Introduction to Agile

Dr. Vadim Zaytsev aka @grammarware
2015

Manifesto

* Manifesto for Agile Software Development, 2001
* http://agilemanifesto.org

Individuals &
interactions

over

processes
& tools

(1)

Agile “Individuals”

* Kent Beck
* Mike Beedle
* Arie van Bennekum
* Ali Cockburn
* Ward Cunn
* Martin Fowler
* James Grenning
* Jim Highsmith
* Andrew Hunt
* Ron Jeffries
* Jon Kern
* Brian
* Ro C. Martin
* Steve Mellor
* Ken Schwaber
* Jeff Sutherland
* Dave Thomas

SUnit

TDD

XP

Wiki

Refactoring

Clean Code

OOA

Scrum

Pragmatic Programmer

Individuals & interactions

over

processes & tools

(1)

Working software

over

comprehensive documentation

Customer collaboration

over

contract negotiation

Responding to change

over

following a plan

Let's try…

* How to choose a language?
* How to deal with deadlines?
* Can we outsource?
* When is the first release?
* How often to deploy?
* What to test and when?

Agile principles

* Satisfy customers by rapid delivery of valuable software
* Welcome changing reqs, even late in dev
* Deliver working software frequently
* Business people and developers must work together daily
* Projects are built by trusted professionals
* Face-to-face conversation above other means
* Working software is the measure of progress
* Maintaining constant pace of dev indefinitely
* Continuous attention to technical excellence & good design
* Maximise the amount of work not done
* Self-organising teams deliver best results
* Regularly reflect and adapt yourselves

http://www.agilemanifesto.org/principles.html

Agile principles

* Satisfy customers by rapid delivery of valuable software
* Welcome changing reqs, even late in dev
* Deliver working software frequently
* Business people and developers must work together daily
* Projects are built by trusted professionals
* Face-to-face conversation above other means
* Working software is the measure of progress
* Maintaining constant pace of dev indefinitely
* Continuous attention to technical excellence & good design
* Maximise the amount of work not done
* Self-organising teams deliver best results
* Regularly reflect and adapt yourselves

http://www.agilemanifesto.org/principles.html

Methods and methodologies

* eXtreme Prg
* Scrum
* Kanban
* Lean
* BDD
* DDD
* TDD
* Pair prg
* Refactoring

Scrum

Logo from https://www.scrum.org

Scrum: the team

* Product owner
  * adopts the users’ PoV
  * adds user stories to the backlog
* Devs
  * deliver PSIs
  * pass the ball back & forth
* Scrum master
  * ensure progress
  * remove all distractions

Scrum: the events

* Sprint
  * timeboxed iteration
* Plan
  * max 4h
* Daily stand-up
  * all come prepared
  * max 15m
* Weekly review
  * demo
* Weekly retro (SM!)
  * possible improvements

Scrum: the artefacts

- Product backlog
  - user stories
  - requested features
  - bugfixes
  - todos
- Sprint backlog
  - same, but concrete & timeboxed
  - todo/wip/proto/done

Scrum: the rest

- Definition of done
  - PO & devs agree beforehand
- Burn down chart
  - plotted weighted backlog
- Spike
  - exploratory mini-sprint
- Planning poker
  - sprint plan gamified
- Scrum of scrums
  - focus on impediments (for others)

Alternatives

- Waterfall
  - Req → Design → Code → Test
- Whatever
  - plan? what plan?
- Hero programmer
  - code in the basement
- Specialised teams
  - server/client, backend/frontend

Scrum at Project
Software Engineering

Pitching

* market fit
* solid idea
* clear focus
* limited time
* match the audience
* only key points
* more inspiration:
  https://medium.com/@JDcarlu/6-different-kinds-of-pitches-5d96a076b6df

* “trust me with your money”

Peter Thiel, Blake Masters, *The Pitch*, 2012.

MVP

* Minimum Viable Product
* one feature
* killer feature
* feature that works
* demoable prototype
* fake the rest

* “you think this is cool? just wait”

Working demo

* clear increment
* new features, old bug fixes
* planned vs completed
* not too long
* impressive
* points of improvement

* “it didn’t work, now it does”

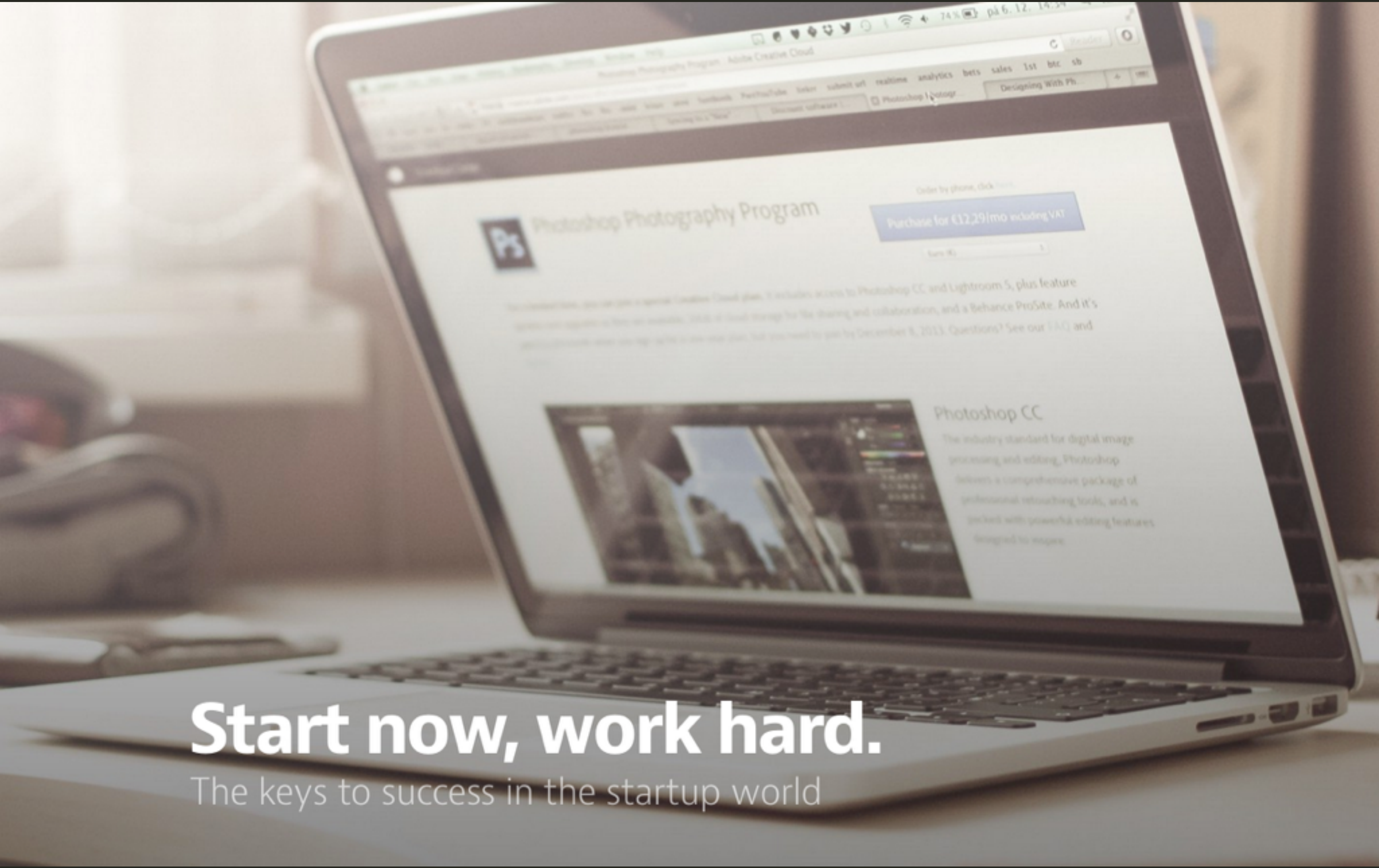
Audits

* Product owner manages
* Everyone participates
* Brag with the best
* Explain the worst
* Think globally

Final

* project website
  * goals, devs, etc
* useful deliverable
  * working, clear increments
* quality work
  * testing, etc

* “we’re done”



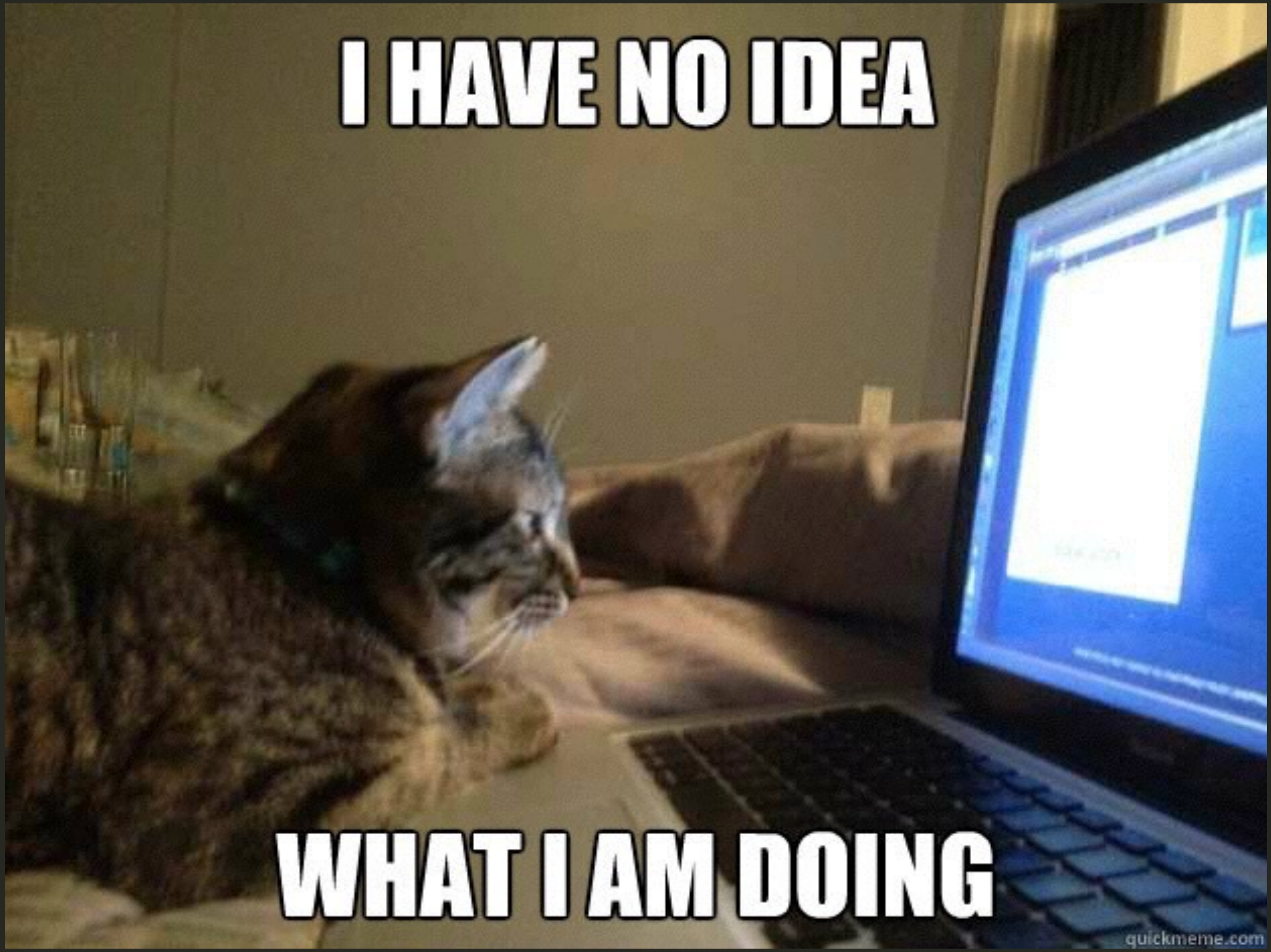
Start now, work hard.

The keys to success in the startup world

https://medium.com/@thomashoneyman/start-now-work-hard-294849ac1992

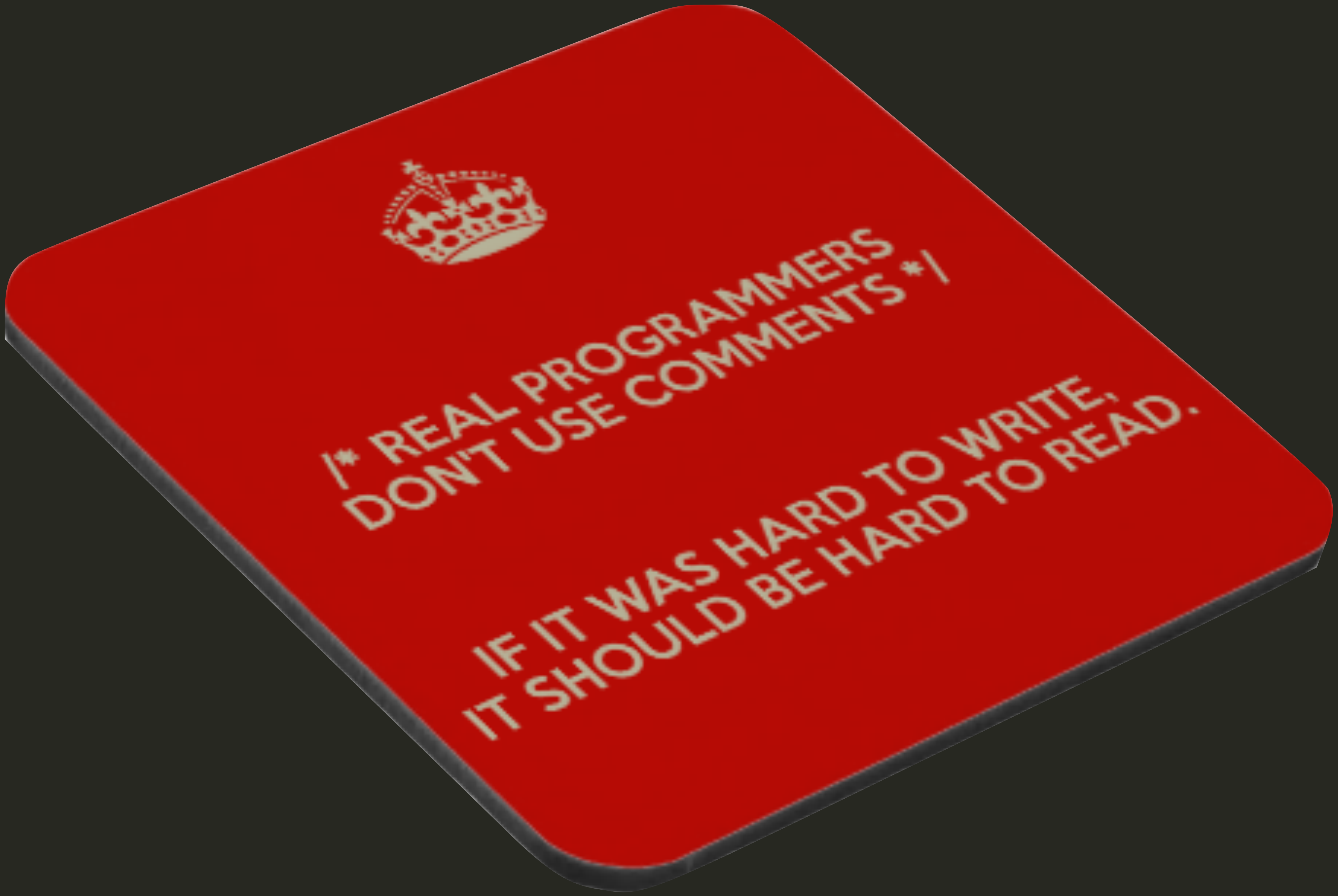
Agile problems

I am Scrum Master…



http://www.quickmeme.com/meme/35ik0t

Real programmers…



http://www.keepcalm-o-matic.co.uk/product/coaster/real-programmers-dont-use-comments-if-it-was-hard-to-write-it-should-be-hard-to-read-2

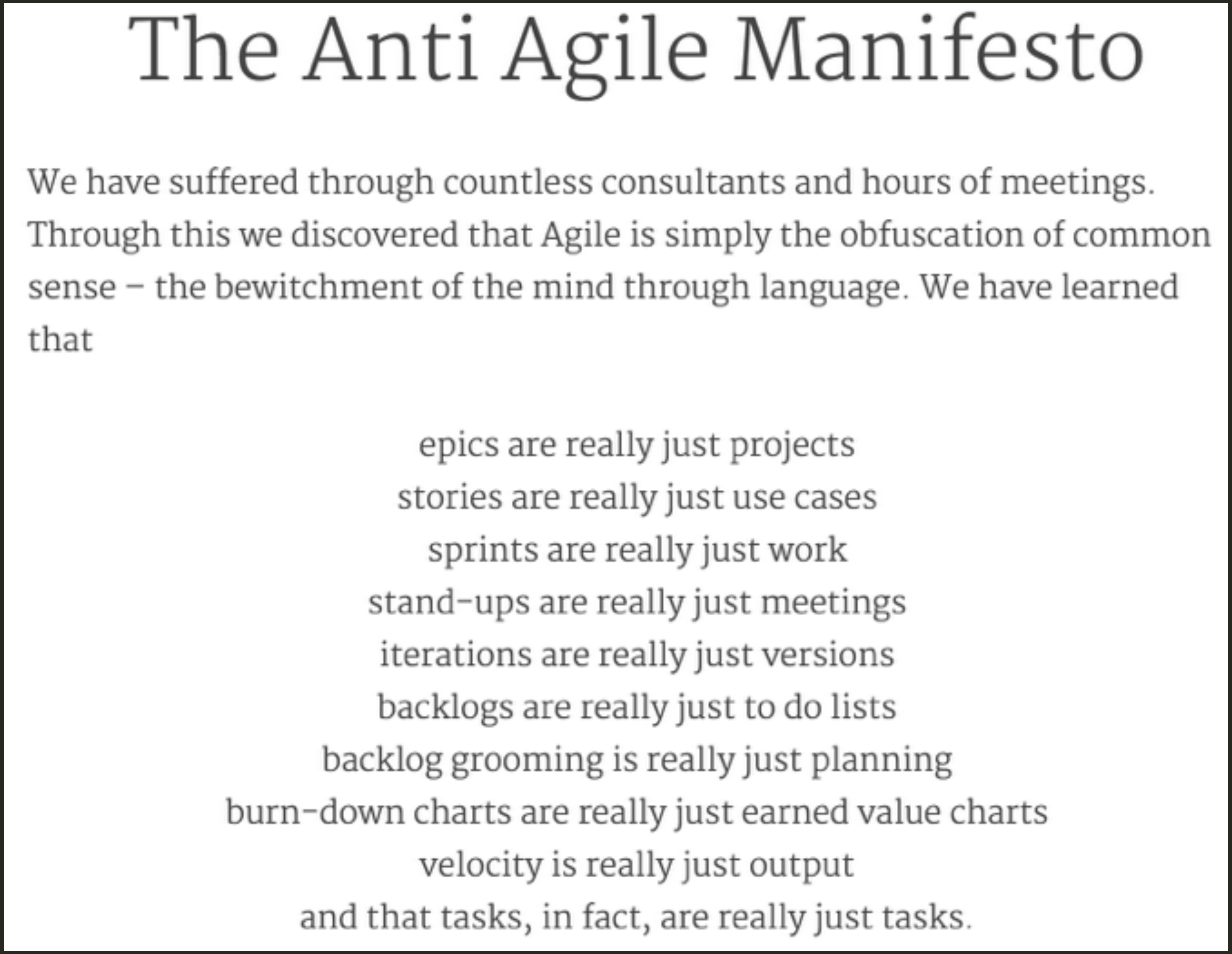
Plan? But Agile!

teh internets

Too much done…

http://www.troll.me/2011/11/23/x-all-the-things/check-all-the-boxes/

Cool kids lingo



The Anti Agile Manifesto

We have suffered through countless consultants and hours of meetings. Through this we discovered that Agile is simply the obfuscation of common sense – the bewitchment of the mind through language. We have learned that

epics are really just projects
stories are really just use cases
sprints are really just work
stand-ups are really just meetings
iterations are really just versions
backlogs are really just to do lists
backlog grooming is really just planning
burn-down charts are really just earned value charts
velocity is really just output
and that tasks, in fact, are really just tasks.

http://antiagilemanifesto.com/

Pro navigators

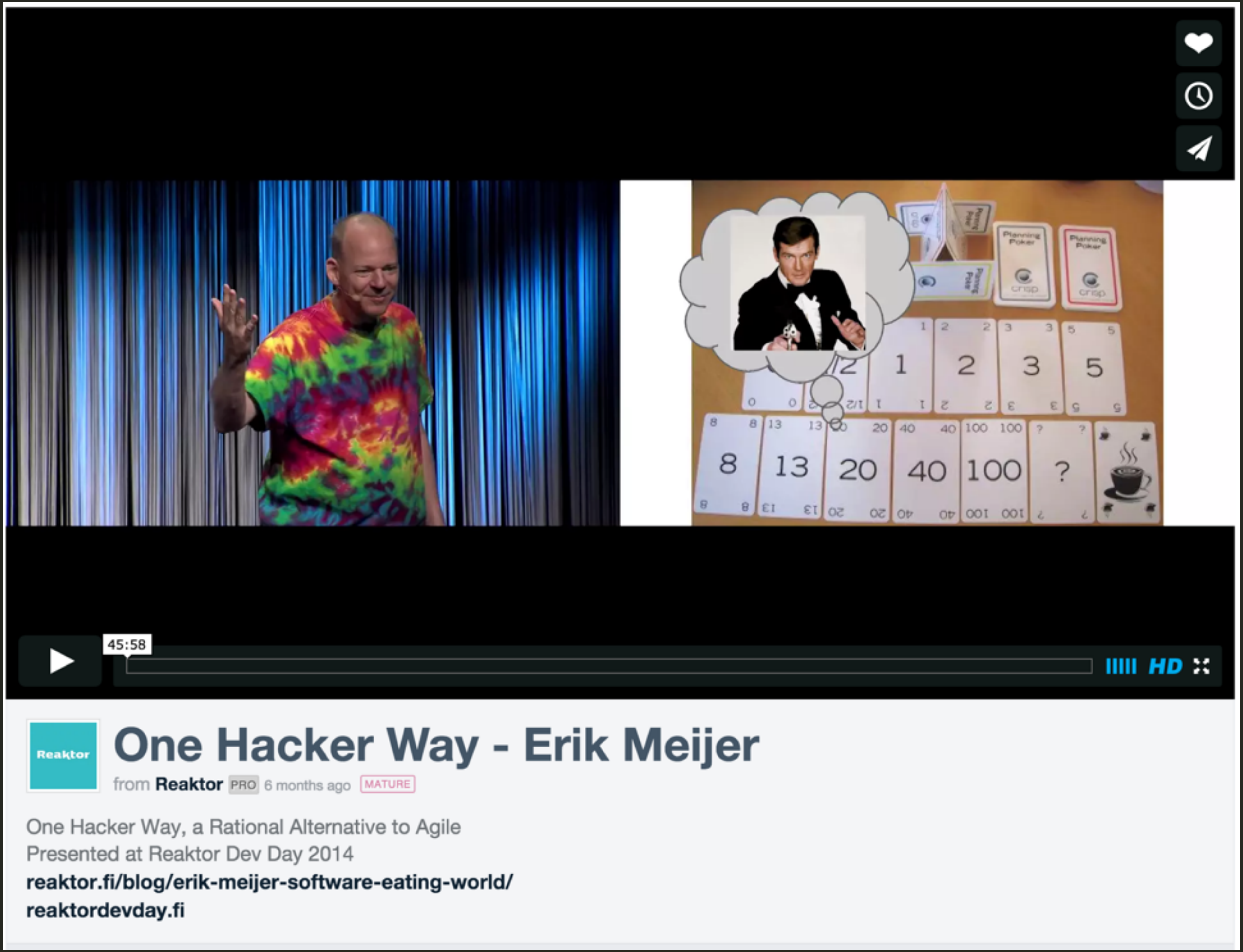
Daily stand-up

Why did I drink so much last night?
I was watching cat videos on YouTube all week…
When was the last time any of you commit any code?

Klean Denmark, https://www.flickr.com/photos/54548484@N08/5424689186; idea of Erik Meijer

Too much poker

Scrumtroopers, http://www.axisagile.com.au/resources/scrumtrooper-images/

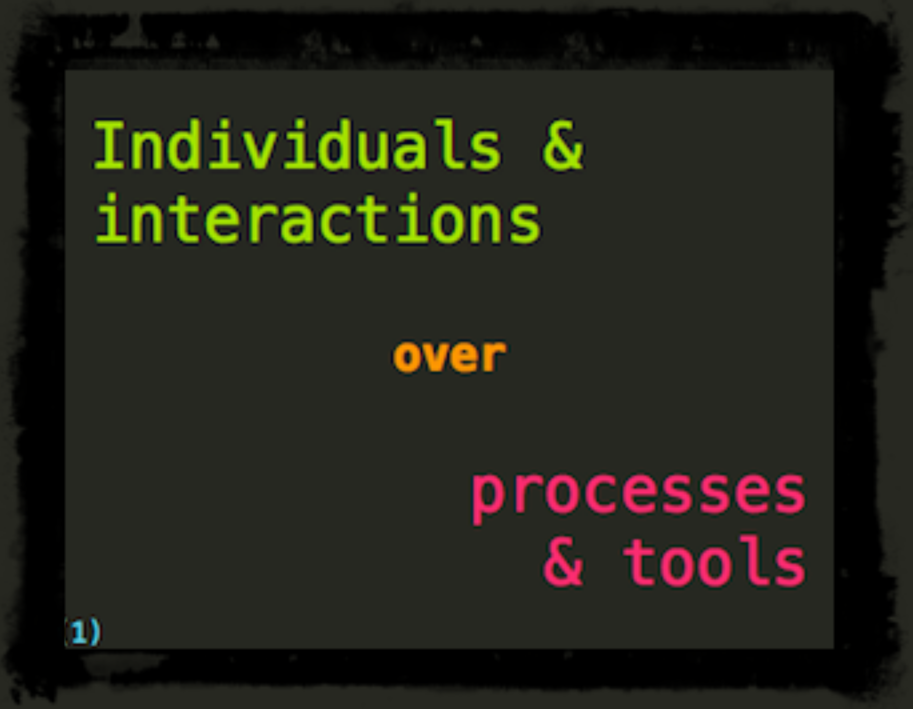


One Hacker Way - Erik Meijer

from Reaktor 6 months ago

One Hacker Way, a Rational Alternative to Agile
Presented at Reaktor Dev Day 2014
reaktor.fi/blog/erik-meijer-software-eating-world/
reaktordevday.fi

Agile Manifesto

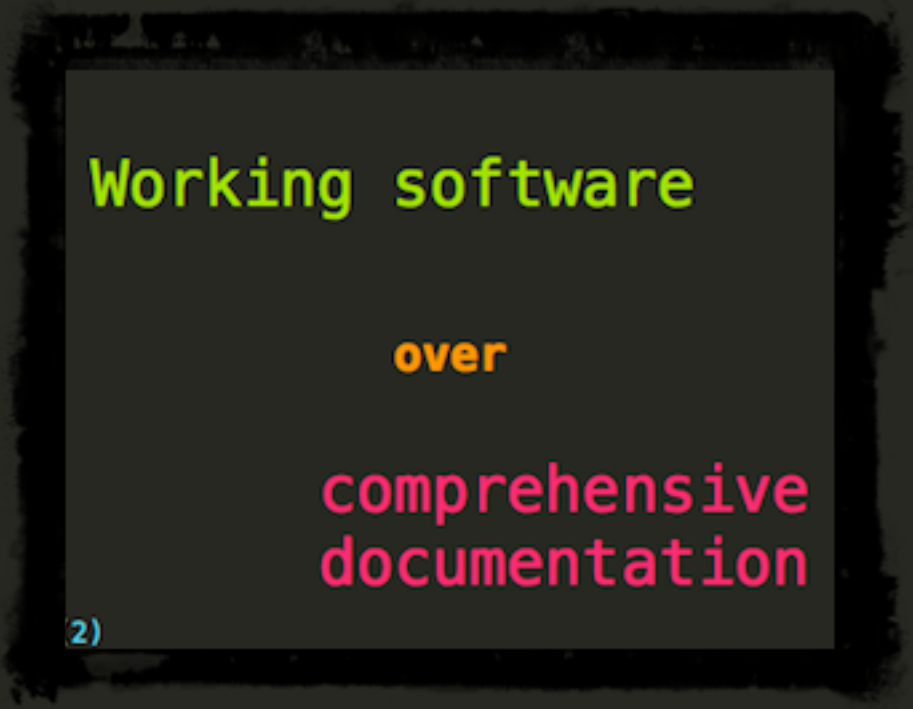


Individuals & interactions

over

processes & tools

1)

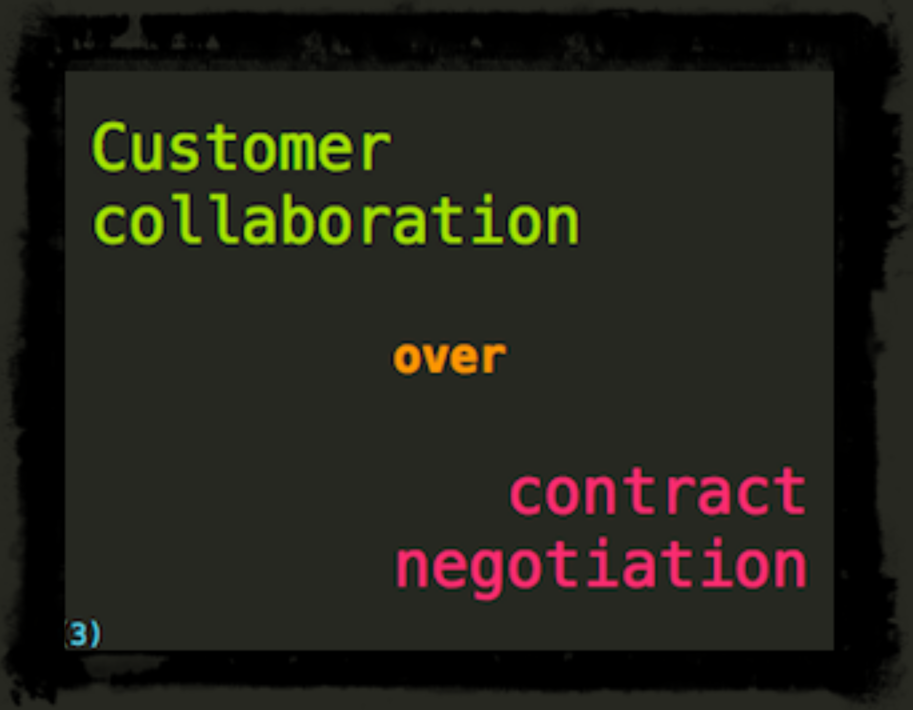


Working software

over

comprehensive documentation

2)

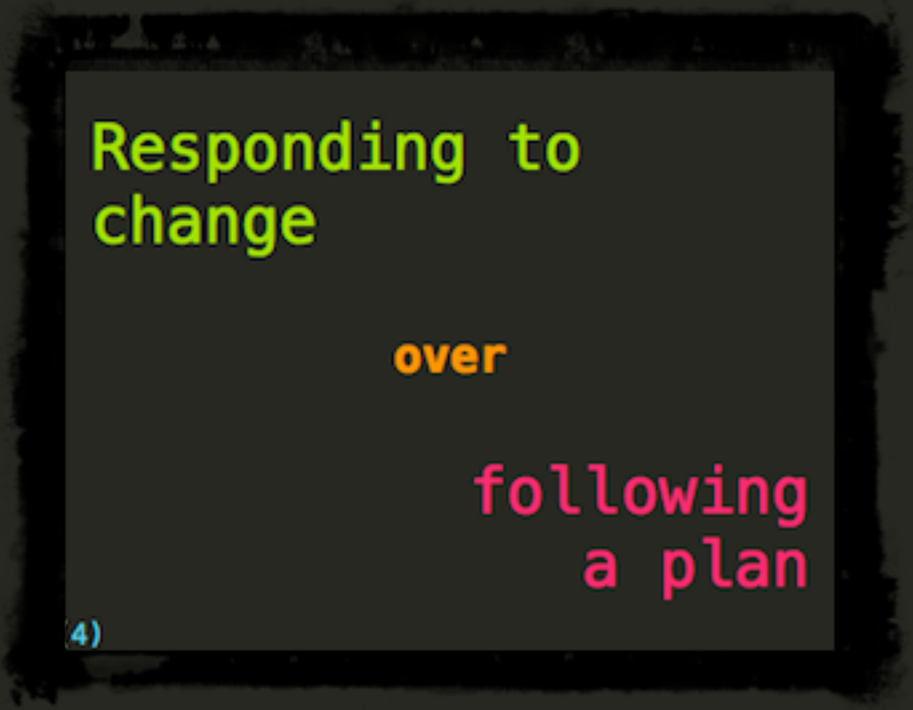


Customer collaboration

over

contract negotiation

3)



Responding to change

over

following a plan

4)

Scrum

8 people per team
1 product owner
1 scrum master
everyone works

weekly sprints
demos
backlog
daily stand-ups
retros

1: pitch
2: MVP
3: working
2&3: audits
4: dry run
4’: final

start now
work hard
deploy early
deploy often
stay cool

Questions?

Don’t forget to tweet-mention @grammarware!