



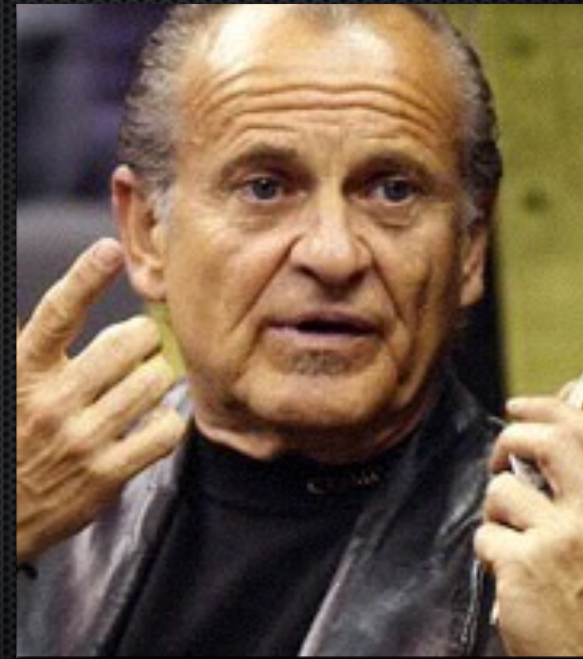
Agile Transitions

From the Trenches



Who is *this* Freakin' Guy?

- Programmer of things
- Teacher of stuff
- Drinker of Beer
- Keeper of Grounds at Hogwart's
- Star of Finding Bigfoot
- Very Serious
- Very Professional



Good agile, Bad Agile

What IS agile? and what is Agile?

- Agile is a brand

 - has mostly lost its meaning

 - describes any number of Scrums, Kanbans or XPs

 - strikes fear into the hearts of BAs (Tom Smykowski) and PMs

- agile is an adjective

 - means nimble or spritely

 - better description of what we're trying to do

 - means we'll be better adept at responding to changes/mistakes

 - strikes fear into CFOs and the hearts of the lazy

Tools Can Help You Become More agile

(But they don't MAKE you agile)

Tools for becoming agile
eXtreme Programming
Scrum – Scrumbut – Scrumban
Kanban – Lean

Common Missteps



“Agile just won’t work here.”

Summarily dismissing agile software development.

Comes from a misunderstanding of Agile/agile usually.

Followed by TDD, Pair-programming, etc. won’t work here

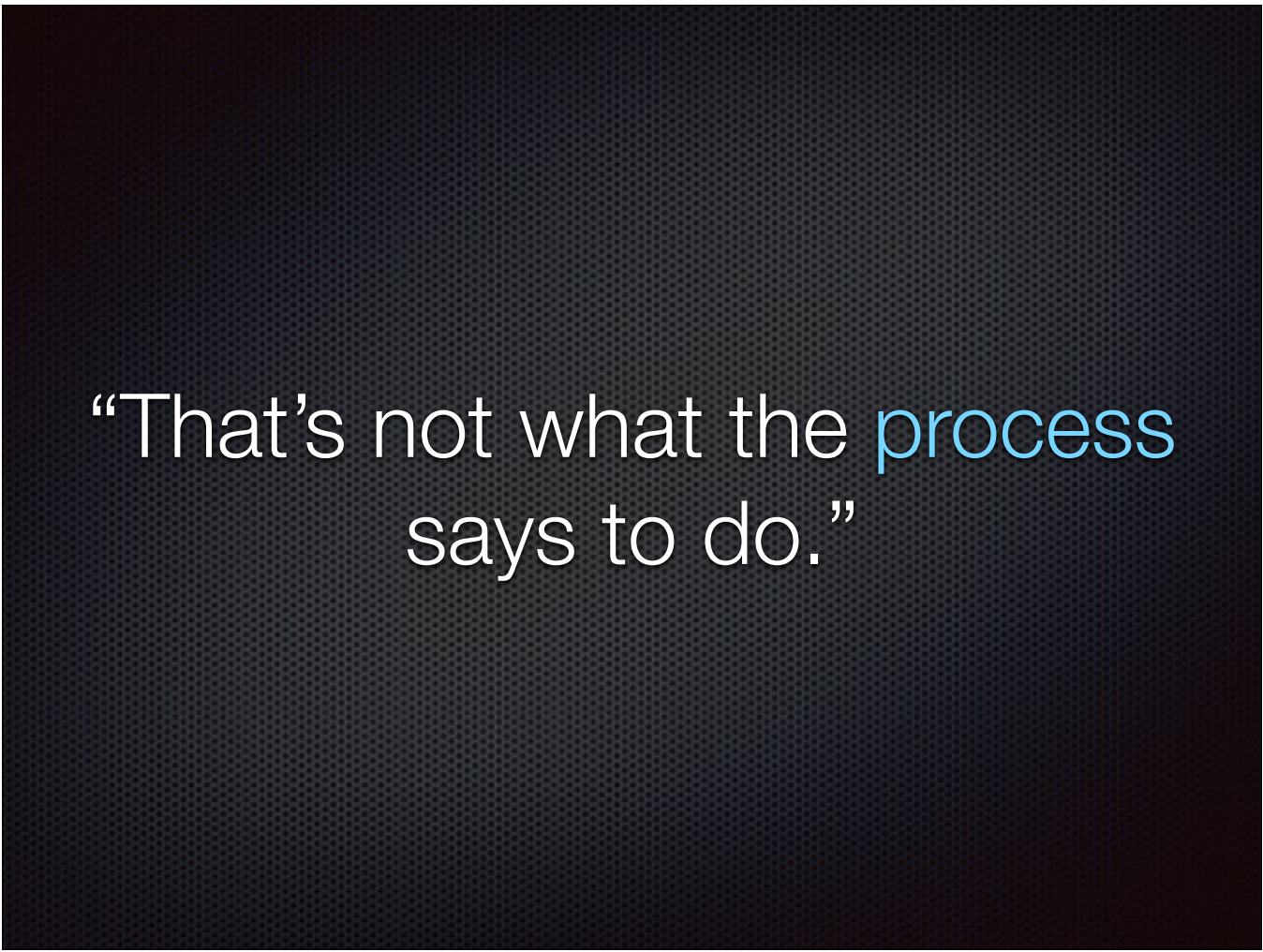
What would have to happen in order for it to work here?



“I am on the QA team.”

No. You are on the team that is building this software.
Stop thinking about your company in functional silos.
Contractor model is not a bad idea.

You can still have those departments, but contract them out to each team.



“That’s not what the process
says to do.”

You are not your process.

Your process exists to serve the organization. Not the other way around.

Your process should be set to improve software delivery, not for the sake of the process.

Understand you were not hired to follow a process. You were hired to make software.

Me: “Everything is always
changing.”
Retort: “Yeah, that’s the problem
with Agile.”

Agile didn't _cause_ the change.

Business must inherently handle change.

Agile is all about how you respond to change.

The process should make change easier, more fluid

When change is painful, you can choose to ignore change, or make change easier.

What's more important: giving the business what they asked for? Or giving them what they need?

If something is painful, do it more often.

"My team isn't meeting their
commitments. I don't know
what to do."

Commit to less.

Success begets success.

Make sure you start with a commitment your team can achieve.

It's easy to pull in more stories if you finish early and it builds confident, successful teams.

If your team isn't meeting its commitments, either you're over committing or your team is lazy and not getting their work done.

If you assume they are lazy, you:

- put websense in place to keep them from Facebook, etc.

- Keep a tighter leash on them to make them do their work.

- (Scientific Mgmt) It creates a subjugated relationship (parent/child).

 - If you act like the parent, your team will act like children.

 - Sociology 101, people fill the roles they think others expect them to fill.

 - If you are authoritarian, you will create "subjects" that don't think for themselves.

 - They ask every time. They run EVERYTHING past you.

If you assume they are over committed, you:

- Back their commitments off a little to the point that they can achieve their commitments.

- Maybe even pull in some extra stories.

- They feel confident. They feel good. You stay out of their way.

- They become empowered to charge ahead, to make improvements, to think for themselves (and make mistakes).

- This is a recipe for innovation.

Which team would you rather work on/with?



“But Agile says we don’t do x.”

Being agile doesn’t mean we stop doing _anything_ except things we don’t have to do.
For the most part, even Agile doesn’t specify that you don’t do something.
The Agile Manifesto only proclaims that some things are more valuable than others.

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**.



“We tried Agile once, it was
a mess.”

Don't make it sound like _spinach_!

It's not about Agile, it's about being agile.

Chances are the mess is already there, and your new process just made it visible.

Agile tools are SUPPOSED to help you find problems, so you can address them.



“What about planning?!”

If we make a plan, and something (requirements, assumptions, etc.) changes, what happens to the plan?

Why spend a lot of time developing a plan?

When reality starts diverging from the plan:

- Developers want to adjust the plan to fit reality.

- Project Managers want to adjust reality to fit the plan.

- Managers/C-Level want to fire someone. (which is probably why PMs want to adjust reality)

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

Communication

Most of the Agile tools out there, are meant to increase communication:

- Stand-Ups
- Retrospectives
- Pair-Programming
- Task Boards
- Team Co-Location
- etc..

Some Good Reads

- Extreme Programming Explained: Embrace Change - Kent Beck
- Agile Software Development With Scrum - Ken Schwaber & Mike Beedle
- Kanban: Successful Evolutionary Change for Your Technology Business - David J. Anderson
- Managing the Design Factory - Donald Reinertsen
- Lessons in Agile Management: On the Road to Kanban - David J. Anderson

Closing Words

- Silver Bullet
- Start With What You Do Now
- Question Everything
- Improve
- :||

Stop looking for a bullet period how about a prism? Magnifying glass and a good, full toolbox.
You can't improve by removing defects.
What would happen if we did/didn't do this?

Thank You