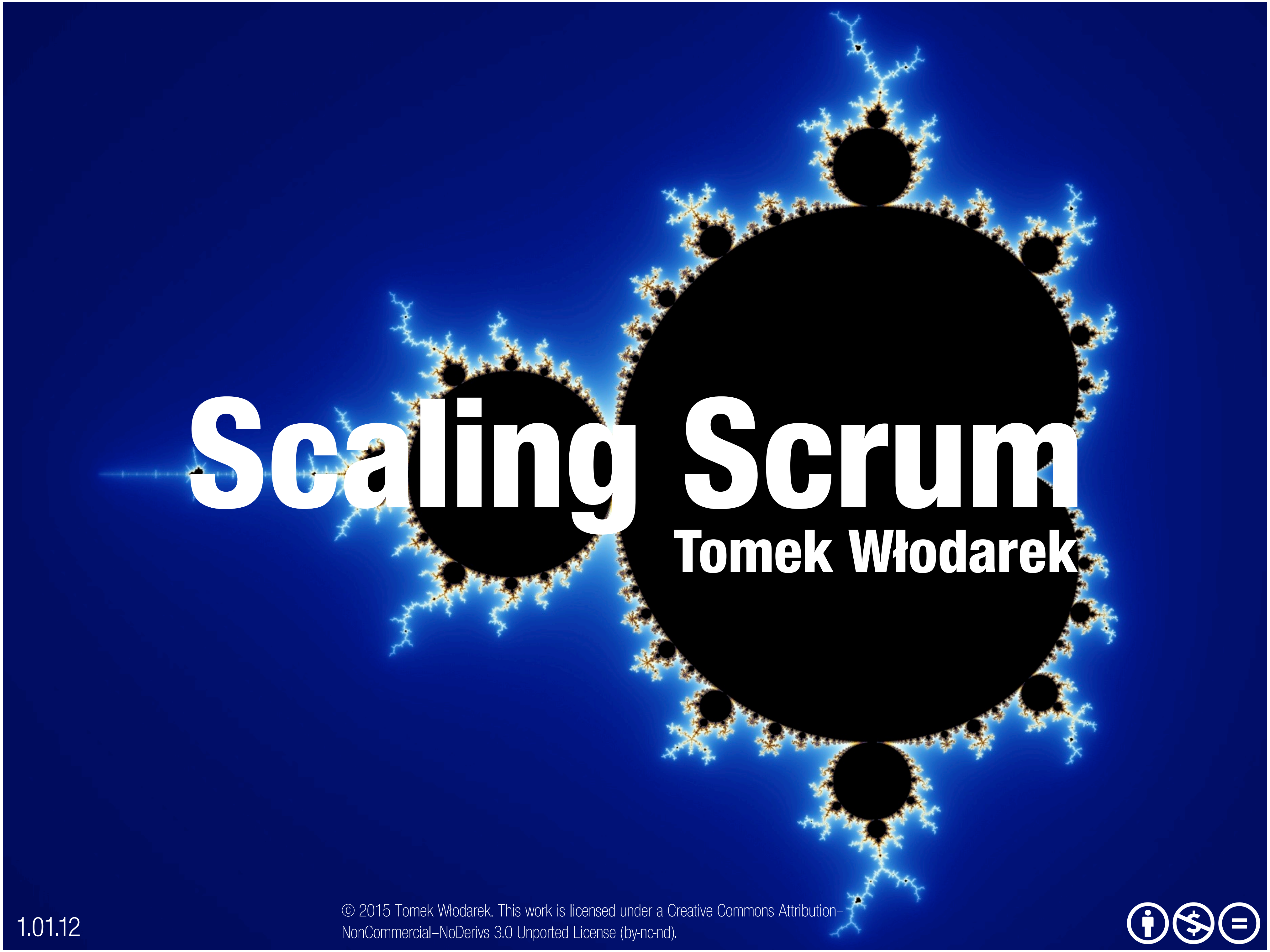
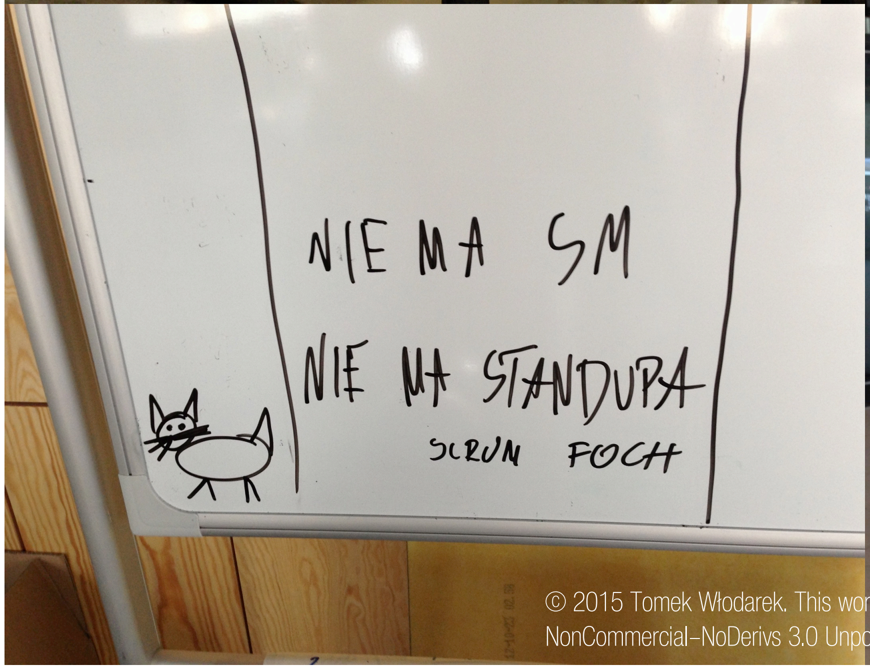


Scaling Scrum

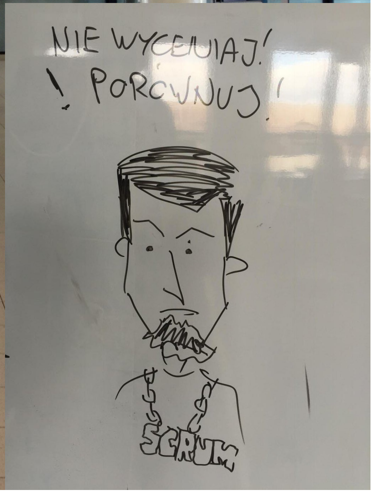
Tomek Włodarek

1.01.12

RELEASE PLAN
DEFINITION OF DONE


NIE MA SM
NIE MA STANDUPA
SCRUM FOCH

Staff only


NIE WYCENIAJ!
! PORÓWNUJ!
SCRUM

Scaling Scrum

> If a problem cannot be solved, enlarge it.
>
> **–Dwight D. Eisenhower**

how long have you been doing Scrum?

more than 5 years

more than 3 years

more than a year

failed or not interested

how long have you
been actually **using** Scrum?

discovery framework

> Scrum is just a simple framework that will identify everything in an organization that gets in the way of optimally building software. Scrum exposes every cultural dysfunction that impedes developing software.
>
> **–K. Schwaber, J. Sutherland**
>
> Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust

Scrum didn’t work out for us,
but we do love the standups.

half-arsed Scrum
half-arsed agile

stam·pede

/stʌmˈpiːd/

1. */masowy pęd/*

A sudden frenzied rush of panic–stricken animals.

2. */uciekać w popłochu/*

To flee in a headlong rush.

scale·pede

/skʌlˈpiːd/

1. */masowy pęd/*

A sudden frenzied rush of panic–stricken companies to scale Scrum because they need more software, now.

2. */uciekać w popłochu/*

To flee in a headlong rush to prescribed methods because scaling Scrum is difficult.

scale·pede

/skʌlˈpiːd/

1. */masowy pęd/*

A sudden frenzied rush of panic–stricken companies to scale Scrum because they need more software, now.

2. */uciekać w popłochu/*

To flee in a headlong rush to ~~prescribed methods~~ **intellectual fads** because scaling Scrum is difficult.

what is scaling Scrum?

organizational transformation towards Scrum?

single team working on several projects?

dispersed teams?

multiple teams working on unrelated products?

multiple teams working on unrelated products using a shared code base?

multiple teams working on a product bundle?

scaling Scrum is...

… any implementation of Scrum where multiple Scrum teams build one product

… any implementation of Scrum where multiple Scrum teams build a product bundle/suite (multiple related products) how many people have you involved?

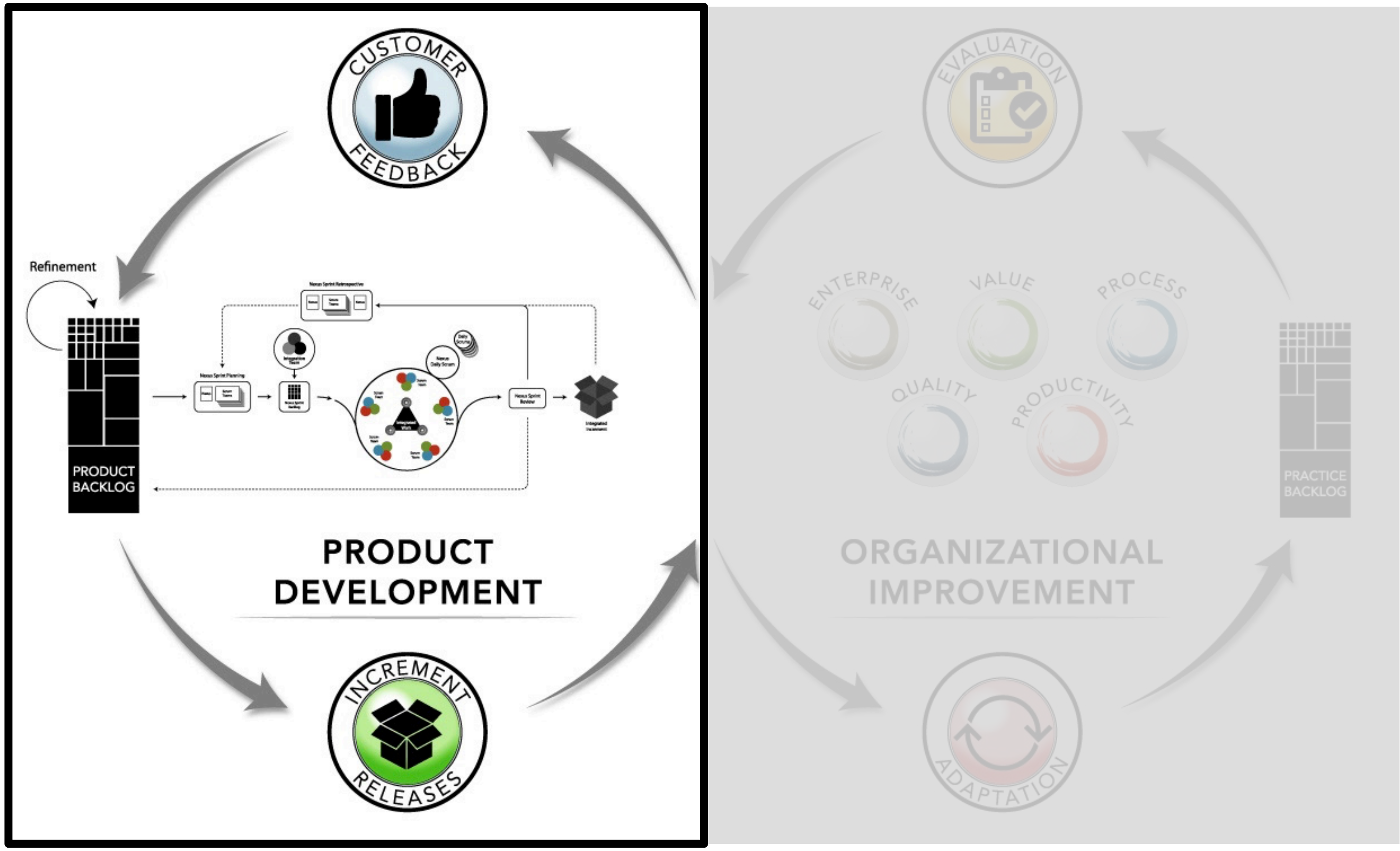
less than 50? (lucky bastards)

50 to 150?

150 to 500?

500 to 1000?

more than a 1000? (OMFG!)


CUSTOMER FEEDBACK
Refinement
PRODUCT BACKLOG
Nexus Sprint Retrospective
Nexus Sprint Planning
Integration Team
Nexus Sprint Backlog
Nexus Daily Scrum
Daily Scrums
Scrum Team
Integrated Work
Nexus Sprint Review
Integrated Increment
PRODUCT DEVELOPMENT
INCREMENT RELEASES
EVALUATION
ENTERPRISE
VALUE
PROCESS
QUALITY
PRODUCTIVITY
PRACTICE BACKLOG
ORGANIZATIONAL IMPROVEMENT
ADAPTATION

why to scale Scrum?

excellent results with single teams, desire to gain similar benefits from larger groups? or

we are so damn inefficient we simply **need more and more people to bloody have the work done**

linearity fallacy

(when it is assumed that a phenomenon functions linearly, overlooking important factors that may produce non-linearity; nothing is linear in the software development world – complex adaptive systems aren't causal) what are the two biggest hurdles when scaling Scrum?

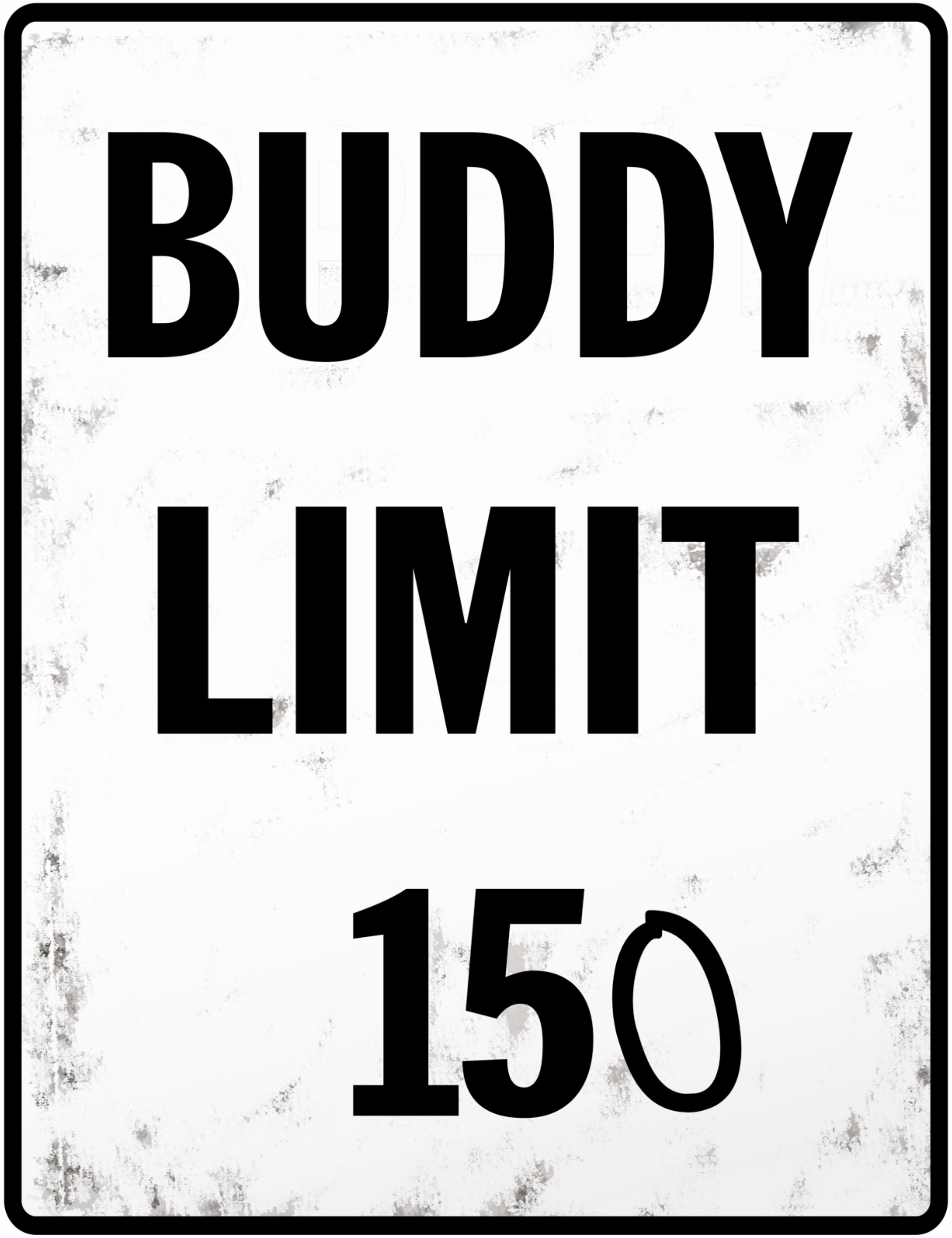
communication dependencies

what causes most scaling efforts fail?

communication dependencies

indivi tions

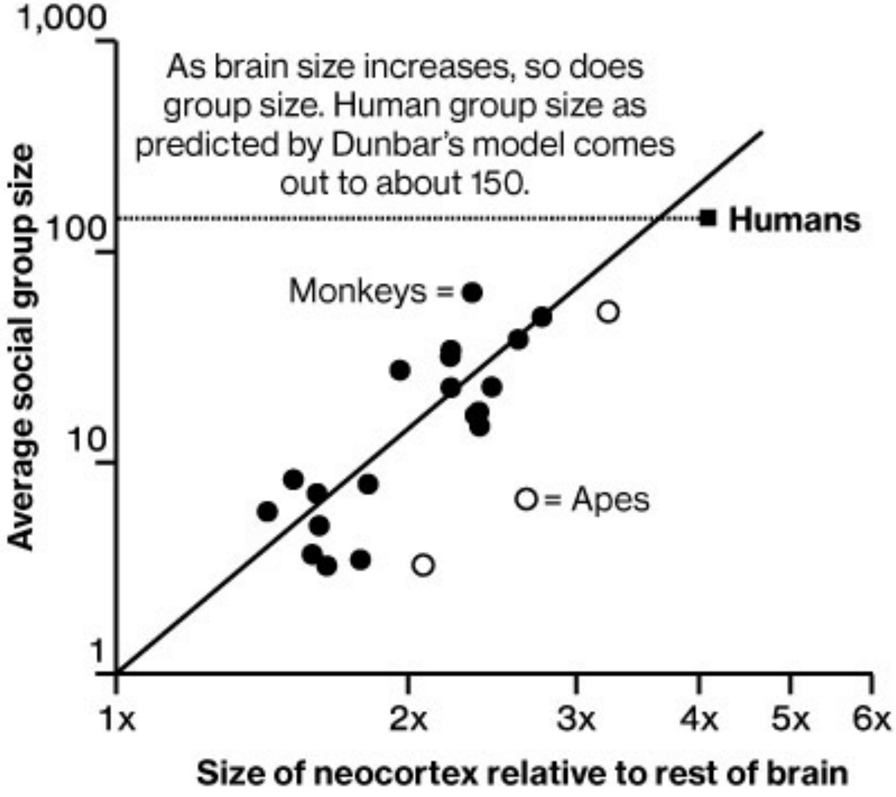
over ools



> [for most of us] it’s a cognitive challenge
> to keep track of more [than 150] people.
>
> **–Robin Dunbar**

https://www.youtube.com/watch?v=ppLFce5uZ3I
https://steveleksonfiles.wordpress.com/2011/08/dunbar-neocortex-size-and-group-size.pdf

The Social Cortex



As brain size increases, so does group size. Human group size as predicted by Dunbar's model comes out to about 150.

DATA: THE SOCIAL BRAIN HYPOTHESIS, DUNBAR 1998

> [for most of us] it's a cognitive challenge
> to keep track of more [than 150] people.
> **–Robin Dunbar**

https://www.youtube.com/watch?v=ppLFce5uZ3I
https://stevelekson.files.wordpress.com/2011/08/dunbar-neocortex-size-and-group-size.pdf

“

Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization’s communication structure.

–Melvin Conway

http://www.melconway.com/research/committees.html
http://hbswk.hbs.edu/item/5894.html

”

(opposite is also true: teams structure tends to lean towards existing system architecture in an attempt to “optimize skillset” and to reach imaginary “100% resource utilization”) dependencies: internal, external, business domains, product lines, PBIs, someone's absence (or presence), relationships, processes, workflows, priviledges, access rights, specialities, competencies, knowledge, source code quality and ownership, code repositories, branches, versions merging, real or perceived module boundaries, frameworks, layers, components, systems, DBs, tools, ...

Dependencies are f*cking evil.

scrumvival.com/20130620

so, how to scale Scrum?

don’t.

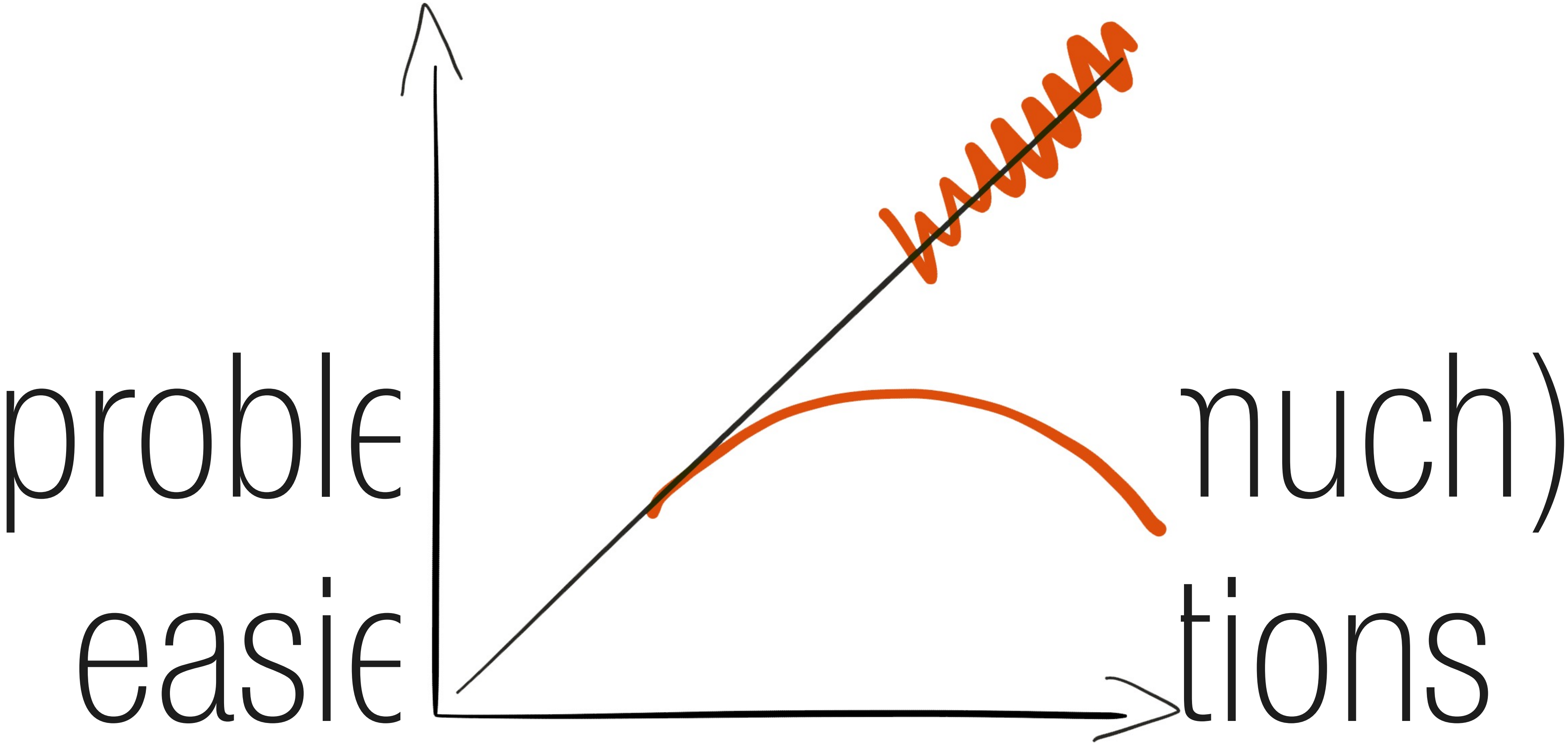
“

Scaling agile methods is the last thing you should do. A better approach is to try to scale down your project.

–Martin Fowler

http://martinfowler.com/bliki/LargeAgileProjects.html

”



(don’t scale up if you struggle with Scrum at a team level; you will end up with even bigger mess)

think of alternatives

> “Before every action ask yourself. Will this bring more monkeys on my back? Will the result of my action be a blessing or a heavy burden?
> **–Alfred Armand Montapert**”

use Scrum to identify and remove impediments at a team and org level

build up teams’ capabilities (teamwork, skills, motivation, relationships, practices and tools)

focus on value, produce less software, scale the scope down

kill multitasking, streamline projects/releases

...

#2

1 Product = 1 Backlog

(and possibly 1 Product Owner to rule them all)

#3

software is invisible

(thank goodness outcomes of running software are visible)

Dependencies are f*cking evil.

scrumvival.com/20130620

enable integration

(think beyond technical e.g. communication, relationships, cultural, environmental) integrate work across all levels **early and often**

build and inspect integrated increments **early and often**

identify and remove dependencies **early and often**

strive for technical excellence (practices, architecture, quality)

#4

don’t share PBIs across teams

(refine, reframe, regroup)

refinement bingo

(don’t split PBIs as it often creates dependencies and/or leads to “waterfallish” approach; **simplify** PBIs instead)

Game rules:

- The whole team plays.
- Pick up a backlog item
- Pick up 3 random numbers from 1 to 12
- Look up the breakers (next slide)
- Player who sees a break up possibility first yells: “Bingo!”
- Player gets 5 points for each new *valuable-usable-feasible* backlog item
- Player gets a penalty of -1 point for each proposal rejected by the team
- Estimate and put the new backlog items to the backlog
- Move on the next backlog item

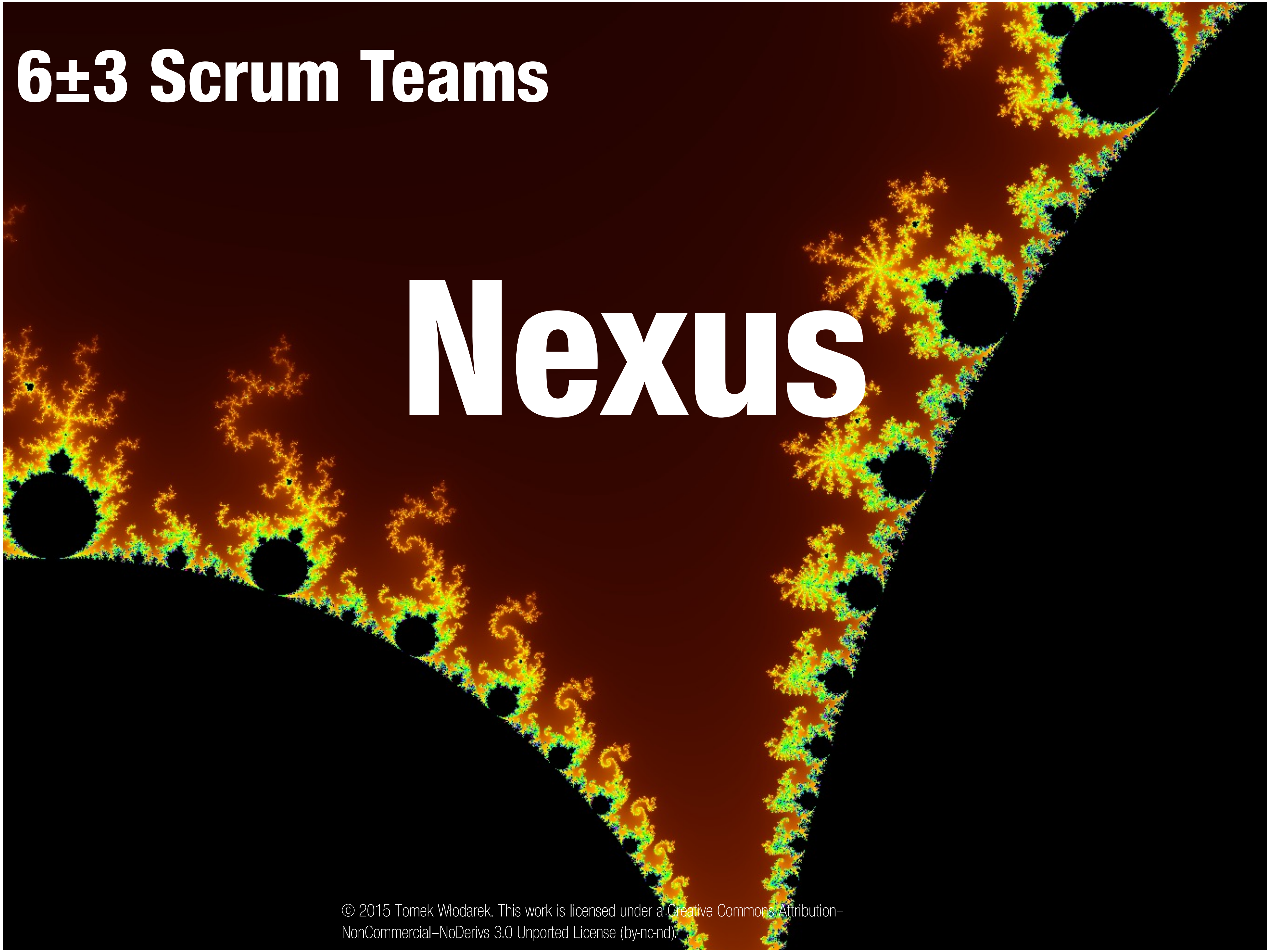
1. **SEPARATE**
2. **SIMPLIFY**
3. **FAKE**
4. **SUBSTITUTE**
5. **REMOVE**
6. **MINIMIZE**
7. **REDUCE**
8. **LIMIT**
9. **DEFER**
10. **DISTINGUISH**
11. **COMBINE**
12. **UNIFY**

1. **DATA**
2. **ERROR**
3. **OUTPUT**
4. **INPUT**
5. **USER**
6. **LOGIC**
7. **WORKFLOW**
8. **PERFORMANCE**
9. **INTERFACE**
10. **DEPENDENCY**
11. **LAYER**
12. **TRANSACTION**

1. **LIMIT(S)**
2. **ROLE(S)**
3. **FORM(S)**
4. **SIZE(S)**
5. **TYPE(S)**
6. **STEP(S)**
7. **NUMBER(S)**
8. **OPERATION(S)**
9. **VARIATION(S)**
10. **VALIDATION(S)**
11. **DETAIL(S)**
12. **FUNCTION(S)**

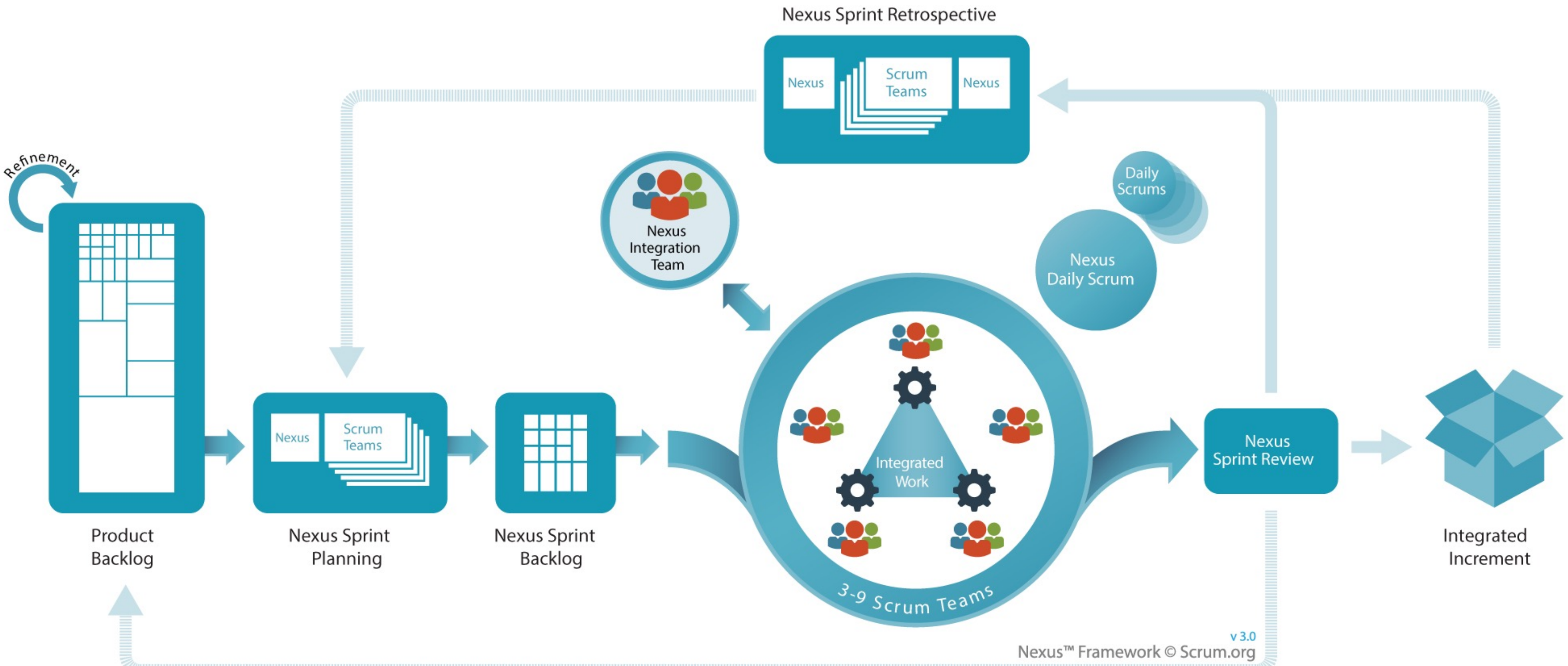
#5

let there be slack



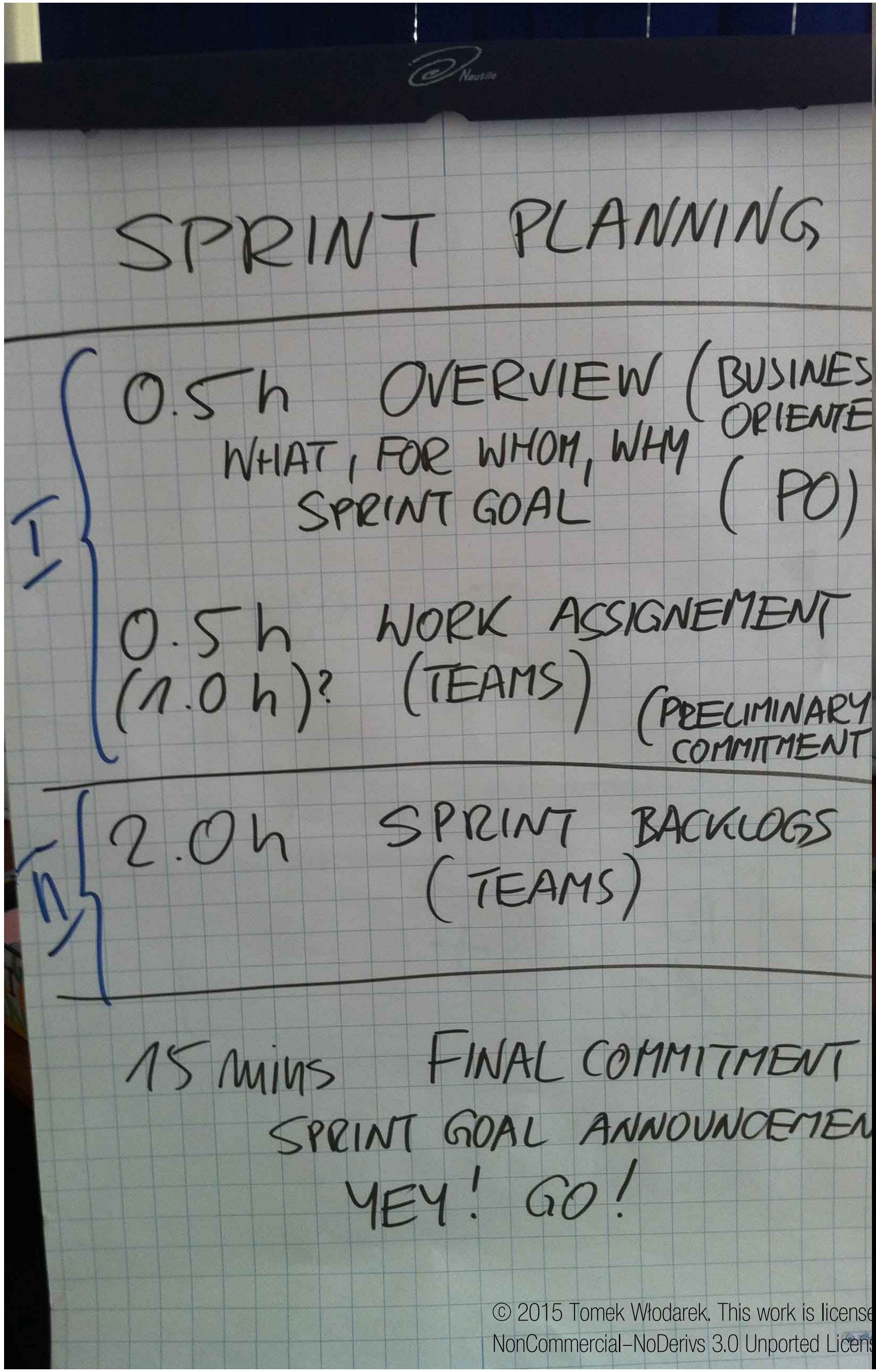
6±3 Scrum Teams

Nexus


Nexus Sprint Retrospective
Nexus
Scrum Teams
Nexus
Refinement
Nexus Integration Team
Daily Scrums
Nexus Daily Scrum
Product Backlog
Nexus
Scrum Teams
Nexus Sprint Planning
Nexus Sprint Backlog
Integrated Work
3-9 Scrum Teams
Nexus Sprint Review
Integrated Increment
v 3.0
Nexus™ Framework © Scrum.org

überscrum

(additive framework for scaling; Scrum foundations – principles and values; opens communication; widens and deepens inspection and adaptation mechanisms; fosters continued transparency; relies on bottom-up intelligence)



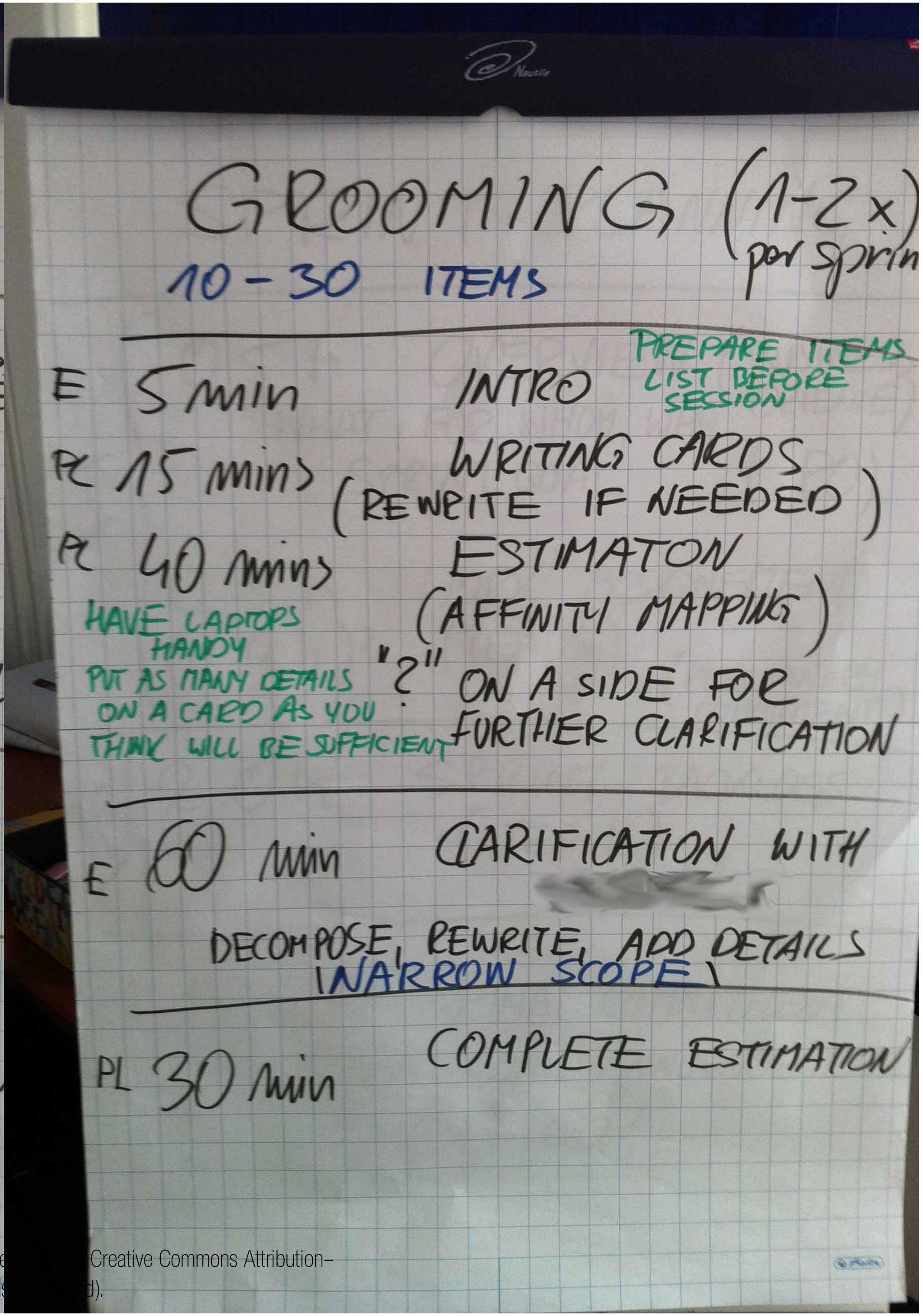
SPRINT PLANNING

I
0.5h OVERVIEW (BUSINES ORIENTE
WHAT, FOR WHOM, WHY
SPRINT GOAL
(PO)

0.5h WORK ASSIGNEMENT
(1.0h)? (TEAMS)
(PRELIMINARY COMMITMENT

II
2.0h SPRINT BACKLOGS
(TEAMS)

15 mins FINAL COMMITMENT
SPRINT GOAL ANNOUNCEMEN
YEY! GO!



GROOMING (1-2x per sprin
10 - 30 ITEMS

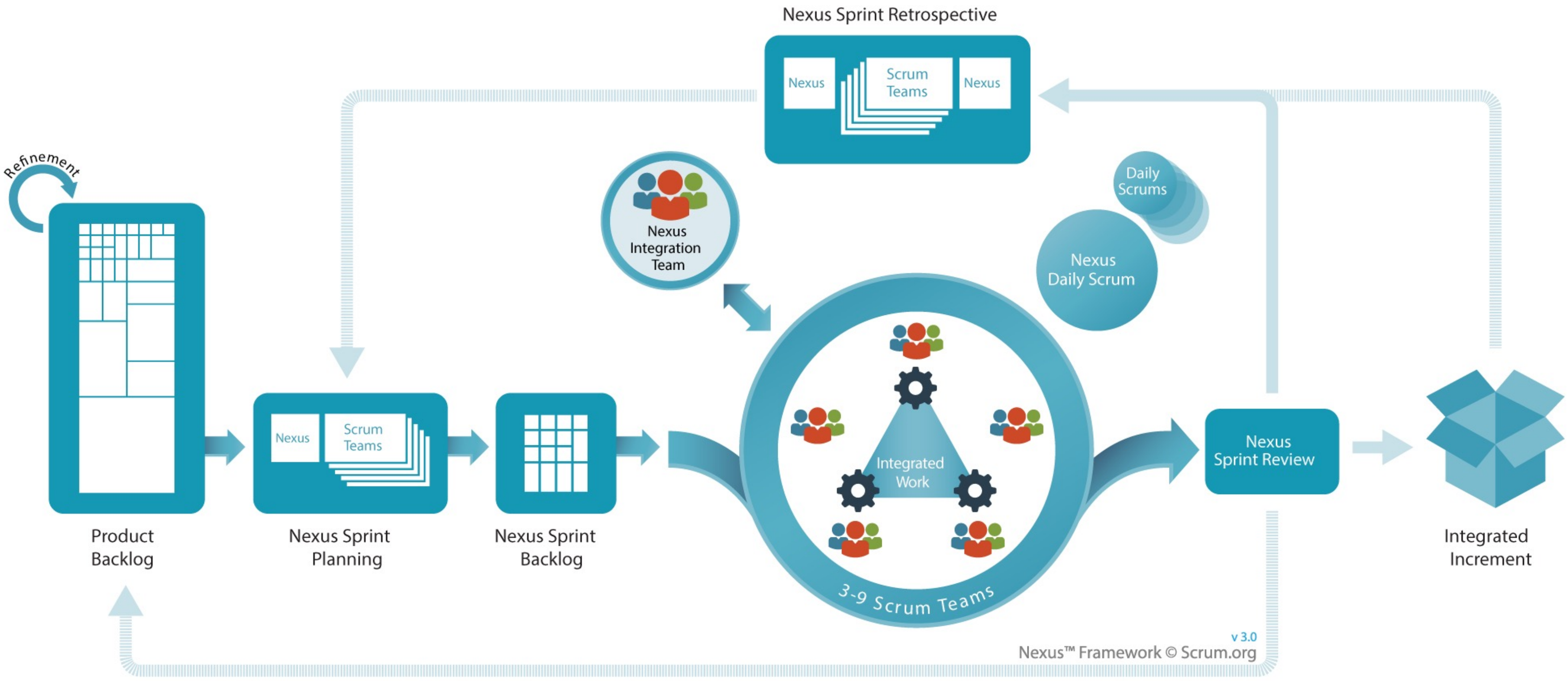
E 5 min INTRO
PREPARE ITEMS LIST BEFORE SESSION

PL 15 mins WRITING CARDS
(REWRITE IF NEEDED)

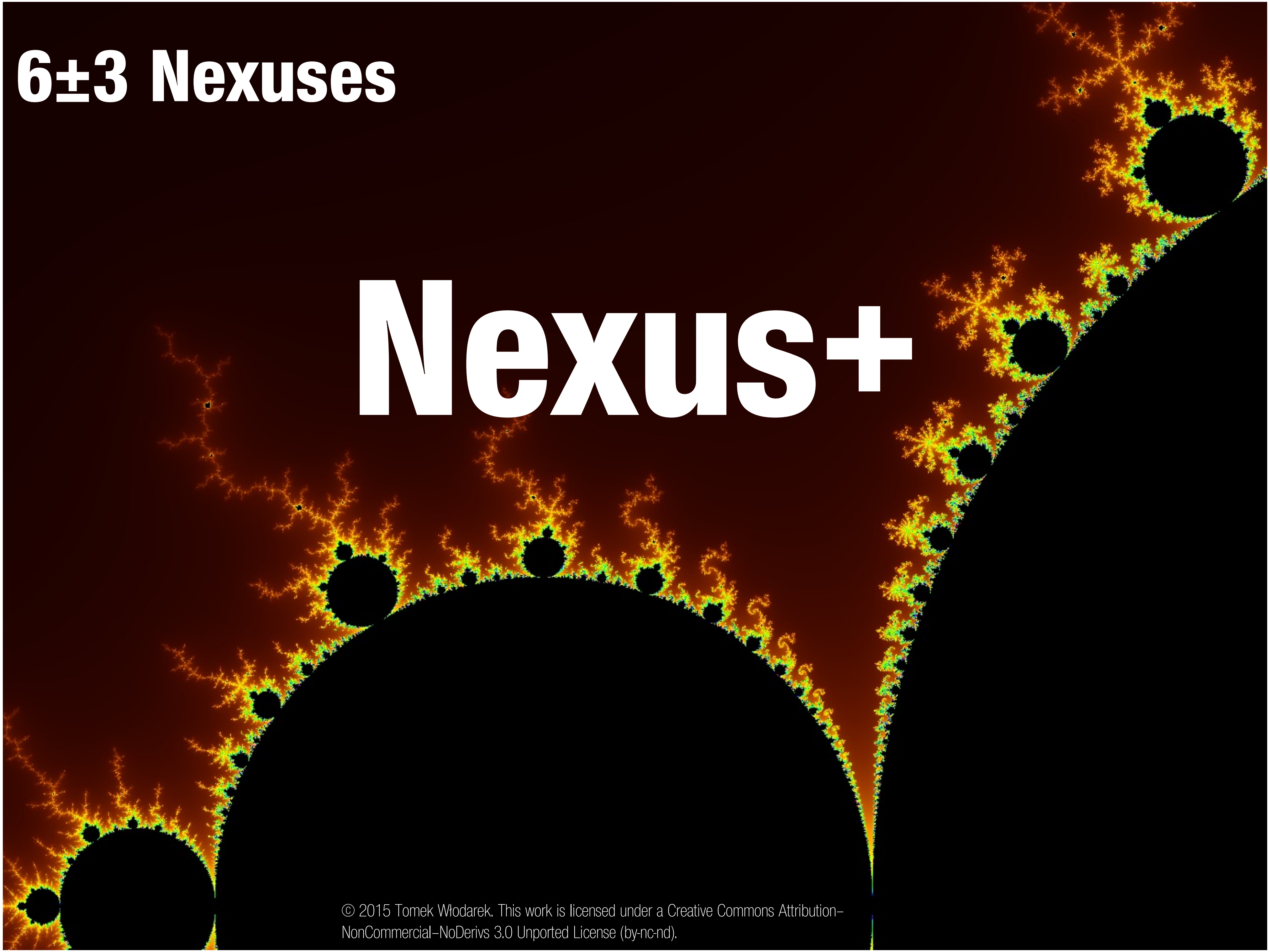
PL 40 mins ESTIMATON
(AFFINITY MAPPING)
HAVE LAPTOPS HANDY
"?" ON A SIDE FOR FURTHER CLARIFICATION
PUT AS MANY DETAILS ON A CARD AS YOU THINK WILL BE SUFFICIENT

E 60 min CLARIFICATION WITH [redacted]
DECOMPOSE, REWRITE, ADD DETAILS
(NARROW SCOPE)

PL 30 min COMPLETE ESTIMATION

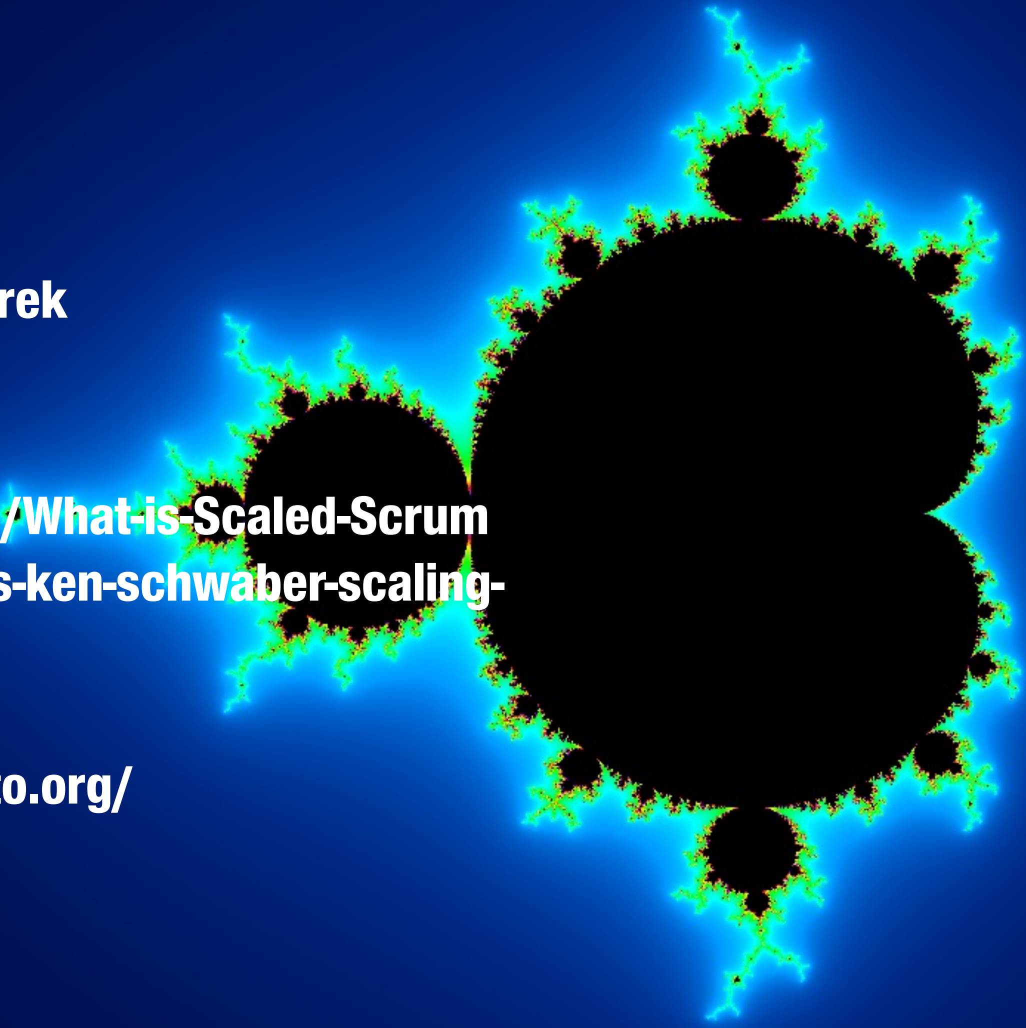

Nexus Sprint Retrospective
Nexus
Scrum Teams
Nexus
Refinement
Nexus Integration Team
Daily Scrums
Nexus Daily Scrum
Nexus
Scrum Teams
Integrated Work
3-9 Scrum Teams
Nexus Sprint Review
Product Backlog
Nexus Sprint Planning
Nexus Sprint Backlog
Integrated Increment
v 3.0
Nexus™ Framework © Scrum.org

6±3 Nexuses



Nexus+

thank you! questions?



tomek@poddrzewem.pl

twitter: @poddrzewem

http://www.linkedin.com/in/wlodarek

http://www.poddrzewem.pl

https://www.scrum.org/Resources/What-is-Scaled-Scrum

http://techbeacon.com/enter-nexus-ken-schwaber-scaling-scrum-future-agile

http://www.scrumvival.com

http://www.halfarsedagilemanifesto.org/