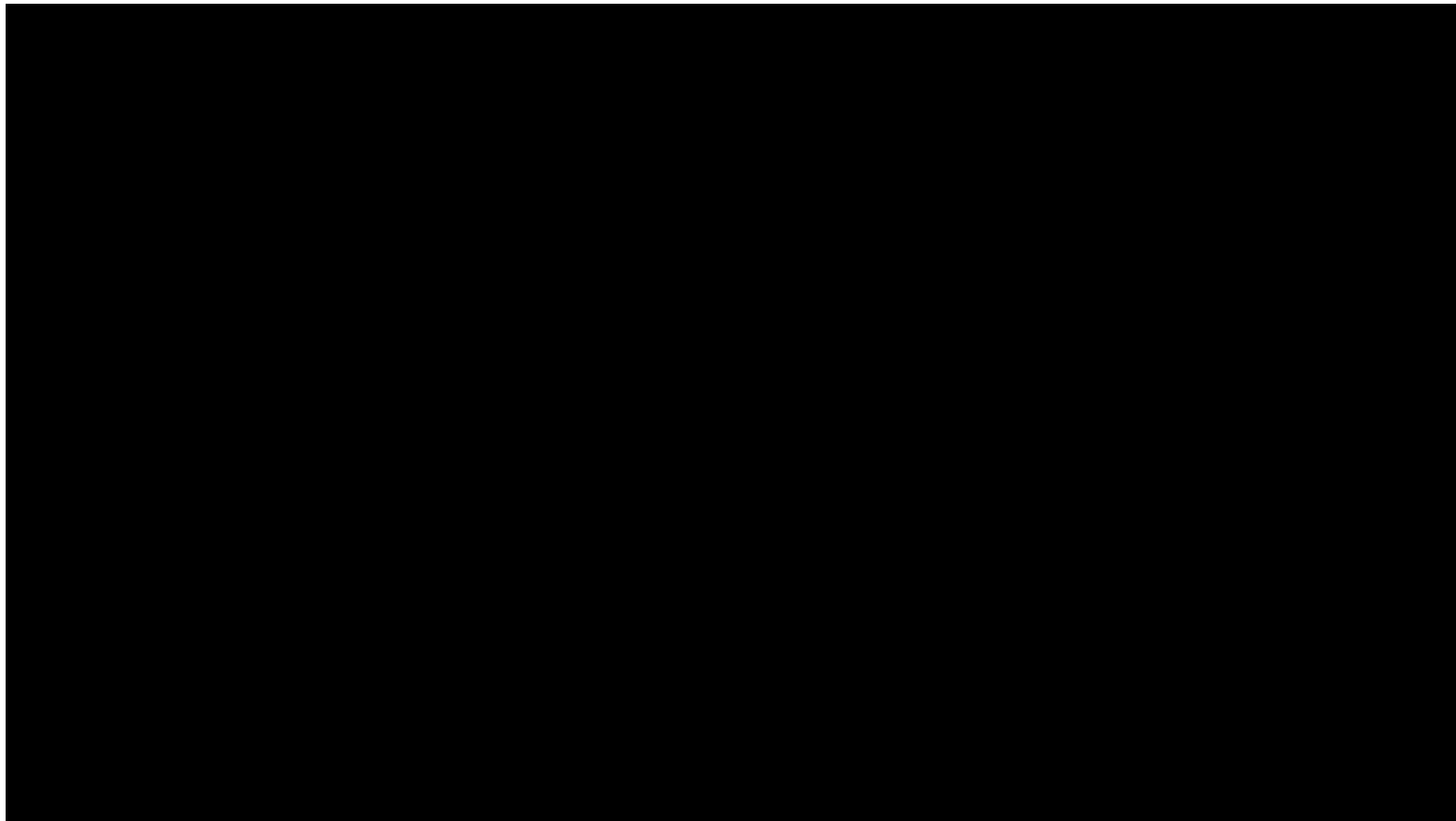


# Arquivo.pt:

Recovering and improving resilience

Daniel Gomes

# Arquivo.pt – the Portuguese Web Archive



<https://youtu.be/dqG0VILi3gs>

# Brief history of Arquivo.pt

- 2007: Project launch to build a service
- 2010: Search prototype publicly available
- 9/2013: Service collapse due to hardware malfunction
  - Data loss of 17% (17 TB)
  - Crawling interruptions
  - Suspension of search service
- 2014, 2015: Recovery and improving resilience

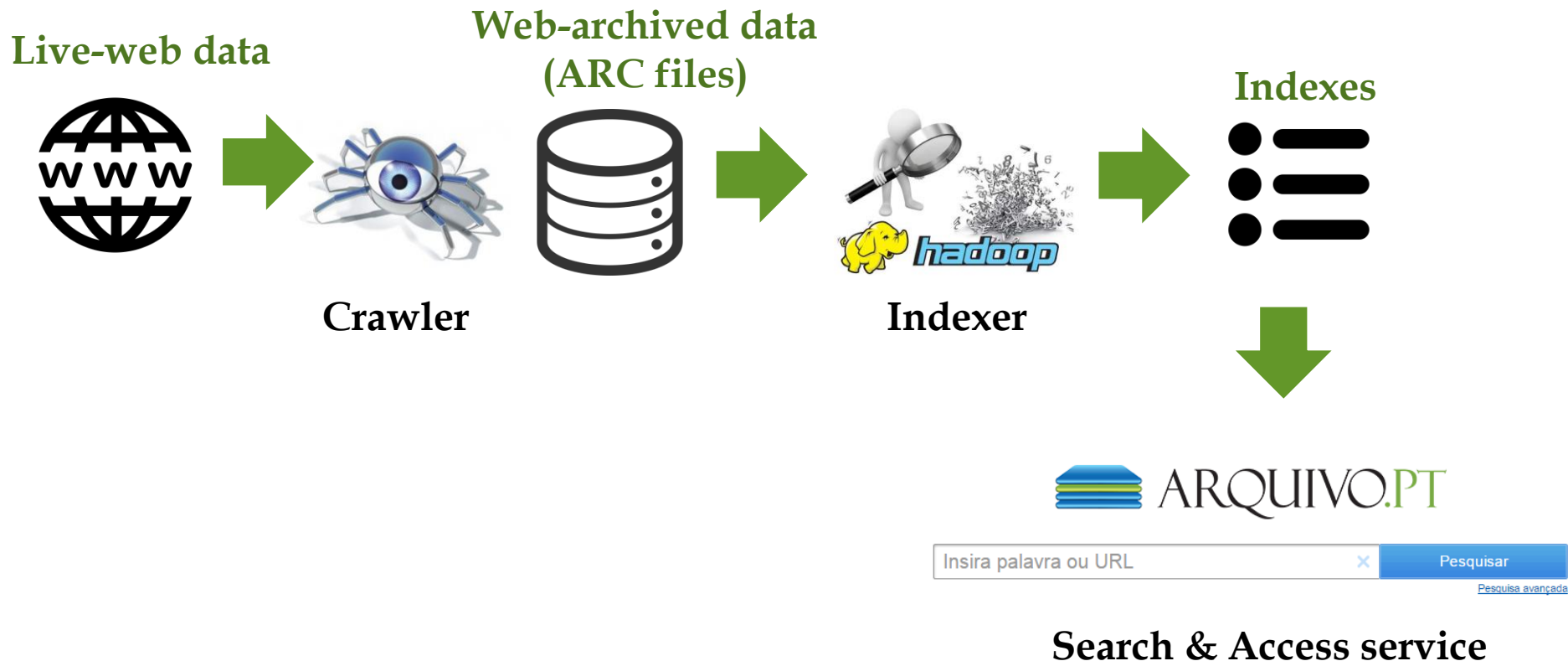
Only disadvantages? **No.**



We are here to share our experience.

# Arquivo.pt system overview

# Our web archiving workflow is mainly automatic



# We are a medium-size web archive

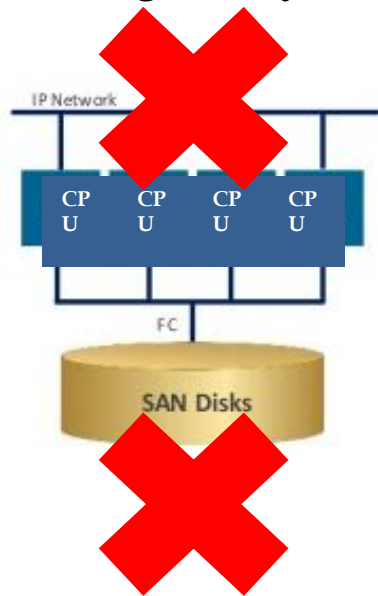
- Hardware and Software
  - 85 servers
- Archived data
  - 2 700 million files
  - 205 TB (ARC files, indexes, replication)
- Estimated data growth
  - 72 TB/year

# #1: Hardware and software architecture shift to *Share-Nothing*

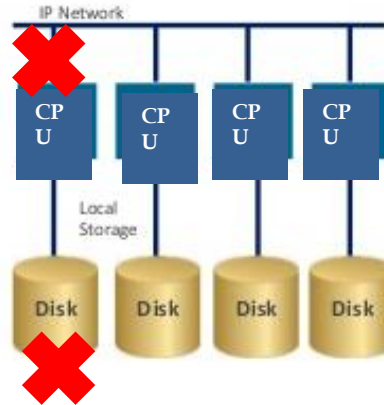


# Hardware architecture shifted to *Share-Nothing*

Centralized: blade server enclosures + storage arrays

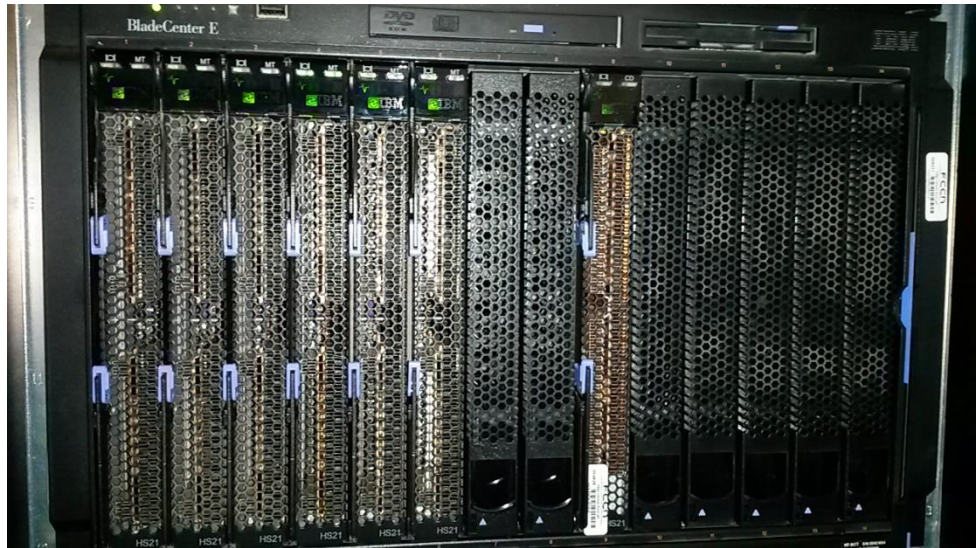


Distributed (*share-nothing*): independent rack servers



**Design-to-fail:** the failure of a single equipment cannot jeopardize the service

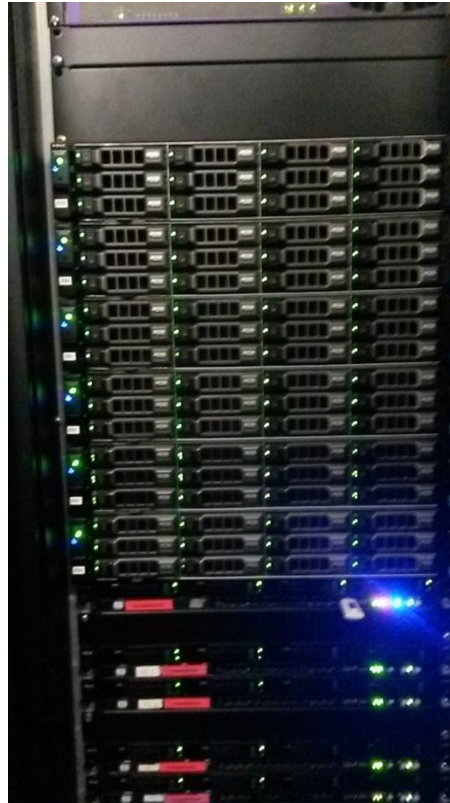
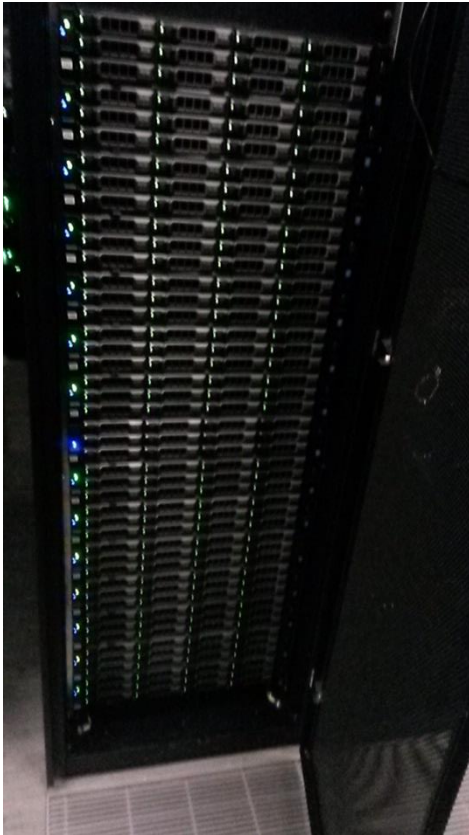
# Inefficient physical space management at the data center with *blade systems*



Space that was never used

Space still occupied  
after servers  
disabled

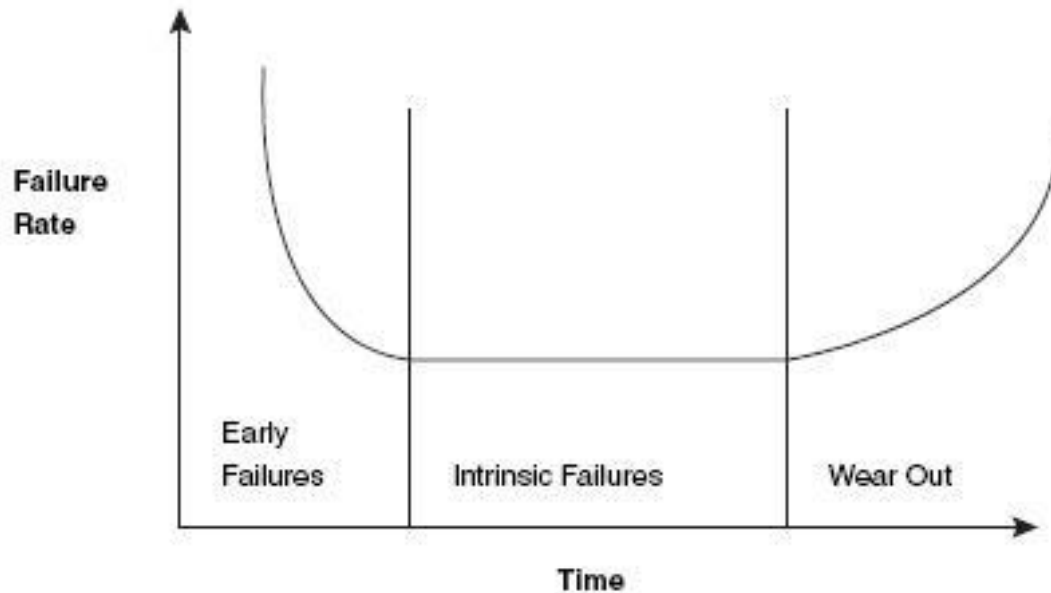
Independent rack servers: physical space is released as servers break



Only the required physical space to host the operational servers is occupied

# Perform stress tests immediately after buying

Figure 6-1. Bathtub Curve



- Bathtub curve
  - We want to identify Early Failures during the warranty period
- Open source tools: *bonnie*, *stress*, *memtest*

# Segregate development from production networks

Private network



Development environment



Gateway between networks



Public network

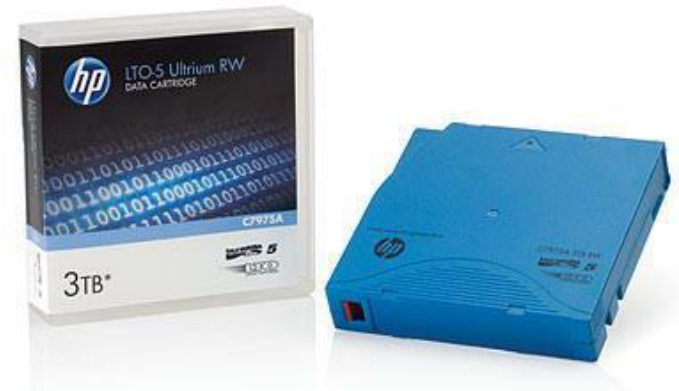


Pre-production and production environments

# #2: Reinforce replication policies

# Tape backups

- Bundle backup to tape every 4 months
  - ARC files, indexes
- Random test recoveries from tape
  - Data recovery from tape is very slow





# Hard disk backups

- Daily copy during crawl on live hard disks
  - Lose at most 1 day of crawled data
- All data is replicated across 2 independent servers
  - ARC files, indexes, software





# Distant location backups

- Tapes moved to distant geographical location
  - Lisbon to Porto: 275 KM
- ARC files copied to the Internet Archive through the Internet
  - Lisbon to California: 9 000 KM



# #3: Monitor the service

# Monitoring tools fail

The service is broke  
but we didn't know

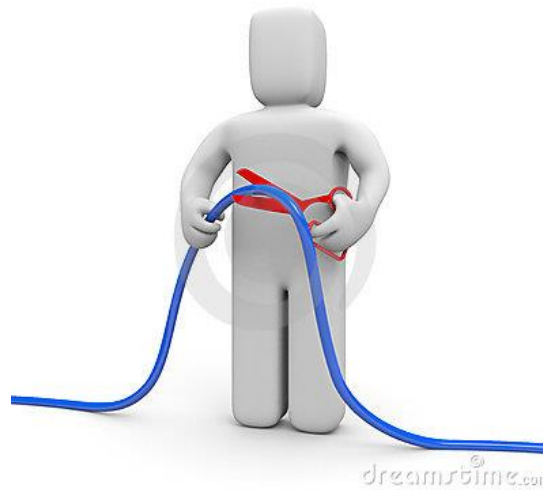


So we didn't do  
anything to fix it

# Use complementary/redundant monitoring tools for the service

- Hardware failures: vendor tools are not enough
- Availability
  - Nagios and Uptime Robot (external)
- Access stats
  - Awstats and Google Analytics (external)
- Resources
  - Cacti and Ganglia

# Induce faults to test monitoring!



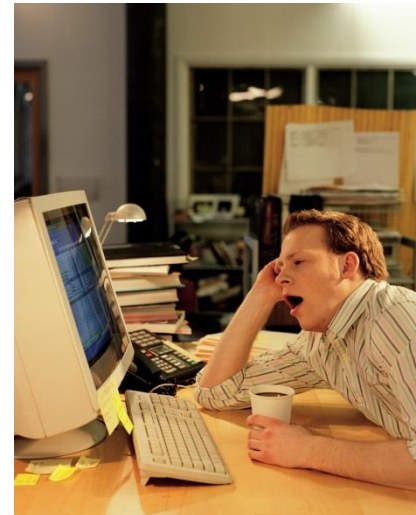
It's better to identify problems when you are ready for them

# #4: Quality Assurance for software development

Regression:  
"when you fix one bug, you  
introduce several newer bugs."

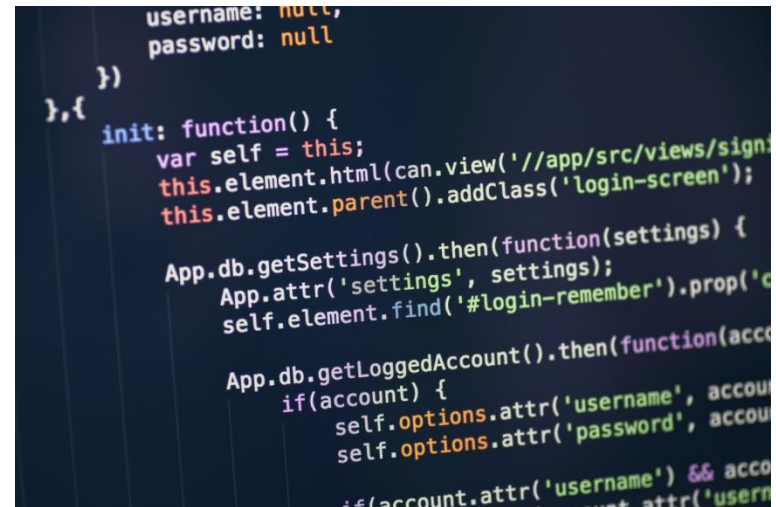


People get tired from doing  
repeatedly the same (testing).  
**Computers don't.**



# Code testing: automatize

- Compilation: the code is well written!
- Unit: does what it supposes to do!
- Functional: makes the service work!
  - Simulate user workflows (e.g. search for an archived page)



```
username: null,  
password: null  
})  
, {  
  init: function() {  
    var self = this;  
    this.element.html(can.view('//app/src/views/sign  
    this.element.parent().addClass('login-screen');  
    App.db.getSettings().then(function(settings) {  
      App.attr('settings', settings);  
      self.element.find('#login-remember').prop('c  
    App.db.getLoggedAccount().then(function(accou  
      if(account) {  
        self.options.attr('username', account  
        self.options.attr('password', account  
        if(account.attr('username') && accou  
        if(account.attr('usern
```



# Workload Capacity testing: automatize

## Minimum thresholds

- Workload average: 3 responses/second
- Speed average: 5 seconds per response



# Security testing: automatize

## Security

- OWASP Zed Attack Proxy (ZAP)



# Usability testing: conducted by skilled professionals

- Identify the problems that really affect the service
- Most Computer Science departments have Human Computer Interaction groups skilled to help you



# #5: Document and test procedures

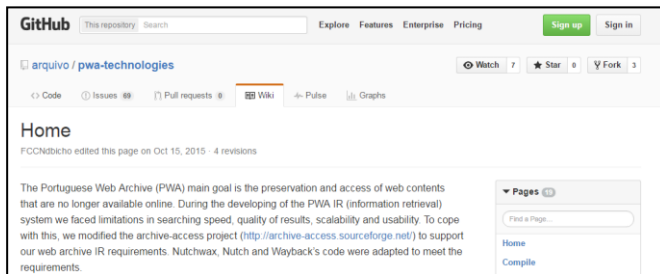
# Document procedures



## A first attempt to archive the .EU domain Technical report

Daniel Bicho  
daniel.bicho@fccn.pt

João Miranda  
joao.miranda@fccn.pt



## Learning Temporal-Dependent Ranking Models

Miguel Costa <sup>1,2</sup>  
miguel.costa@fccn.pt

Francisco M Couto <sup>2</sup>  
fcouto@di.fc.ul.pt

Mário J. Silva <sup>3</sup>  
mjs@inesc-id.pt

<sup>1</sup> Foundation for National Scientific Computing, Portugal

<sup>2</sup> Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal

<sup>3</sup> INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

## Several types of documentation with different purposes

- Wiki: internal procedures
- GitHub: software
- Reports: analysis
- Internal and external presentations: collaborations
- Scientific and technical publications: peer-review

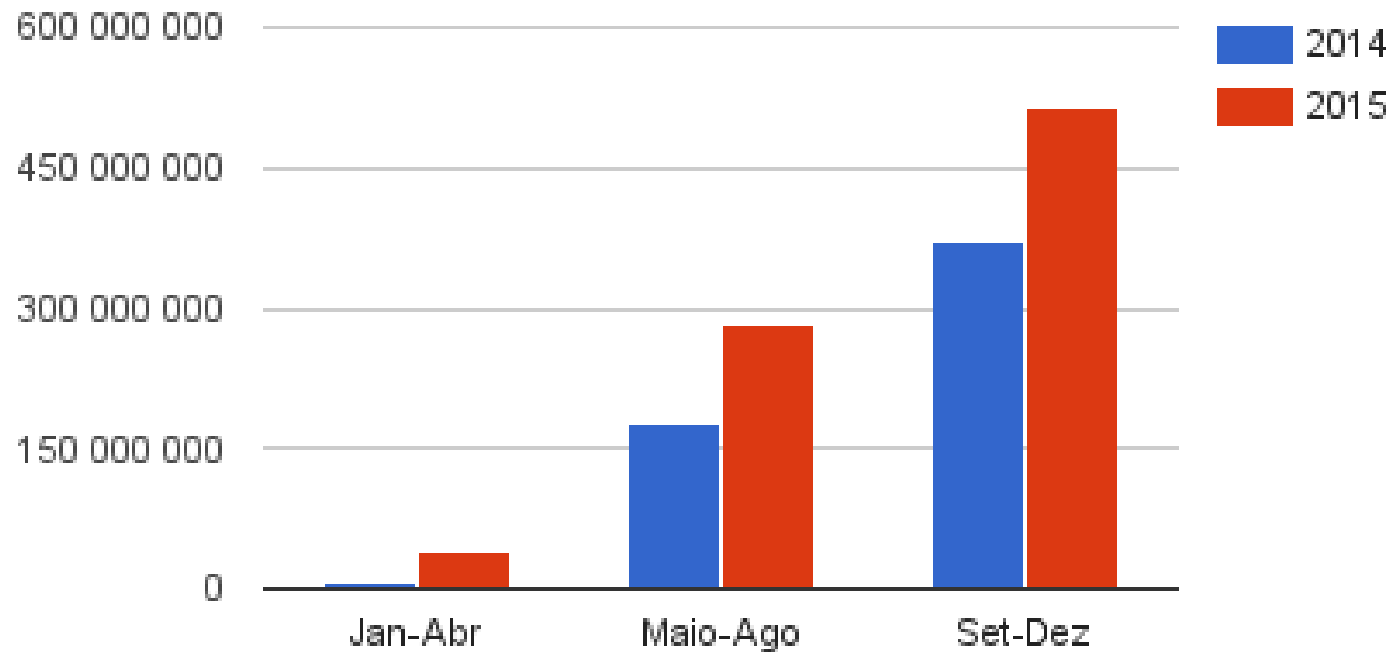
# Test the documentation

- Installations of software components from scratch
- Procedures executed by colleagues based on existing documentation without any help
- Open source everything we do
  - Increases responsibility
  - Increases quality
  - GitHub: pwatechnologies

# Results



# Crawling is stable

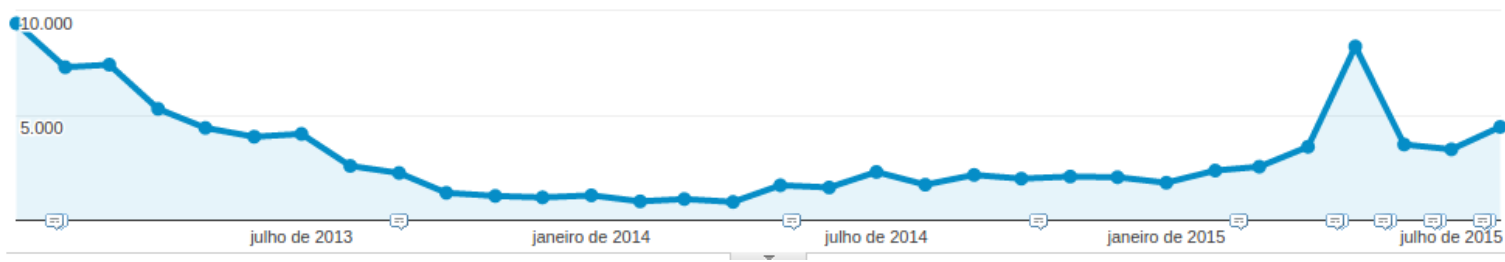




Search and Access availability in 2015

100%

# Recovering our users



- 4 090 users per month (average)
- Gaining new users
  - 90% are new users

# Lessons learned

- *Share-nothing* architecture for hardware and software
- Backup to Tapes is useful **but**
  - Recovery is very slow and prone to errors
  - Replication on live hard-disks is a must
- Test everything every time
  - Automate testing
- Development without proper Quality Assurance leads to waste of resources
- Accept staff rotation and proactively prepare for it