

An Agile Approach to a Legacy System

Chris Stevenson
skizz@thoughtworks.net

Andy Pols
andy@pols.co.uk

The Chess Clock

Replace a broken legacy system

- Trading application
- All business logic in stored procedures
 - 1600 of them
 - 3000 lines long
 - multiple versions
 - no foreign keys
 - 1 database user (sa)
 - lots of external apps talk directly to the db
- Batch
 - 4GL, C, shell

Don't reproduce legacy code

Tried to rewrite
the system in Java

Failed

Refactor Legacy Systems
by
Delivering New Business Value

Always ask the users
what the problem is

Incrementally build trust

Treat Politics
as a
User Requirement

Don't say no, say "Later"

Brainstorm Antibodies

Use Live Data

Involve the Whole Team
in
Large Refactoring/Redesigns

**Build a Small,
Self-selected Team**

Effective Teams
need
Break Points

Don't get hung up on Process

Variable size Iterations

No pre-allocated tasks

Team Estimating

If you get it right,
and make it compelling,

users won't turn it off,
and
will fight your battles for you

"They re-wrote the system
without touching the code"

Department Manager
(now promoted)