

The background of the slide is a grayscale aerial photograph of the University of Porto's campus, showing a dense cluster of modern and traditional university buildings.

M.EIC, 2022-23

Large Scale Software Development

Filipe Correia, Daniel Pinho, João Pedro Dias



How do we measure software development?

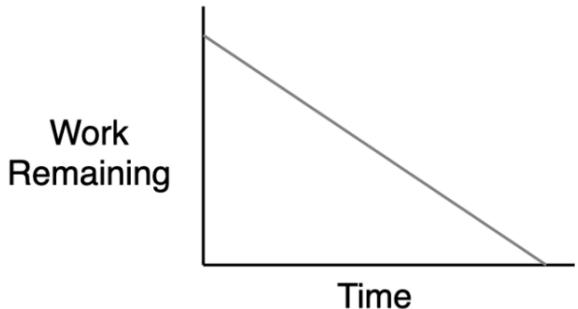
- Important to support “*inspect & adapt*”
- Productivity in software development is difficult (and pointless) to measure
 - Lines of code? Man-hours? Velocity? Utilization?
 - We can be really productive in building the wrong thing...
- Metrics are surrounded by pitfalls
 - Turning a metric into a *goal* can create the wrong incentives
 - Can be misinterpreted and misapplied

Recommended reading: *Cannot Measure Productivity*, by Martin Fowler

Burndown charts

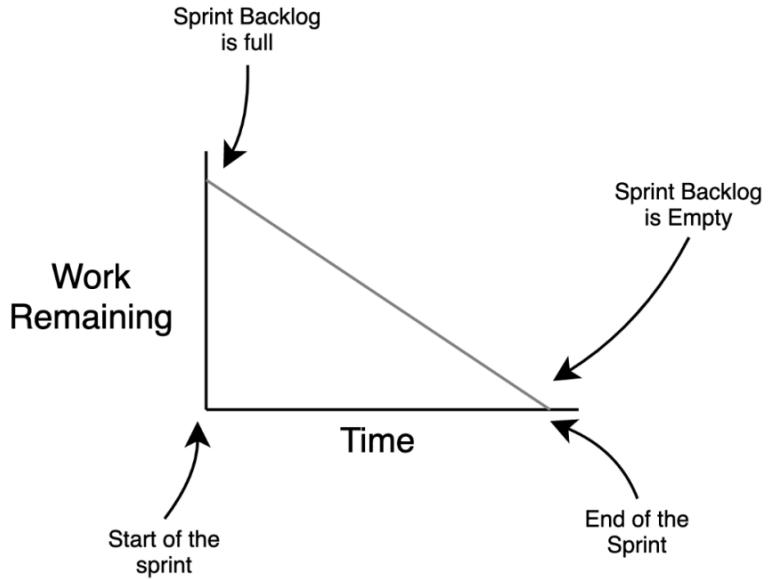
Burndown charts

- Visual representation of work left to do versus time.
- Provides more context to *velocity*
- Provides clues about things that may be happening in a project.
- But beware: it is easy to jump into the wrong conclusions!

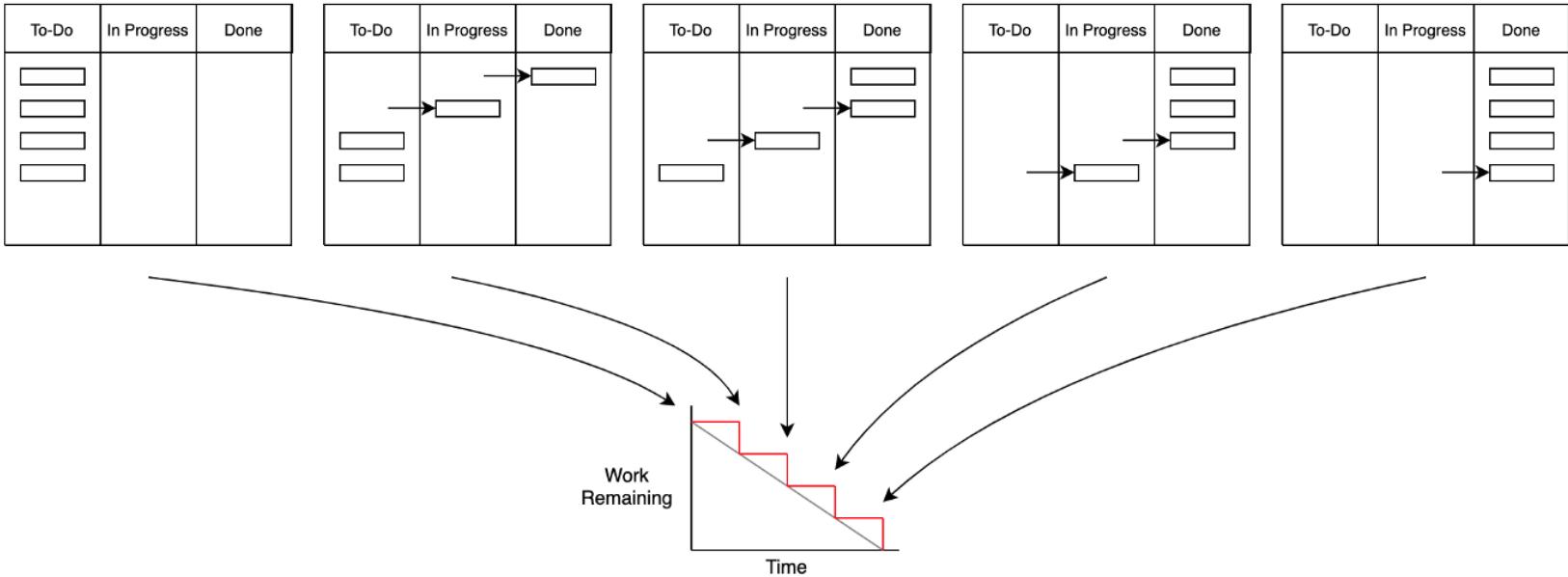


Burndown charts

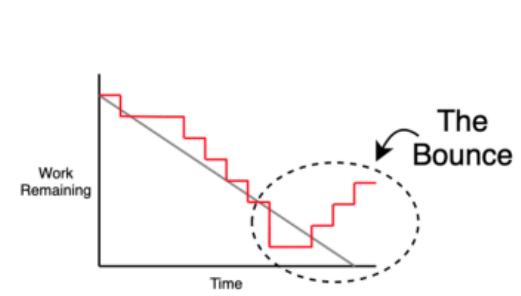
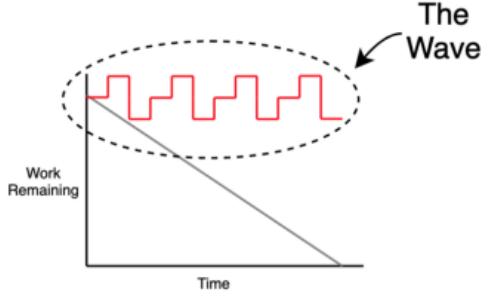
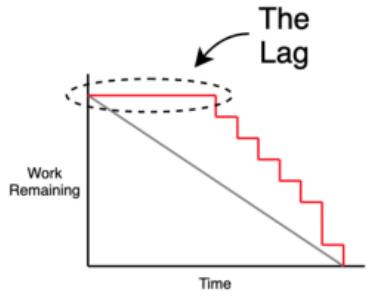
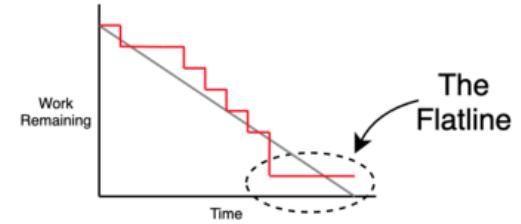
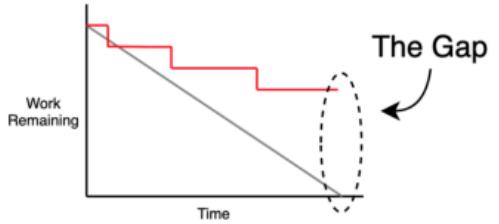
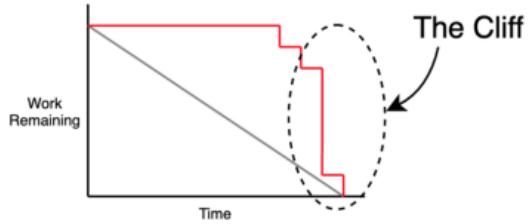
- Visual representation of work left to do versus time.
- Provides more context to *velocity*
- Provides clues about things that may be happening in a project.
- But beware: it is easy to jump into the wrong conclusions!



A straightforward flow



Flows that tell us something





Sign In Get started

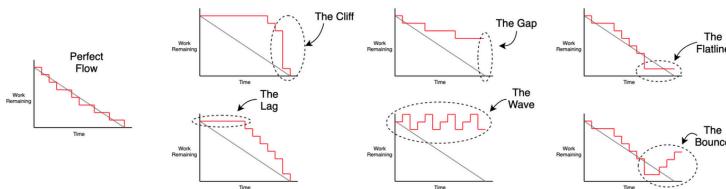


Chris Nielsen [Follow](#)

Aug 9, 2020 · 8 min read · ⚡ Member-only · [Listen](#)



Sprint Burndown Charts Gone Wrong



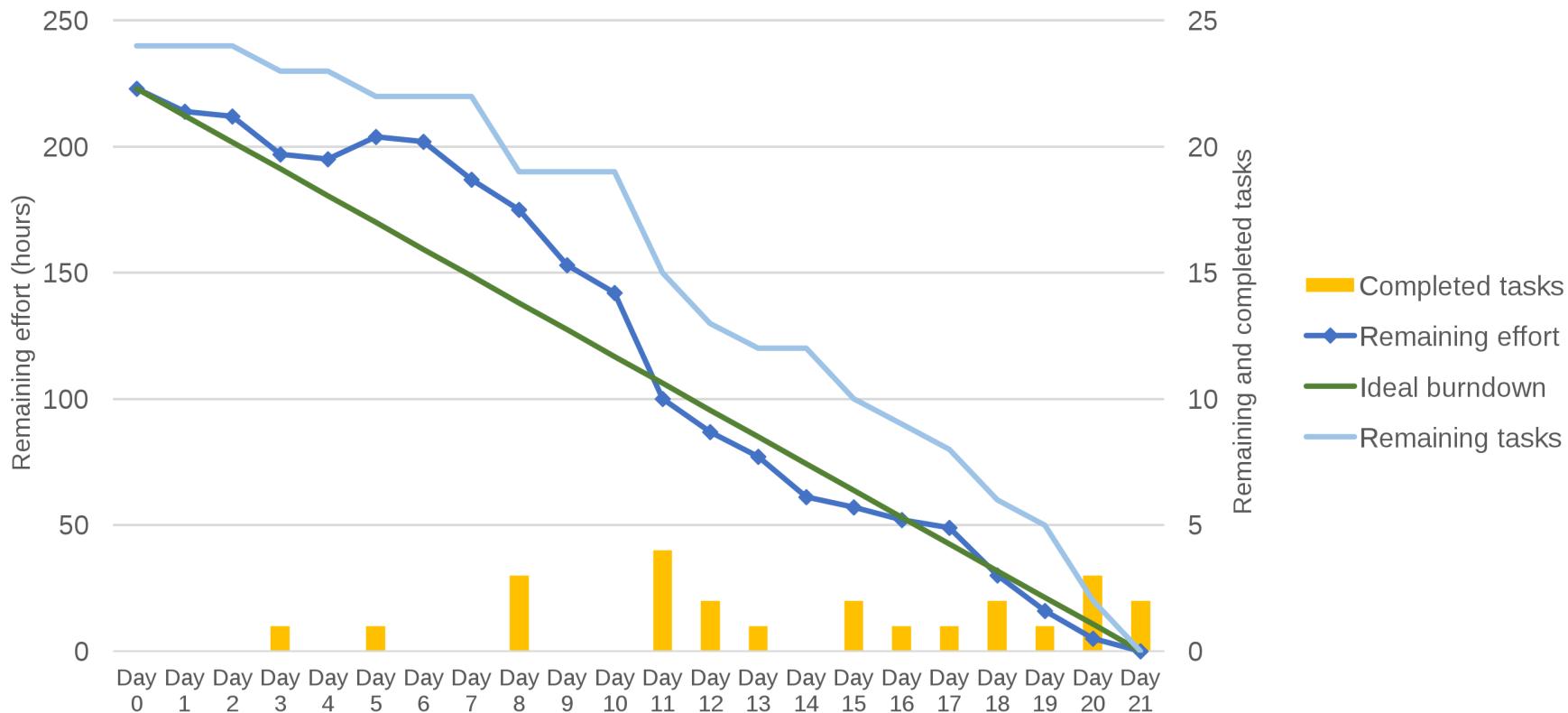
You can learn a lot about a team from their sprint burndown chart. If you know how to read it.

After years of coaching and leading agile teams I've found 6 common patterns that appear over and over again in sprint burndown charts when something is going wrong. Unfortunately, burndown charts are just too high level to tell you specifically what a team is doing. But they'll point you in the right direction. Then you'll know what questions to ask. And quite frankly, the most important thing you can do to help a self directed agile team grow, is to ask the right questions.

The basics

But before we start looking at problems, let's make sure we understand the

Sample Burndown Chart

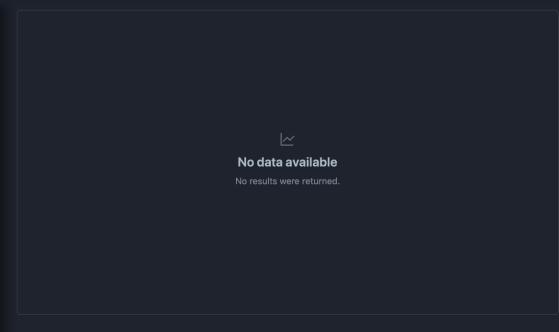


Burndown charts on GitHub

(almost...)

The screenshot shows a Jira interface with two main panels. On the left is the 'Project Board' for the project 'case-management-platform'. It displays four columns: 'Product Backlog' (30 items), 'Sprint Backlog' (0 items), 'In Progress' (9 items), and 'Done' (9 items). The 'Product Backlog' column contains several user stories, one of which is highlighted with a red box. The 'In Progress' column also has a red box around its header. On the right is the 'Insights' section, specifically the 'Burndown Chart'. The chart tracks progress from October 14 to November 10. It shows three sprints: Sprint 1 (blue line), Sprint 2 (green line), and No Sprint (purple line). Sprint 1 starts at level 10 and ends at level 10. Sprint 2 starts at level 10, drops to level 10 by Oct 29, and ends at level 24. The 'No Sprint' line starts at level 0, peaks at level 10 on Oct 29, and ends at level 0 on Nov 10.

DS 2022/23 Project Burndown charts (10 Nov @ 11:30)



DevOps

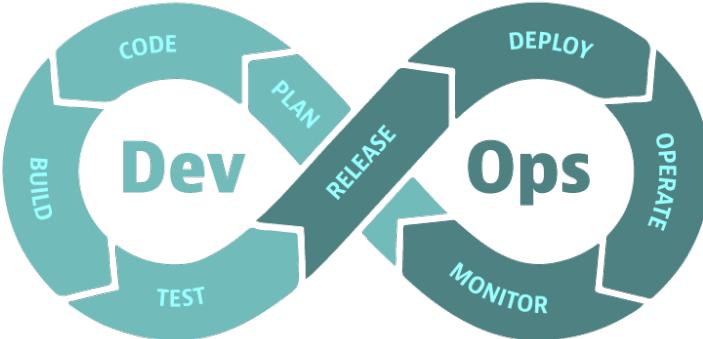
Defining DevOps

“a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality”

- Len Bass, Ingo Weber and Liming Zhu

Defining DevOps

- Cultural approach to releasing software quickly and reliably into production
- Based greatly on the *agile* mindset, with a focus on automation and tooling
- Depends on close collaboration between *development* and *operations*
- ... and everyone, really (DevSecOps, DevBizOps, AIOps, DataOps, ...)



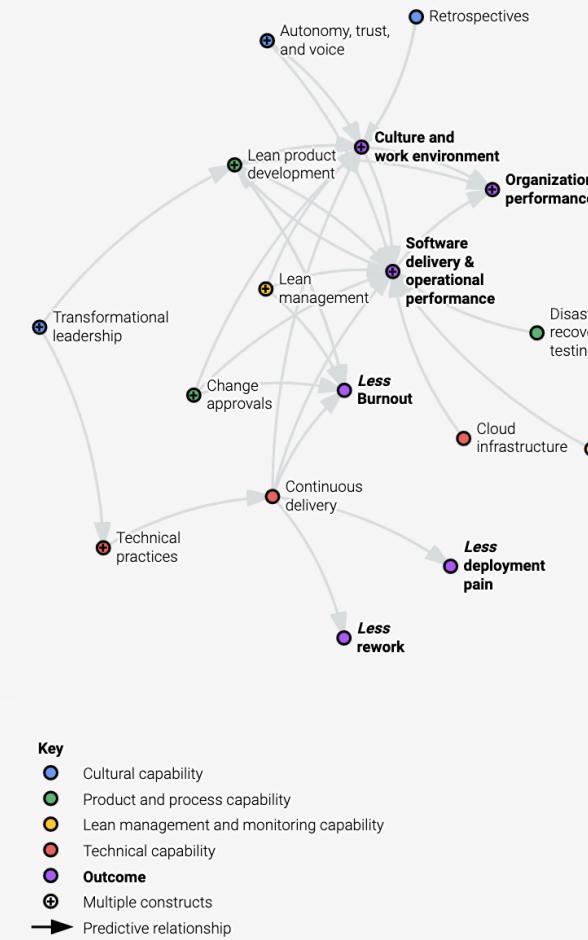
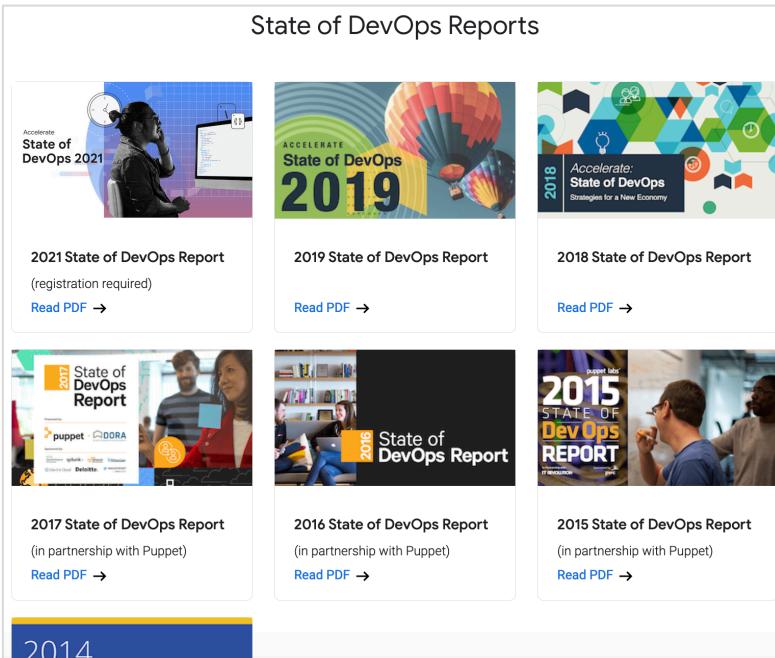
Some DevOps Practices

- Build automation
- Test automation
- Infrastructure as code
- Deployment automation
- Deployment/integration pipelines

Four Key Metrics

Measuring Software Delivery

DevOps Research and Assessment (DORA)



Four Key Metrics

Speed
(throughput)

- Deployment frequency
- Lead time for changes

→ How often we successfully release to production
→ The time it takes for a commit to get into production

Stability

- Mean time to recover (MTTR)
- Change failure rate

→ The time it takes to recover from a failure in production
→ The % of deployments causing a failure in production

- Focused on outcomes (rather than outputs)
- About how well teams are developing and delivering software
- Try to capture intention
- Global – they imply collaboration

State of DevOps Report 2021

Software delivery performance metric	Elite	High	Medium	Low
⌚ Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
🕒 Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one hour	Between one day and one week	Between one month and six months	More than six months
⌚ Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week	More than six months
⚠ Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hot fix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

Findings of the State of DevOps Reports

- *There is no tradeoff between speed and quality – perfectly possible to have both!*
- *Excellence in software delivery and operational performance drives organizational performance.*
- *Elite performers now make up 26% of teams in our study, and have decreased their lead times for changes in production.*
- Comparing elite and low performers, we find that elite performers have:
 - **973x** more frequent code deployments
 - **6570x** faster lead time from commit to deploy
 - **3x** lower change failure rate (changes are 1/3 less likely to fail)
 - **6570x** faster time to recover from incidents

<https://cloud.google.com/blog/products/devops-sre/announcing-dora-2021-accelerate-state-of-devops-report>

Take the DORA DevOps Quick Check

Measure your team's software delivery performance in less than a minute! Compare it to the rest of the industry by responding to **five multiple-choice questions**. Compare your team's performance to others, and discover which DevOps capabilities you should focus on to improve. We don't store your answers or personal information.

[Take the Quick Check](#)

QUESTION 1 OF 5

Lead time



For the primary application or service you work on, what is your lead time for changes (that is, how long does it take to go from code committed to code successfully running in production)?

- More than six months
- One to six months
- One week to one month
- One day to one week
- Less than one day
- Less than one hour

QUESTION 2 OF 5

Deploy frequency

For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?

- Fewer than once per six months
- Once per six months

References

- Cannot Measure Productivity
Martin Fowler, 2003
- Sprint Burndown Charts Gone Wrong
Chris Nielsen, 2020
- Accelerate: The Science of Lean Software and DevOps
Nicole Forsgren, Jez Humble, Gene Kim
- Continuous Delivery: Reliable Software Releases through Build, Test, & Deployment Automation
Jez Humble and David Farley, Addison-Wesley, 2010
- Explore DORA's research program
- 2021 Accelerate State of DevOps report addresses burnout, team performance
Dustin Smith – DORA Research Lead, 2021
- What is DevOps? (video)
Dave Farley
- If you don't know where you're going, it doesn't matter how fast you get there (video)
Nicole Forsgren, Jez Humble @ DOES18

FEUP.DEI



DEI - DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
dei.fe.up.pt | secdei@fe.up.pt | +351 225 082 134
Director **João MP Cardoso** | Full Professor
jmpc@fe.up.pt | +351 220 413 284

FEUP - FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Rua Dr. Roberto Frias | 4200-465 Porto | PORTUGAL
www.fe.up.pt | feup@fe.up.pt | +351 225 081 400
 @DEI_FEUP