Continuous integration

in real life

Vladimir Alekseichenko

About me

- *BigData* @ Base CRM

- I'm **passionate** about **good practice**
- Hobbies: *Haskell, Go, Brainfuck...*

Contacts

- @slon1024
- github: slon1024
- vova@vova.me

My experience*

- Release manager
- I did CI a few real project
- Also in big (enterprise) project with (big) technical debt.

* - *in this context*

I'll tell you

- What is CI?
- Why CI is cool?
- Why you should use CI?

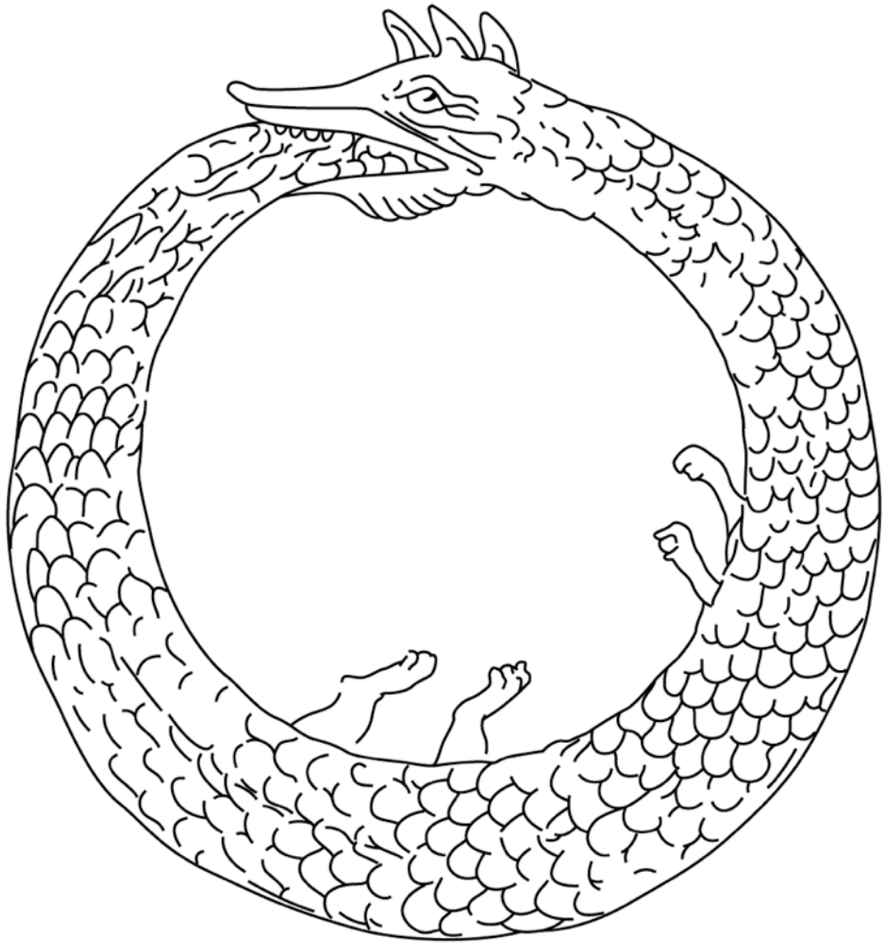
I won’t tell you

- About tools
- About technical details
- About trends
- ...

What is **continuous integration**?

Captain obvious says

What is continuous?



Uninterrupted in time, sequence, substance, or extent.

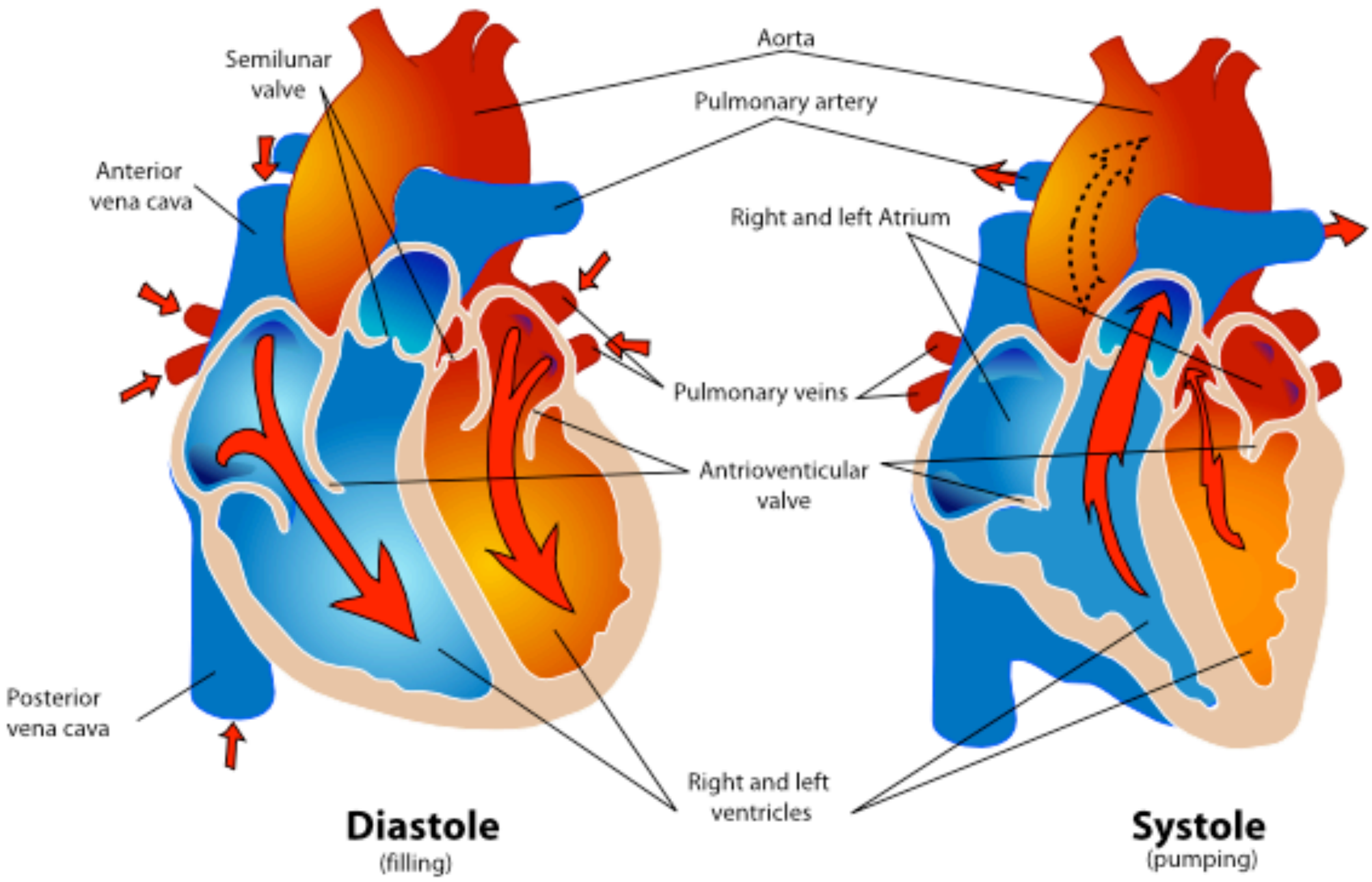
http://en.wikipedia.org/wiki/Ouroboros

In **nature**

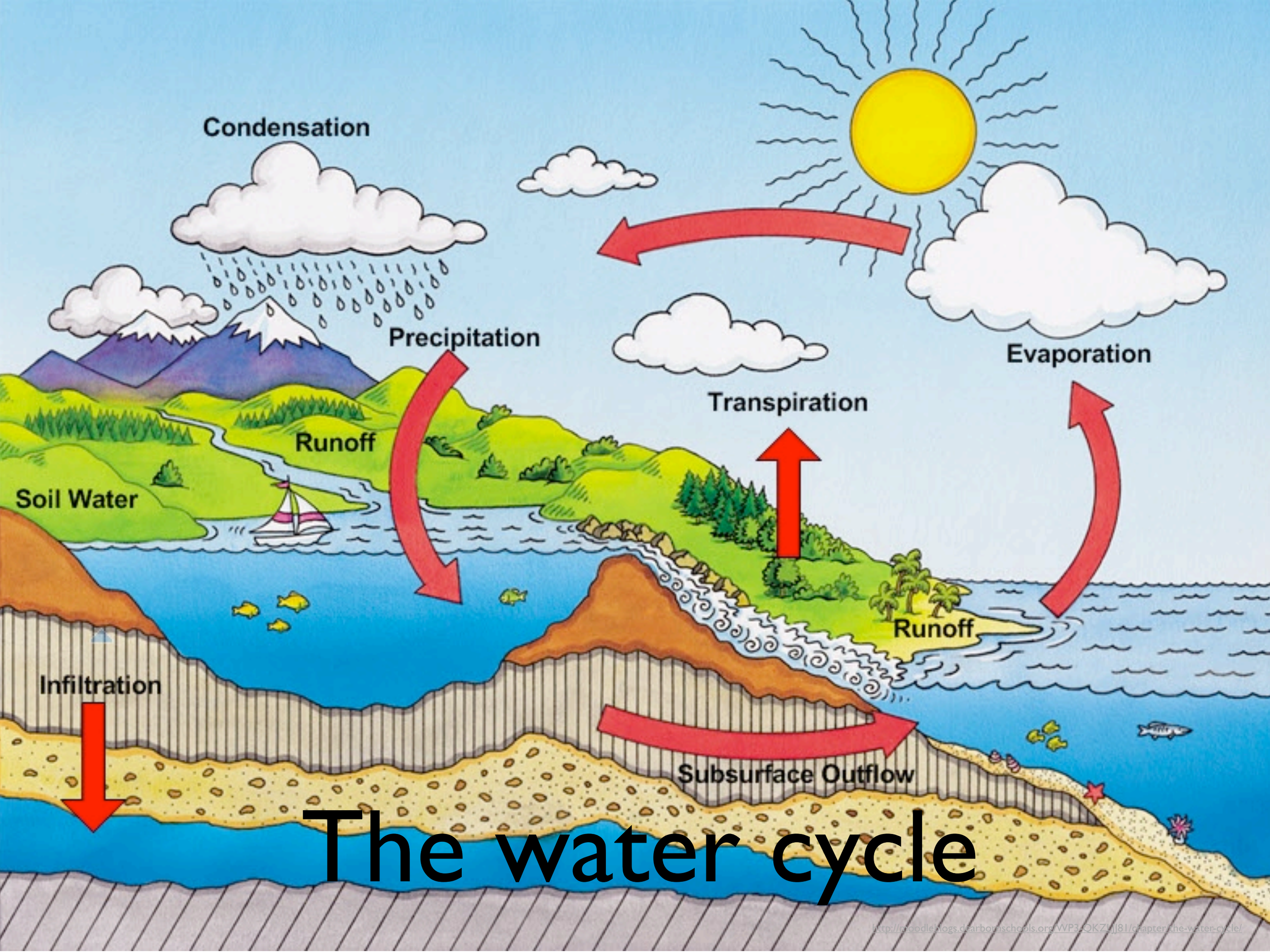
everything and whole

is **continuous**!

Heartbeat



http://www.clker.com/clipart-49488.html



The water cycle

http://moodleblogs.dearbornschools.org/WP3-OKZUJ81/chapter-the-water-cycle/


Season
http://all-wallpapers.ru/archives/223

What is integration?

The **act** of combining or adding parts to make a **unified whole**.

http://www.czesci24.pl/

Examples of integration

- Social integration
- Racial integration
- Economic integration
- Educational integration
- ...

Lego bricks

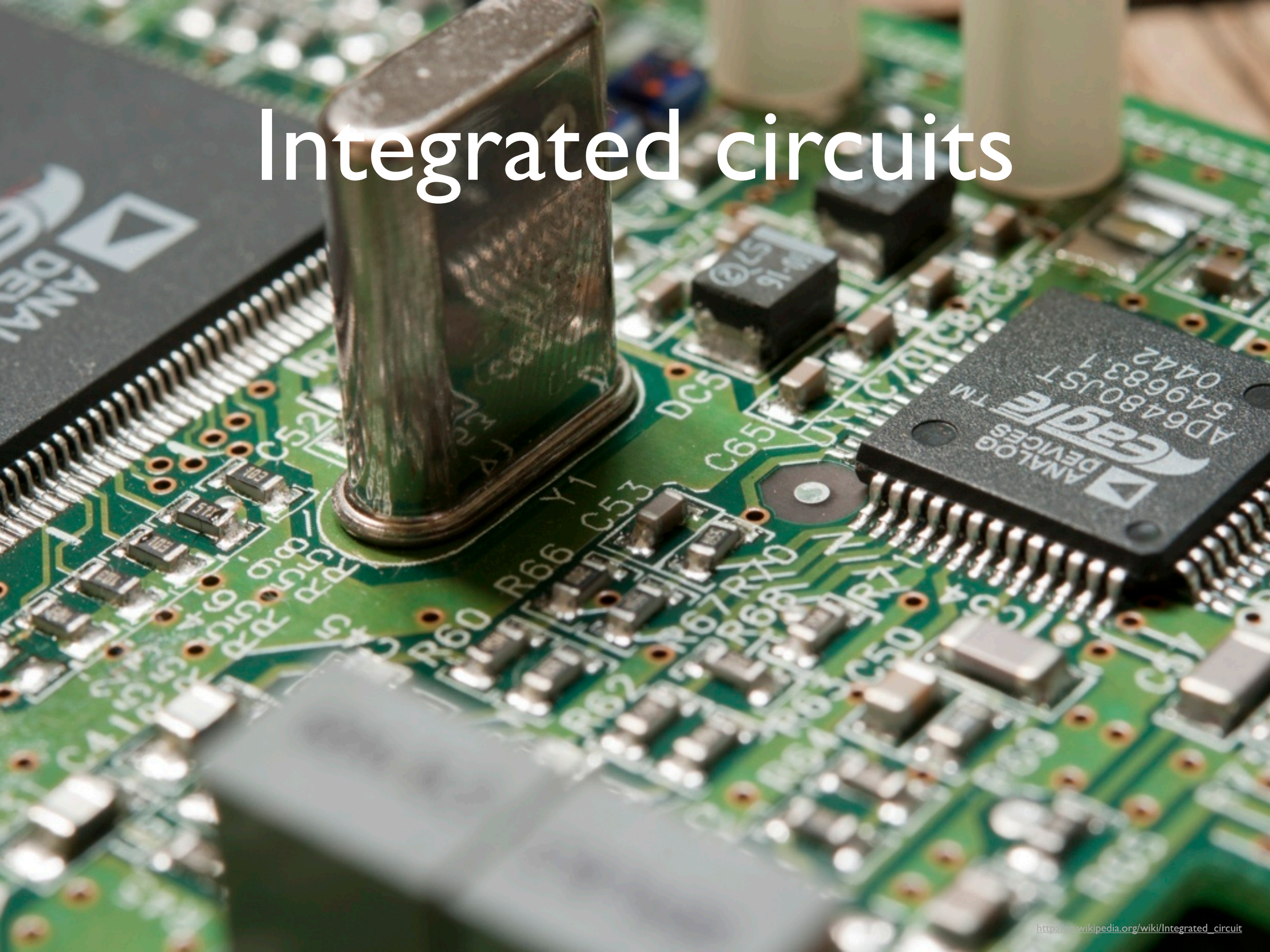
http://en.wikipedia.org/wiki/Lego

Start of integration

http://365thingsaustin.com/event/lego-kids-fest-texas/

Finish of integration

Integrated circuits



http://en.wikipedia.org/wiki/Integrated_circuit

Smoke testing

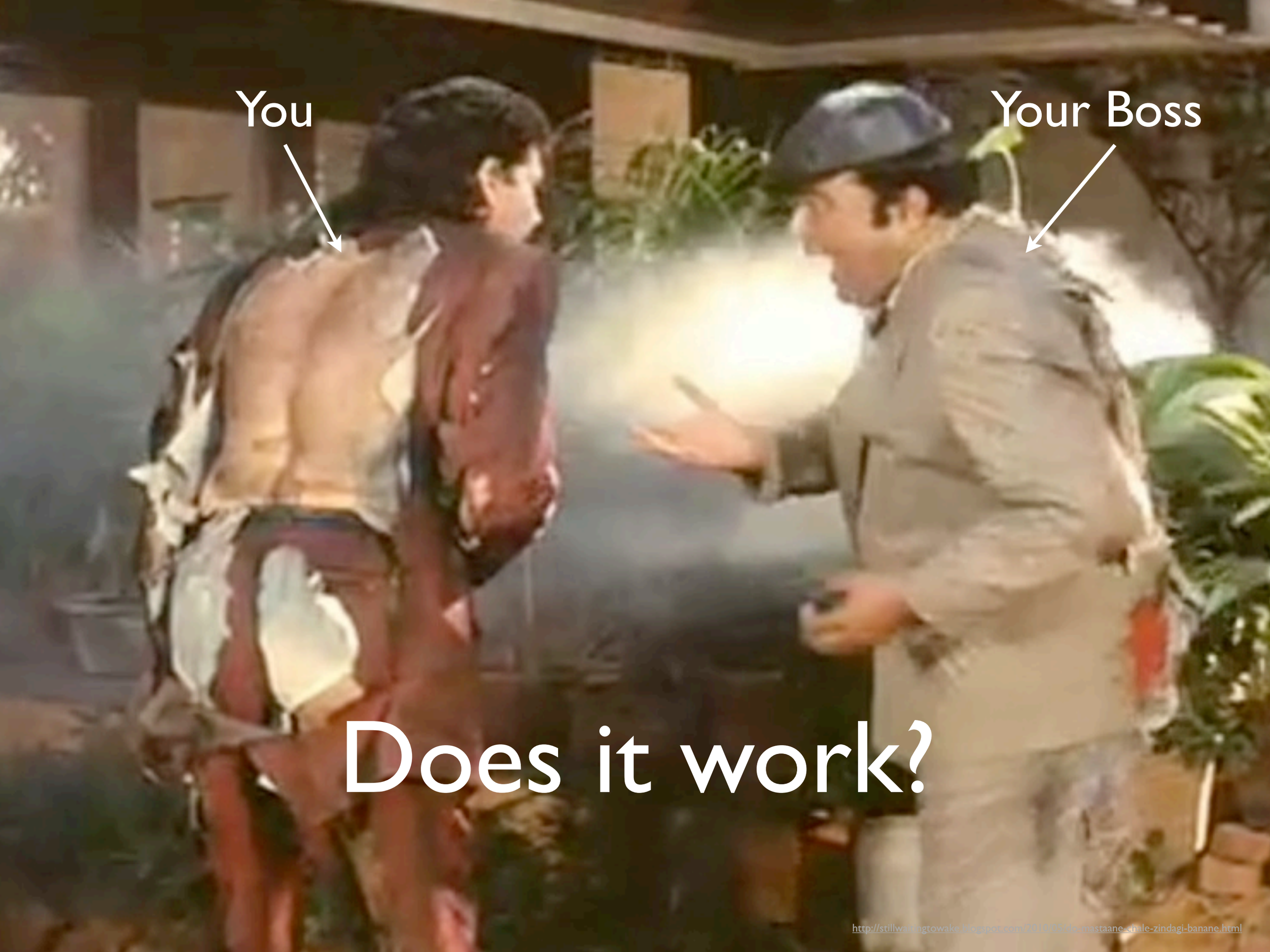
You plug in a new board and turn on the power. If you see smoke coming from the board, turn off the power.

The phrase smoke test comes from electronic hardware testing.

What is the most **useful** in integration?

Feedback

It’s works or not.



Does it work?

http://stillwaitingtowake.blogspot.com/2010/05/do-mastaane-chale-zindagi-banane.html

Secret of life

Continuous + "*some action*"

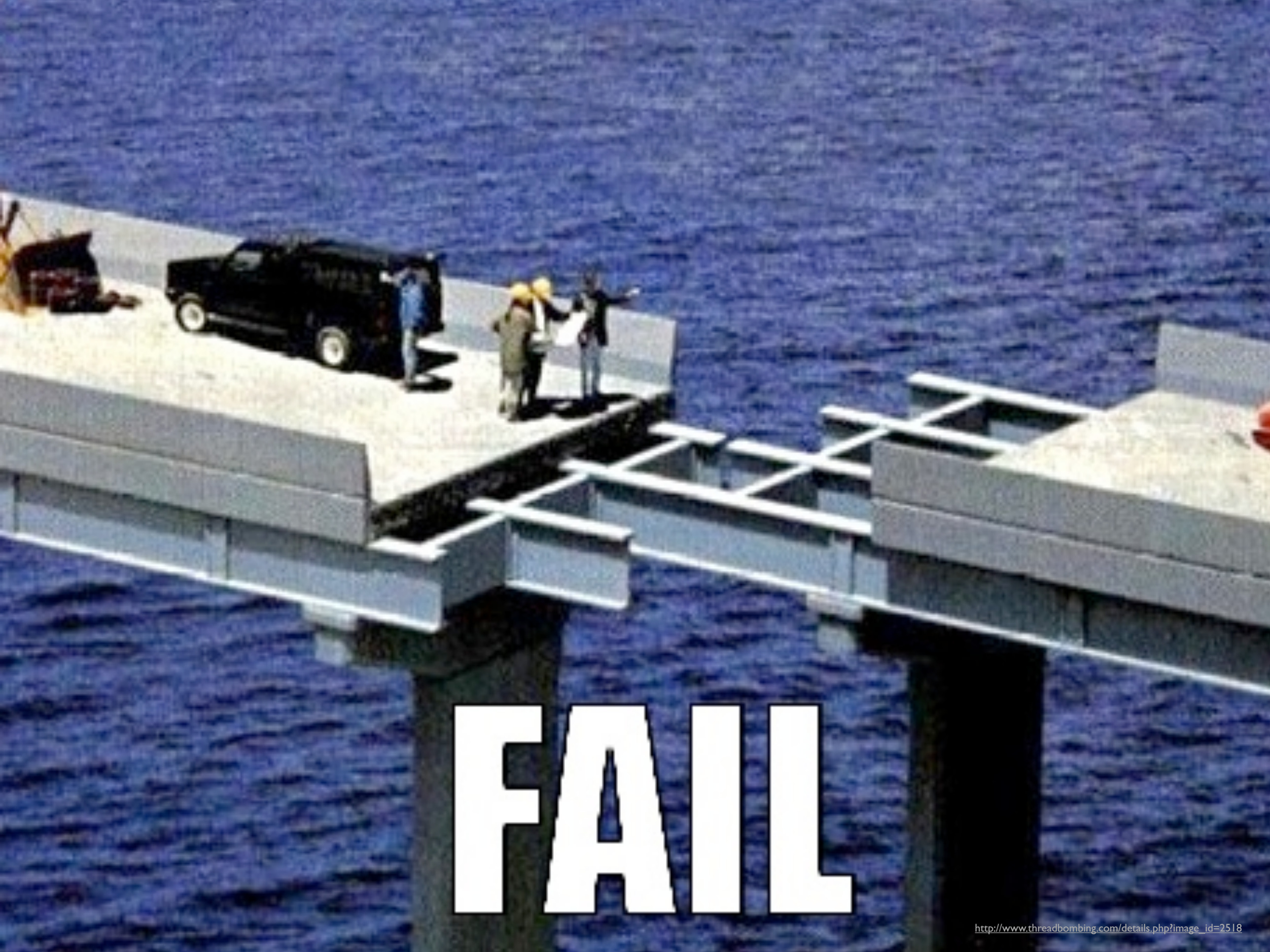
= it's power

Little strokes fell great oaks


Вода камень точит
Water cuts through stone
http://vk.com/aether_quinta_essentia


水滴石穿 (Shǔīshíchuān)
Water cuts through stone
http://www.glzjzx.com

What happens if the **integration** is **rarely** done?

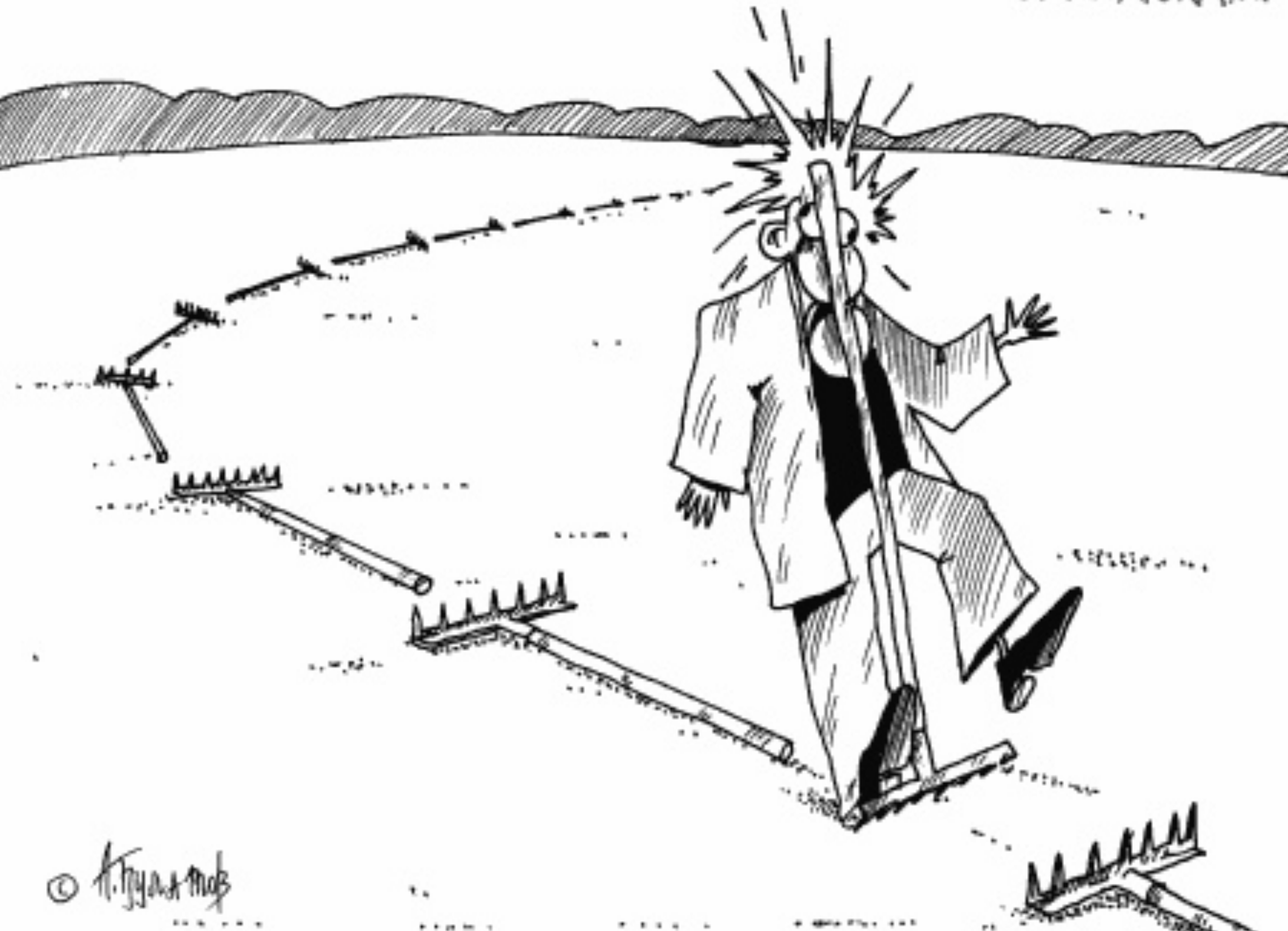


http://www.threadbombing.com/details.php?image_id=2518

Fail

http://deartheonion.wordpress.com/2011/03/01/engineers-fail-to-bridge-gap-between-rich-and-poor/

And how does it look in the software?



Project I

- **Banking** software
- Team*: 32 people
- Age**: 2+ years

** - all the people who touched the project*

*** - when I started to make CI*

Project 1

- **Big technical debt**
- Line of code: 100+k
- Unit test: 1 *(just only one)*
- Independent projects: 5
- Build project: 20+ min.

What was done?

- Compiling*
- Run tests**
- Notifications (when build failure)

* - *it turns out, was a challenge.*

** - *not only unit tests.*

Project II

- **Insurance** software
- Team: 9 people
- Age*: 7 months

* - *when I started to make CI*

Project II

- Technical debt (not big)
- Line of code: 7k
- Unit tests: 400+
- Independent projects: 2
- Build project: 20+s.

What was done?

- Compiling
- Database integration
- Run unit tests
- Notifications (when build failure)

Project III

- Startup
- Team: 5 people
- Age*: 0

* - *when I started to make CI*

What was done?

Two independent branches of the CI:

- backend
- frontend

Backend (1)

- Compiling
- Database integration
- Run unit tests
- Duplicate code analysis
- Static code analysis

Backend (II)

- Comprehensive statistics reports
- Test history
- Build failure conditions (more powerful)
- Produced artifacts

Frontend

- Build project*
- Run unit tests
- Statistics reports and test history
- Produced artifacts

* - *tasks like: managed dependencies, minification...*

What I learned?

Lesson 1

Never too late to **start**

Lesson II

But it is **easier** to start at the **initial stage**

Lesson III

Big **technical debt**

is very **bad** thing :)

Lesson IV

To create a **habit** in a **team** to CI is a big challenge

Lesson V

CI - it's a living organism

If you stop taking care of it,

it quickly dies.

Lesson VI

CI should be **fast**

(a few minutes)

Lesson VII

CI helps to **stabilize** the **release** process

A few tips

How to start?

- Do not wait!
- Do not hesticate!
- Do not panic!

What to do?

- Source code compilation (if it's possible)
- Database integration
- Testing
- Inspection
- Notification

When do build?

- At every check-in (commit)
- Every time a dependency changes
- Every X minutes

How do build?

- Use a single build script
- That can run from console
- Do not depend on an IDE

What is CI?

- It’s a process
- It's patience (to people, to code, ...)
- It’s set of tradeoffs (reasonable decision)
- It's a big effort at the beginning and then a lot of benefits

What is **not** CI?

- Nightly builds
- Once made and forgotten about it :)
- The way when broken build can be corrected at the end (of the day, of the week, ...)

Books (1)

- **Continuous Integration: Improving Software Quality and Reducing Risk** by *Paul M. Duvall*, *Steve Matyas*, *Andrew Glover*.
- **Software Configuration Management Patterns: Effective Teamwork, Practical Integration** by *Stephen P. Berczuk*.

Books (II)

- **Continuous Integrationin .Net** by *Marcin Kawalerowicz, Craig Berntson.*
- **Configuration Management Best Practices: Practical Methods that Work in the Real World** by *Robert Aiello, Leslie Sachs.*