FORBES DIGITAL COMMERCE

Continuous Evolution in an Agile Development Team

(Gabriel Dinant, April 8th 2014)

WHAT ABOUT ME?

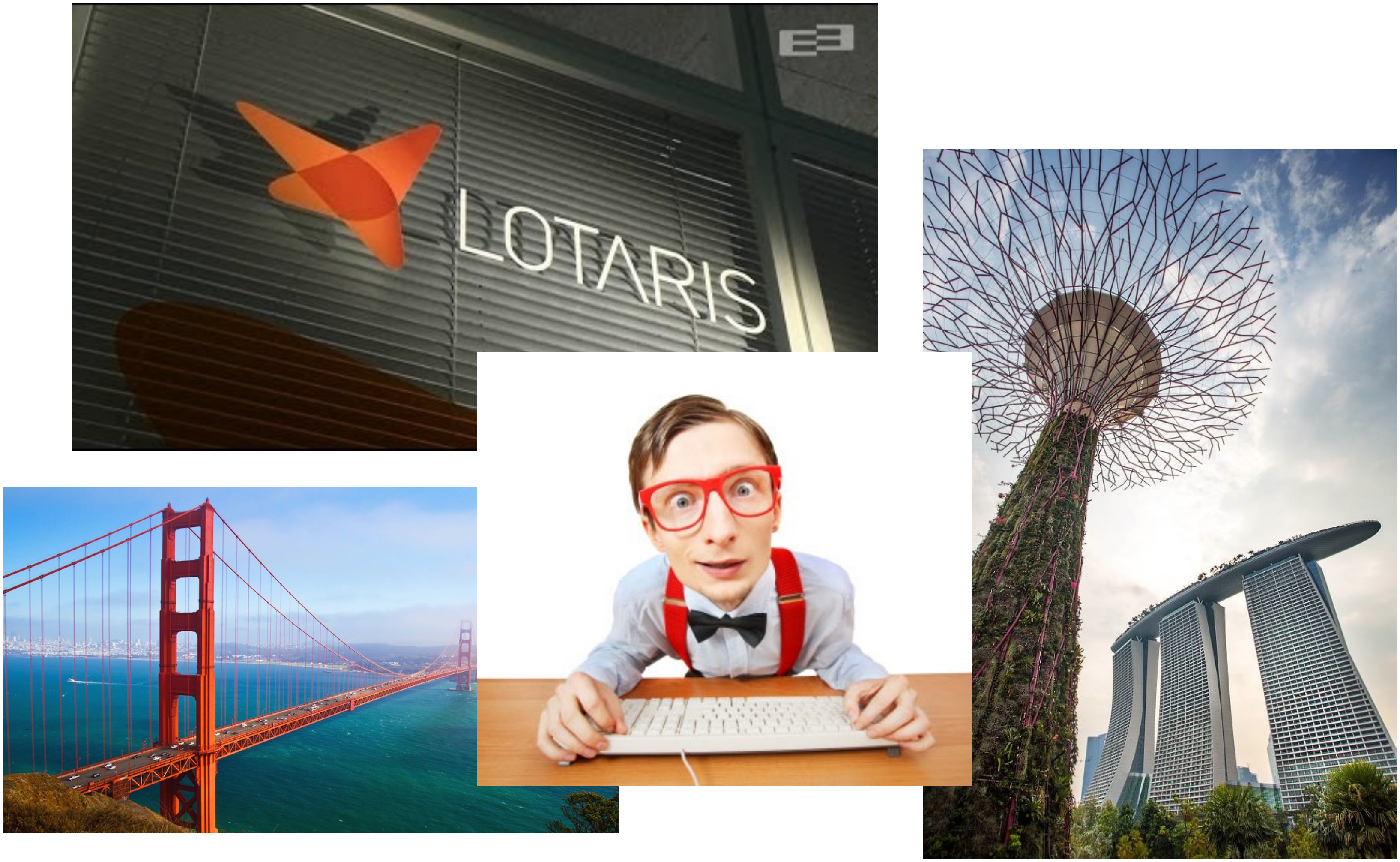
Gabriel Dinant,
27, M, Neuchâtel

Software Engineer & Agile Development Master
@ **F**orbes **D**igital **C**ommerce

Started at Lotaris / FDC in 2011

AGENDA

- **LOTARIS?**
- **AGILE METHODOLOGIES**
  - Quick overview of Scrum, Kanban and Lean
- **AGILE @ LOTARIS**
  - History
  - From Scrum to “our” ScrumBanLean, why?
  - Nowadays
  - Pitfalls from Scrum to Lean
- **AGILE TOOLS – LIVE**


LOTARIS

Lotaris [SA | Pte. Ltd | Inc.]

- **A COMPANY**, founded in 2007
  - ~30 employees
  - Headquarters in Yverdon (Y-Park)
  - Offices in Singapore, San Francisco
- **INTERNATIONAL ENVIRONMENT**
  - Customers spread around the world
  - Engineering in Switzerland / Singapore / Romania
  - Sales in USA / Singapore
- **HIGH EXPECTATIONS**
  - Service Level Agreement (>99.5%)
  - PCI Level 1 compliancy

BEHIND THE SCENES

- **THE STARTUP FLAVOUR**
  - Permanently growing team
  - Permanently evolving products
- **OPERATING THE PLATFORM**
  - Live customers across the world
  - Service Level Agreement (99.5%, focusing 100%)
  - PCI Level 1 compliancy
- **TEAMS**
  - Developers: 17 devs (10 CH, 5 RO, 2 SGP)
  - Quality Agents: 5 (1 CH, 4 RO)
  - Operations: 3 OPS (3 CH)
- **ENVIRONMENTS**
  - QA, Preprod, Prods, DRPs
  - Demo, Ad-hoc servers

SCRUM

KANBAN

LEAN

SCRUM

Most popular Agile Methodology

Getting better day by day

Scrum Team

Simple & Flexible

Scrum Master

Iterative

Product Owner

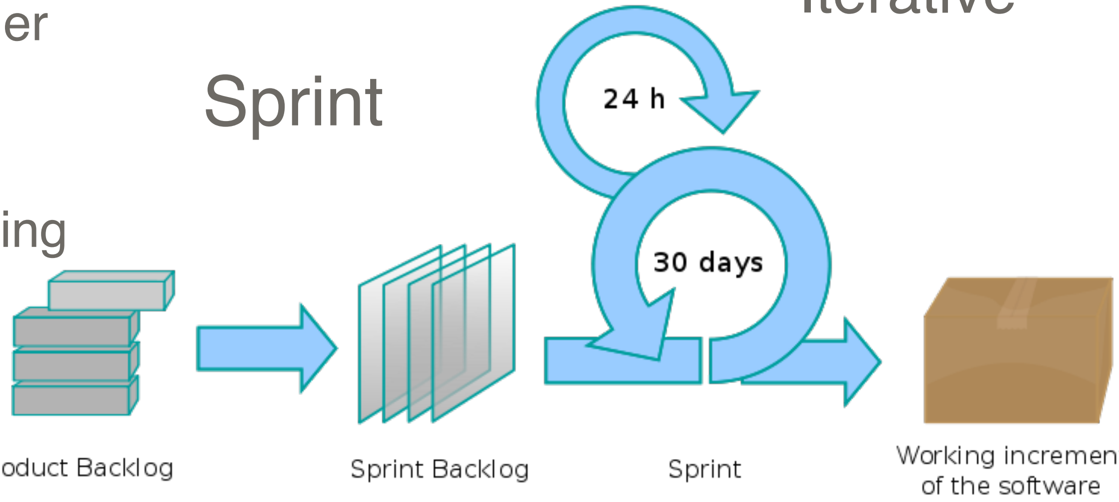
Sprint

Daily Scrum

Sprint Review Meeting

Story Points

Team velocity



KANBAN

Continuous Delivery

Incremental

Just-in-time

Feature lifecycle

Enforce a process

Evolutionary change

Limited pull system

Involve everyone

1 *To Dev.* 4	1 *In Dev.* 4	1 *To QA* 4	1 *In QA* 1	*To Deploy*	*Deployed*
			F1.1		***F1.1***
			F2		
				F3	
	F4.2	***F4.1***			
	F5				

LEAN, a mindset

Decide as late as possible

Empower the team

Deliver as fast as possible

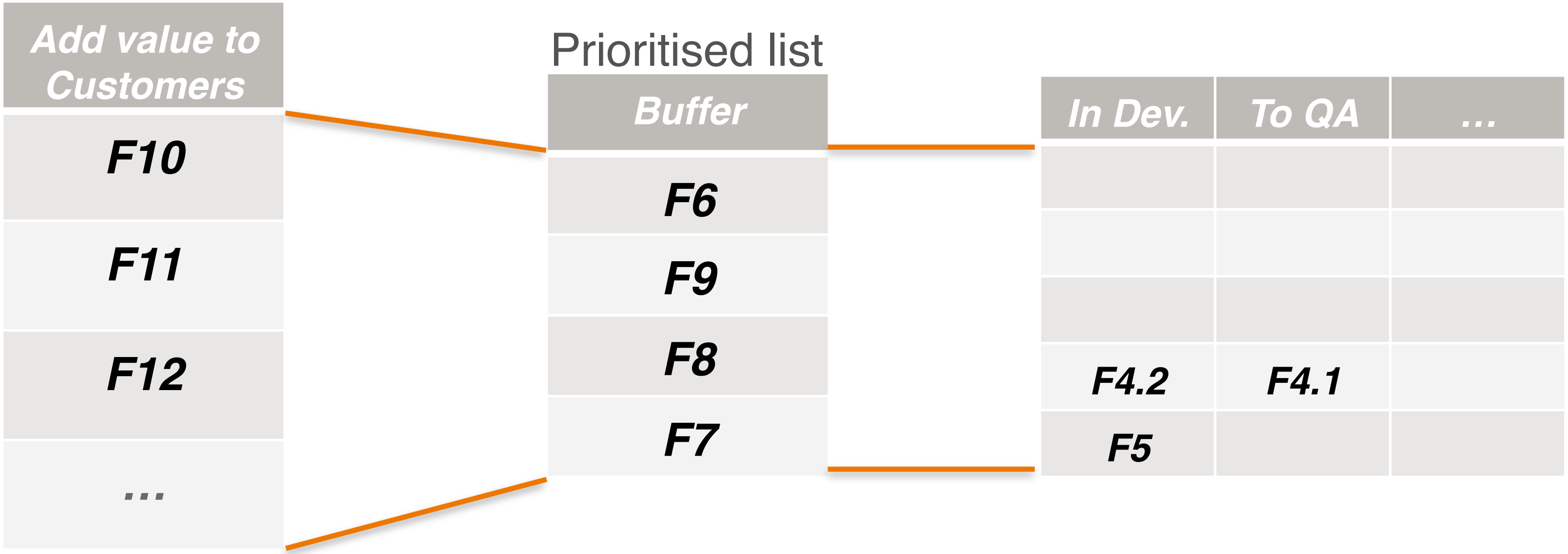
Value stream

"Think big, act small, fail fast; learn rapidly"

Amplify learning

Decide as late as possible

Eliminate waste



Add value to Customers
F10
F11
F12
…

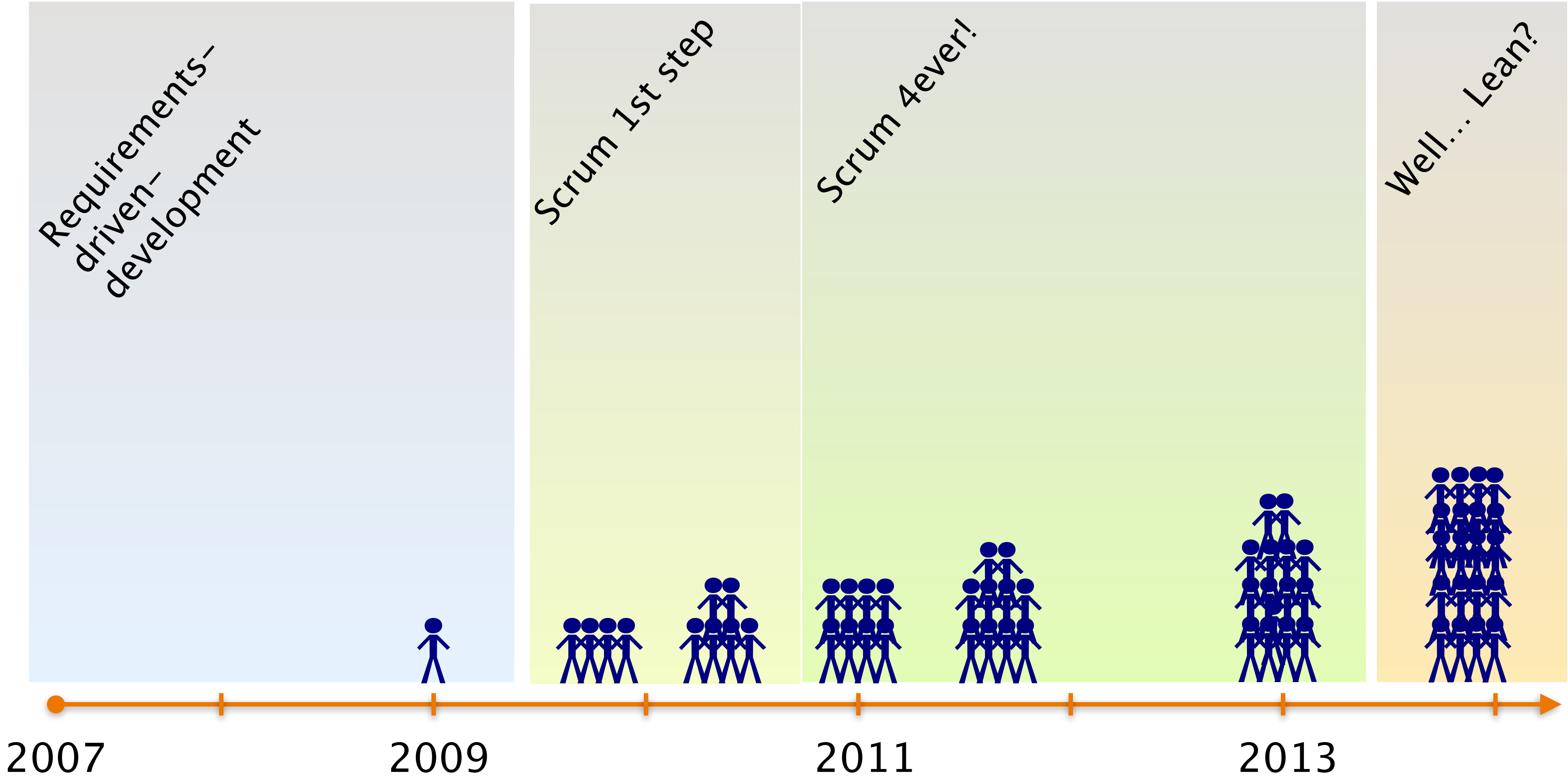
Prioritised list

Buffer
F6
F9
F8
F7

In Dev.	*To QA*	…
F4.2	***F4.1***	
F5		

AGILE DEVELOPMENT @ LOTARIS

AGILE @ LOTARIS: HISTORY



Requirements–driven–development

Scrum 1st step

Scrum 4ever!

Well… Lean?

2007

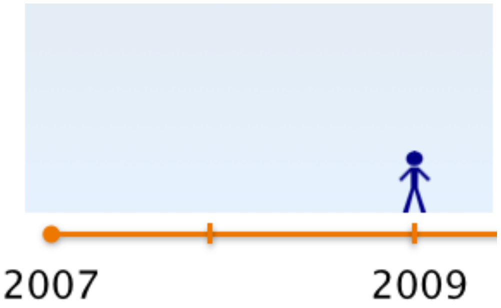
2009

2011

2013

MID 2009

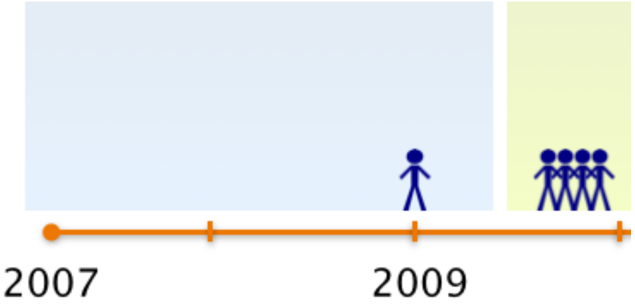
- CONTEXT
  - Delivery team: 1 dev
  - Deadlines: yesterday
  - Requirements coming from the boss
- AGILITY
  - Not needed at that time



Flavor:
"So let me remember… what do I have to do. I have 10 works in progress,
9 of them are high priority, but the 10th is so cool! Anyway, I'm 3 weeks late. The customer can wait a bit…" (A developer)

END OF 2009

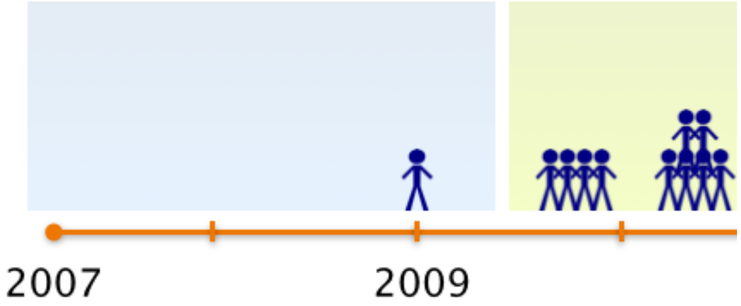
- **CONTEXT**
  - – Delivery team: 4 devs
  - – Deadlines: ASAP
  - – Requirements coming from the CTO
- **AGILITY**
  - – 1st attempt of applying Scrum, using Excel
  - – Task board sticked on the door
  - – Daily scrum meetings lasting between 15 minutes and 3 hours



Flavor:
"Damnit… I forgot to report my remaining work on the door! But who care anyway?" (A developer)

MID 2010

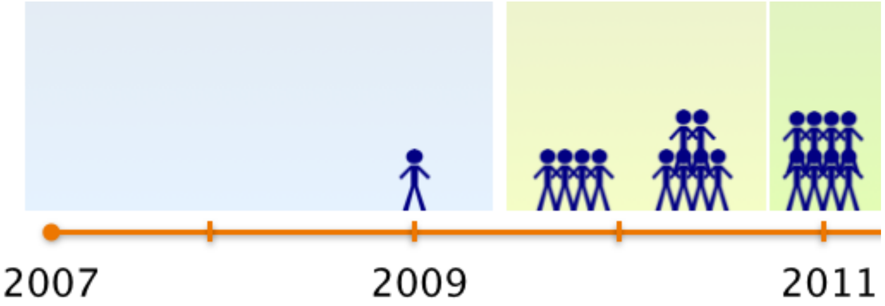
- CONTEXT
  - Delivery team: 6 devs, 1 QA
  - Deadlines: "Do the best you can"
  - Requirements coming from the CTO
- AGILITY
  - Too much pressure, no time to do that…
  - "RT" used as ticketing system
  - Daily Scrum meetings … gone?



Flavor:
"Am I dreaming or we used to have daily meetings? And what is this useless piece of paper sticked on the door?" (A developer)

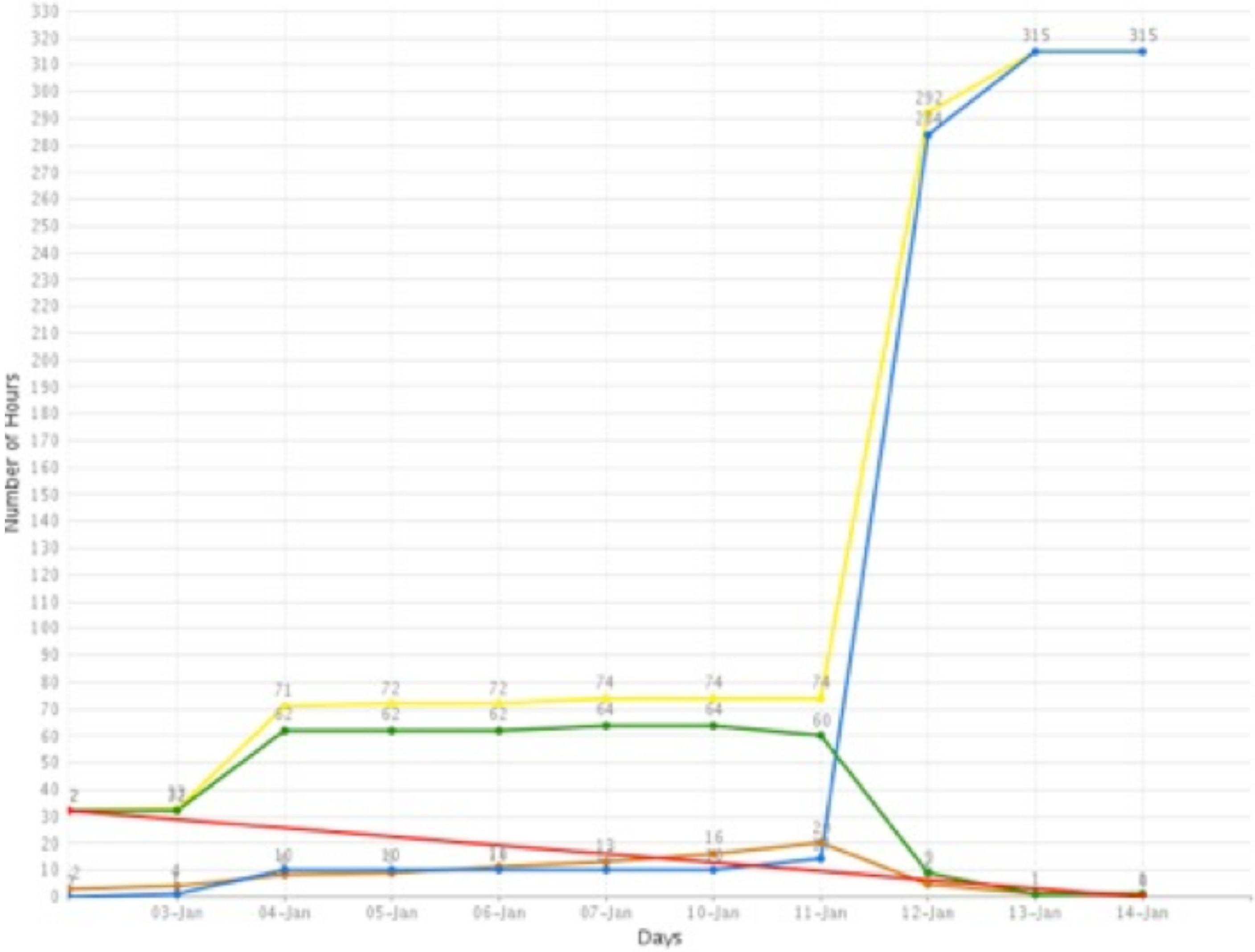
END OF 2010

- CONTEXT
  - Delivery team: 8 devs, 2 QAs
  - Deadlines: “Do the best you can”
  - Requirements coming from the CTO
- AGILITY
  - Daily Scrum meetings,
  - 2 weeks sprints, but weekly deliveries
  - JIRA! (explorative way)



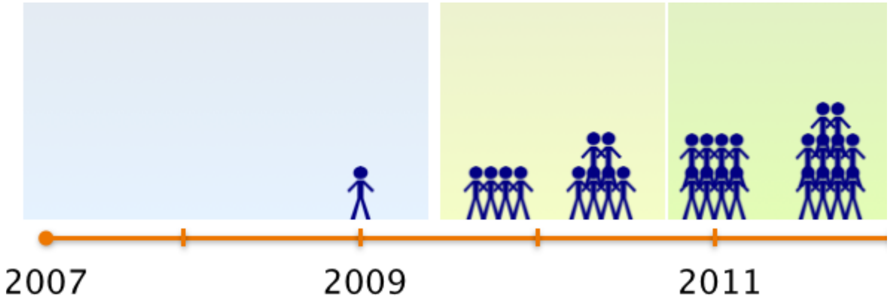
Flavor:
“Wow, JIRA is so cool… RT was too blue. I think I will log again next week to check if something has changed. Cool, I can even drag-and-drop stuff!” (A developer)

END OF 2010


Number of Hours
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
03-Jan
04-Jan
05-Jan
06-Jan
07-Jan
10-Jan
11-Jan
12-Jan
13-Jan
14-Jan
Days
315
315
292
284
71
72
72
74
74
74
62
62
62
64
64
60
2
32
2
4
10
10
16
13
16
20
9
1
0

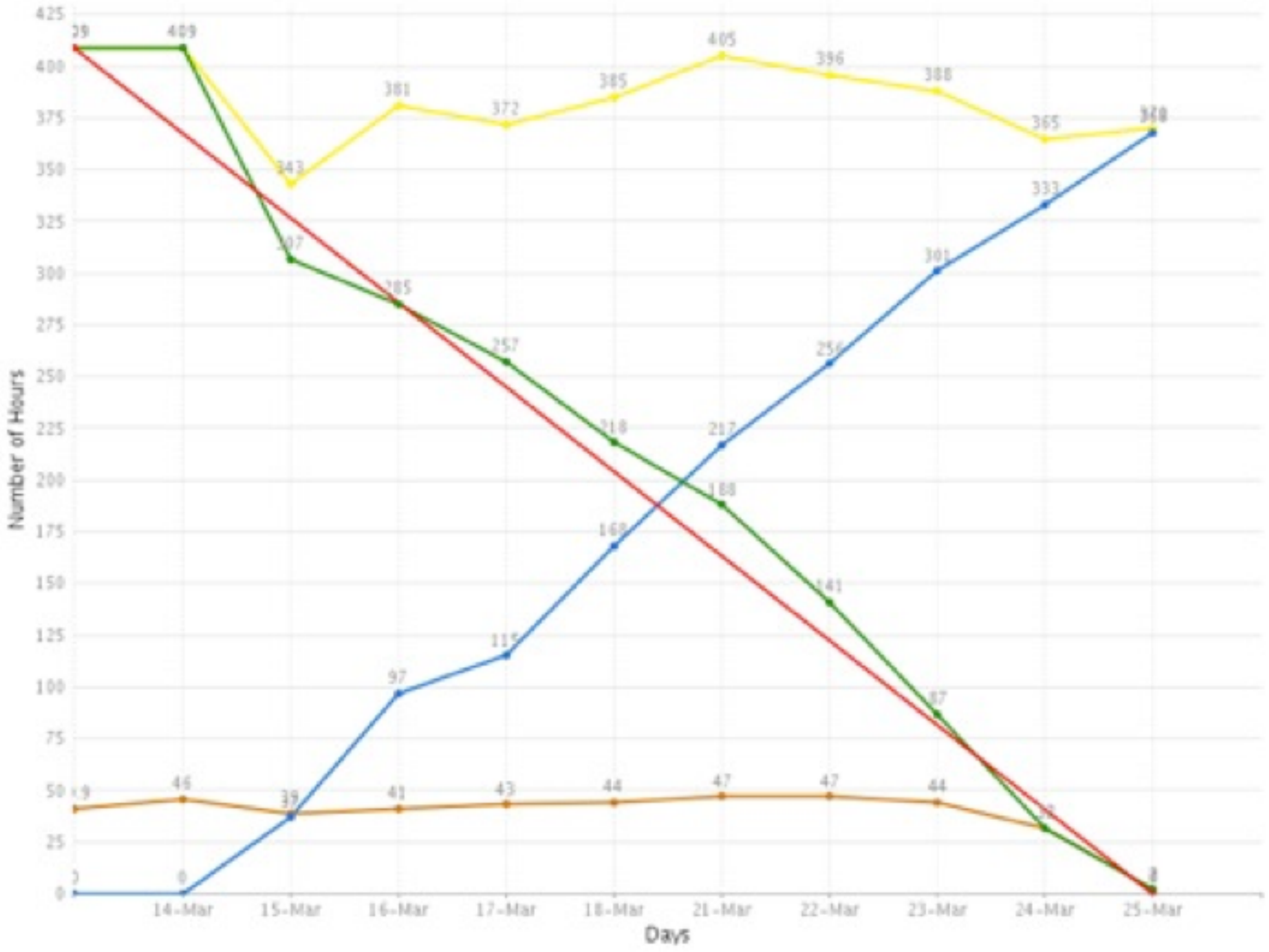
MID 2011

- CONTEXT
  - Delivery team: 10 devs, 4 QAs
  - Deadlines: End of sprint
  - Requirements coming from the product owners & project managers
- AGILITY
  - Daily Scrum meetings & planning meetings
  - Devs understand the basics of JIRA and use it (under pressure)
  - Scrum is adopted by the company and the partners



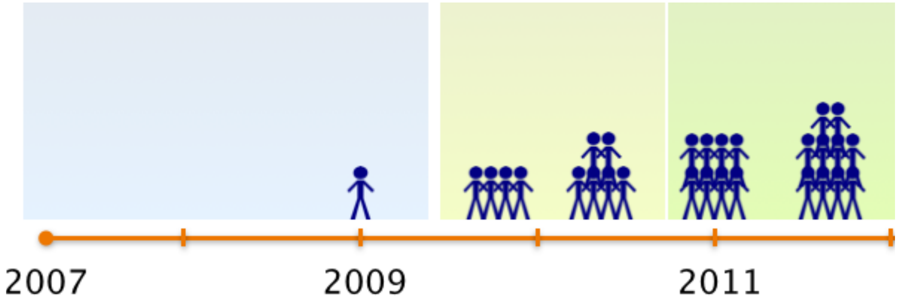
Flavor:
“He is again late for the daily scrum… it looks like he will need to bring croissants tomorrow! Hmm but there is already someone bringing some tomorrow. Arg, we really need a diet…” (The whole team)

MID 2011


Number of Hours
Days
425
400
375
350
325
300
275
250
225
200
175
150
125
100
75
50
25
0
14-Mar
15-Mar
16-Mar
17-Mar
18-Mar
21-Mar
22-Mar
23-Mar
24-Mar
25-Mar
409
405
396
388
381
372
385
365
370
368
343
333
307
301
285
257
256
218
217
188
168
141
115
97
87
46
41
43
44
47
47
44
39
37
0
0
2

END OF 2011

- CONTEXT
  - Delivery team: 10 devs, 5 QAs
  - Deadlines: End of sprint
  - Requirements coming from the product owners & project managers
- AGILITY
  - Introduction of the review meetings
  - Code reviews in pair programming
  - Team starts to be big, people get bored in the meetings

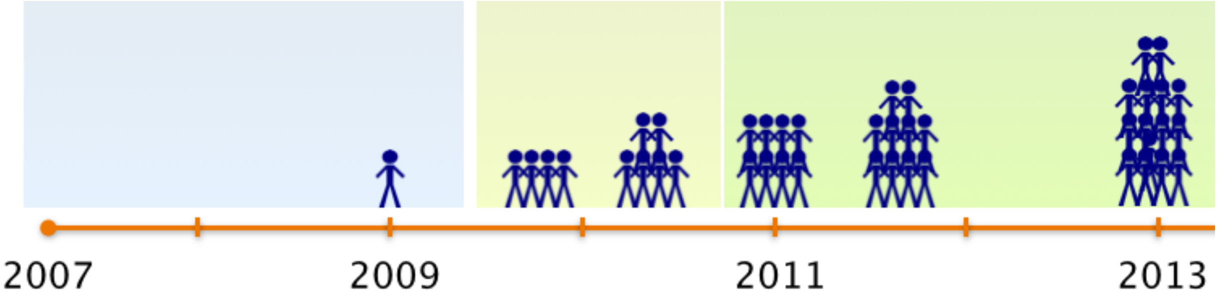


Flavor:
"Ok, the sprint ends in 3 days, we have 45 stories to test and there are 12 bugs waiting to be fixed. So please, do your code reviews and tell us when we can deploy on the QA server." (A QA)

END OF 2012

Flavor:
"- We need all this features, it's very important." (Product Owner)
"- Yes, but we really need to do some maintenance… and we are lacking automated tests. We start to be worry about the platform." (Scrum Master)

- **CONTEXT**
  - Delivery team: 14 devs, 6 QAs
  - Deadlines: End of sprint
  - Requirements coming from the product owners & project managers
- **AGILITY**
  - 3 weeks sprint, we can not do ALL this in 2 weeks…
  - Quality review meetings compliant to non-techies
  - Retrospective meetings
  - More and more issues, well let's do a "consolidation sprint"?

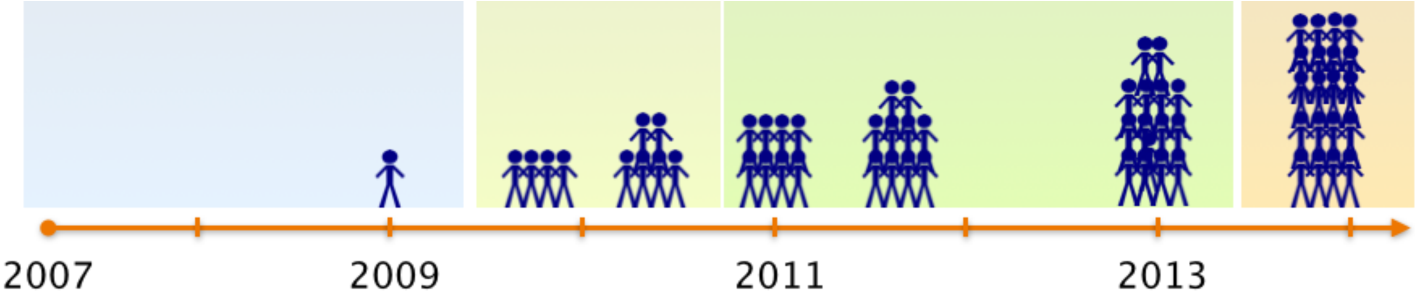


MID OF 2013

Flavor:
"WTF, who did this? Nobody knows how it works? Well we need to deliver on time, so hell that I'll do a quick fix then…" (A developer)

- **CONTEXT**
  - Delivery team: 15 devs, 4 QAs
  - Deadlines: End of sprint(s)
  - Requirements coming from the product owners & project managers
- **AGILITY**
  - Daily Scrum with >15 persons
  - Who made this @#Ç!* feature that isn't working anymore?
  - Deadlines pushed forward

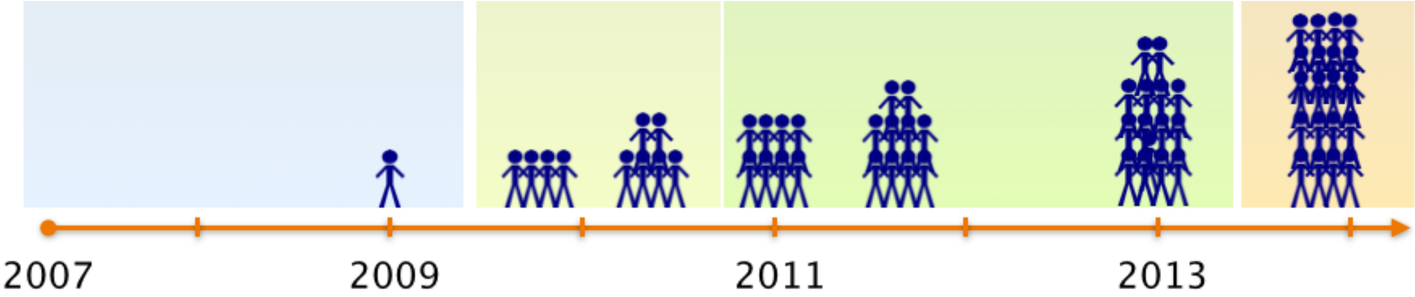
Is Scrum still working?



NOWADAYS

Flavor:
"Let's schedule a design review tomorrow, we will start the implementation in two days and we aim to get this done in 2 weeks. Has a tester reviewed the BDD stories already?"
(A Feature Team)

- **CONTEXT**
  – Delivery team: 17 devs, 6 QAs
  – Project Deadline: We would like this set of features ready for…
  – Feature Deadline: "As fast as possible"
  – Requirements: features adding value to customers
- **AGILITY**
  – ScrumBanLean!
  – BDD
  – Everything **must** be automated tested
  - GIT
  - Enforcement of Reviews
  - Empower the Team



WHY THESE SHIFTS?

- **FROM NOTHING -> SCRUM**
  - Get structured and efficient
  - Deliver short iterations
  - Team velocity, easy way to calculate the duration of a project
  - Time tracking (burndown charts)
  - Communication channel (Scrum Master, Product Owner), …
- **FROM SCRUM -> LEAN**
  - Continuous delivery
  - Quality over quantity!
  - Compromised agility, what was a Sprint already?
  - Maintenance, …

"Ok, how about this motto: 'If you are unhappy for any reason, we will feel really bad'."

NOWADAYS : SCRUM-BAN-LEAN

- **SCRUM MEETINGS**
  - Daily Scrum, Review Meeting, Retrospective
- **WORKFLOW WITH KANBAN**
  - Bring everybody in the workflow
- **LEAN MINDSET**
  - See waste
  - Deliver as fast as possible
  - Quality is important
  - Add customer value
  - …

NOWADAYS : NEW EXCEPTIONS

- **EMPOWER THE TEAMS**
  - Feature Owner / Dev Lead (+team) / QA Lead (+team) / OPS Lead (+team)
- **ENFORCEMENT OF DESIGN REVIEWS**
  - Make it approve by the Team
- **ENFORCEMENT OF (BIG) CODE REVIEWS**
  - Ensure the quality, teach best practices, teach Lotaris conventions
- **ENFORCEMENT OF FEATURE APPROVALS**
- **AUTOMATED TESTS** (100% QUALITY REQUIRED!)
- **BDD** (BEHAVIOUR DRIVEN DEVELOPMENT)
- **REORGANIZATION OF THE COMPANY**
  SOLUTION/ PRODUCT TEAMS

PITFALLS FROM SCRUM TO LEAN

- **THINK DIFFERENT**
  - Make understand that Scrum is not the solution anymore
- **SIZE OF WORK & FOLLOW-UP**
  - Since we don't have deadline based works, what is the best size of a story?
  - How do we handle projects with deadlines?
  - How to measure progress?
- **TEST EVERYTHING**
  - Yes… but it's taking time, and time is money, and… NO, this is not waste!
- **COMMUNICATION**
  - Empower not a person but a Team, communication is key!
  - Crossing Dev / QA / OPS boundaries might be difficult…

IS SUCCESS ONLY ABOUT METHODOLOGIES?

- **A GIVEN METHODOLOGY DOESN’T FIT TO ALL COMPAGNIES**
  IT DEPENDS ON:
  – Client expectations
  – Type of the product
  – State of mind of the employees
  – The way you adapt the methodology to your problems
- **KEEP THE TEAM MOTIVATED** 🔑
  – Use Retrospectives!
  – Share everything!
  – Cross the boundaries of Dev / QA / OPS

JIRA IN ACTION

TOPICS THAT STILL NEED TO BE CLARIFIED

- **WORKFLOW**
  - The current workflow ends at the feature’s release. That is really all?
  - How to manage project plans with JIRA?
- **MEETINGS**
  - Code reviews are taking lots of time, should we use a tool for helping us?
- **DOCUMENTATION**
  - How to keep a documentation useful that doesn't become waste?

DISCUSSION

YOUR TURN