

Agile Project Management for RUN3?

Giulio Eulisse



Foreword

- It's very difficult to keep separate the hype from the useful stuff when talking about Agile project management, thanks to the multi-billion project management training industry behind it
- I do feel ridiculous when using Agile buzzwords as in this presentation...
- ... however, like in all fairy tales, there is some truth behind the buzz and some useful things to be picked up, IMHO
- Some of the material I present actually comes from the (excellent) CERN Course about Agile and Scrum

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

© 2001, the Agile Manifesto authors

12 Agile principles

Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must **work together** daily throughout the project.

Build projects around **motivated individuals**. Give them the environment and support they need, and **trust them to get the job done**.

The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.

Working software is the primary measure of progress.

Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention **to technical excellence** and good design enhances Agility.

Simplicity—the art of maximizing the amount of work not done—is essential.

The best architectures, **requirements, and designs emerge from self-organizing teams**.

At regular intervals, **the team reflects** on how to become more effective, then tunes and adjusts its behavior accordingly.

What Agile is not

Adopting Agile practices does not mean...

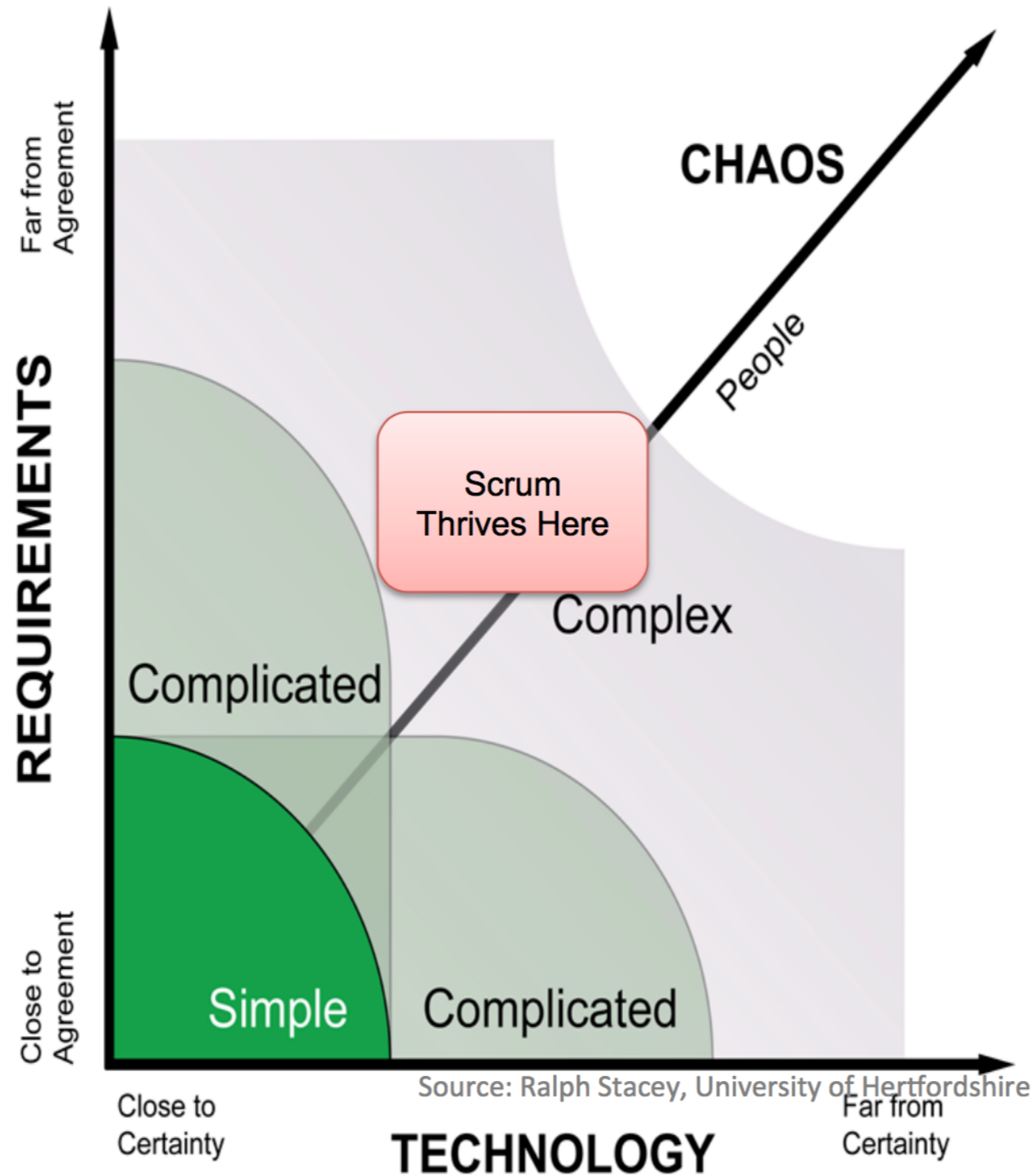
- No longer doing any project management
- Being a chaotic organisation without any governance
- No longer doing documentation
- Leaving the team all by itself
- Doing half-baked work
- Changing everything overnight
- There will be no more problems



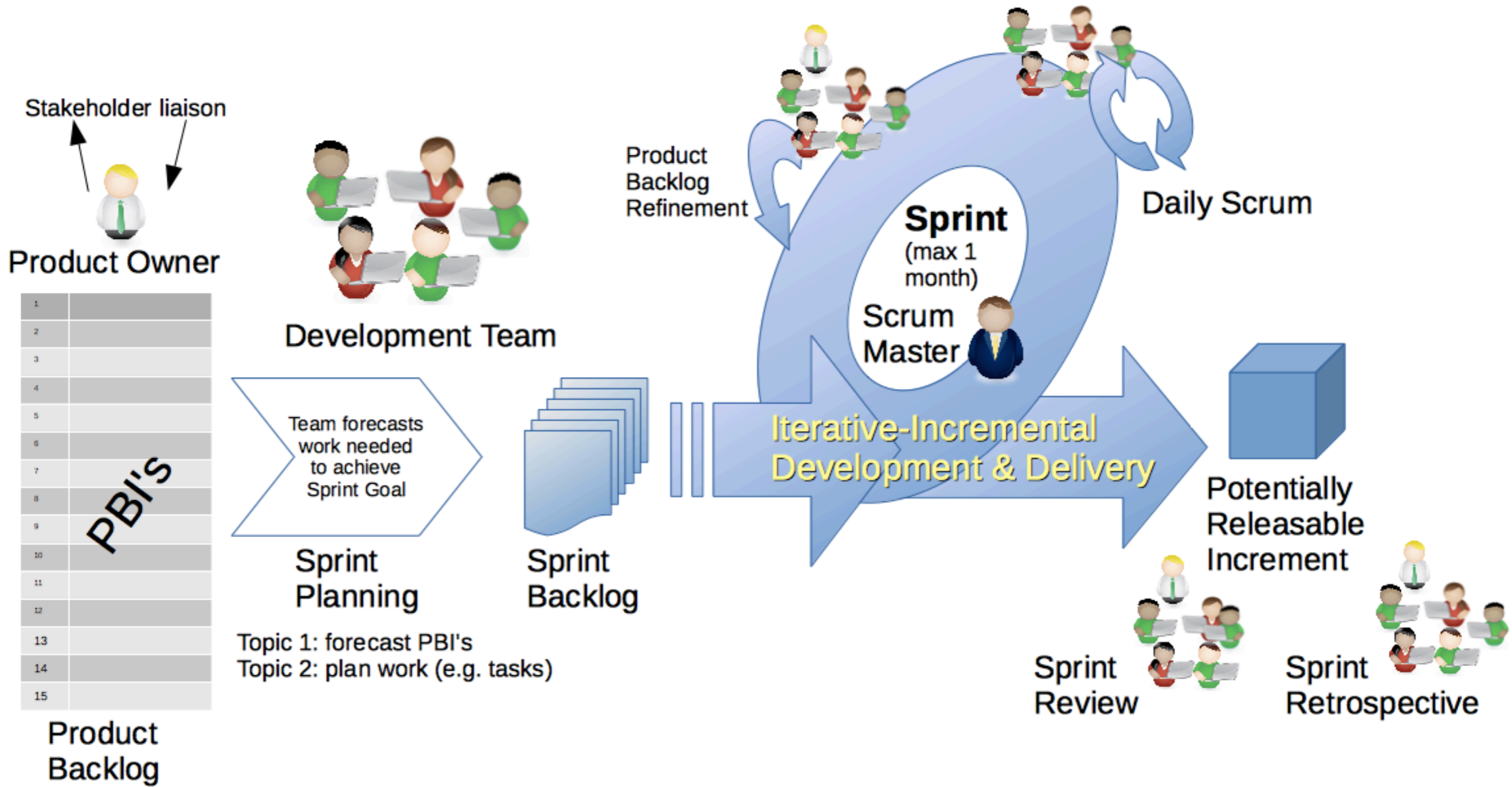
Project Management with Scrum

- **Scrum** is an **iterative and incremental** agile software development framework for managing product development. Dates back to 1995.
- It is widely adopted in the software industry and (for bad and for good) is the way many software projects are managed
- It takes Agile values and implements them in an actual workflow to be followed during the development process
- While there is actually a formal set of rules to be followed, however the idea is that practice should dictate what works for each team (Agile!)

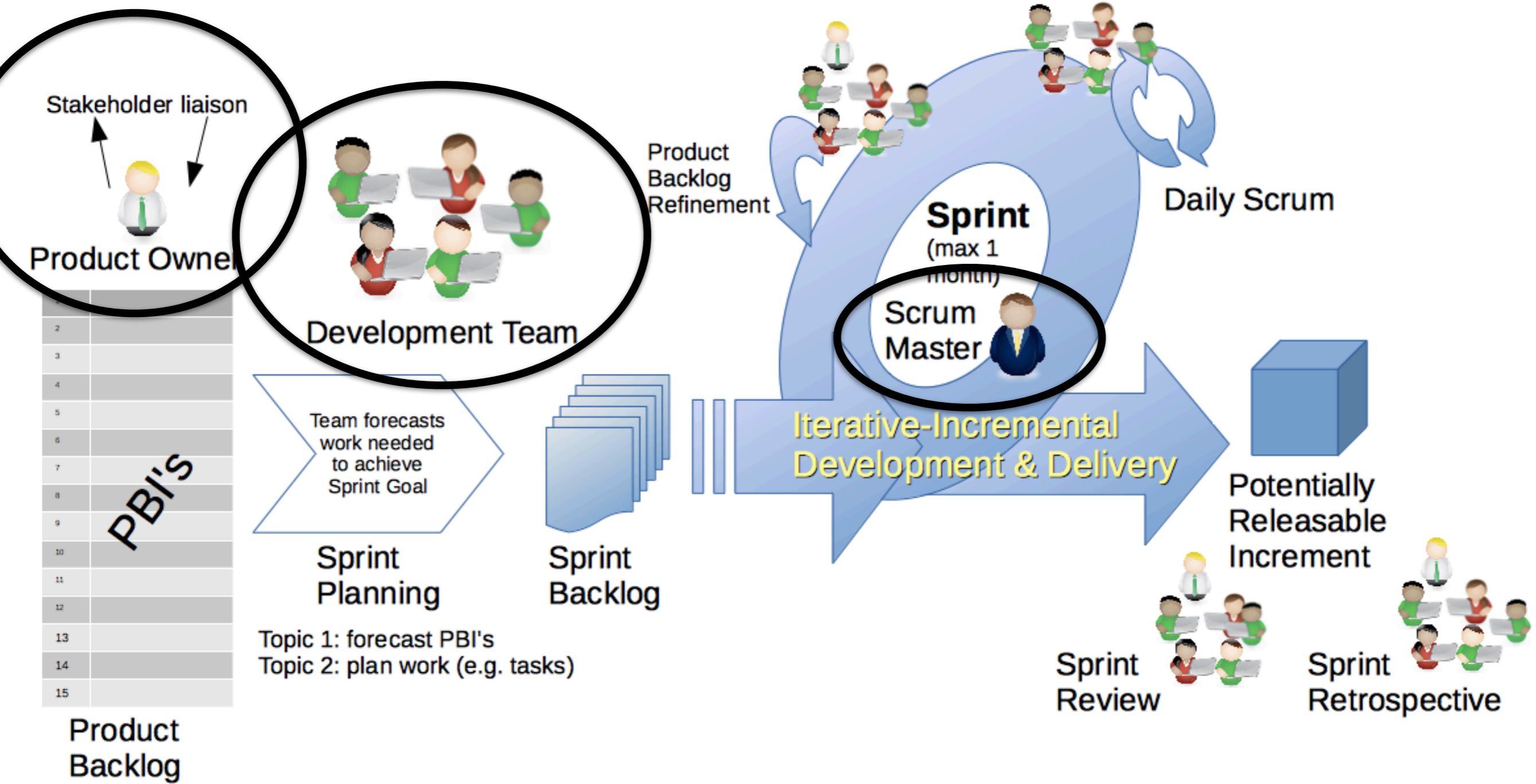
When adopting Scrum



Scrum



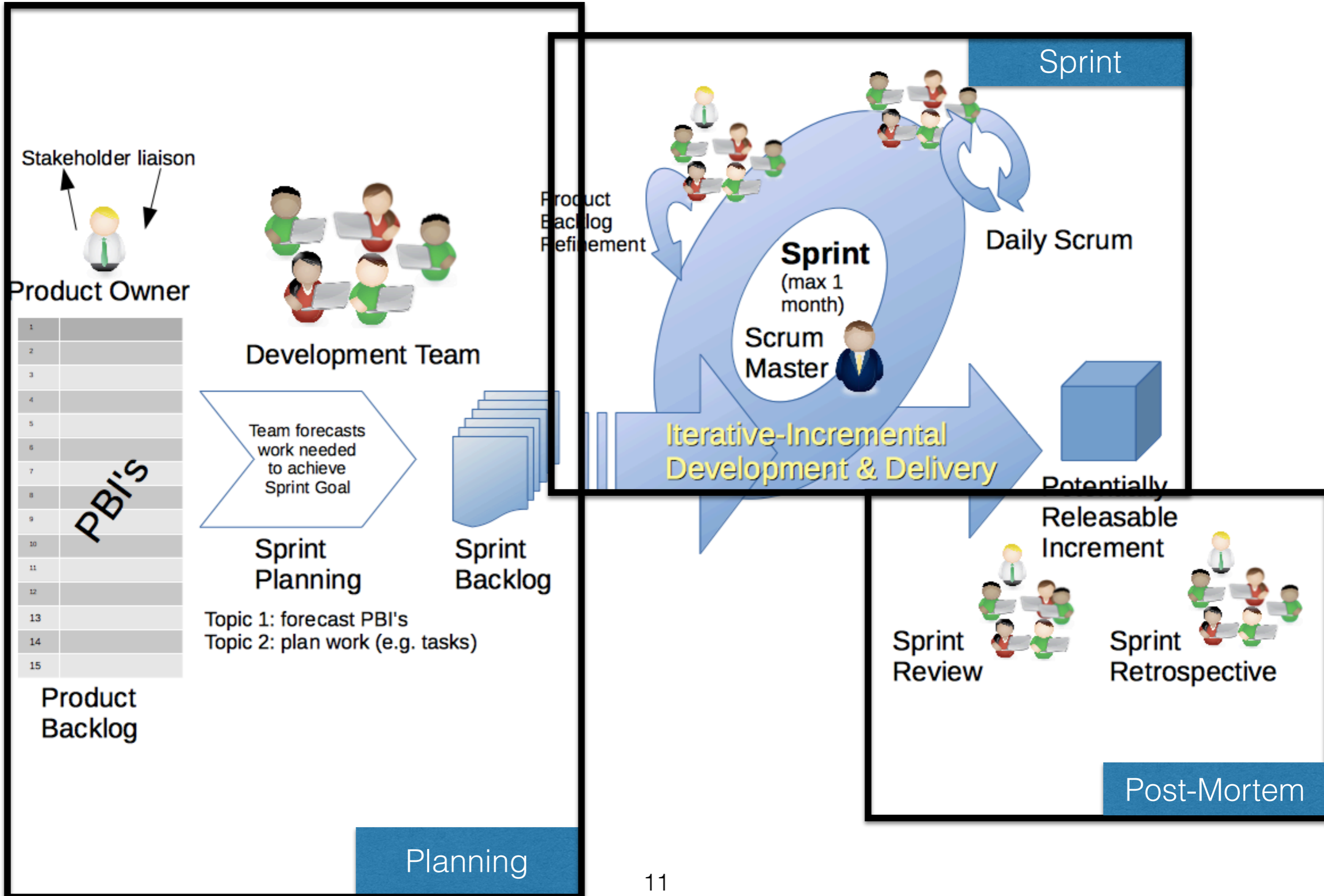
Scrum: roles



Scrum roles

- The ***Development Team***: 3 - 9 people working on the same product. Sets the complexity of deliverables and decides technical aspects of their implementation.
- The ***Product Owner***: the one who mediates the stakeholder feedback and sets the **value** of deliverables. Does not dictate tools or technical choices.
- The ***Stakeholders***: those who will profit / use the product
- The ***Scrum Master***: the guardian of the process orthodoxy. Usually part of the development team, he is responsible to keep the ball rolling and resolve conflicts, but not necessarily a people manager or a project lead.

Scrum workflow



Scrum workflow (preparation)

- Feature requests from the Product Owner or one of the Stakeholders are added to a "**Product Backlog**" most likely in the form of "*User Stories*". *Product Owner* should evaluate the stories and give them a value.
- Product Backlog Items should be deliverable like entities, not implementation details.
- A **Sprint Planning Meeting is held**. The *Team* and the *Product Owner* decide a *Goal* for the *Sprint*, and they refine the *Definition of Done*. Team is ultimately responsible to pick up Product Items from the Product Backlog, split them in actionable changes, and put them in the Sprint Backlog.

Scrum workflow (sprint)

- Sprints are supposed to be **timeboxed** and **short** (from two weeks to one month).
- **The sprint starts.** The team subdivides tasks and holds **daily standup meetings** to monitor progress. Some sort of Sprint Board could be populated with actual tasks to track progress.
- Halfway a sprint there should be a **Product Backlog Refinement meeting** where the Team and the Product Owner decides if something needs to change in the Product Backlog itself or if the sprint needs to be adjusted to match the deadline.

Scrum workflow (post-mortem)

- At the end of the Sprint, a **Sprint Review** meeting is held between the Product Owner, the rest of the Stakeholders, and the development team. The increment is demonstrated and feedback is provided, eventually amending the Product Backlog.
- After collecting feedback on the increment, the Team and the Scrum Master discuss how the actual process went, in a **Sprint Retrospective**.

Being Agile without actually knowing it

The Github transition we just did is a good example of a project managed in a Scrum-like way (without actually formalising the process):

- One person effectively playing the product owner role and mediating the feedback of the other stakeholders
- A developer team effectively taking all the technical decisions
- Iterative approach to the new workflow documentation and the implementation of the various parts of the infrastructure

People complaining about the transition in the first week: 0

What to keep

- Working in **short, well focused, sprints** is a good idea. Having them timeboxed and result in a deployable increment is a must
- Having a "**Product Owner**" like figure helps making sure we do not wander in the wrong direction
- **Trusting developers for technical decisions** rather than micromanagement
- A **formalised process** helps keeping the focus
- **Evolution** is a core part of the Development Process. Sticking to some bad idea simply because a lot of work was invested in a given solution is deadly
- Having to **deploy an actual increment** is a fundamental to keep people transparent about their work

Scaling Scrum

- Major SCRUM criticism is of course that it uses a language, practices, which are ok for a somewhat small people team on a single project.
- One empirical solution is "Keep Scrum at the single-team level, do traditional project management above that".
- LeSS (<http://less.works/less/framework/index.html>) is a rewording of Scrum to scale up to manage large software organisations. LeSS stands for *Large Scale Scrum*.

LeSS BOOK CHAPTER 2:
INTRODUCTION



LEAN THINKING



SYSTEMS THINKING



PRINCIPLES



COACHING



ADOPTION



CONTINUOUS IMPROVEMENT

FEATURE TEAMS



TEAMS



STRUCTURE



ORGANIZATION



COMMUNITIES



CONTINUOUS INTEGRATION

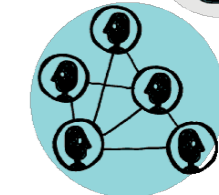
TECHNICAL EXCELLENCE



ARCHITECTURE & DESIGN



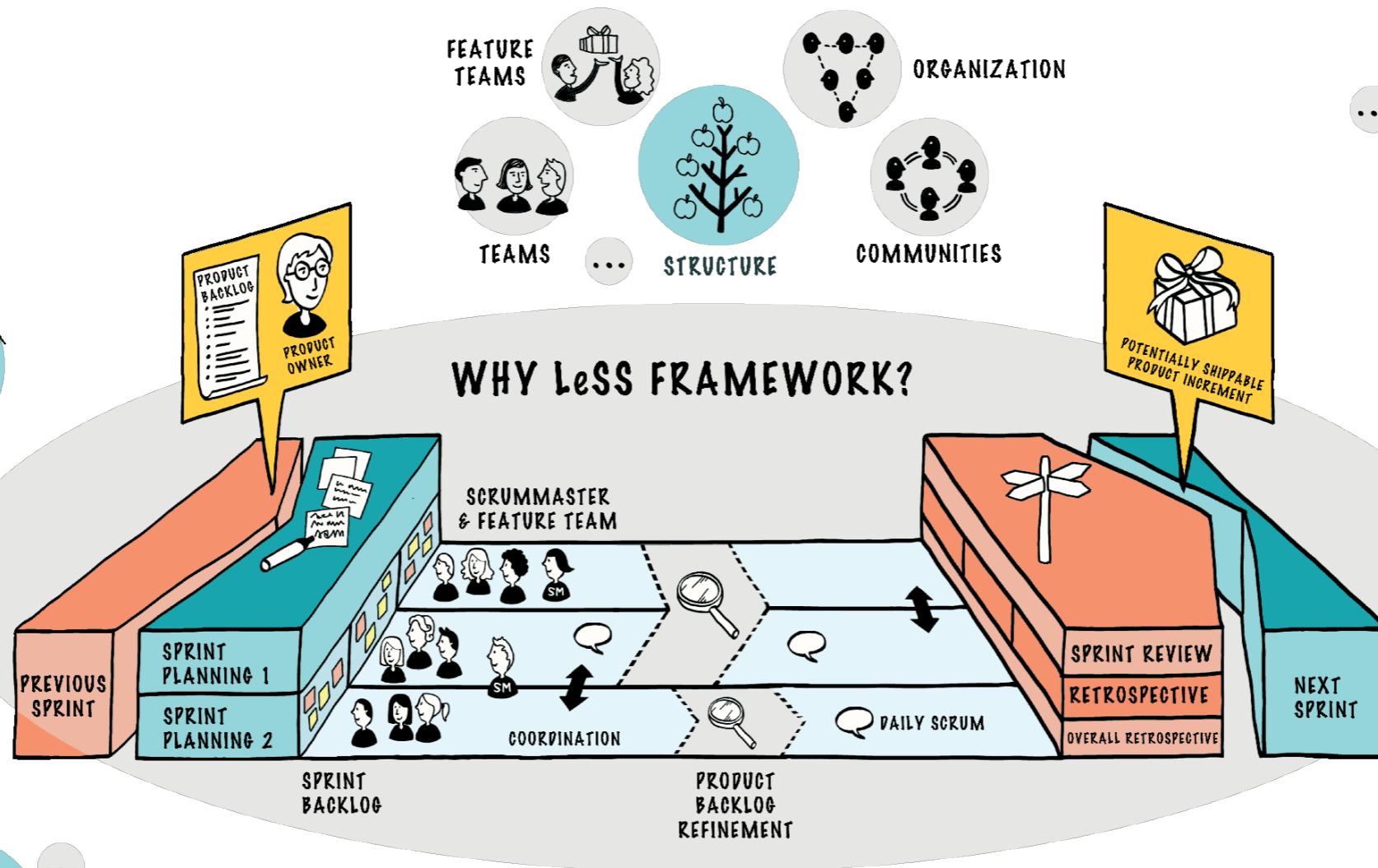
ROLE OF MANAGERS



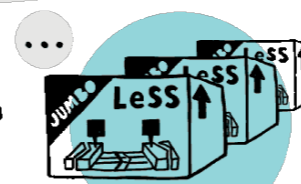
MANAGEMENT



GO SEE



PRODUCT OWNER TEAM



LESS HUGE



REQUIREMENT AREAS

CC BY-NC-ND

<http://less.works>

Adopting Scrum in O2?

Requires a cultural shift, both from management and from developers point of view

Finding the right balance is the difficult part:

- For sure some of the practices are a genuine improvement over some common HEP project planning and development idiosyncrasies
- At the same time one needs to be pragmatic and make sure not to alienate experts
- If some people are happy and delivering on time, why changing their way of working?

Use the Framework project / Test Timeframe effort as a guinea pig for some of the practices (partially already happening)?