AGILE BASICS

Continuous improvement

We as a team (including Ops and Helpdesk) work together to continuously improve our collaboration. For example, retrospectives are done regularly. Actions are followed up in a regular, measurable, and official way

#CI

AGILE BASICS

Backlog management

The Product Owner regularly updates an ordered and qualified backlog, which is shared with the team and stakeholders. Before development, we, the entire team slice epics into User Stories and work together on the acceptance criteria. There are a DoR and a DoD

#grooming #backlog

AGILE BASICS

Synchronisation

On all organizational levels, synchronization points are set up, known by all teammates and carried out, including: daily meetings, backlog grooming, inter-team sessions, demonstrations. These synchronizations are supported via clear efficient visual management

#daily #demo #ritual

METROLOGY

Business indicators

Indicators linked to product delivered value are available in dashboards, which allow all actors to get feedback about their activity so we may adapt and enhance performance

#value

METROLOGY

Strategy & tools

We (Business and IT) rely on a common strategy helping us to improve our measurement capacity, thanks to tools supporting indicators time-based analysis and correlations

#datalake #ELK

METROLOGY

Usage tracking

In order to detect potential issues, monitor activity and product usage, we (all IT actors) have access to shared real-time dashboards

#monitoring

CD PIPELINE

Build & Package

Artifacts building and packaging are not hand-made. They are uploaded automatically on a shared server, secured, and dedicated. Traceability is ensured

#continuousIntegration

CD PIPELINE

Provisionning & Deploy

To decrease error and deploy time, environment setup and artifact deployment are not hand-made. Dev and Ops share the same tools, which are used from Dev to Prod environments

#infraAsCode #IAAS

CD PIPELINE

Continuous tests

In order to provide fast feedback on changes and decrease testing time despite application code base growth, tests are triggered automatically in the software factory

#fastFeedback #testAuto

TESTING

Strategy

We keep our testing strategy up-to-date, allowing everyone to have a somewhat clear view of objectives and means in order to ensure acceptance and product quality

#testPyramid

TESTING

Frameworks & Tools

We have testing tools and frameworks designed for all kinds of tests, and those fit our technical and functional context

#unitTest #intTest #perf

TESTING

Practices

We understand TDD and BDD practices, and tests typology (unit, integration, E2E, performance). We have implemented and maintain the best technique according to the context

#TDD #BDD #testFirst

CODE QUALITY

Practices

The developers know theorical principles of code quality, such as CleanCode, Legacy refactoring, and SOLID. Developers apply the Boyscout rule on the code base

#cleanCode #refactoring

CODE QUALITY

Community

The developers regularly do code reviews, where they can share and challenge their points of view, and update common rules

#codeReview #pair

CODE QUALITY

Tools

Code analysis tools are used to tangibly measure quality, detect rule violations, and compare the metrics on a temporal axis

#sonar

COLLABORATION

Transparency

In order to globalize the efficiency, we are committed to expose our objectives, constraints and priorities to our partners. Mutual transparency allows better collaboration

#systemThinking

COLLABORATION

Collective ownership

We as a team particularly pays attention to our resilience. We are co-responsible and practice sustainability by avoiding concentrated roles and knowledge held by one single person

#oneTeam #noHero

COLLABORATION

Spec by example

Specifications, tests and user guides are one single document, produced by BAs, developers, testers and support teammates who iteratively explore user needs and provide examples. This living documentation allows automated testing and saves time

#BDD #executableDoc

PARTNERSHIP

PO onboarded

The Product Owner is part of the team, they listen to the members and accept to experiment. They contribute to continuous improvement and take part in retrospectives

#productOwnership

PARTNERSHIP

Business value model

In order to maximize delivered product value, we (business and IT) use an efficient framework and a unified vocabulary when discussing prioritization. The reflection is based on business objectives, and incorporates: context, costs, benefits, risks, quality…

#value #BVM

PARTNERSHIP

Slicing

A particular attention is given to slicing allowing us to deliver increments that bring value to the clients. All teammates (including the PO), regularly take part of slicing sessions

#storyMapping

LEADERSHIP

Decentralized decisions

In order to avoid bottlenecks and centralized control, mandates from the hierarchy are limited. Those responsible should clearly define areas where a decision can solely be taken by the hierarchy or is in team's hands

#delegationPoker

LEADERSHIP

Space for sharing

Managers seek to facilitate communication, sharing of knowledge and crosstraining of teammates both internal and external. Time is dedicated to sharing: BBL, onboarding of new arrivals, mentorship pairings, strengthened influence of most knowledgable colleagues, etc…

#BBL #hackathon

LEADERSHIP

Start with why

Instead of making decisions on "what" and "how", the manager communicates the intention and the vision, so that the team may propose the solutions with confidence

#hostLeadership

PEOPLE

Develop People

Trust takes place in the team's culture. Regular feedback is natural and constructive, without judgement. One's skills are known and training paths tailored to the individual's needs exist

#trust #training

PEOPLE

Respect

Experimentation and right to failure are core principles of our team. Raising an issue is considered valueable as it drives continuous improvement. We adhere to a high standard in order to provide the best quality of product possible to clients

#rightForFailure

PEOPLE

Role modeling

Team values and rules exist and are built and shared by the entire team (manager included). Instead of pointing fingers, we try to lead by example

#leadByExample