[Yata] game

A DevOps serious game

Adrien Muller | in

Yoan Thirion | in

agilepartner

Roll the game

Objective: The goal of the game is to aim a maximum number of points by delivering features to production (deployment). We will use wooden bricks to do so.
This game will demonstrate major principles behind DevOps.

Number of participants: 6-10 persons

Equipments:

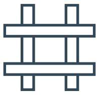
- 1 set of 150-200 bricks/caplas
- 2 pens

Setup:

- Print the pages below
  - ✓ Delivery cards
  - ✓ Score sheets (1 Dev / 1 Ops)
  - ✓ Environments
  - ✓ Kanban board
- Distribute the equipment as described below for each team
- Explain the roles described below

Once the facilitator explained the game:

- Separate physically the Dev and Ops teams
- Put the "Dev" environment on the dev table
- Put the "Pre-production" and "Production" environments on the ops table
- Add the base structure for tower bases on each environment. The base must be as in this image: (base structure represents the environments (DB, Frameworks, Languages, ...))



Sprints: There will be 4 sprints during this game. Each sprint will be organized as described in the sprint image. For each sprint, the facilitator must distribute the cards corresponding (# sprint number).

Roles: Divide the team as described below

	Dev team 	Ops team 	Client
Number of persons	4-6	2-3	1
Environments	Dev / Pre-production	Pre-production / Production	Pre-production
Equipments	Black cards 1 set of 150-200 bricks 1 pen 1 kanban board 1 score sheet	Red cards 1 set of 15-20 bricks 1 pen 1 score sheet	1 Ruler

SPRINT

Before the sprint

The facilitator :

- Distributes dev cards (black) to the dev team
- Distributes ops card (red) to the ops team
- Read the facilitator guid for the sprint

Development [3 min.]

- Develop as many features as they can on the **dev** environment
- Deliver to the **pre-production** environment at the end of the sprint

Client validation

- Validate or not the features delivered on the **pre-prod** environment

Deployment

- Maintain / monitor the environments
- Deploy the result of each sprint **from pre-production to production**
- Measure the deployment time (sec.)

Update score sheets

- Update dev + ops score sheets

Retrospective

The facilitator facilitates the retrospective

- What's happened ?
- What went well ?
- What went wrong ?
- Why
- …

Delivery Cards

#1	Flat	100 pts
A smooth surface without holes.		

#1	Balanced	300 pts
Half of the bricks horizontal and half vertical.		

#1	10 cm	300 pts
The structure is over 10 cm tall.		

#3	Paper	200 pts
The structure can support a piece of paper.		

#3	Design	300 pts
The structure is rotated 45 degrees from the base.		

#3	Single	200 pts
The last floor is a single piece.		

#4	30 cm	400 pts
The structure is over 30 cm tall.		

#4	Tower	100 pts
The last floor is higher as possible single piece.		

#4	Stone	200 pts
The structure can support 1kg.		

#1	Stable	500 pts
Dev has to document how to deploy the structure in prod. No tower is deployed in production unless it is documented.		
#2	Patch	500 pts
Modify the base in pre-prod and prod with a T-shape (2 bricks with one in vertical).		
#3	Collaborate	500 pt
Ops and Dev must collaborate, the structure is deployed in production or loose 1000 pt.		
#4	Automate	1000 pts
Automate the deployments between pre-prod and prod.		

DEV

PRE-PRODUCTION

PRODUCTION

DEV

To do	In progress	Done

Instructions (DEV)

Earn as many points as possible by delivering as many features as you can.

Score sheet

Iteration	Number of cards to pre-prod	Number of cards accepted*	Points	Cumulative points
1				
2				
3				
4				

Formula for points: sum of points accepted, -500 points for any regression (i.e. an "accepted" card would no longer pass).

*accepted/validated by the client in the pre-production environment

Instructions (OPS)

Earn as many points as possible by validating all your challenge cards.

Score sheet

Iteration	Deployment (sec)	Points	Cumulative points
1			
2			
3			
4			

Facilitator guide

Sprint 1. Documentation approach vs collaboration

Build and deliver to pre-production in 3 minutes, Deploy to production in 3 minutes, testing, results, points, (retrospective).

- DEV: 3 minutes for building and deliver to pre-prod
- OPS: Refuse any non-documented tower

Retrospective (key messages)

❑ Stop starting, start finishing / WIP limits
❑ KISS
❑ Production deployment requires **collaboration** and reveals problems.
❑ Silo breaks the collaboration

Sprint 2. Silo again

No collaboration: **it is forbidden to communicate with Ops (by the top management)**

- DEV: Document the delivery + no cards for them
- OPS: T shape base in pre-prod + prod

Retrospective (key messages)

Delivery and deployment in the presence of both teams.

❑ **Opposite objectives** between Dev & Ops
❑ Definition of done ⇔ targeted environment

Sprint 3. Culture of collaboration

Move from siloed delivery to collaboration: everyone in one room

- DEV: construct, starting by taking back the previously tower + cards for Sprint 3
- OPS: facilitates the deployment in pre-prod + prod

Retrospective (key messages)

❑ **Collaboration** makes it possible to deliver
❑ Focused on culture/collaboration
❑ Collaboration saves time

Sprint 4. Automation

- DEV: construct, starting by taking back the previous tower + cards for Sprint 4
- OPS: automate the deployment between pre-prod and prod

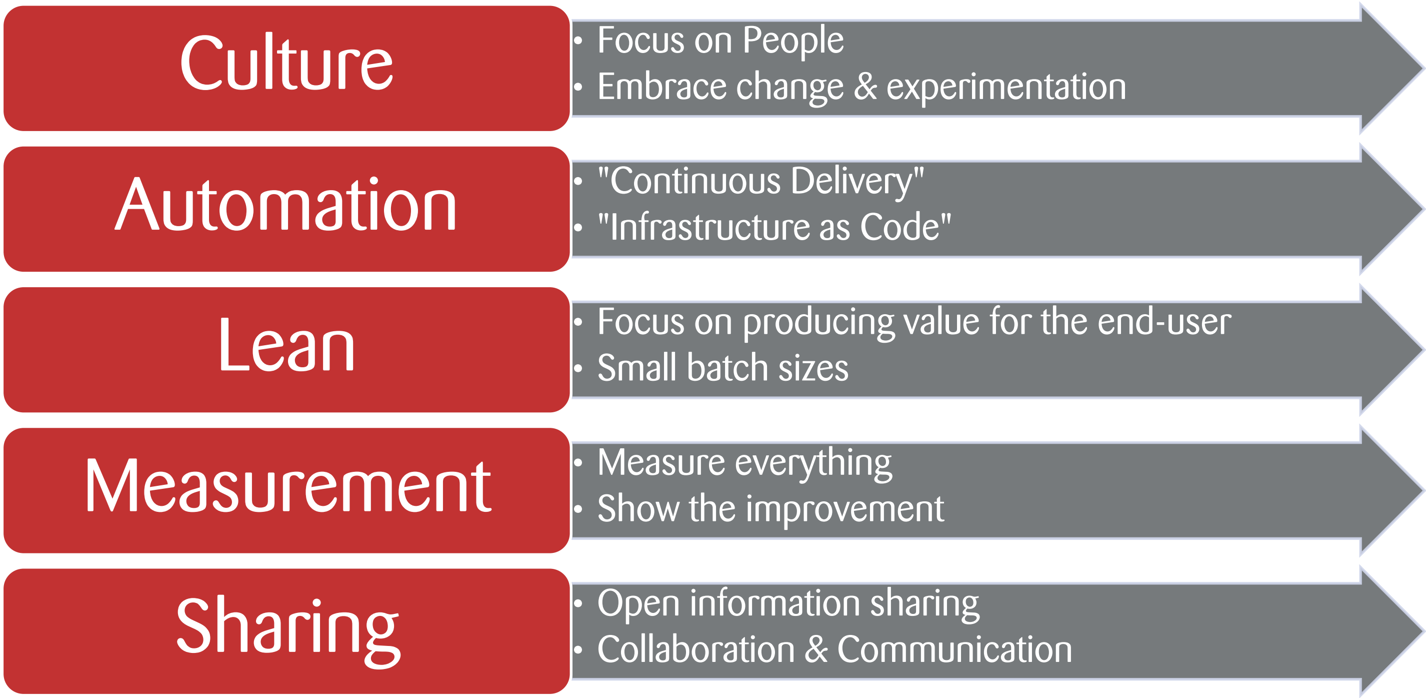
RETROSPECTIVE (KEY MESSAGES)

- ❑ Clone of production (**blue/green deployment**), could be simulated by swapping two pre-production and production post-its
- ❑ Automate deployment and acceptance testing, instead of manually doing it
- ❑ Pre-production environment for practicing before production
- ❑ **Continuous deployment card by card**, ops manager sums the deployment times
- ❑ Automation saves time

<u>Global retrospective and debrief</u>
Review the score sheets + devops product board

<u>Conclusion</u>

- Evolution of the metrics used on the boards, and aggregation into one shared board (performance, process, people, KPI linked to events)
- Be C.A.L.M.S



Culture
- Focus on People
- Embrace change & experimentation

Automation
- "Continuous Delivery"
- "Infrastructure as Code"

Lean
- Focus on producing value for the end-user
- Small batch sizes

Measurement
- Measure everything
- Show the improvement

Sharing
- Open information sharing
- Collaboration & Communication