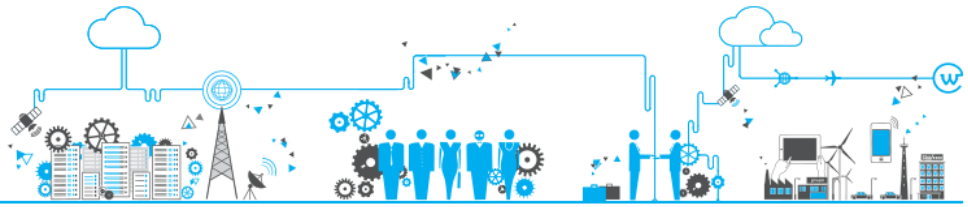
Doing Agile in an ISO-20000 World

a story from the trenches



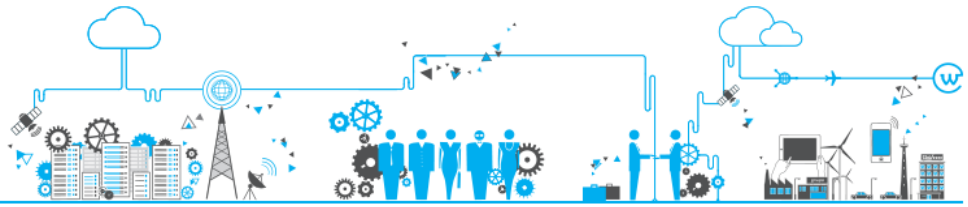
Manuel Padilha
CTO, VP Engineering

«I don't think anyone could object to a ban on the word [Agile] when it is used as a noun. That's just plain wrong. (…) Agile is not a noun, it's an adjective, and it must qualify something else. "Do Agile Right" is like saying "Do Orange Right."»

Dave Thomas http://pragdave.me/blog/2014/03/04/time-to-kill-agile/

So, lets change the title of this presentation

Developing software with *Agility* for an ISO-20000 Telco

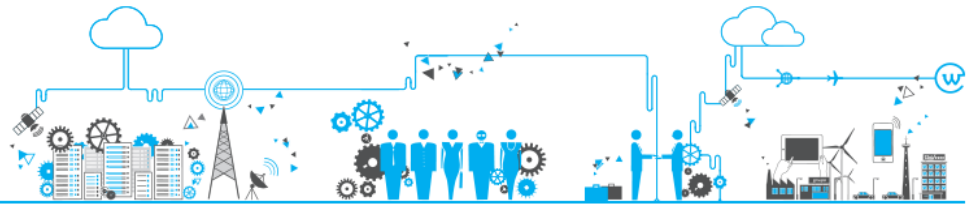


Project Scope

"Replace a set of legacy applications with a brand new, distributed, fault-tolerant, high-performance system, keeping all existing functions intact [and, by the way, we also need this and that]."

So, translating from marketing-speak:

1. build something from scratch
2. mimic existing functions (that were organically added over the course of the last 15 years)
3. dramatically change architecture
4. add new stuff while we're at it



The Customer

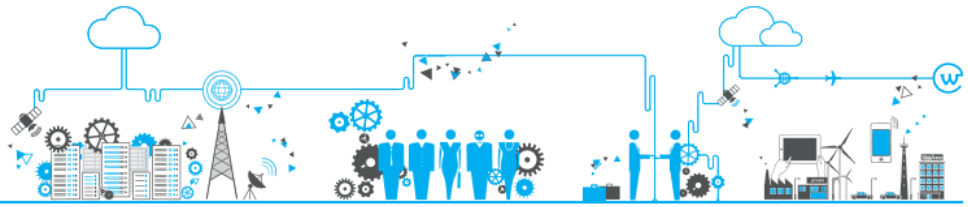
A film distribution and exhibition company which is part of a larger Telco

Two very different cultures

- one is a small (<500 people), mostly autonomous company with a vertical structure
- the other a telco behemoth, with rigidly defined processes and a mostly horizontal structure

For this project

- The small company provides “requirements”
- The large company manages the project and deals with the IT side of things



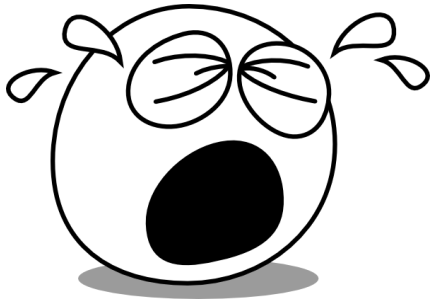
Project kick-off conversation

We need this **done** by mid June. You must start right away and we'll clear things up as you move along.

small company sponsor

First we must write the **spec**. And the **test plan**. Then we'll discuss **migration**, **deployment** and **roll-out** strategies. After that we can refine the **project plan** and agree on **milestones** and **deliverables**. After that is done we'll ask IT to **provision** Dev and QA environments, and once that's set up you can start building software - this should happen around February. Oh, and we need 3 months to do **acceptance** and **load tests**, a production **pilot** and a **phased roll-out**, so you'll need to be finished by March.

large company project manager



small company sponsor

has a hard deadline
is comfortable with flexible scope

trusts partner
doesn't care about document signoffs

cares a lot about the end product quality
and maintainability (he will be paying for it!)

large company project manager

will enforce hard deadline
needs a fully defined scope

distrusts partner by principle
document signoffs are part of the process

cares a lot about by-products quality
maintenance is someone else's job

Pack your bags and go home? Give up and do waterfall?

No. No. No!

Let your development team use Scrum and keep them isolated from Project Management.

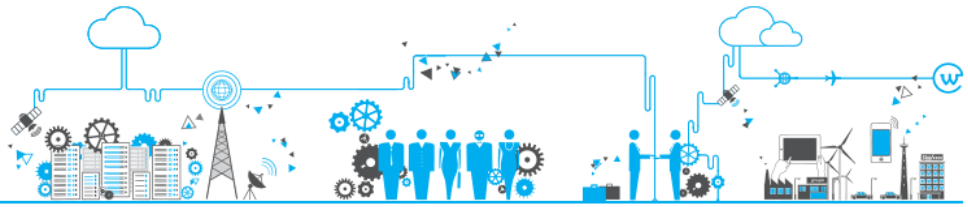
Allow customer interaction at sprint reviews, but make sure there is a separate meeting with the PM.

Prioritize the backlog according to the milestones agreed with the PM. Be prepared to negotiate changes.

Never forget

You'll be judged by what you deliver,

not how well you play the game*



*but you still have to play it

The Project Plan vs. The Product Backlog

Scrum says you should keep a product / project backlog and feed a sprint backlog from it. There is no master plan.
Project Management procedures say we need a Project Plan with deliverables and milestones that is signed out by the Project Manager and key stakeholders **before** the project starts.

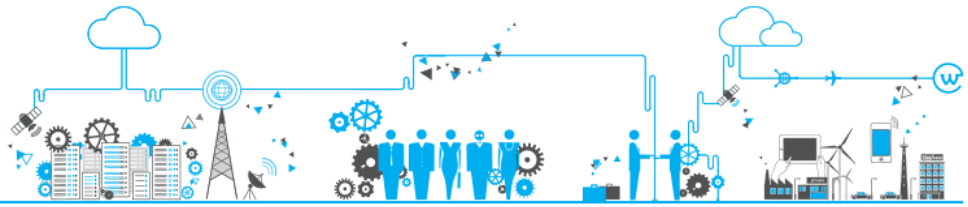
So, what now?

You can't escape the Project Plan, so **guess**. Guess the best you can, but with no fear of failing.

Project Plans need to be signed out, but they can be **changed** later on.

Use the Project Plan as a kind of informal Product / Project Roadmap and stick it to your Scrum wall.

Do not impose it on your team. **Listen.**



Change Management

a) We need to replace our current self-vending eletronic payment provider. Your software should interface with the hardware directly.

- No, it would take forever to obtain the mandatory software approval from the payment broker.
- Yes, we have inside connections, we can do it “really fast”

(The jury is still out on that one, but there’s no time left to do it).

b) We need to change the way we perform eletronic payments online.

- Oh wait, we can’t, it’s not safe.
- Yes we can, just do it already.

c) We need the same supported operating system across the entire platform, it’s going to be X.

- Ooops. X won’t run on that hardware. Let’s replace it!
- Ooops. No budget.
- Hang on, there’s a huge hardware maintenance budget. We can slash it if we buy new hardware...

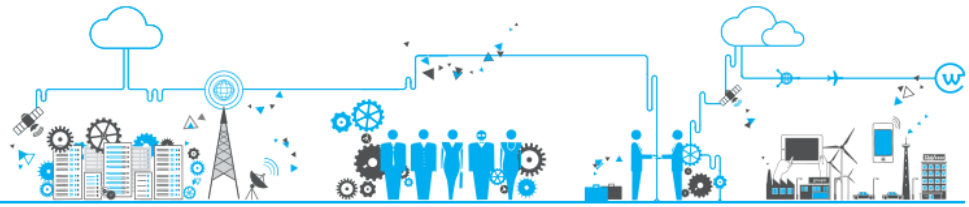
Time Keeping

The **Project Manager** will ask for detailed work logs by requirement.
Your **Team** will spend time refactoring, experimenting, prototyping and generally building software, not orderly checking off items on a requirements document.

So, what now?

Lie. In good faith, but lie. Provide bulk work logs and remind your project manager that they bought a final product, not a bag of hours / resources (this will be tough).

Do keep a record of **time spent on work items**, whatever they may be. JIRA is more than enough for this. That will help in retrospectives and might, on occasion, be useful feedback for the Project Manager.
Just make sure your development team isn't hindered by it.
Their time is precious. Yours, not so much.



Summary

Agile = Good
Planning and Processes = Bad

Right?

Wrong!

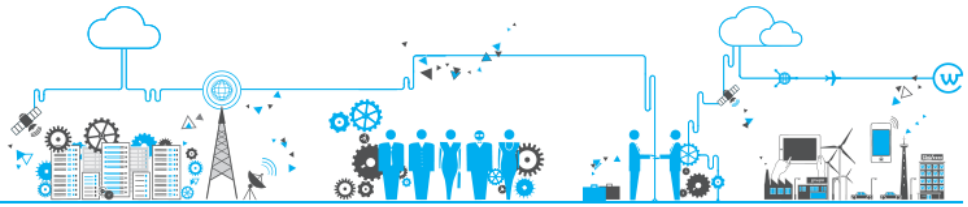
A professional Project Manager was essential for this project.

He helped keep the customer in check and put some healthy pressure on us.

Once he adapted (ha!) to an agile team, it was mostly smooth sailing.

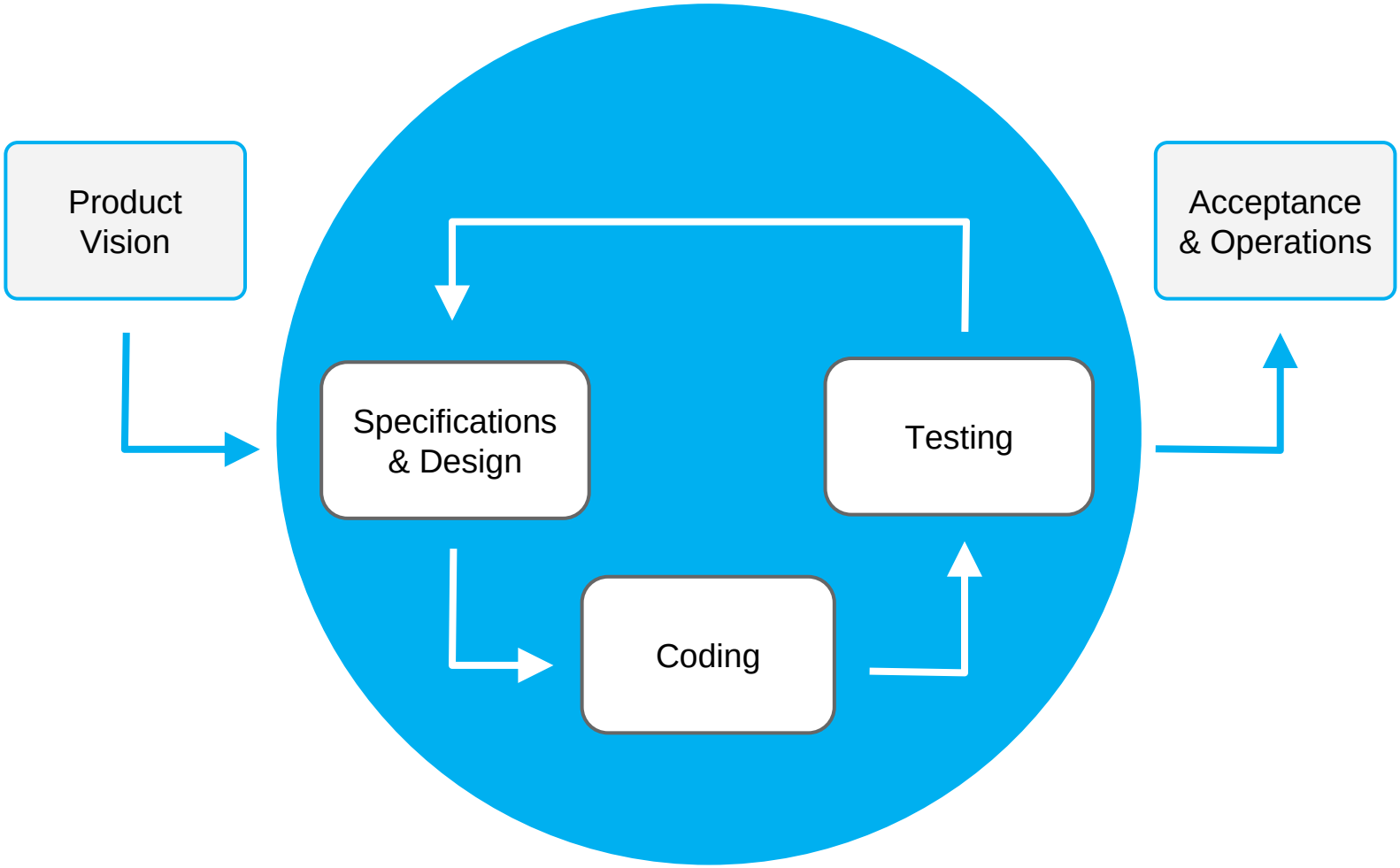
The processes were experience-tested and proved invaluable, even for a *rogue* team as ours.

Document templates were too complex, but contained good ideas for the ones we ended up using.



Summary

WaterScrumFall



Thank you!