Continuous Delivery - a few whys and a how

Willem van den Ende
WWW.QWAN.EU
@mostalive

Central Questions

1. Can we have our cake and eat it?

2. Can we ship more often without breaking the bank, or shooting ourselves in the foot?

How can I benefit?

People who do actual work:

- find better ways of delivering software

Consultants:

- find better ways of convincing others to find better ways of delivering software

All: maybe a better tool for your box

Real Artists Ship

"An old saying at Apple Computer,
attributed to Steve Jobs,
meaning that while producing stuff
that is cool, elegant, innovative,
and Insanely Great is important,
it is even more important
to actually produce working software
that can be delivered to users."

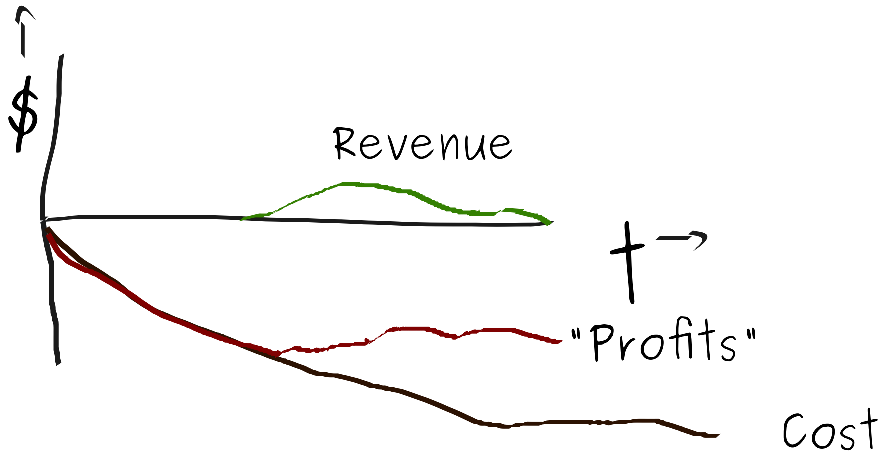
http://c2.com/cgi/wiki?RealArtistsShip

http://www.folklore.org/StoryView.py?story=Real_Artists_Ship.txt

Some terms

- Continuous Integration
  - Integrate (build, test, package etc.) software all the time
- Continuous Delivery
  - a button to push to production
- Continuous Deployment
  - automatic promotion to production

Why think?



Because Software Development fails
More often than you think

Therefore we deliver value early
And fail fast

What is Agile Software Development?

"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."

www.agilemanifesto.org

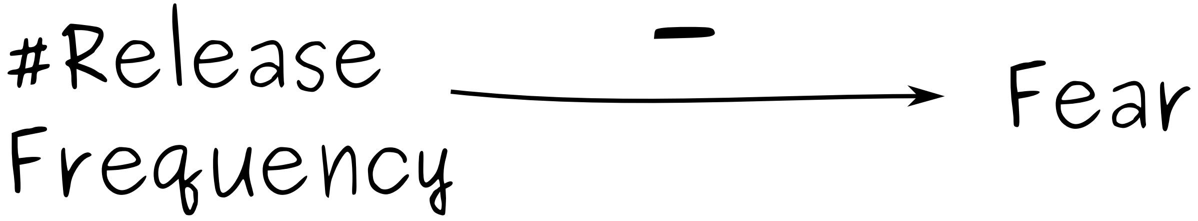
http://agilemanifesto.org/principles.html

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

* Working software is the primary
measure of progress.

*Deliver working software frequently [..]

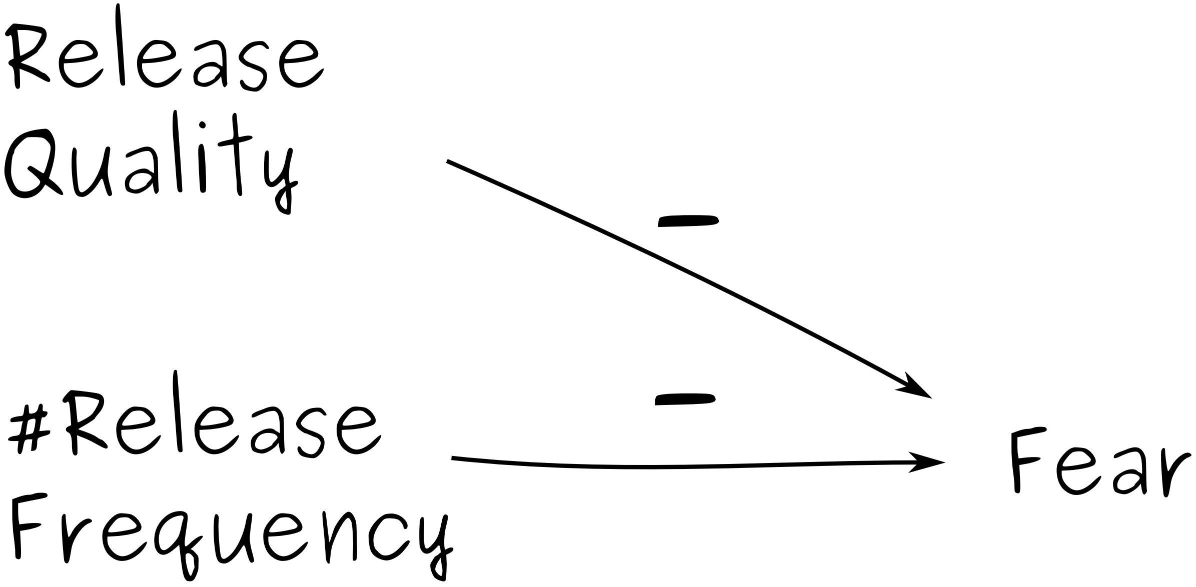
Q: Map out the pipeline for a a one line change to production



#Release Frequency

-

Fear



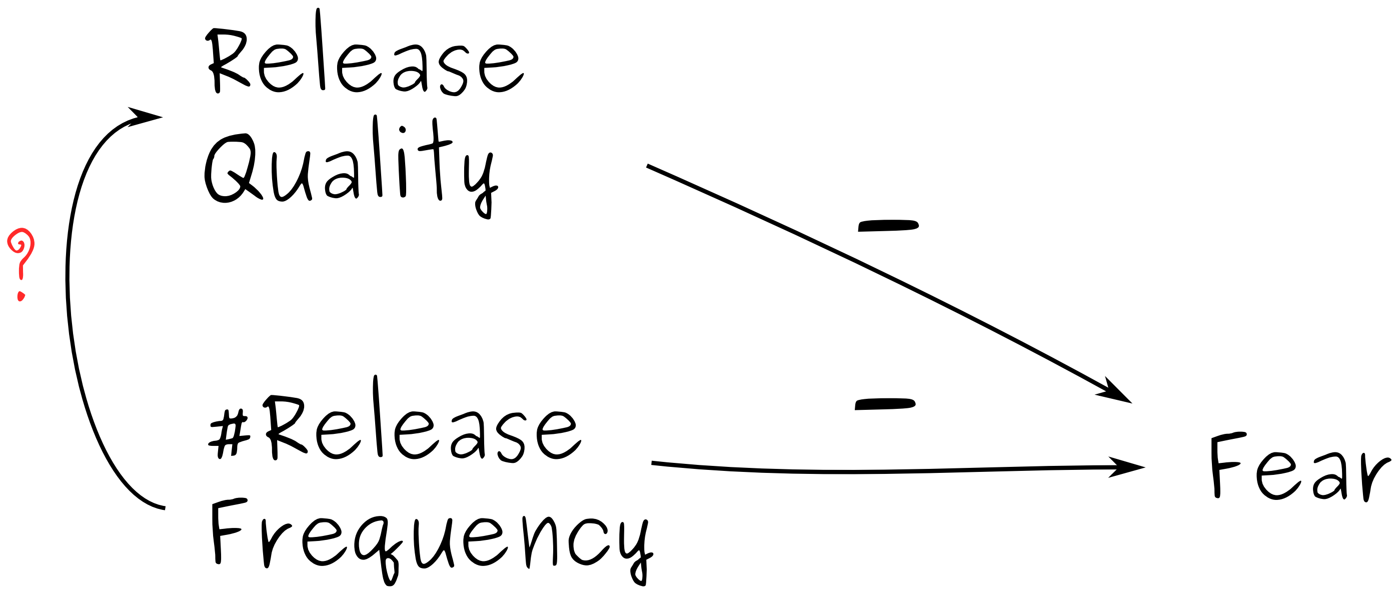
Release Quality

–

–

#Release Frequency

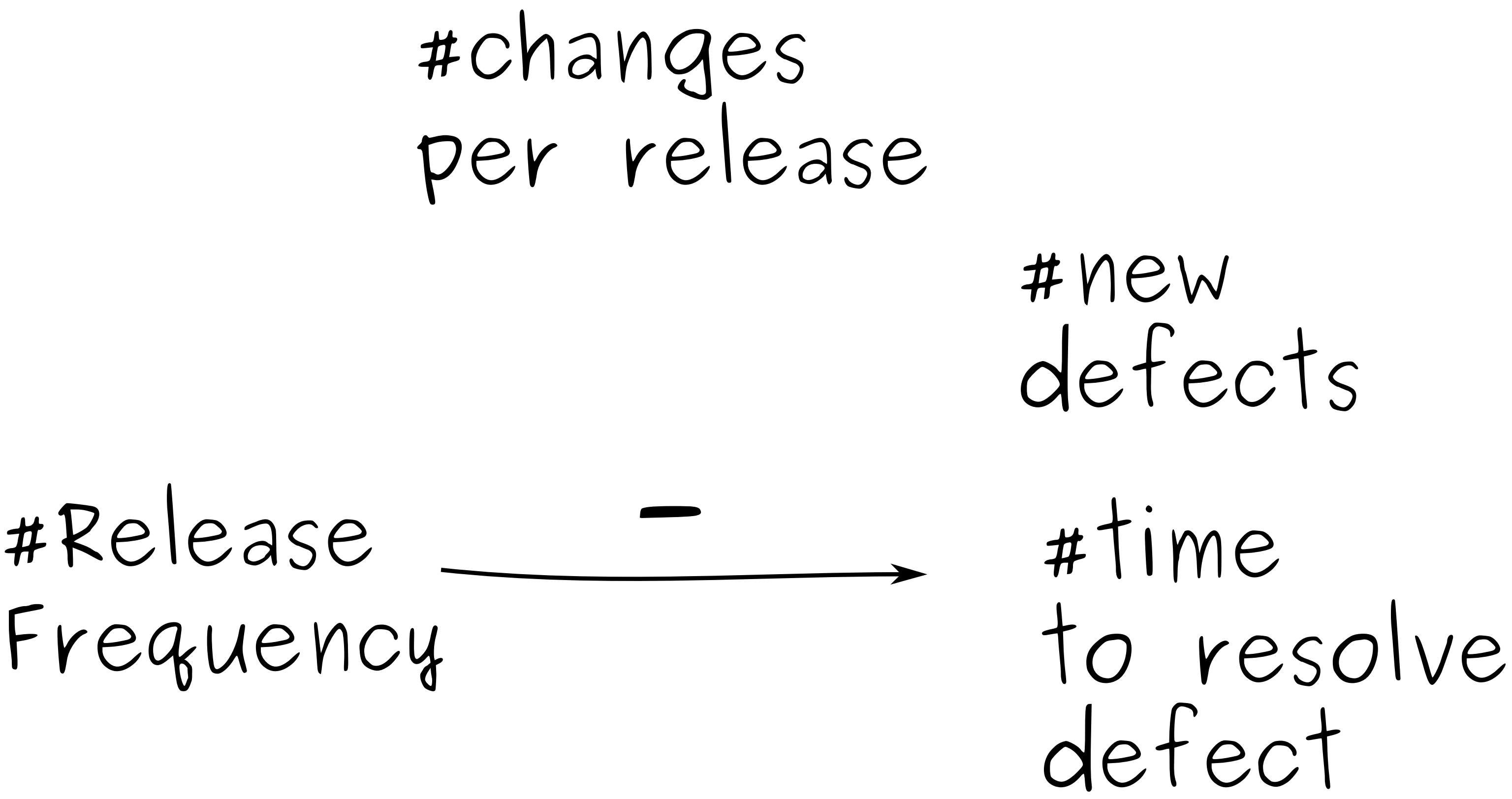
Fear



Release Quality

#Release Frequency

Fear



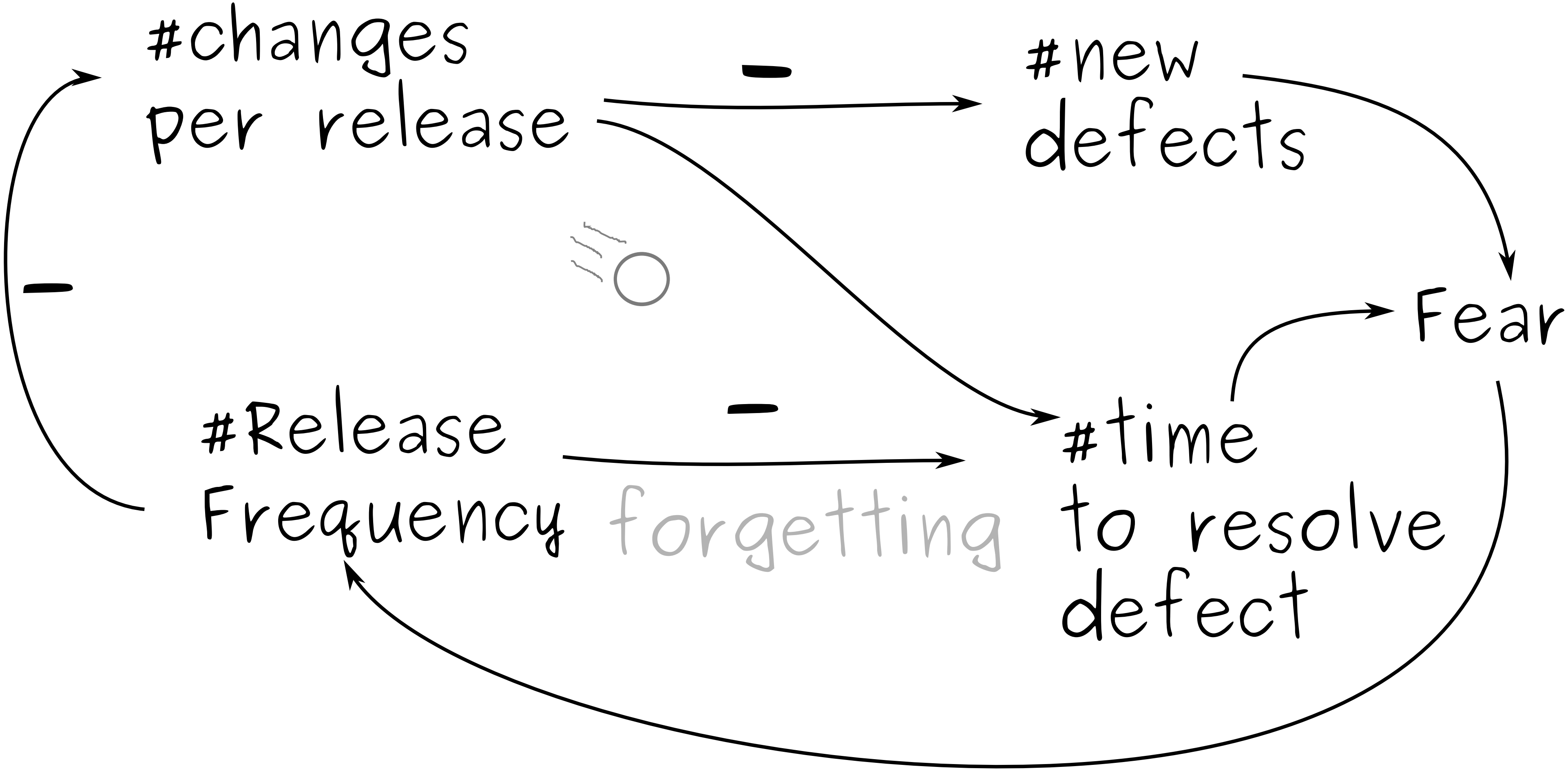
#changes per release

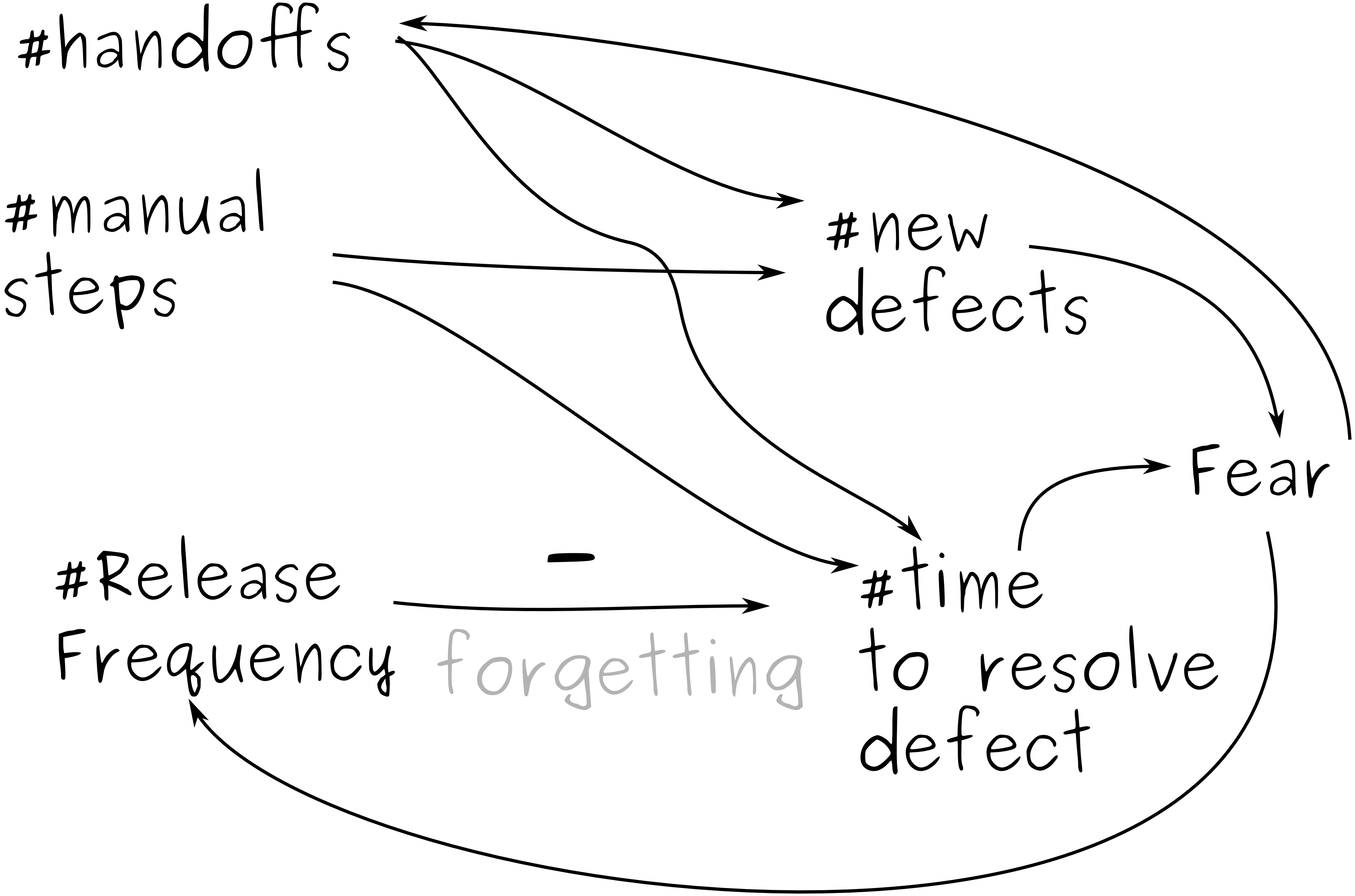
#new defects

#Release Frequency

-

#time to resolve defect


#changes per release
–
#new defects
Fear
–
–
#Release Frequency
forgetting
–
#time to resolve defect



#handoffs

#manual steps

#new defects

Fear

#Release Frequency

–

forgetting

#time to resolve defect

Q: "How often do you ship?"

A: "we work in sprints of two weeks,
and deploy every six weeks,
to acceptance"
Q: "How often do you ship?
A: "We haven't yet, not really"

Q: "How often do you ship?"

A: "in a few months, after 1.5 years of development"
W: "I predict you'll need another three months to go live, and we'll all be here next year"
A: "I don't believe you"

Q: "How often do you ship?"

A year later most of us were still there.
We shipped. Eventually.

Q: How to improve?

How to improve in this situation

Show you can be trusted

Eliminate manual steps and handovers
in all processes under your control.
One by one.

Generate hand-over documentation from
the scripts you use for your (development)
environment. E.g. literate python / haskell
Run your documentation in a build.

How to improve in this situation

Make your co-workers aware:

Painful repetitive manual steps are optional.

Most people prefer working on
things that work.

Even without automated tests,
automating deployments has value.

Story 2: Develop in production

Q: Who is developing this system?
A: Look on the other side of the hallway
Q: Where is the server?
A: Under the developers' desk.

Story 2: Develop in production

Q: How long does it take to deploy a one line change to production?

A: As long as it takes to write the line and press save.

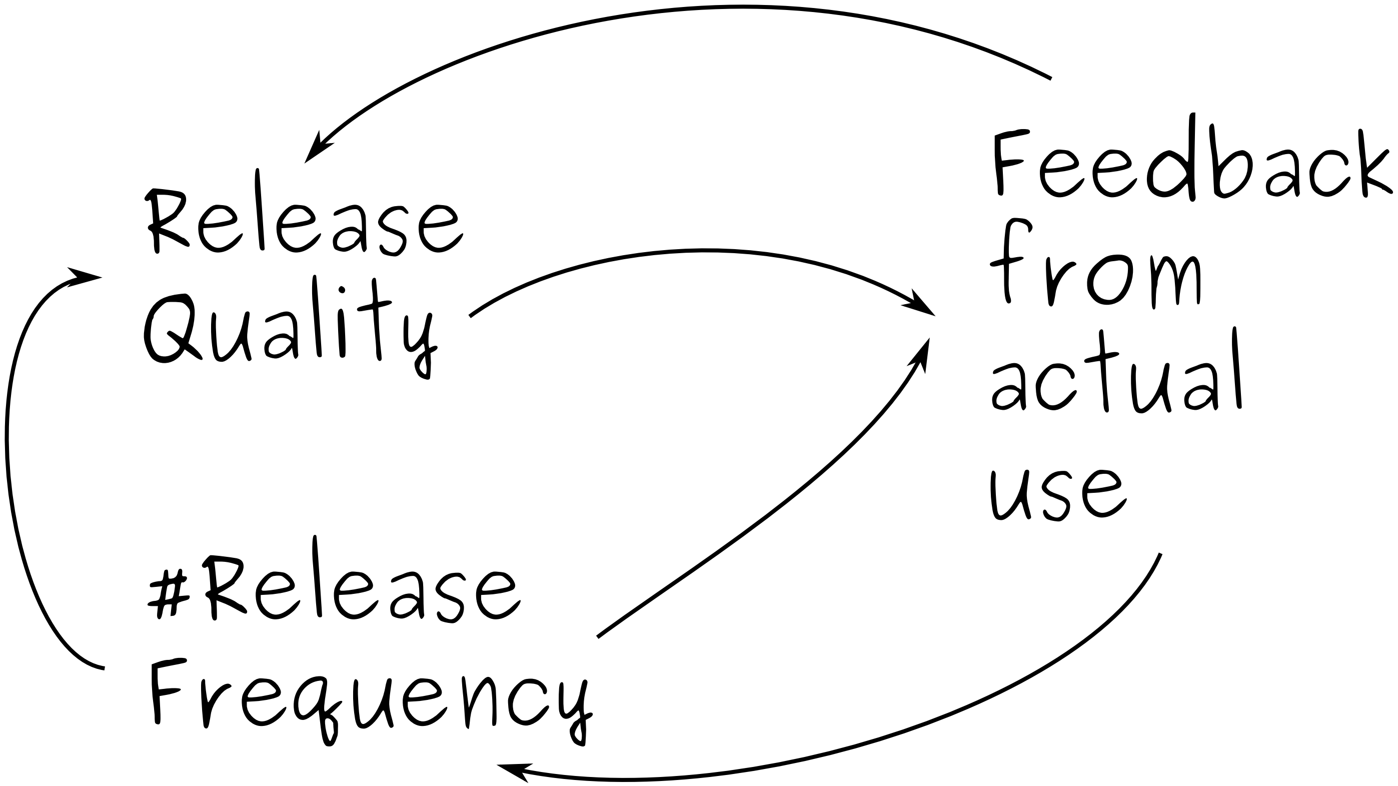
Story 2: Develop in production

Examples:

* wordpress, drupal, other CMS
* labs.qwan.it/photosuggest
* Several clients
* Anything that has the source code available in a production environment and an easy way to let saved source be picked up

Story 2: Develop in production

"there's no environment
like the environment"

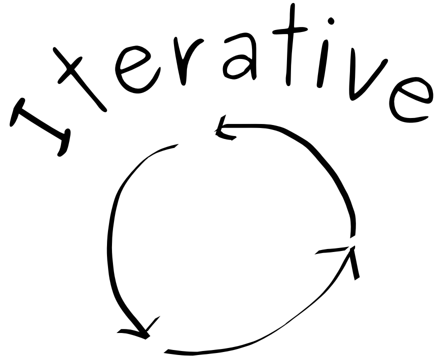


Release Quality

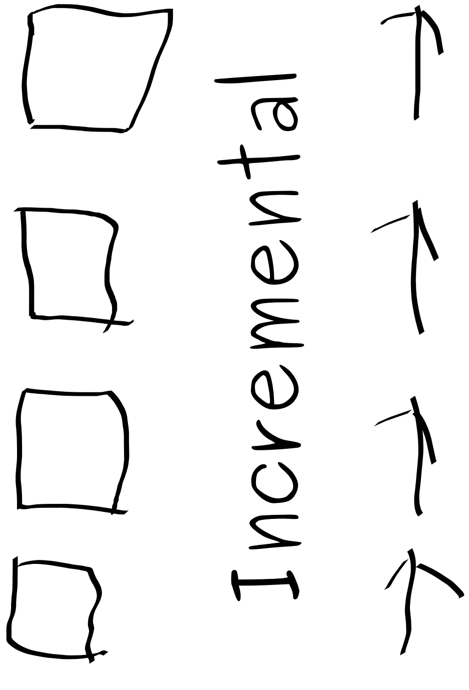
Feedback from actual use

#Release Frequency

"Quality is value to some person"
— Gerald Weinberg



And

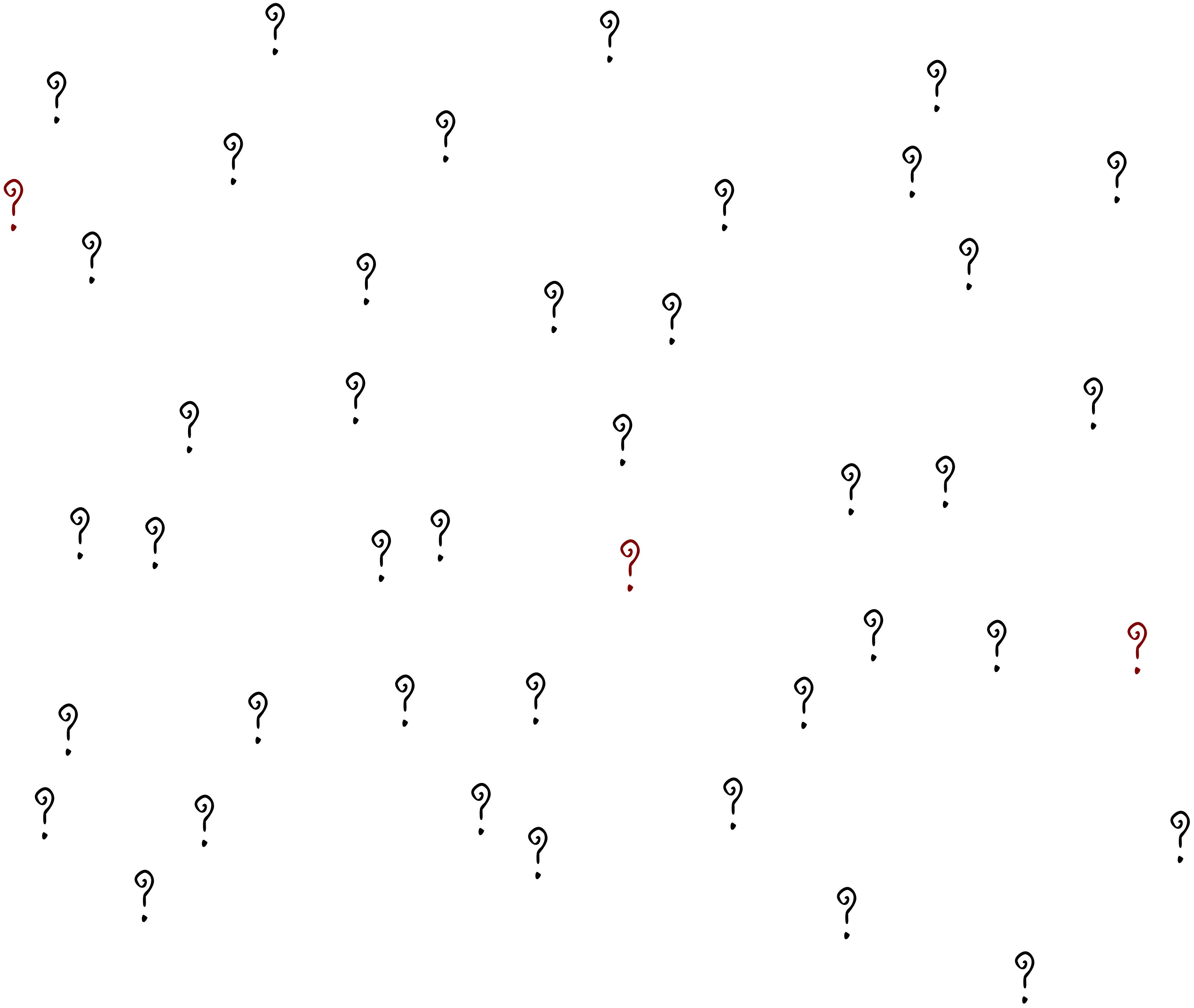


A way out for both stories

1. Ship
2. Ask yourself:

   were we faster than last time?
   did we make fewer mistakes?
   can we do this faster?
   can we do this more sustainably?

Remember: sometimes it gets worse
before it gets better.



Further reading

1. Move Fast - Break Nothing
http://zachholman.com/talk/move-fast-break-nothing/

2. The art of agile development
http://www.jamesshore.com/Agile-Book/

3. Continuous Delivery book
http://continuousdelivery.com/