
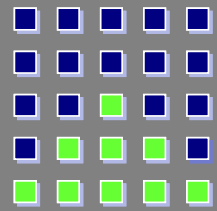


www.software4software.com

A blue robotic hand is shown holding a computer keyboard. The screen of the keyboard displays the word "software" repeated multiple times in a grid pattern. The background is a dark gray gradient.

IBM's 1962 recruiting ad said:
" IBM programmers ... are
devising programs that in
turn use machine capability for
formulating new programs. "

Source: Gerald M. Weinberg, "The Secrets of Consulting"



DELTA
software
technology

The Generator Company

Automatically error-free – a vision?

MDD&PL 2009 Leipzig, 24.3.2009

Rüdiger Schilling

Delta Software Technology GmbH

MDD, PL and GP vs. maintenance and modernisation

MDD, System Families & Generative Programming

Primarily concepts for **new development**

Questions

Can **models** be derived **afterwards**?

What are the roles of **system families/product lines**?

Can **GP** be used **in maintenance**?

What do we do with **millions** of 'old' **Cobol programs**?

The most important question: **Why?**

MDD, PL and GP vs. maintenance and modernisation

Software development from scratch

Biggest attention in publications and research

Many tools

Only 10-20% of Software-TCO

Maintenance, further development and modernisation

Low reputation → necessary evil

Almost no research

A few tools

80-90% of Software-TCO

Possible fields of application

Mass changes

Year 2000

UTF-16 rearrangement for EU harmonisation

Modernisation

Change of platform

Framework adjustment/exchange

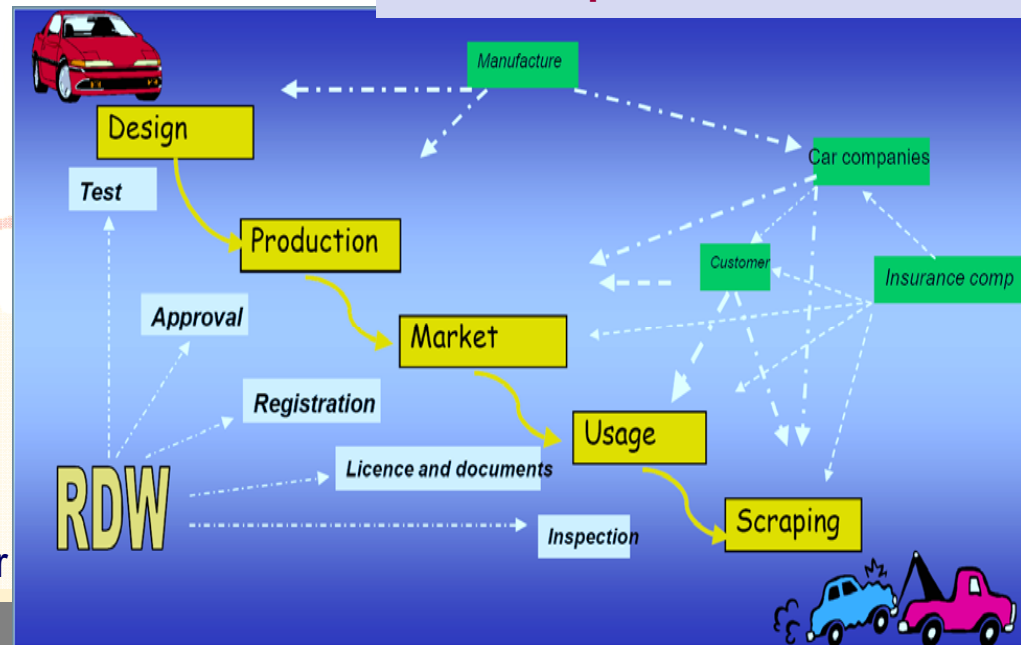
Architecture transformation

Practical example

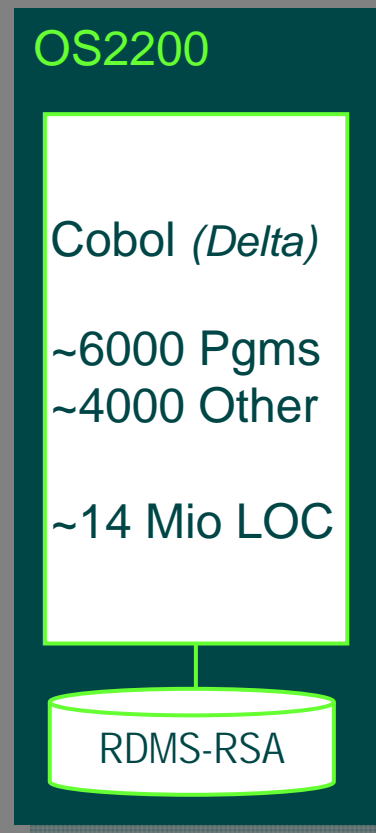
The customer: RDW (Netherlands)
≈ Agency of permission + TÜV + ..

Combination of **Technical Services Department** of the National Transport Inspectorate and the State Department for **Registration of Motor Vehicles and Trailers**, part of the **Public Works Department**

9.600.000 active vehicles
9.700.000 driver licenses
6.000.000 transactions per year
3.500 telephone inquiries/day
7.200.000 APK-inspections/year
750 car inspections/day
23.000 type approval & test reports/year
60.000 exemptions for special transports per year
150.000.000 inquiries by police/year



The task: An easy migration project?



Problem

Unisys OS2200 is too expensive
and futureless

Big, critical application

Task

Changing to new platform
Windows (possibly Unix)

Minimisation of risk

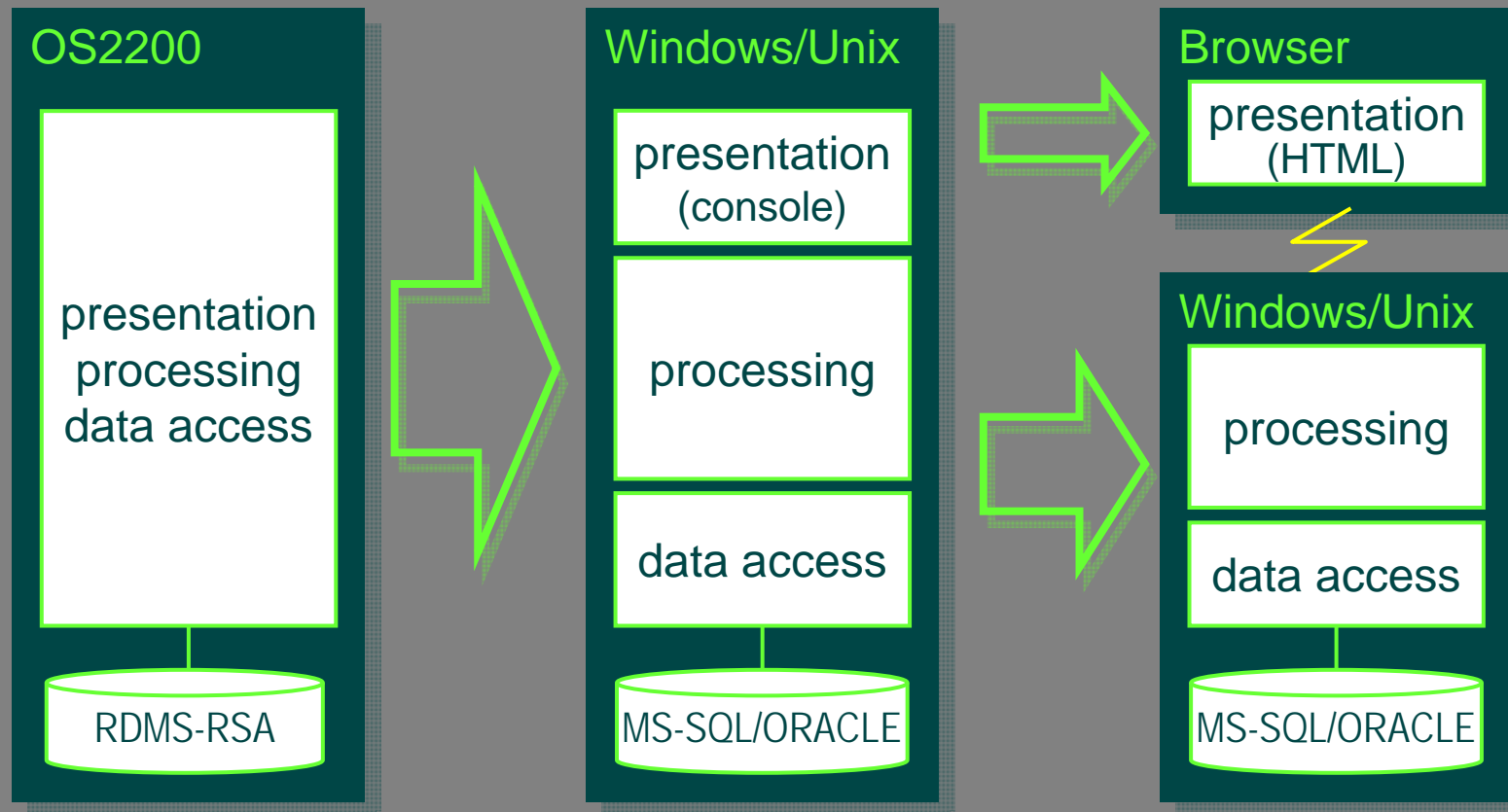
Long-ranging solution

Goal&Path (lesson learned)

Platform onafhankelijkheid

(platform independency)

"Platform onafhankelijkheid" = architecture transformation



Recommendation and application

Our proposal

automatic transformation

Model based

Platform specific elements encapsulated in generators

Grade of automation 80-90%

Requests

At least 95% automation or even 98%

No risks acceptable for production

Details of the task

Unisys OS2200

9-Bit hardware (9 bit/byte, 36 bit/word)

→ All numeric definitions have to be checked and adjusted where necessary

→ The result: structural conflicts

RDMS-RSA

“Old” SQL call interface, non-standard

Monolithic code

DBMS and dialogue operations are not separated

DBMS declarations and operations

Convert to **platform-independent** models

Separate, completely **generated persistence adaptors**

Screen definitions and dialogue operations

Convert to **platform-independent** models

Separate, completely **generated screen modules**

Dialogue operations encapsulated in platform-independent “macros”

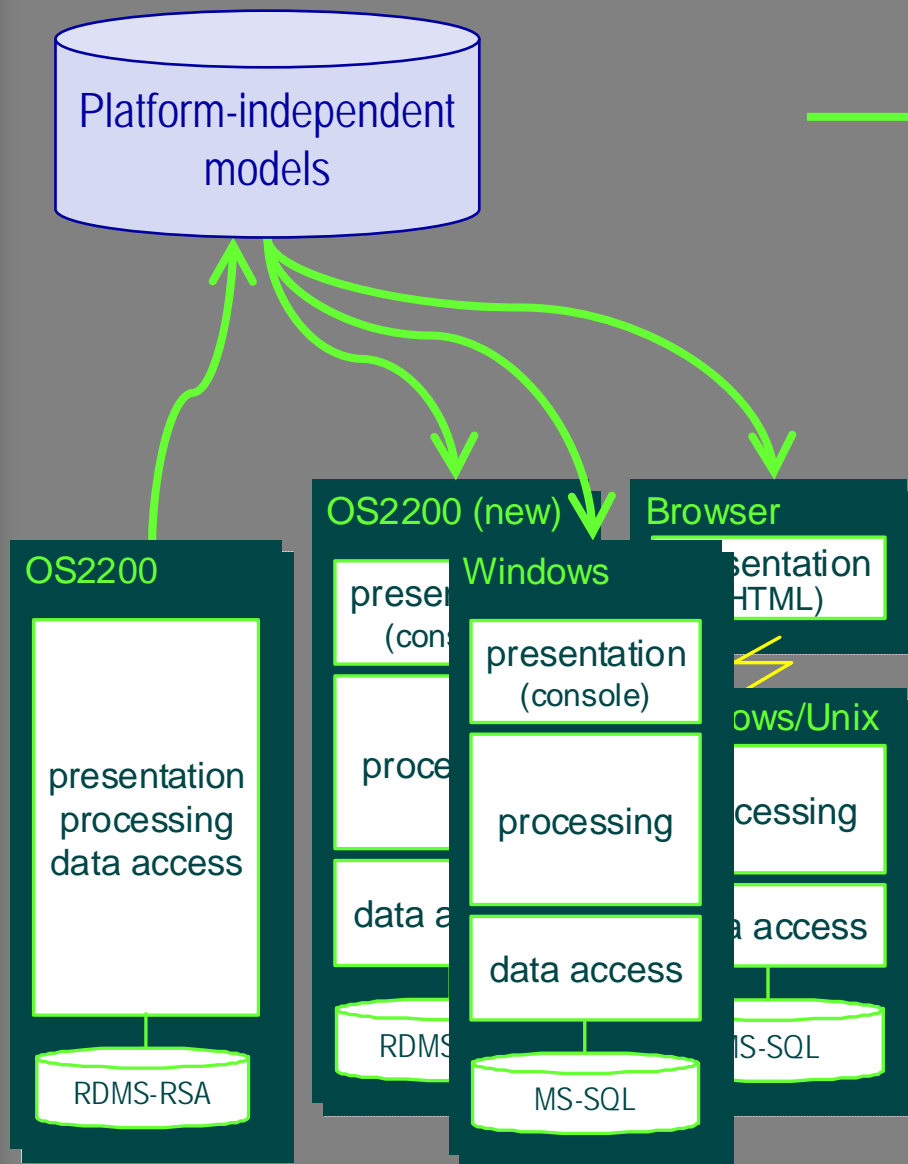
Critical data definitions

Adjusted where possible → complex static **code analysis**

Otherwise encapsulated **generated “façades”**

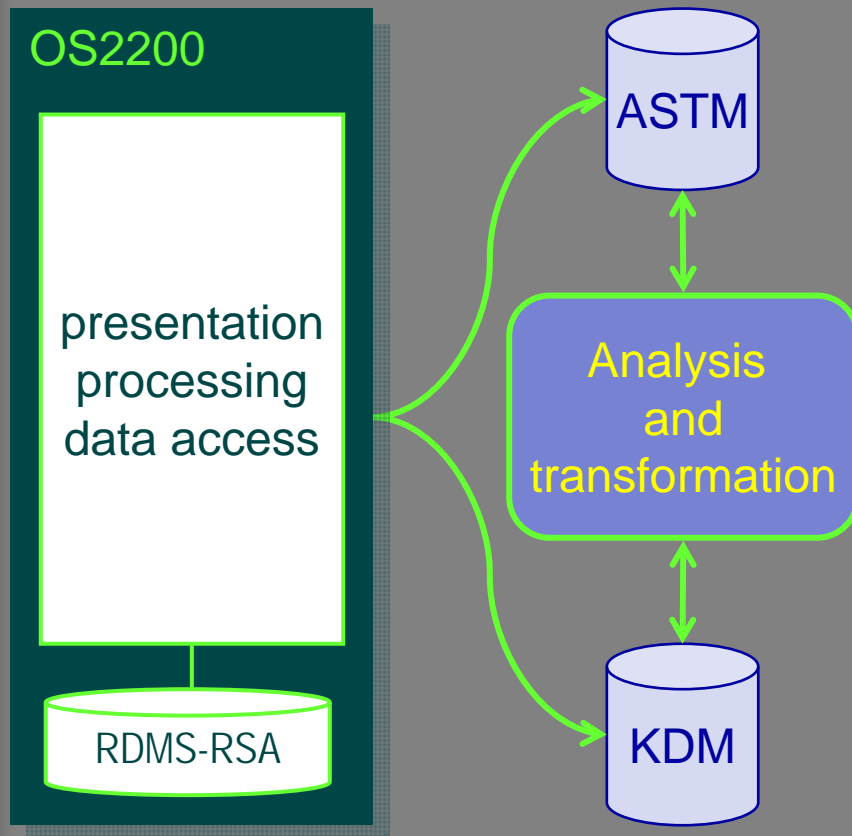
In-place migration

In-place migration



1. The application will be transformed into a platform-independent form
And will be maintained in this form
2. Via generation for the old platform...
“In-place migration“
3. ... and for new platforms
Without further interventions
“Single source – multiple target“

Process in 4 steps



1. step

Transformation of application into neutral formats

ASTM* – Abstract Syntax Tree Model

KDM* – Knowledge Discovery Model

2. step

Analysis and manipulation within the models

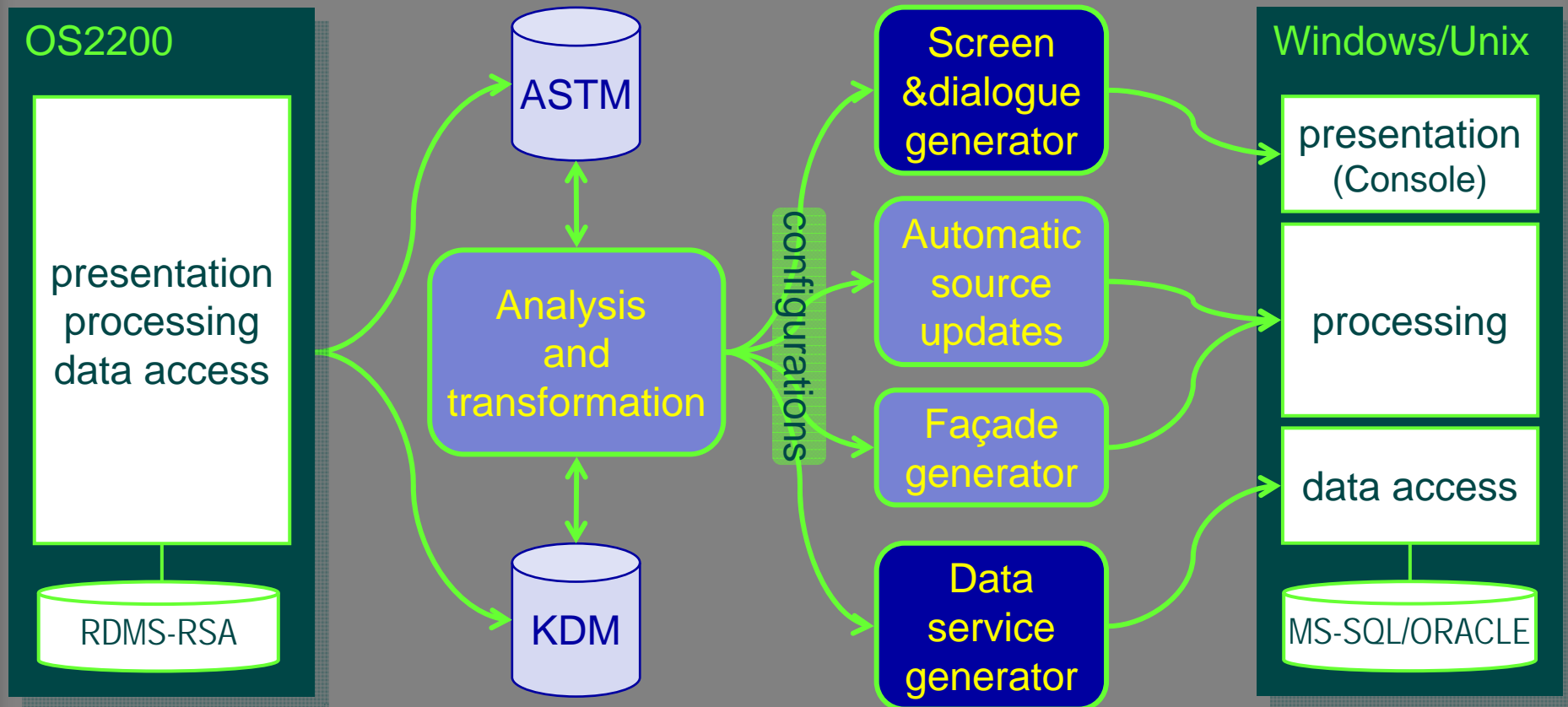
**ASTM and KDM*

Concepts from the OMG Standard ADM (Architecture Driven Modernisation)

Process in 4 steps

3. step

4. step



Amelio® Transformation Platform

Base of transformation factory
discovery, analysis, transformation,
updates

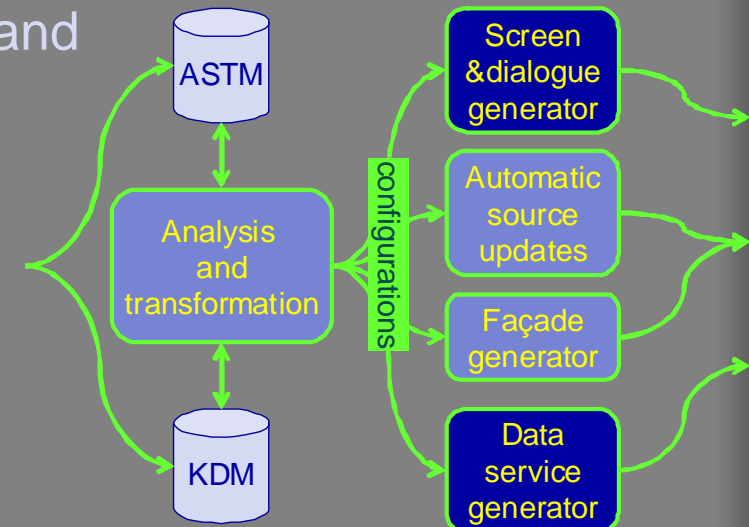
SCORE® Data Architecture Integration

Modelling and generating data services and
persistence adaptors

Realized with:

HyperSenses

Generator development system
For individual generators

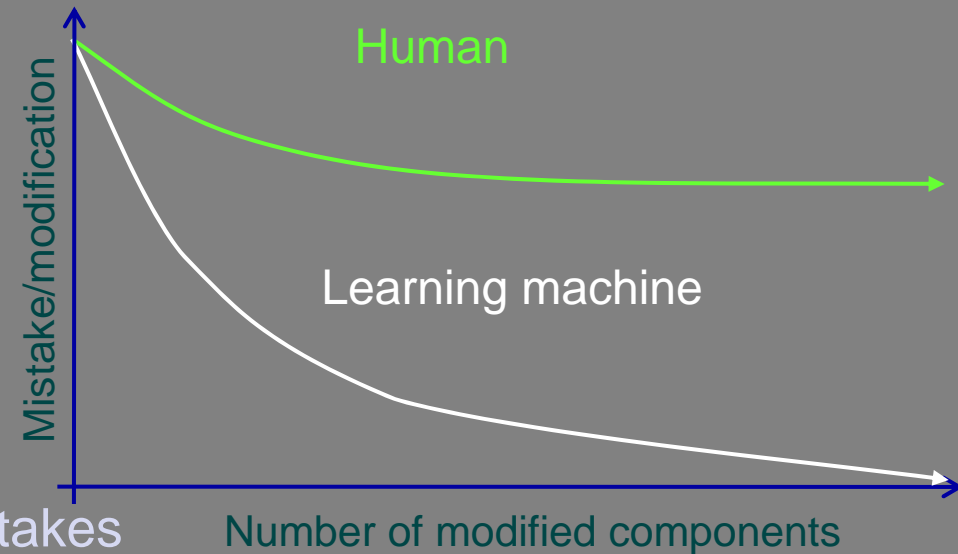


Learning effects (1)

Sometimes humans make mistakes

Machines do the same mistake again and again

Humans and machines are able to learn from their mistakes



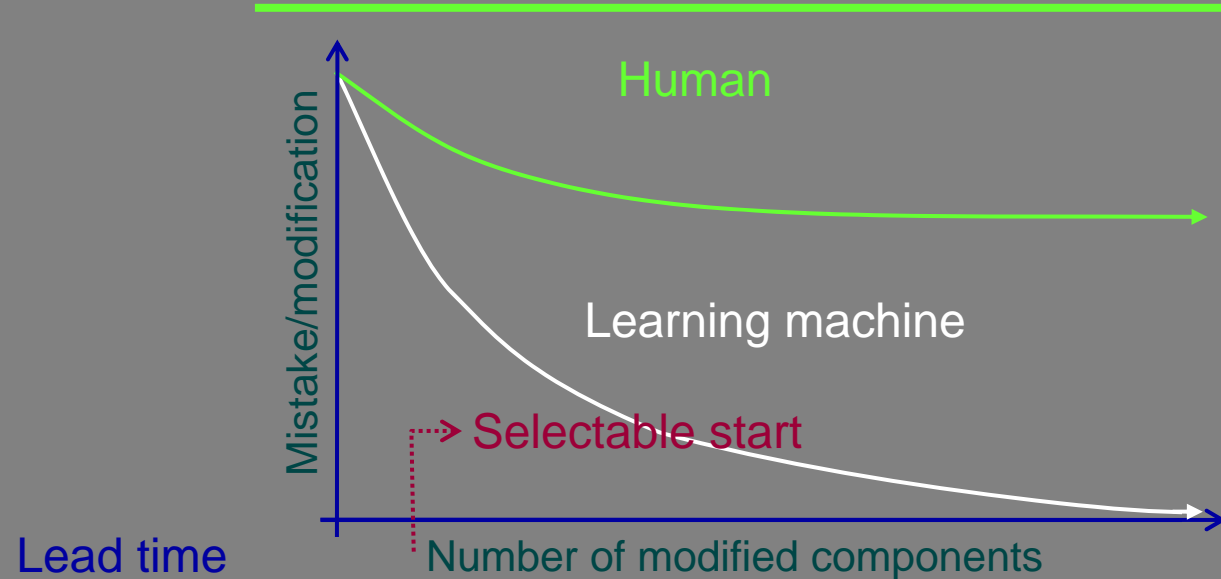
For humans: the larger the team, the less the learning effect
Careless mistakes and the like do not decrease

Learning machines are able to remove every identified mistake completely and it will never appear again

Machines do not make careless mistakes

Learning effects (2)

Automatic processes can be **repeated quickly, as often as necessary** and with less effort



Therefore:

Tests of the automatic modification can be **brought forward arbitrary** With **no influence** on the ongoing development or production

An **error probability** of (almost) **zero** is achievable - before the actual modifications start

Results in figures

Input

OS2200

Cobol (*Delta*)

~6000 Pgms

~4000 Other

~14 Mio LOC

RDMS-RSA

Output

~ 10.000 Modified components

~ 2.500 Persistence adaptors (data services)

~ 1.500 Screens & dialogue modules

~ 1,3 Mio Modifications in the source code

~ 2,0 Mio extra LOC generated

Errors reported at test and deployment

30 at modified components

10 at new components

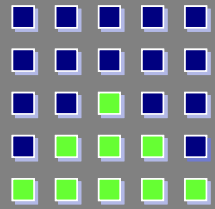
Error rate

30 errors / 1,3 Mio modifications

$$= 0,000024 \rightarrow \underline{0,0024\%}$$

10 errors / 2.0 Mio "new" LOC

$$= 0,000005 \rightarrow \underline{0,0005\%}$$

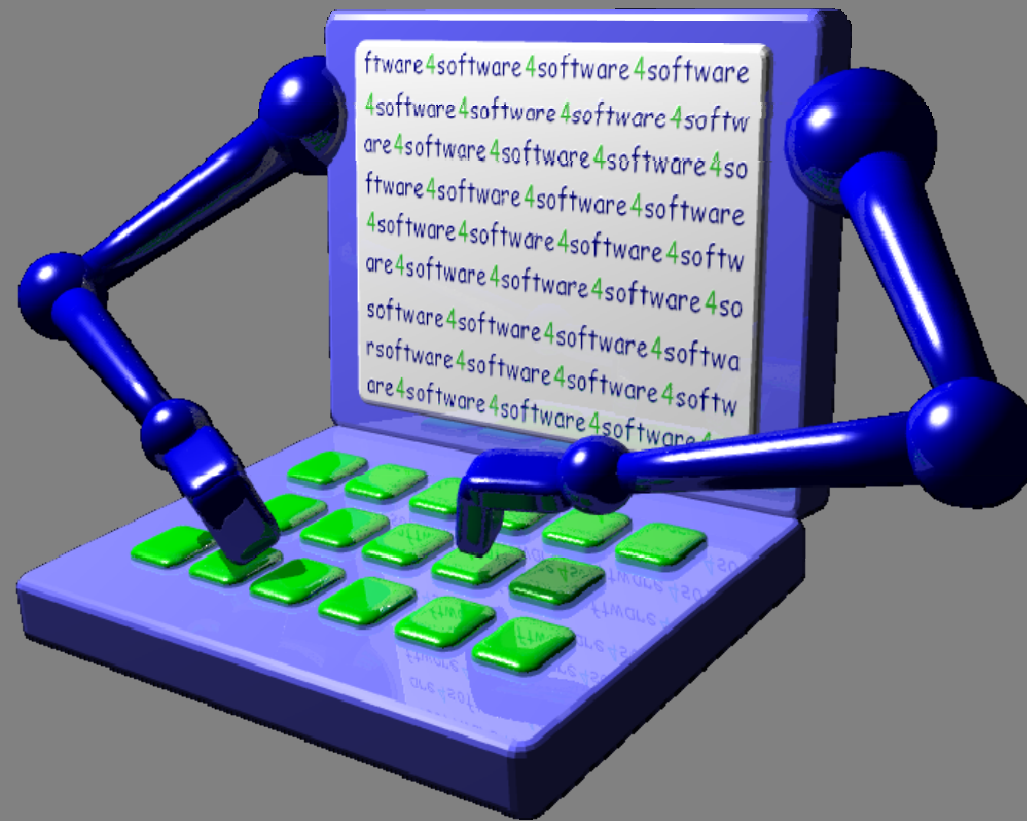


DELTA
software
technology

The Generator Company

Automatically and error-free
– a realistic vision?!

q.e.d.



www.software4software.com