



Release acceleration by combining Agility, Requirements and Continuous Delivery

PagesJaunes.fr Feedback

#PagesJaunes

Fabien Grellier
CTO Pagesjaunes.fr
fgrellier@pagesjaunes.fr
@FabGrellier

POLARION USER CONFERENCE – 13/10/2015 - Stuttgart



What is PagesJaunes ?

- French leader in local advertising and information, PagesJaunes designs and develops a wide range of new content and services on several media, web, mobile and printed.
- Pagesjaunes.fr is the reference portal for local practical information. It brings together :
 - Research services (contact information, reversed),
 - Geographical representation (maps, directions, 3D, aerial views, etc.

Key numbers

936.2

- million € of revenue,
- 632,5 millions € on Internet

700,000

- Customers in France

4000

- employees

1.48
billion

- visits on desktop in 2014

5th

- French website in traffic

4,301,200

- Pages viewed per day

50

- developers

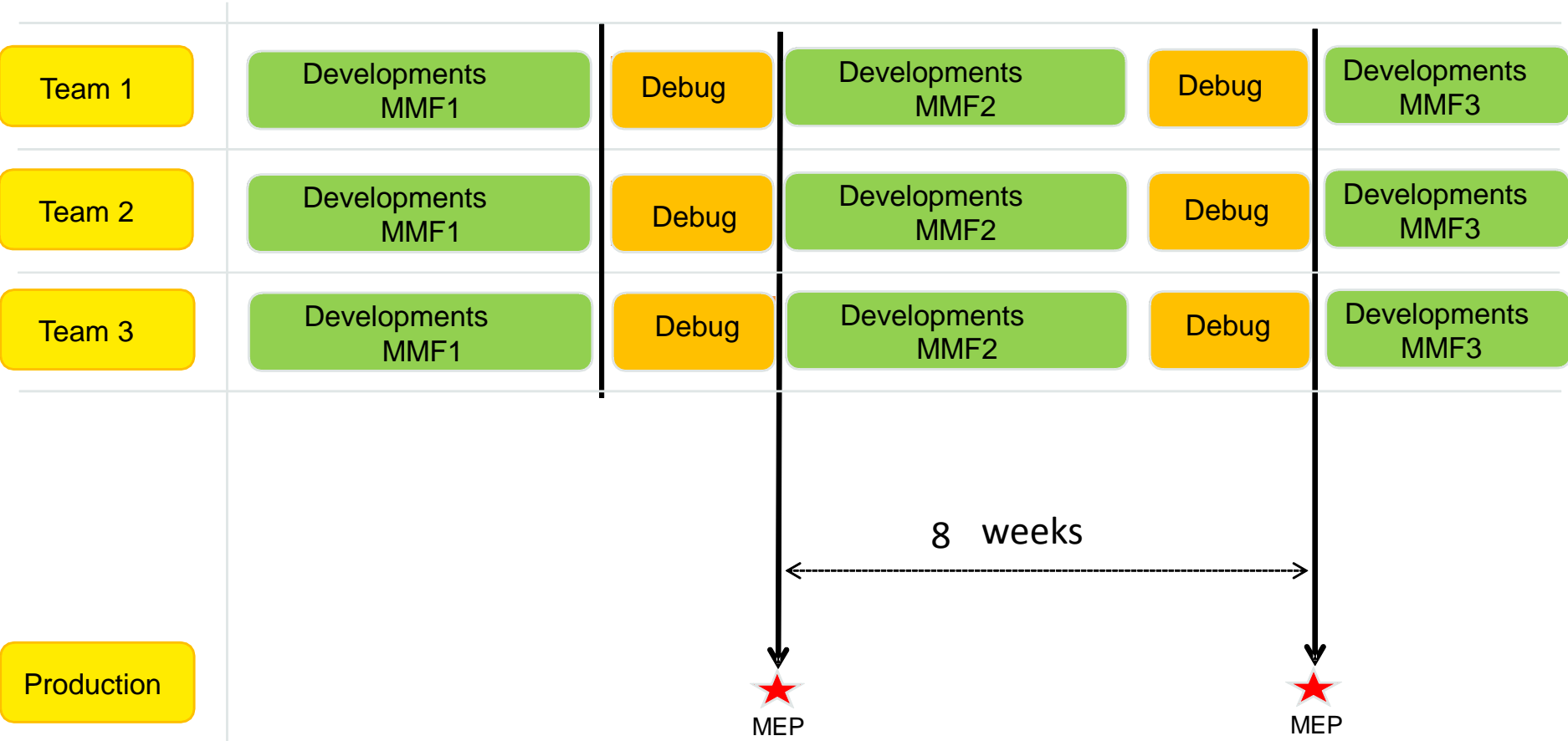
350,000

- lines of code

12,000

- Functional tests cases

Vintage deployment cycle



NewPagesjaunes.fr : targets

Shorten TTM and TTR (*)

- Deliver into production several times a day

Reduce costs

- Computerize at maximum
- Improve the maintainability of automated tests
- Focus the impacts of changes on the testing assets

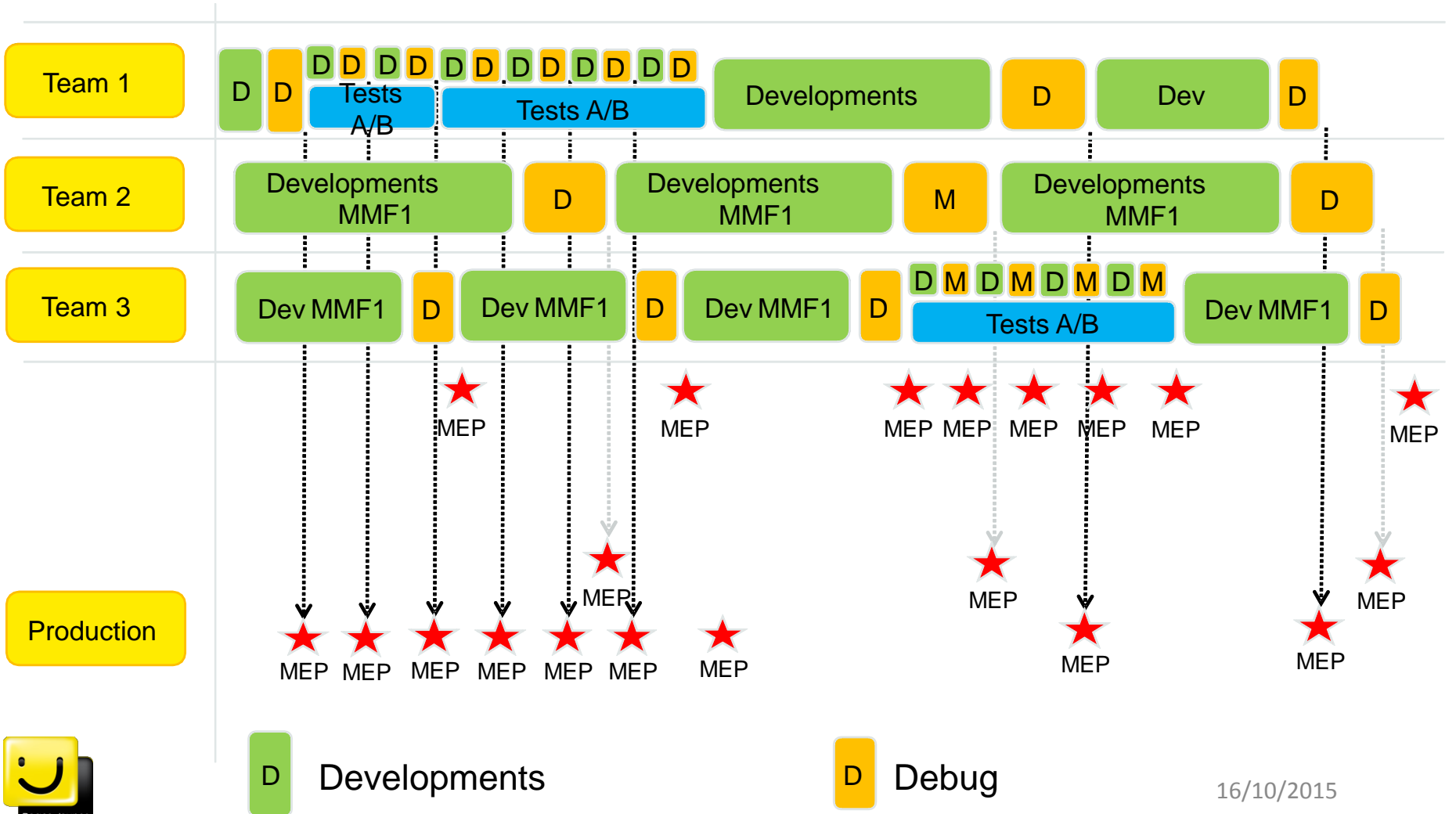
Improve the overall quality of service

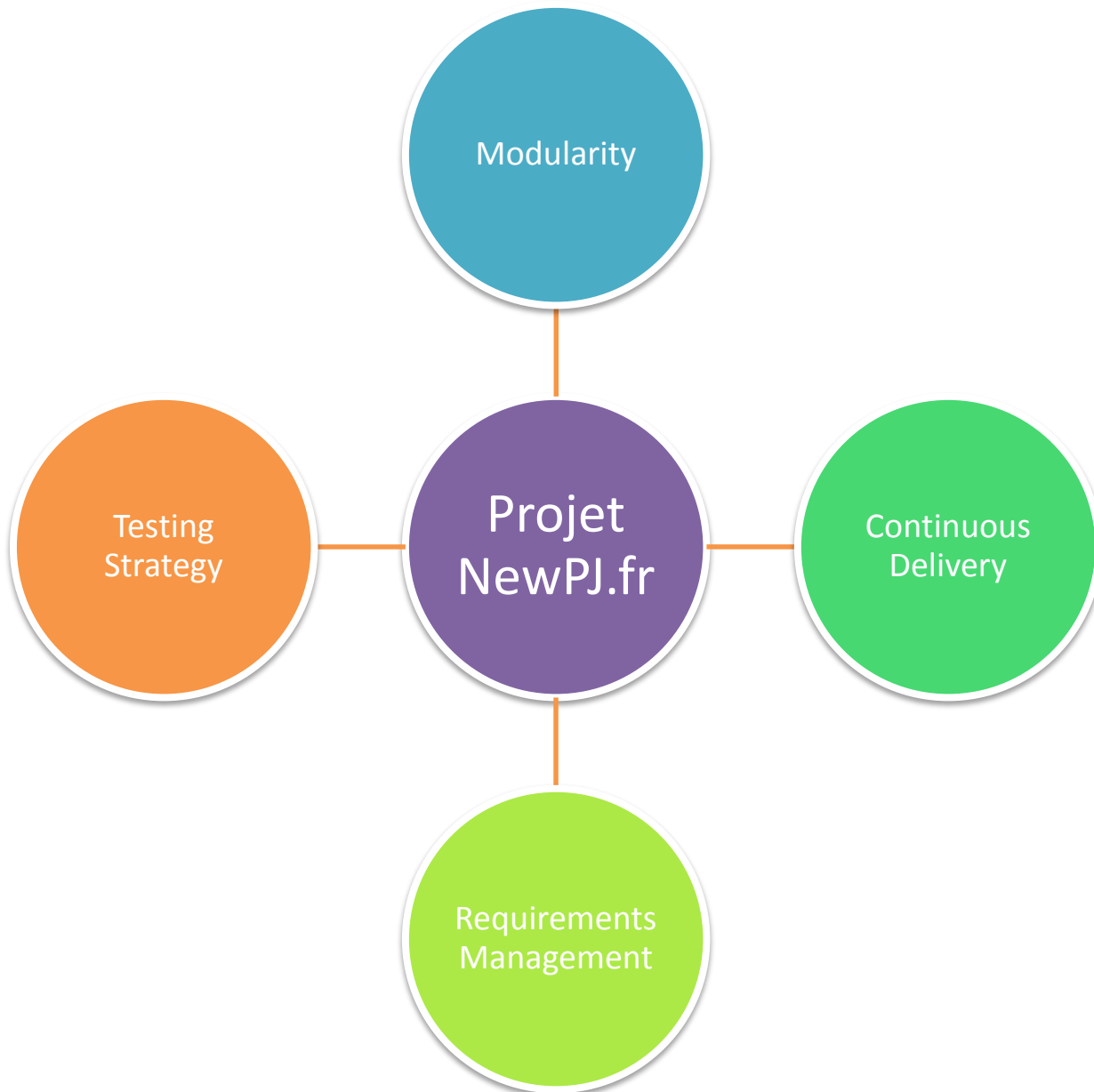
- Reduce the number of detected bugs into production
- Define a test strategy maximizing cost constant quality

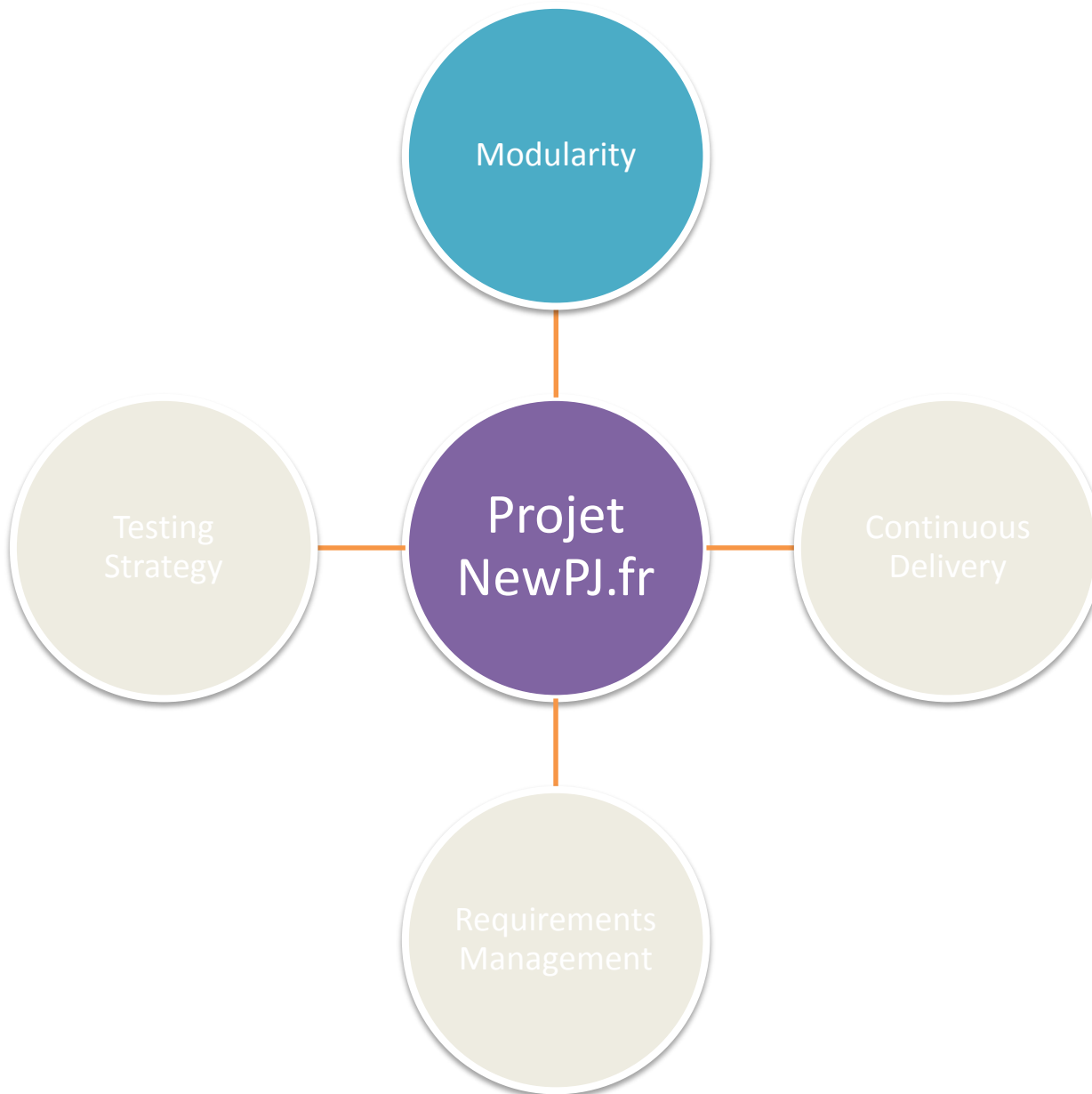
(*) TTM : Time To Market
TTR : Time To Repair



NewPJ delivery cycle



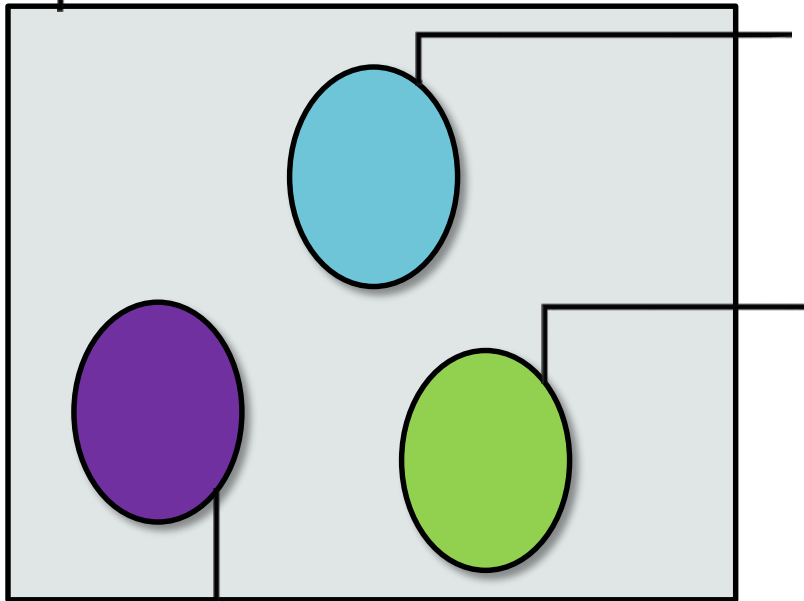




Definition of a module

❑ Complete and autonomous modules, communicating in REST

- Each one contains code, requirements and tests



❑ Requirements



- Written by TechLeader and Product Owner
- Expresses all the technical and functional requirements perennially

❑ Tests



- Verify compliance to module requirements
- Technical : unit, integration, performance
- Functionals : sélénium, REST Assured

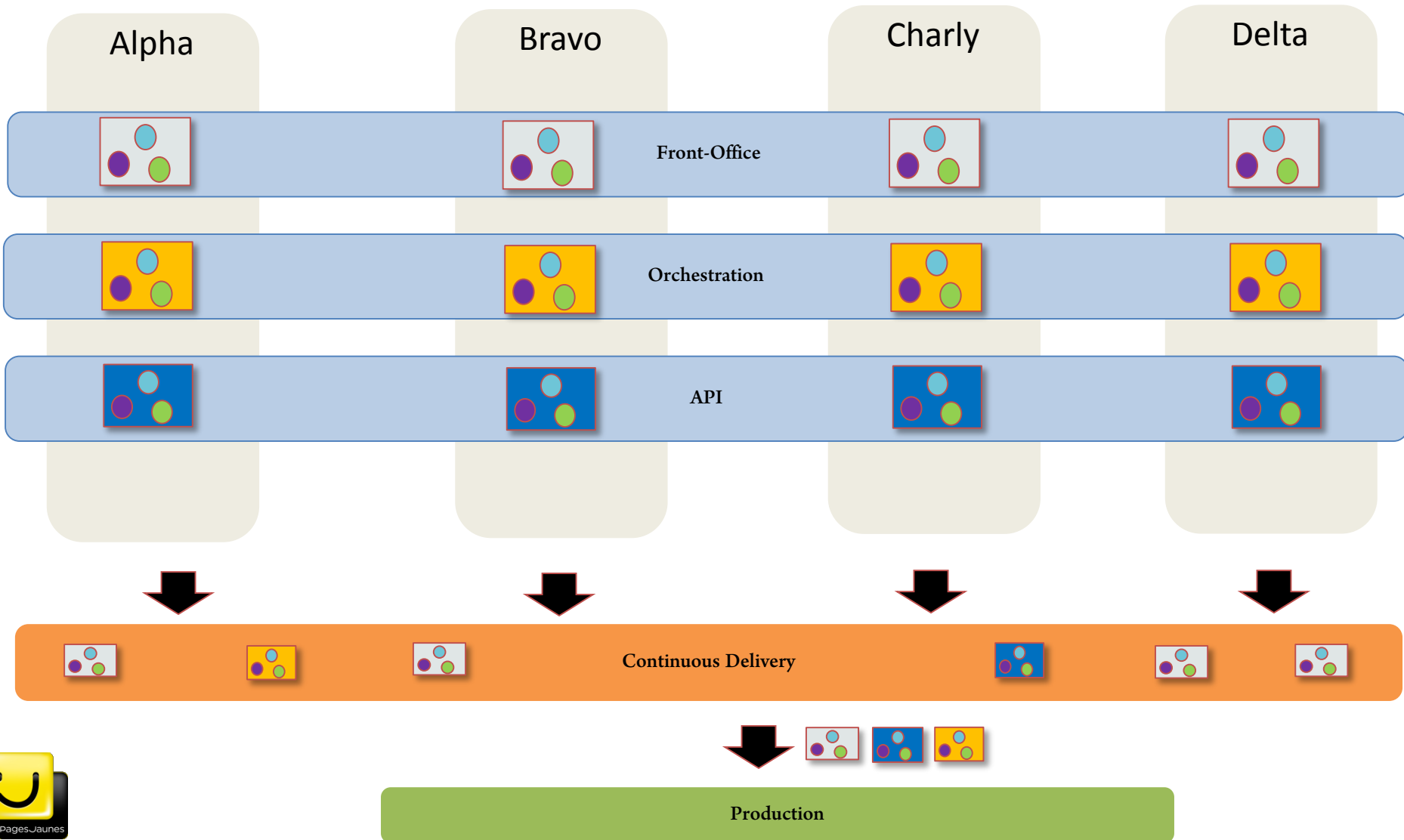
❑ Code

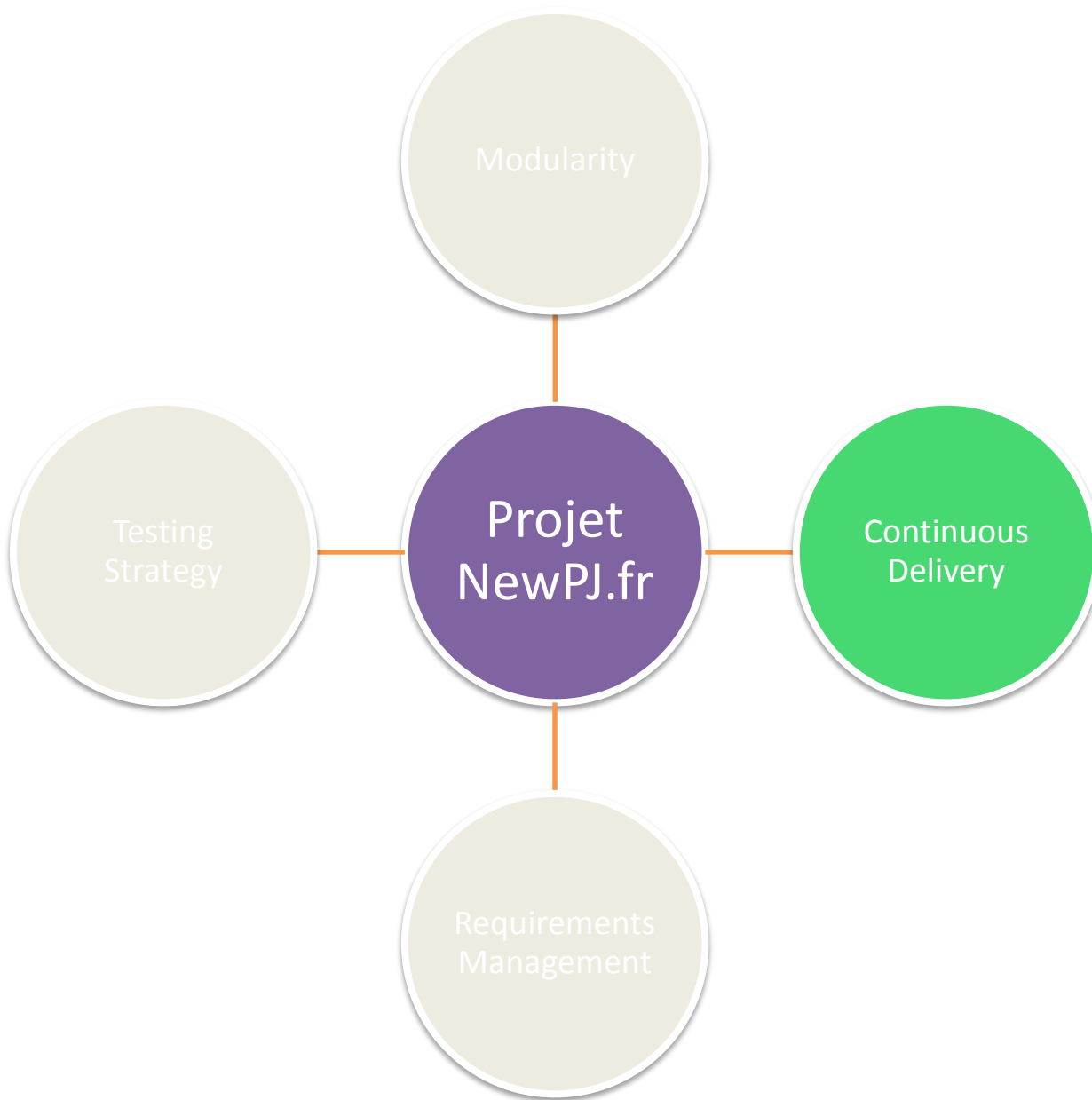
- Written by the developer
- Unit tested by the developer

**Automated tests for
Continuous Delivery**

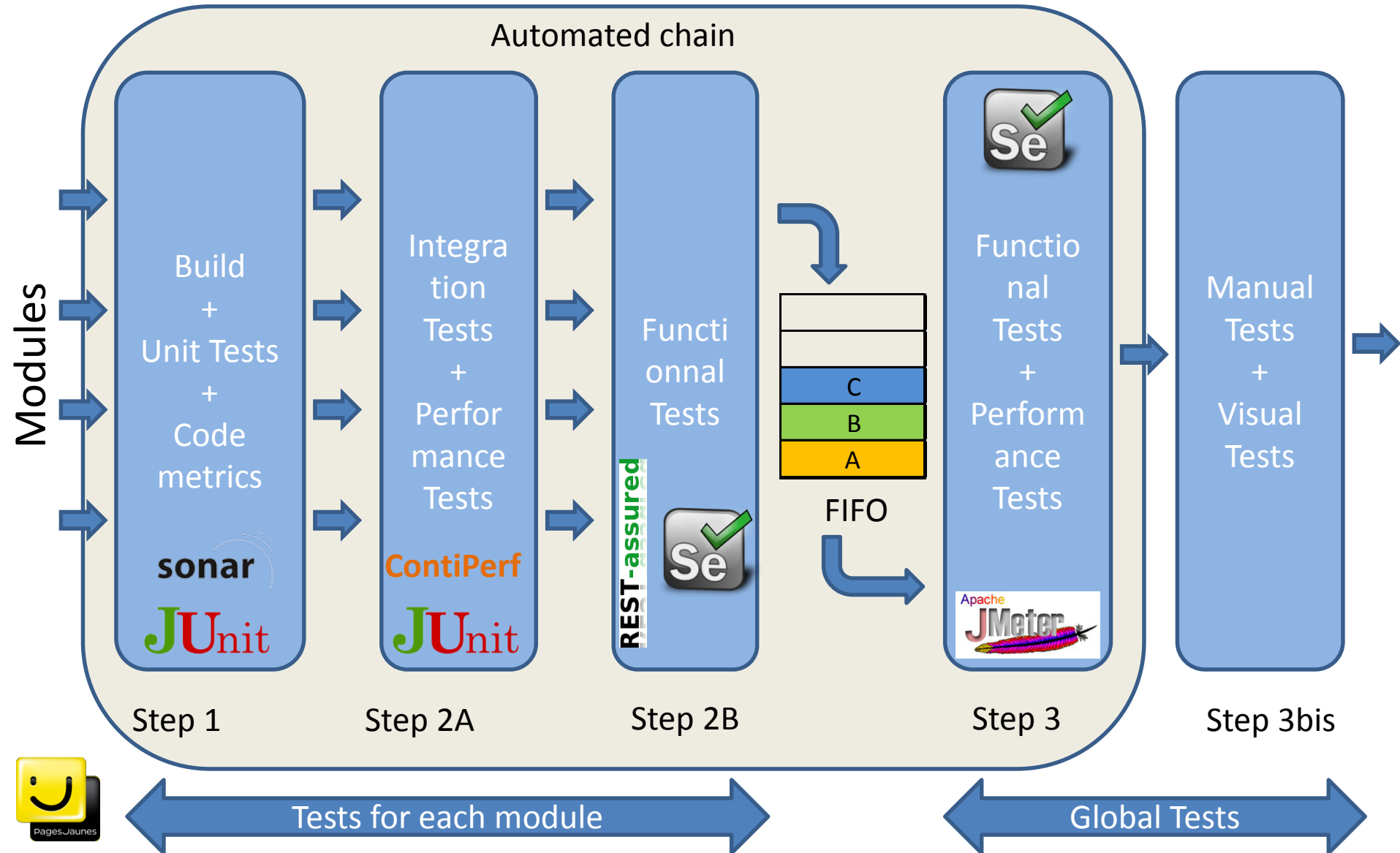


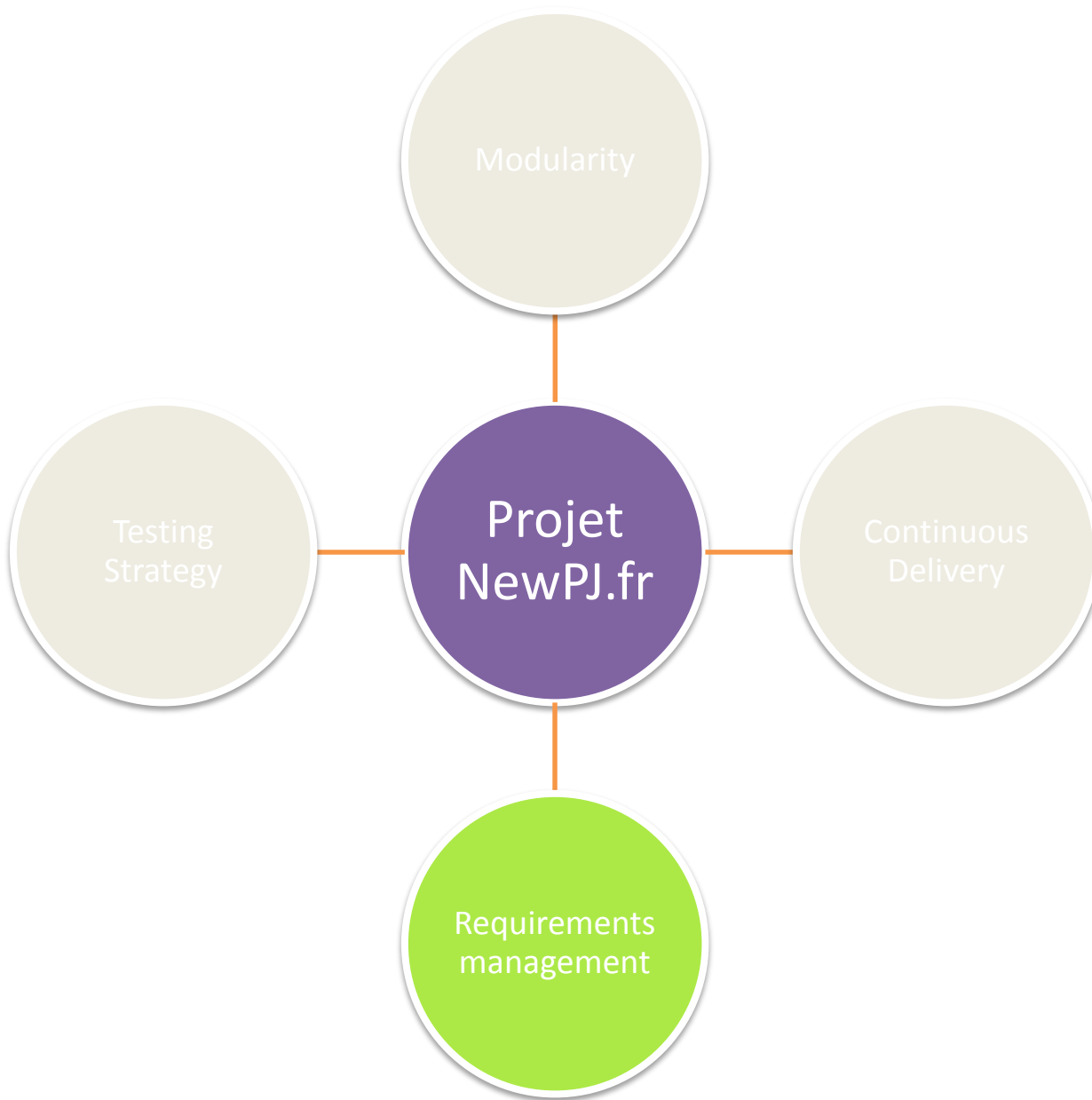
Micro-services and Agile teams





Macro vision of the Continuous Delivery chain





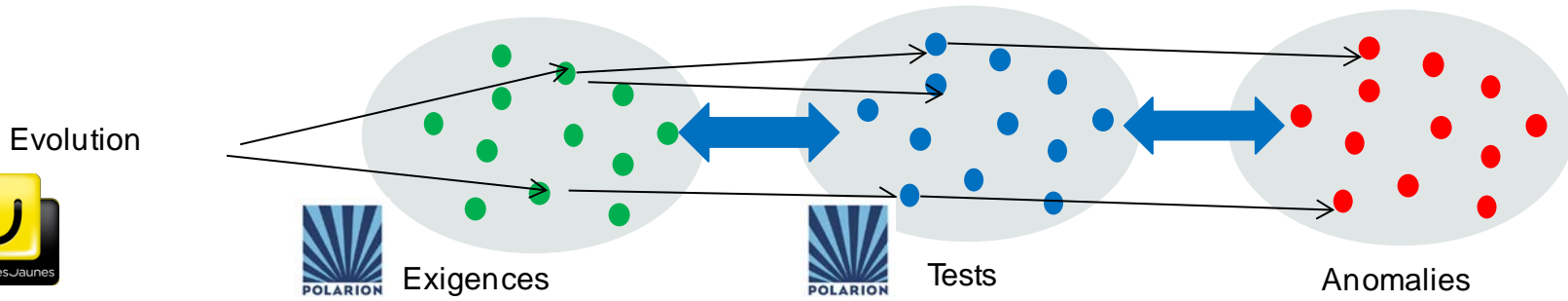
Why a requirements management?

Formalization of the functional

Traceability Requirements / Tests / Defects

Impact analysis

Requirements coverage by tests



Polarion's interactions



Requirements
Criticality



TestRuns

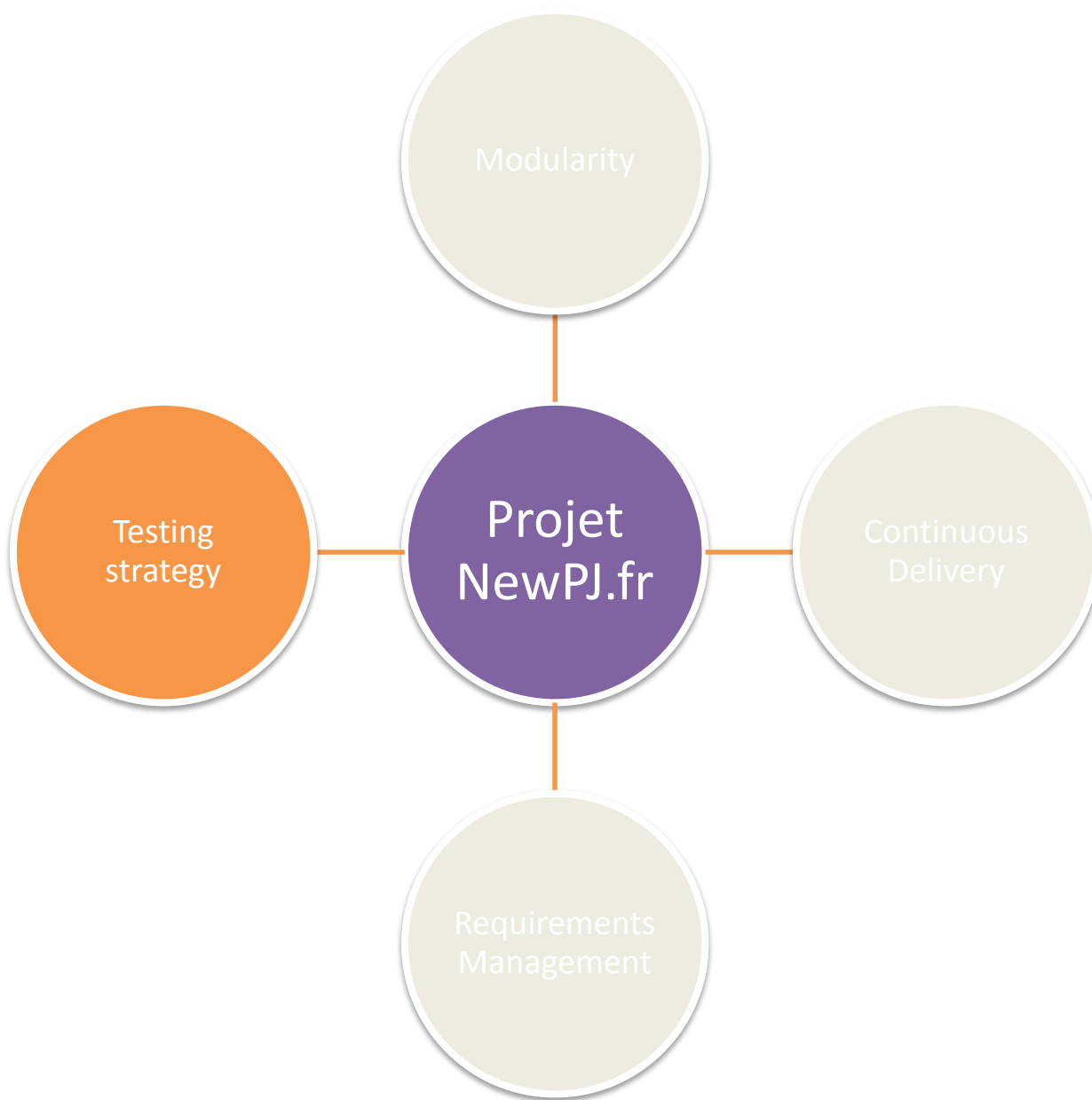


Automated Tests

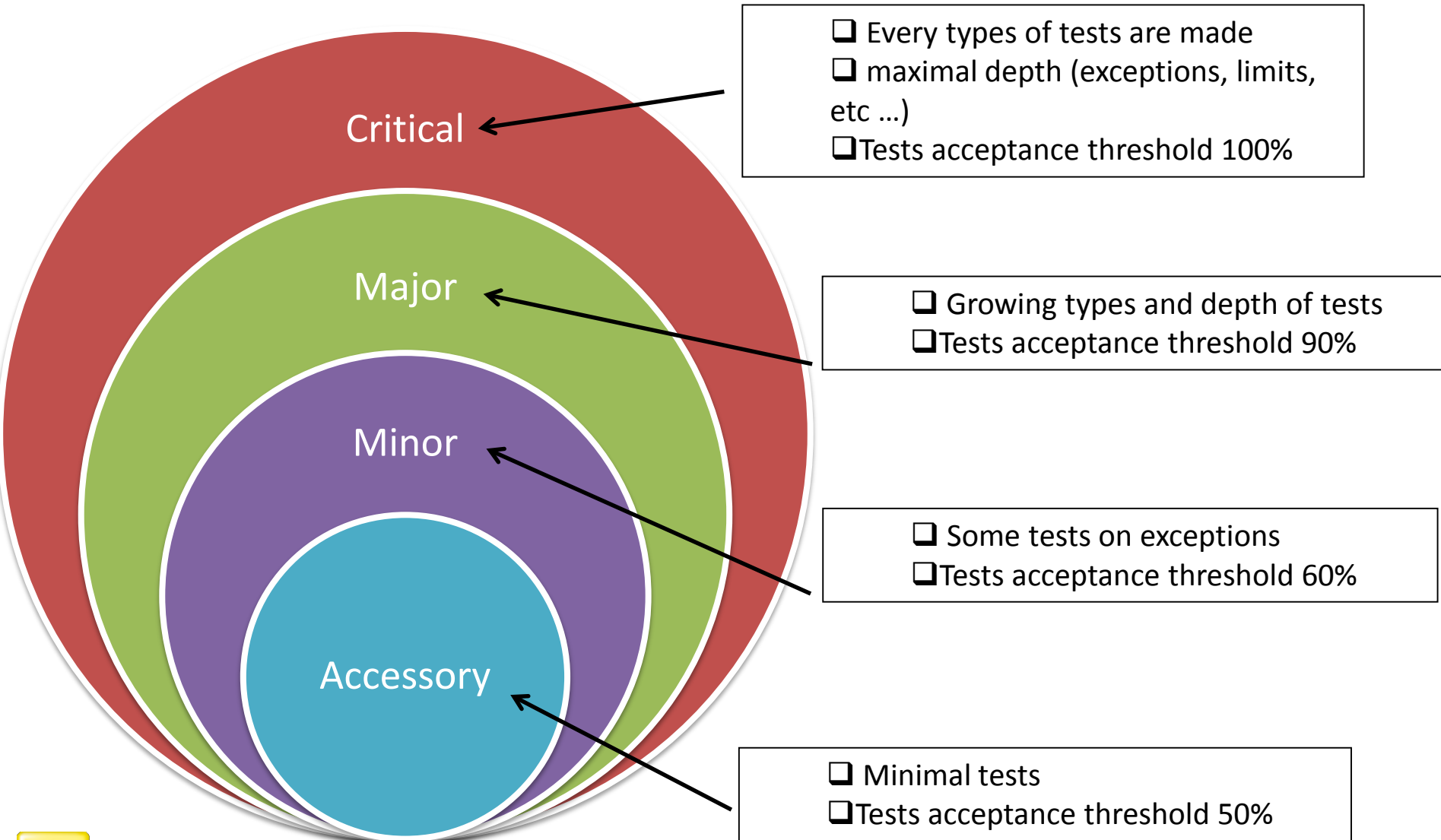


Execution



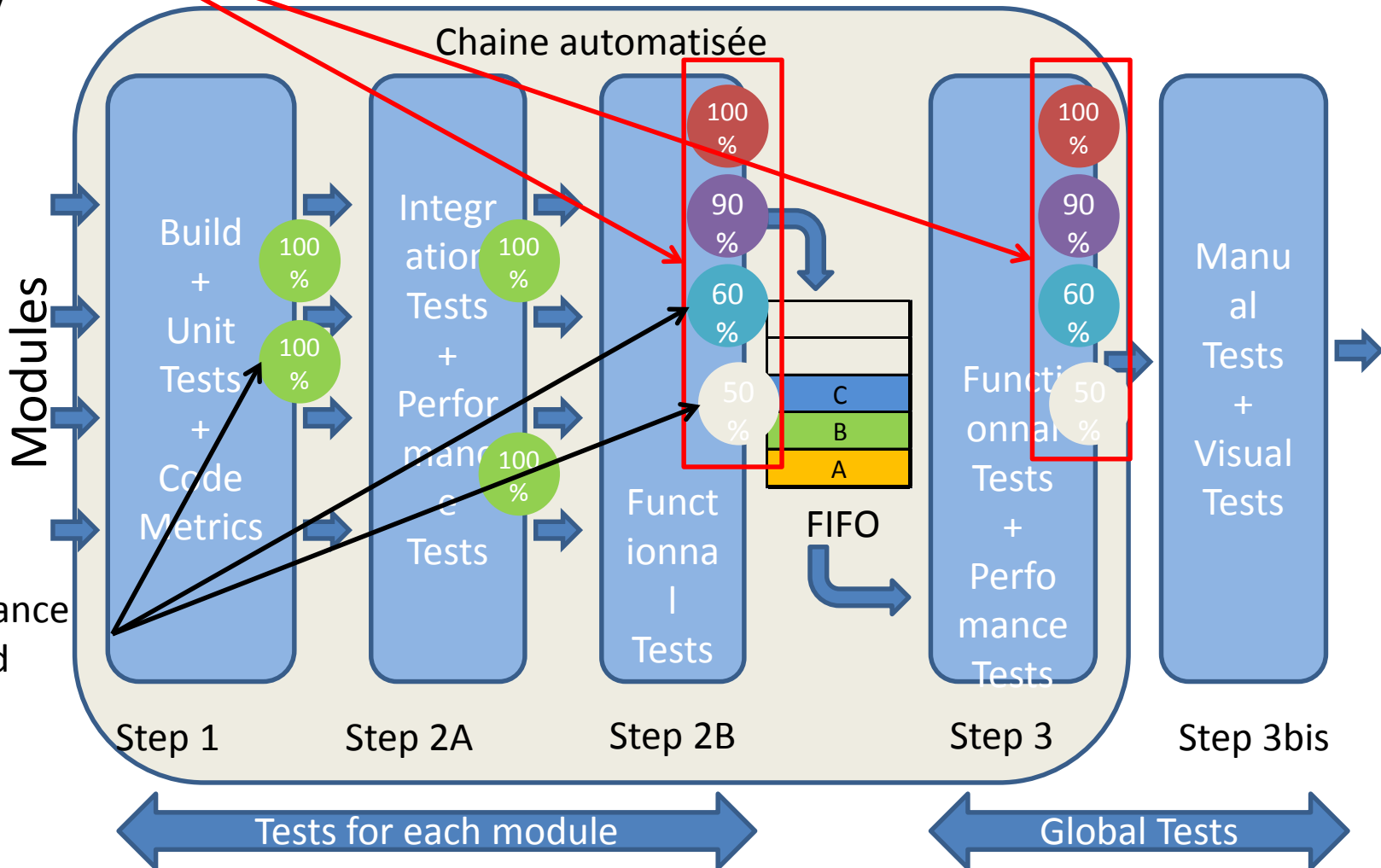


Tests depth depending on criticality of requirements

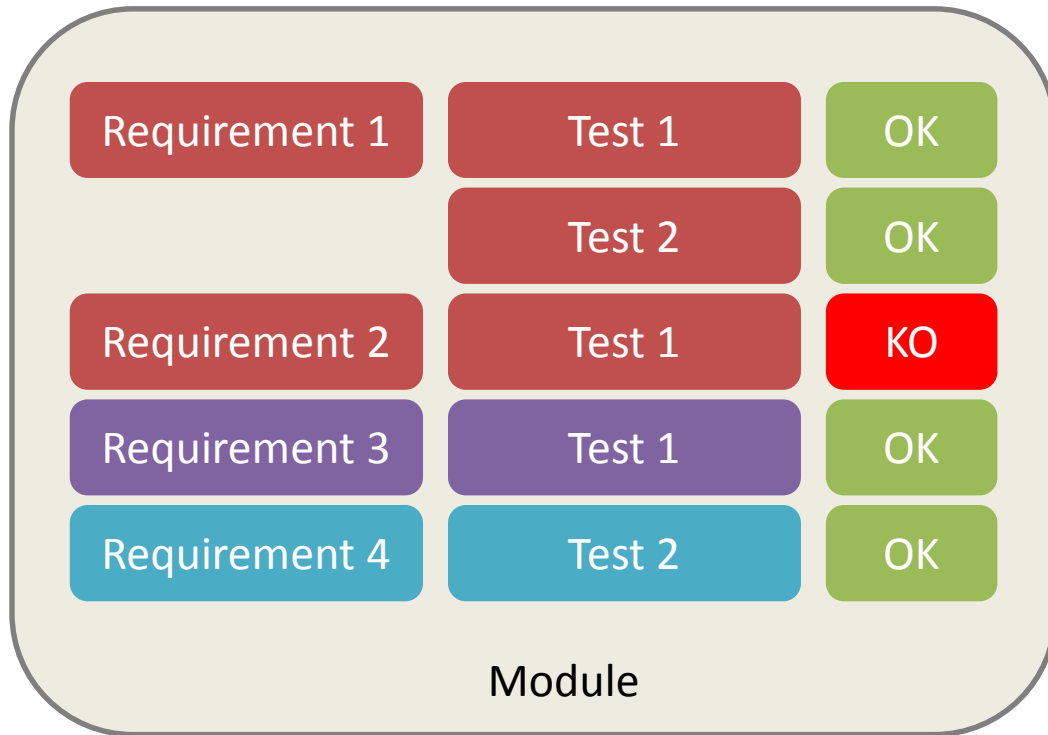


Links between requirements and Continuous Delivery

Values depending on requirements criticality



Module acceptance in the chain



Criticality	Rate	Thres hold
Criticals	66%	100%
Majors	100%	90%
Minors	100%	60%
Accessory	100%	50%



Module rejected

Benefits and concerns



Benefits

- A single centralized tool
- Links between requirements
- Tracability User Stories / Requirements / Tests / Defects
- Existing API to develop a Jenkins plugin to import / export data
- Capacity to develop macros



Concerns

- Too big Granularity : It is not possible to version a specific requirement, but only a document, which is a set of requirements -> We chose to duplicate requirements
- Too many wrong access conflicts proposed by the tool (but evolutions are proposed in SP1 version)

Conclusion

- We deliver several times a week
- A large majority of users want to continue using the tool and sees gains
- Many adaptations was/are necessary to adapt the tool to a Continuous Delivery mode, generating complexity in daily use

QUESTIONS ?

fgrellier@pagesjaunes.fr

@FabGrellier



THANK YOU !

