Making (large) legacy systems beautiful

Lars G. T. Jorgensen
lj3@sanger.ac.uk
• Participated in the Human Genome project
• CERN of biological data
• Open Data, Open Science, (Open Source)
• Ranked UKs most influential research institute.
There are currently 136 tracks switched off, use the menus above the image to turn them on.
Warning:
Contains personal opinion
Legacy?
Large?

• > 600 Gb - 1 Tb OLTP databases
• > 75 Tb of raw data per week
• > 10000 samples
• Complicated analysis
• Novel applications => Extreme amounts of change
Beautiful?
Time is money
Refactor or recycle?
Make mistakes early
ATTENTION
Your mother doesn’t work here.
Please clean up your own mess!

Remove old code
Ignore software fatigue

See the project with fresh eyes
Set up a staging environment

- Take thing to bits
Is it testable?
Adding tests

• Start from the top

• Be creative when it comes to mocking

• Use language features

• (See a TDD/BDD talk)
See all the code
Under utilized technology

• Are there some features in the current technology stack that can help you.
• Views, PL/SQL, Partioning, Stored procedures
• Facade design pattern
Try and reducing the problem size
Is feature useful or have it just become “the standard” way
Talk to the users
Talk to the developers 
(if they are still around)
My Legacy dogma

• Work towards a complete mental model of the system
• Keep it simple
• Break it early
• Always move forward
• Choose small targets
• Don’t take it personal
Come work with Perl

• Come and help human kind by programming Perl (and other things)!