Small steps towards better technical presentations

Dirk Haun, ACCU 2016

The Problem with Bullet points

- people will start reading from the top
- won't listen while you talk about the first items
- then they already know what's coming
- so they won't listen for the rest either

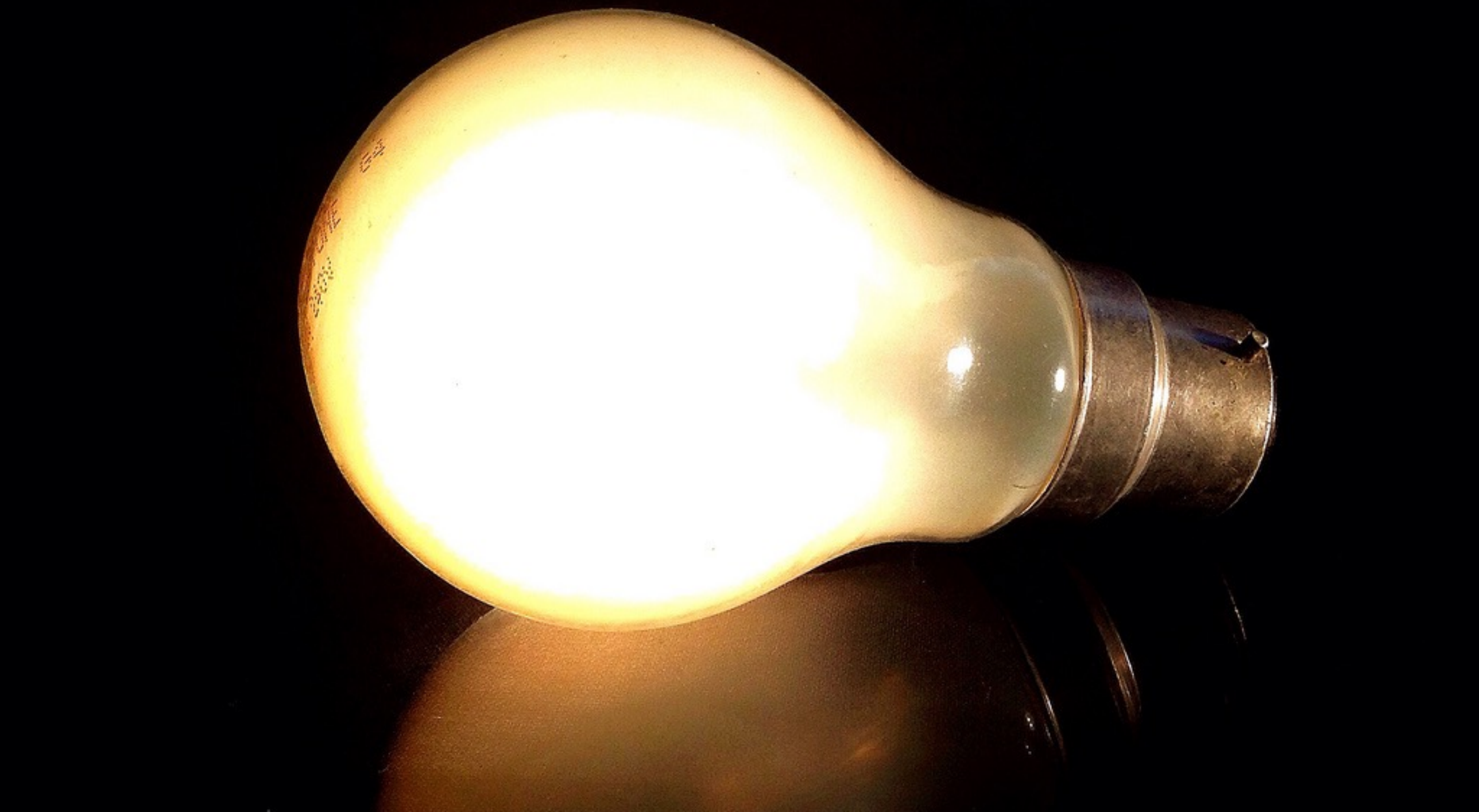
Solution?

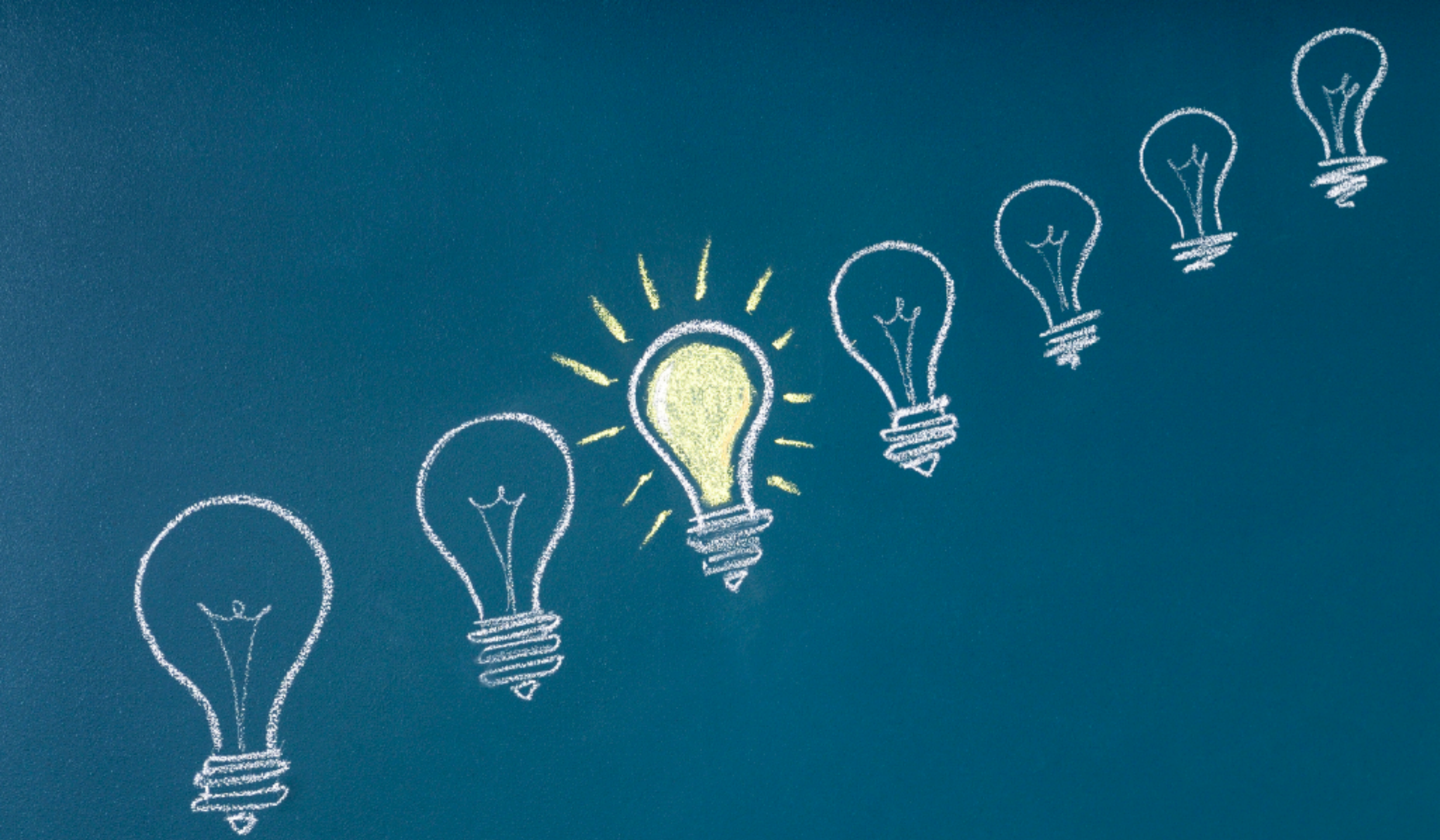
- List only one idea per slide

One idea per slide

One idea per slide

One idea per slide





One idea per slide

Just to be clear ...

- (short) lists are okay
- code is okay

Know what to call slides with a lot of text on them? Your notes! If you feel you need slides with text, it's probably because you need notes. **Don't show the audience your notes.**

Susan Weinschenk

presentation

"What is the use of a ~~book~~,"
thought Alice,
"without pictures or conversation?"

go for the
easy targets

Further lessons

- Take the easy option (always!)

Take the easy option (always!)

Hooks

Hooks

C++11

- Is a better approximation of my ideals for support of good programming
  - Significantly better than C++98
- Has tons of distracting "old stuff"
  - Going back to C in 1972
- We must focus on the essentials
  - And the "good stuff"
  - "Elegance ***and*** efficiency"
- C++11 is not the end, we can do much better still
  - Anyone who says *I have a perfect language* is a fool or a salesman
- Stability/compatibility is an important feature in itself
  - And not free

BJARNE STROUSTRUP

THE CREATOR OF C++

Using
C++ 11
and
C++ 14

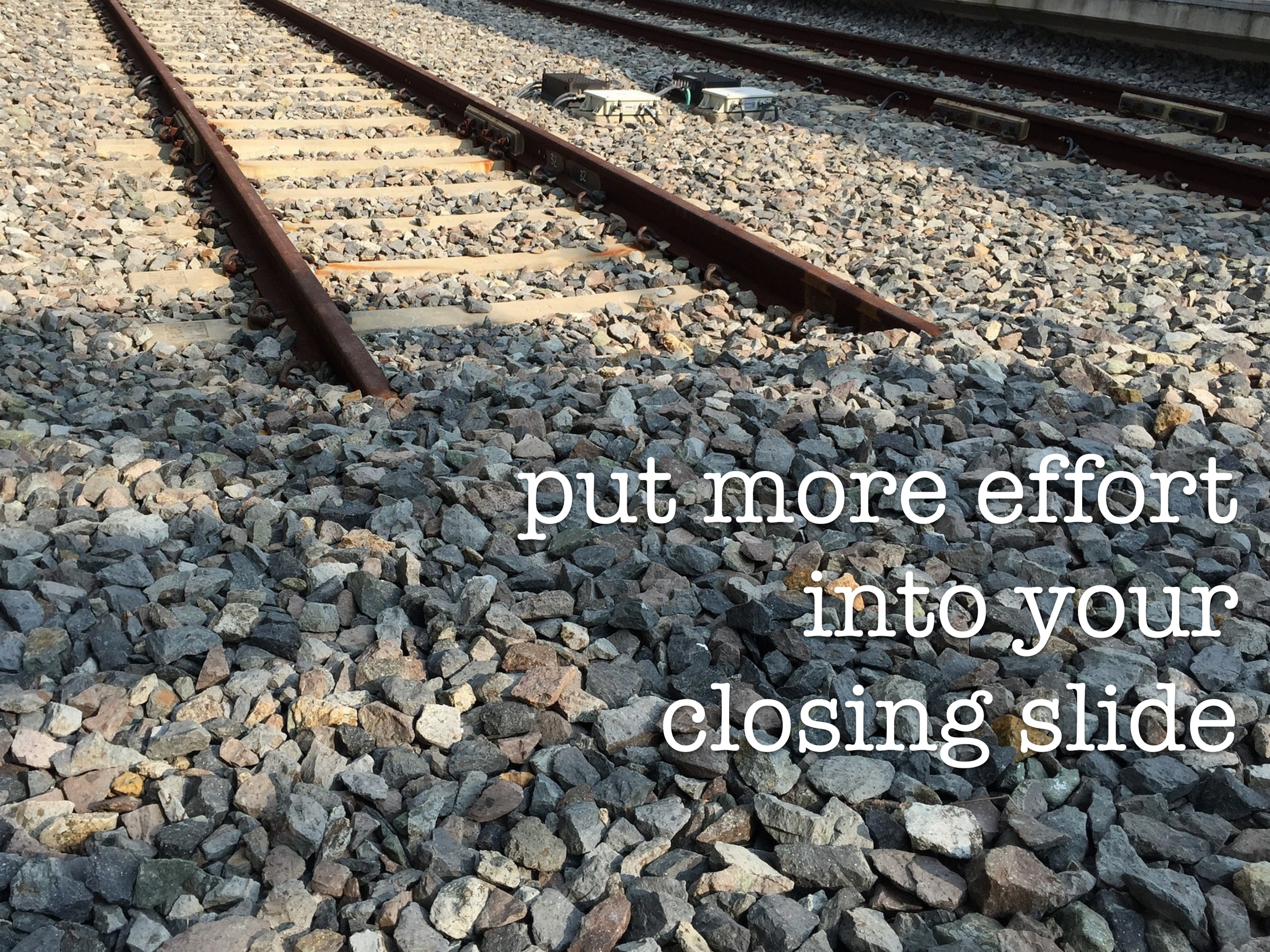
PROGRAMMING

Principles and Practice Using C++

SECOND EDITION

The standard committee not breaking your 20-year-old code is a **feature**.

We can do much better still.

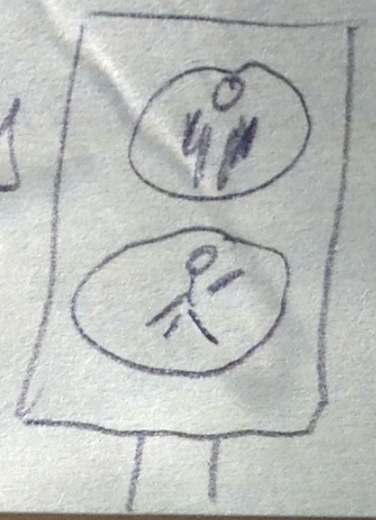

put more effort
into your
closing slide

DTRT

- Worry!
- Don't worry!

(Logo)

Worry



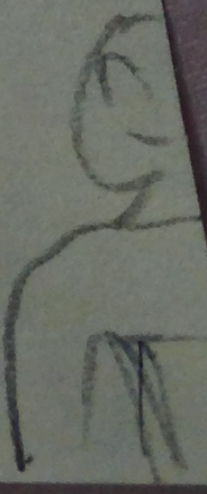
Don't worry

Sometimes you have to worry and sometimes you don't

Worry

Don't worry

Do The Right Thing



Sometimes

you have to

worry

and sometimes you

don't

Do The Right Thing

worry

don't
worry

DON'T

WORRY

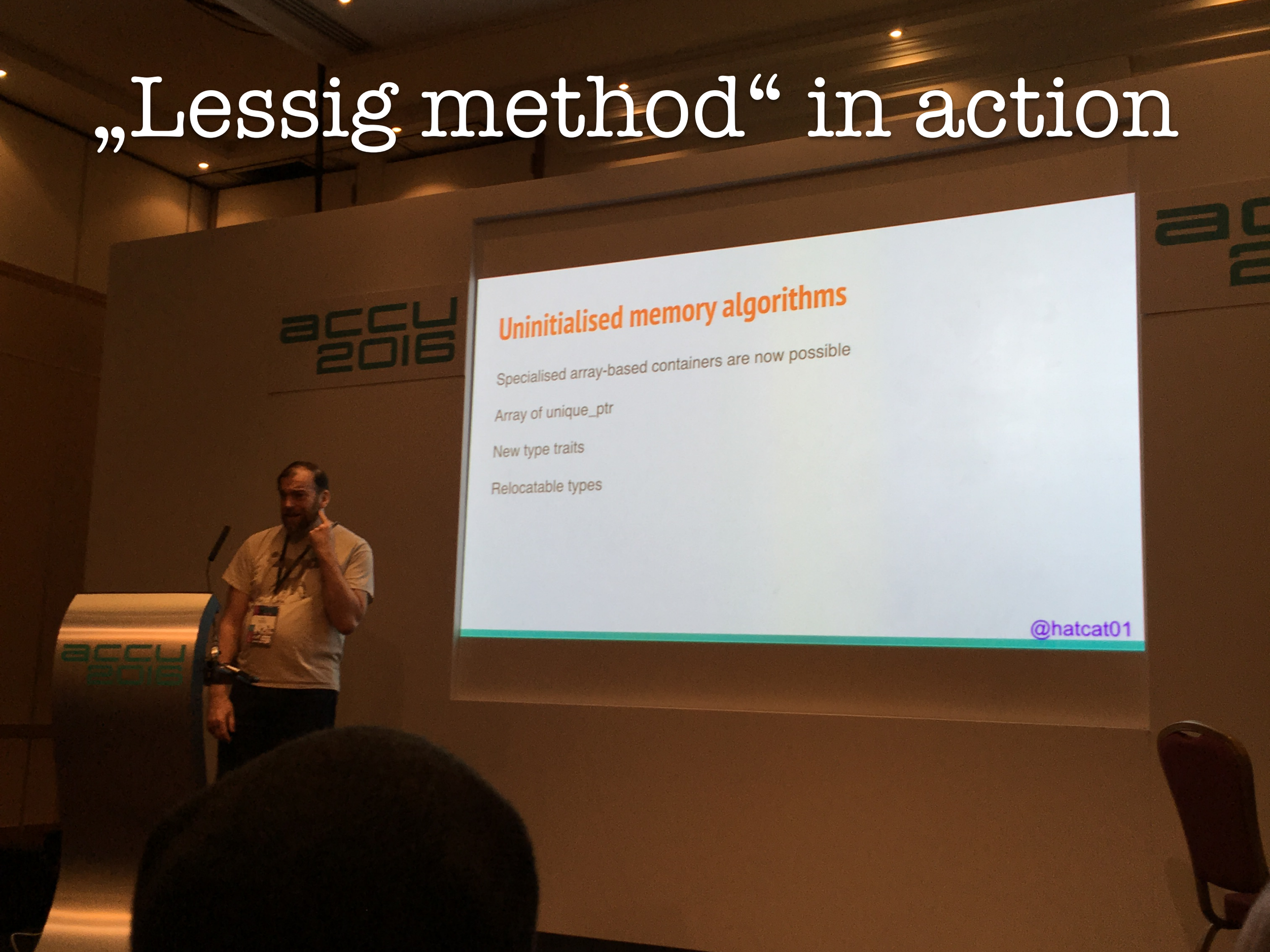
WORRY

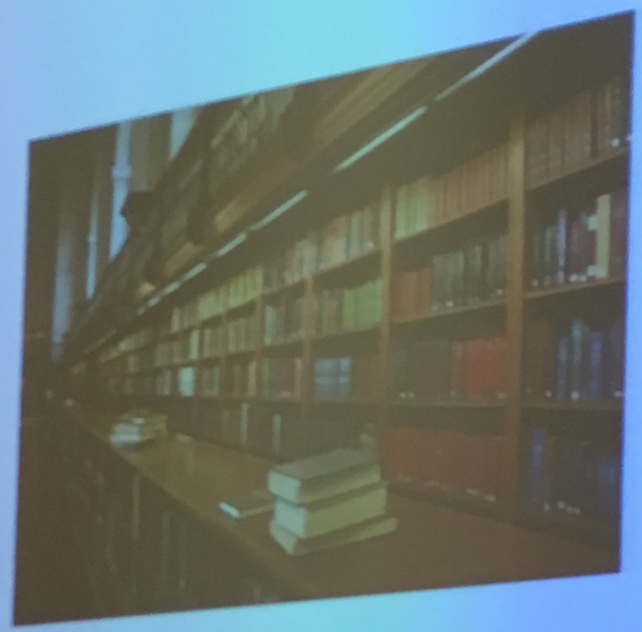
Hara hachi bu

„80% full“

Bonus: Observations from ACCU 2016

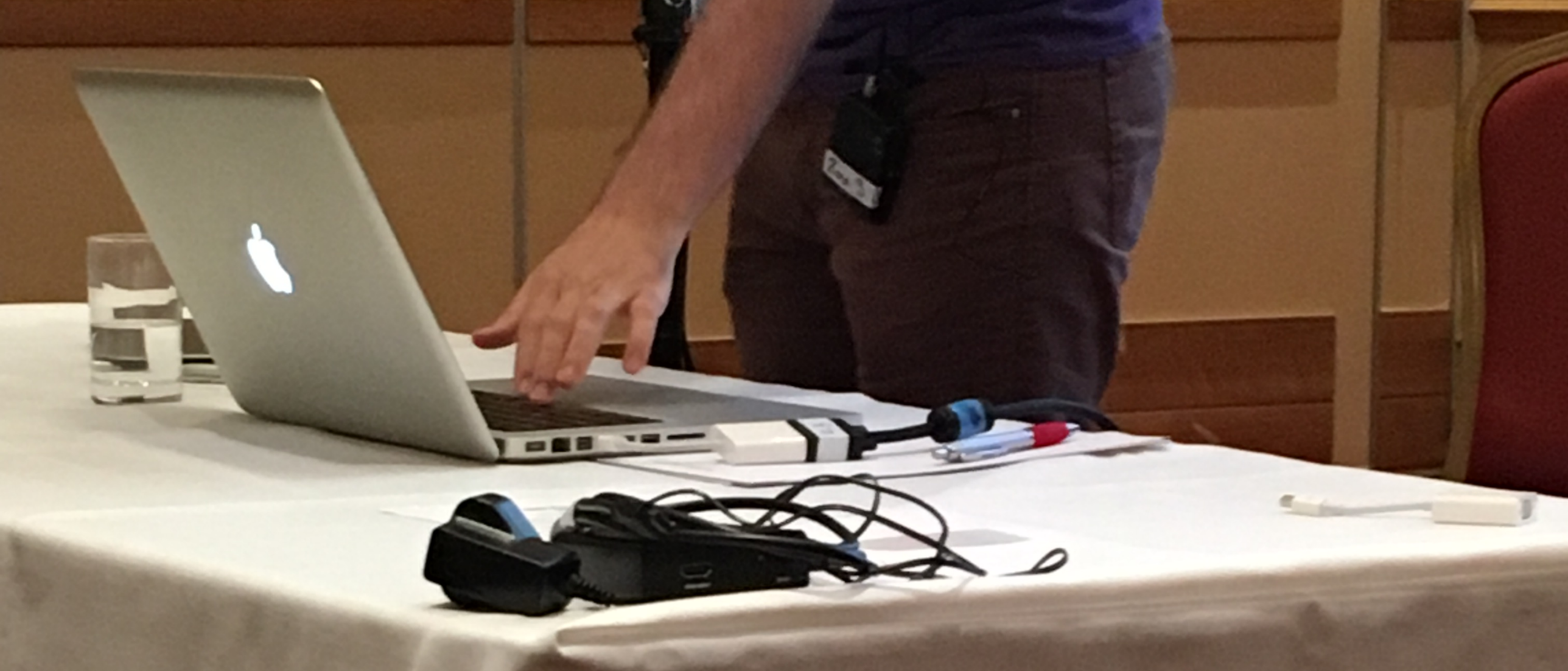
„Lessig method“ in action





Why so small?

Time Management



Use a remote, please

Beware of the laser pointer

Both class and Context instances can play Roles

- … in which case, *we really don't need classes any more*
- *They never were part of the OO vision*
- The **trygve** language supports more or less arbitrary scope nesting (anything inside anything)
- Classes become truly *operational models*. exactly the Piagetian ideal that Kay strove for

thankyouanyquestions?

Don’t show the audience your notes!

Go for the easy targets.

Hara hachi bu (80%).

www.TheMobilePresenter.com
dirk@haun-online.de
@dirkhaun

Hello
World!
Dirk Haun
Presenting for
Geeks

Photo Credits

Footprint Discovery by J N, from Flickr, CC-BY,
Lightbulb by James Bowe, from Flickr, CC-BY,
"Magic lightbulb" by Grant Rambojun, from Flickr, CC-BY,
Light Bulbs Sketched on Blackboard by Lisa-Blue, iStockphoto file #18887141,
Susan Weinschenk portrait from her website,
Alice illustration, originally by John Tenniel, now in the Public Domain,
Dart Board by skeeze, from Pixabay, CC0
Prickly Hooks by Derek Gavey, from Flickr, CC-BY,
C++ book cover (c) Pearson,
Road Marking by 44833, from Pixabay, CC0,
End of the railroad by happyday79, from Pixabay, CC0,
Teenage Girl Scratching Head with Thinking Expression by DaydreamsGirl, iStockphoto file no. #8930276,
Rettet die Ampelmännchen! by Kyle Gradinger, from Flickr, CC-BY-NC-SA,
The Sun and the Moon by Toshihiro Oimatsu, from Flickr, CC-BY,
Meeting by Alexander Kaiser, from Flickr, CC-BY,
Laser pointer by Paul Reed, iStockphoto file #2699367,
all other photos by Dirk Haun

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.