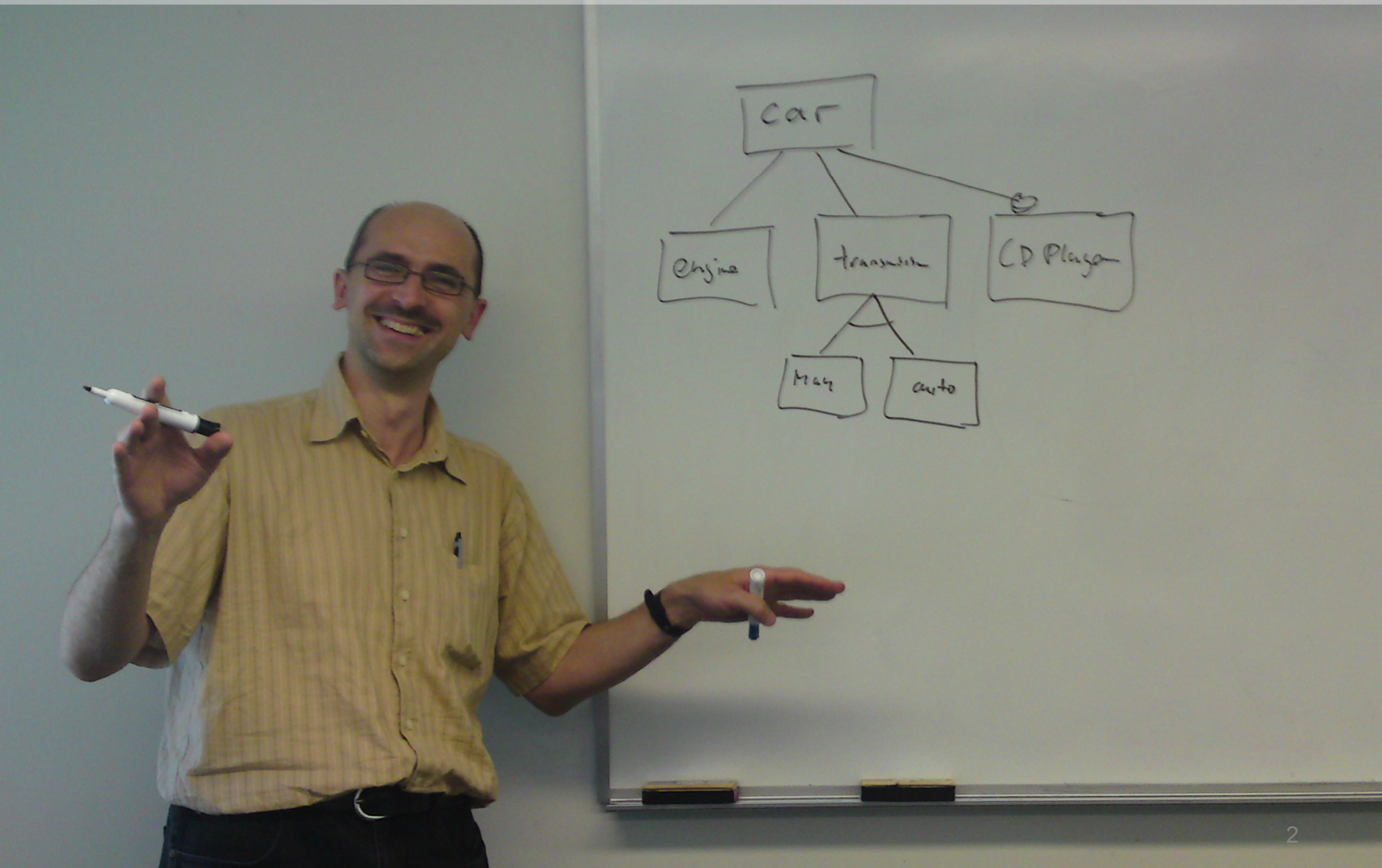
what is a feature?

a qualitative study of features in industrial software product lines

Thorsten Berger, Daniela Lettner, Julia Rubin, Paul Grünbacher,
Adeline Silva, Martin Becker, Marsha Chechik, Krzysztof Czarnecki

Professor, what is a feature?



we found 35 definitions of “feature”

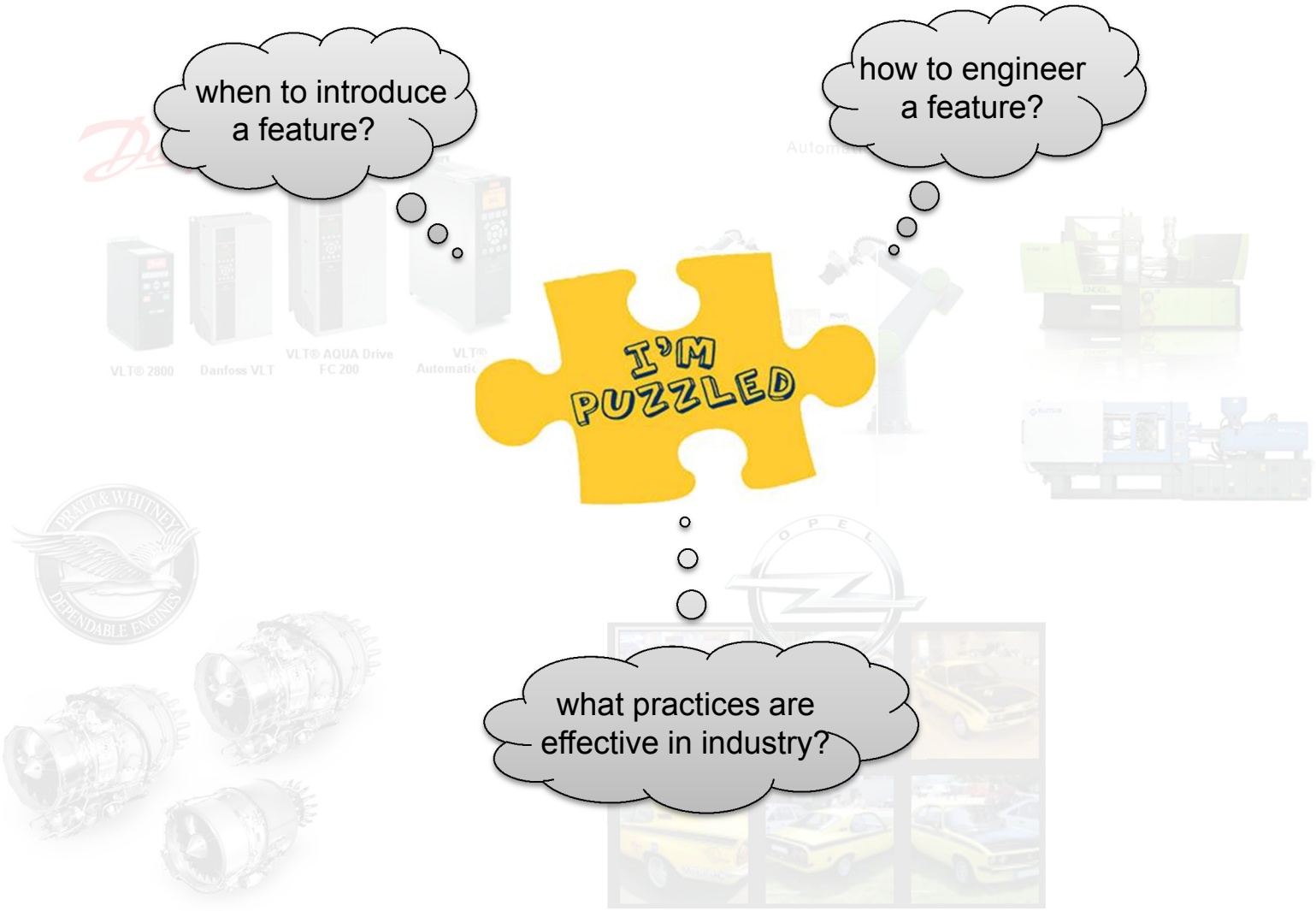
A feature represents an aspect valuable to the customer. [Riebisch03]

A Feature F of a product P is a product requirement $R \subseteq D$ that is visible to a user of the product P. [John10]

A feature is an increment of functionality, usually with a coherent purpose. [Zave99]

Customers and engineers usually speak of product characteristics in terms of the features the product has or delivers. [Kang++02]

features in industry

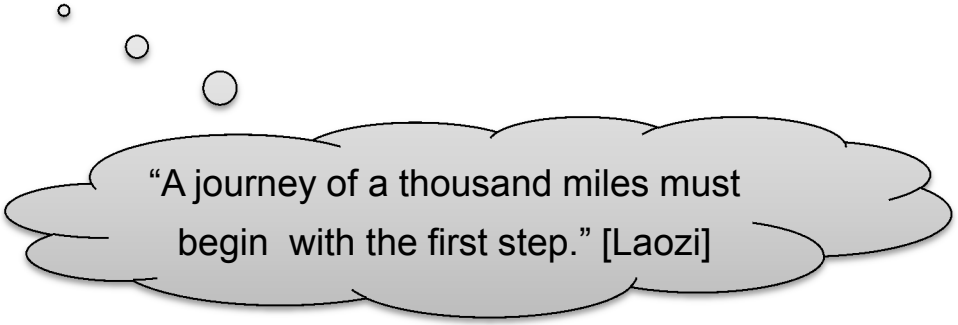


long-term goals

develop a model of what features are

design a feature prediction model

provide a more operational definition



we qualitatively study real features

research questions

- RQ1. What reasons cause companies to perceive a feature as typical, atypical, good or bad?
- RQ2. What are important characteristics of features?

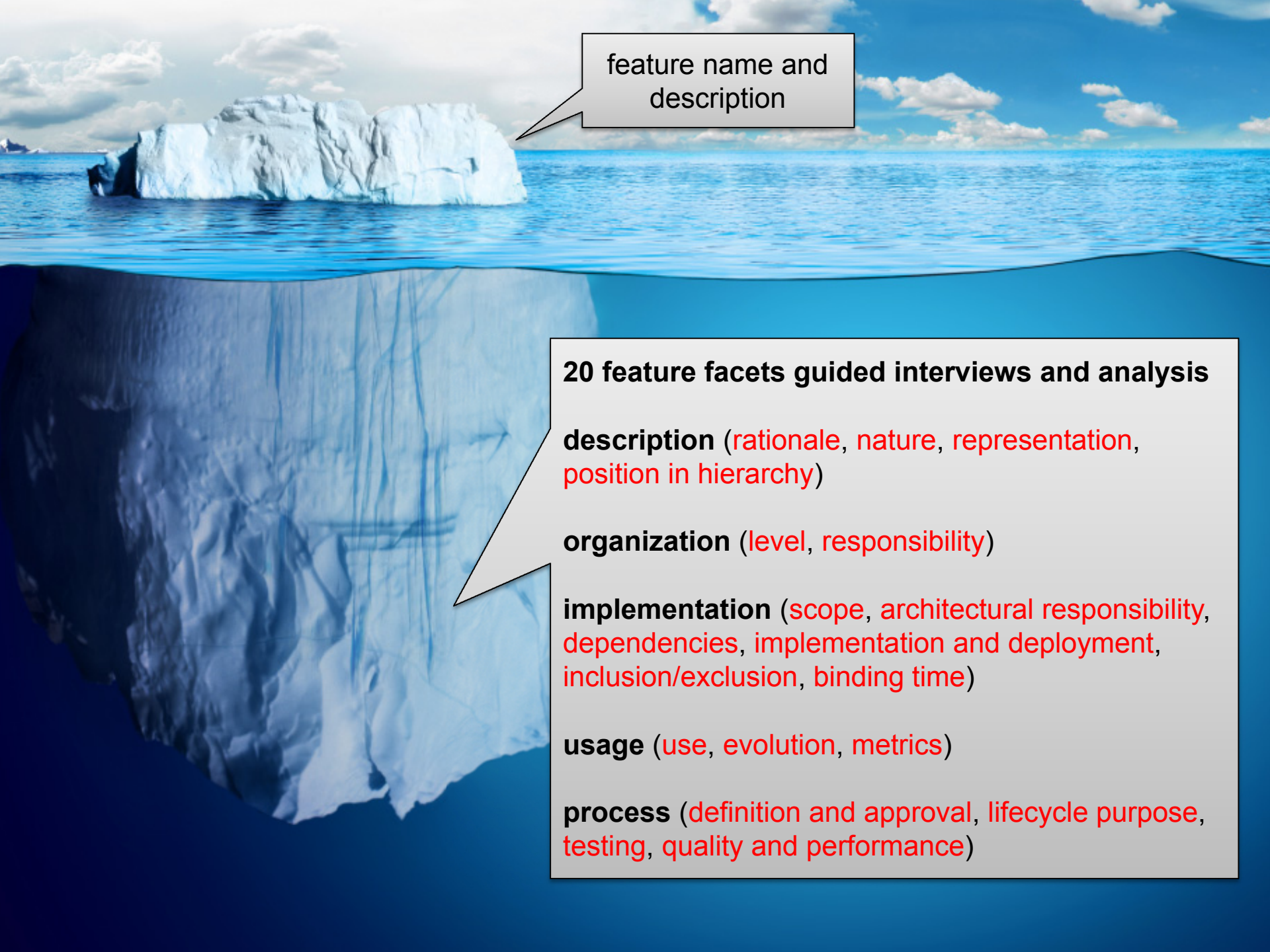
subjects

- three companies, six interviewees

selected 23 features

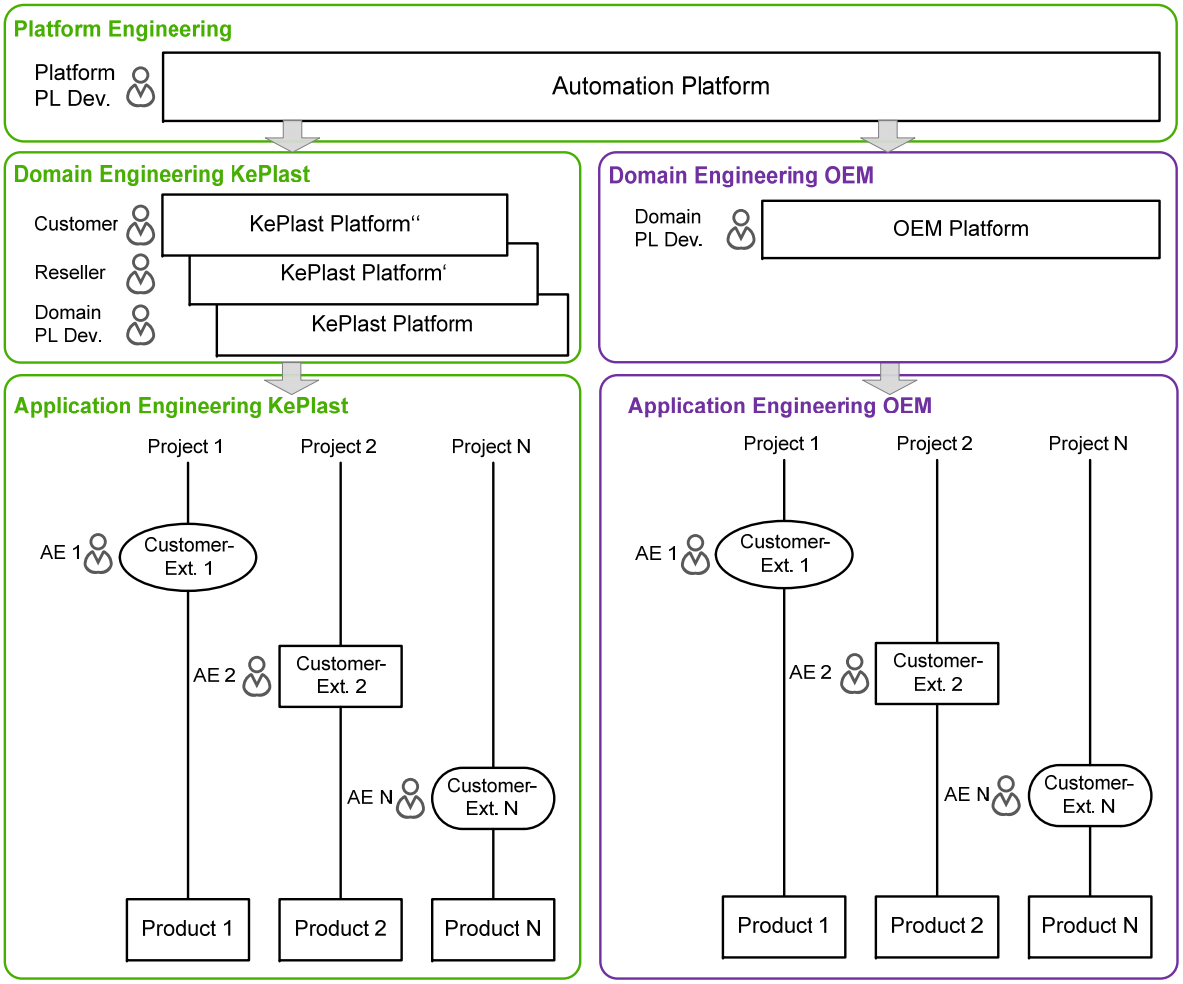
- typical, atypical (outlier), good, bad

interviews (~1.5h) and analysis guided by feature facets


feature name and description
20 feature facets guided interviews and analysis
description (rationale, nature, representation, position in hierarchy)
organization (level, responsibility)
implementation (scope, architectural responsibility, dependencies, implementation and deployment, inclusion/exclusion, binding time)
usage (use, evolution, metrics)
process (definition and approval, lifecycle purpose, testing, quality and performance)

SUBJECT COMPANIES

Keba



clone-and-own reuse

ecosystem with internal and external developers

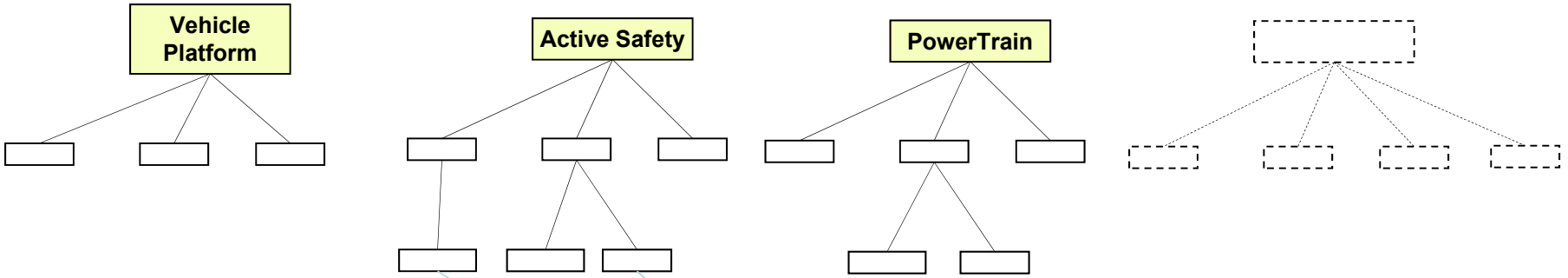
diverse feature representations

- product maps
- configuration tools
- code-level mechanisms

Opel (General Motors)

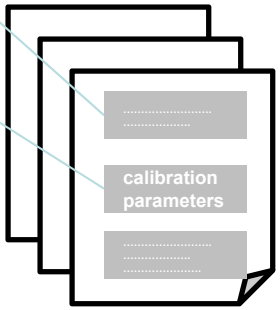
Gears

feature level



requirements level

feature mapping



IBM Rational Doors

logical/physical architecture and deployment level

C/C++ code

logical design blocks

Simulink models

AUTOSAR components

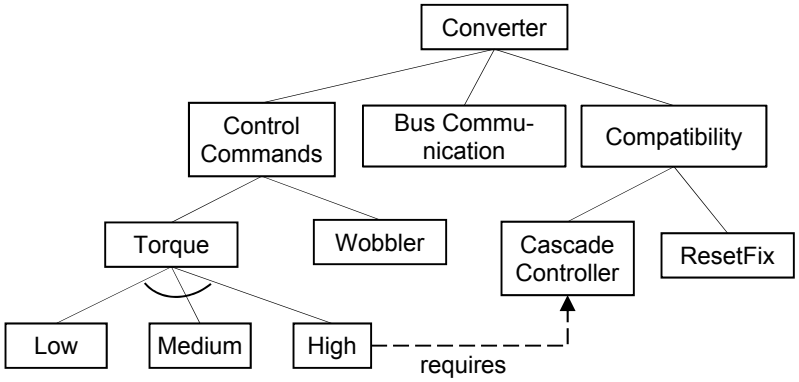
R. Flores, C. Krueger, and P. Clements. *Mega-Scale Product Line Engineering at General Motors*. In SPLC, 2012

Danfoss

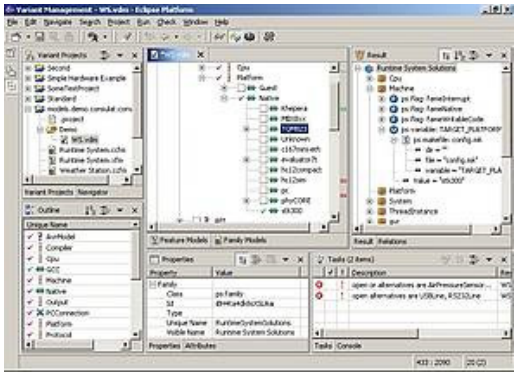
C/C++ files
…
#IFDEF TORQUE
…
#IF defined(T1 & T2)
…
#ENDIF
…
#ENDIF

integrated platform

1.5M lines of code
41,409 variation points
(#IFDEFS)



variability model: ~900 features
(illustrative example)



configurator
(pure::variants)

power-electronics
firmware

interviewees

	part.[1]	role	exp.[2]	features
Keba	A	developer	12	LIN_Movement, Oscilloscope, Euromap, Silent_Mode
	B	product manager	19	ProfiNet_Slave, Wizard, Manual_Configuration, User_Guidance
	C	developer	3	Language_Translation, Production_Overview, DataManager, Heat-Up_Optimization
Opel	D	team lead/ architect	5	Lane_Keeping, Park_Assist, Emergency_Braking
Danfoss	E	architect	4	Torque, Cascade_Controller, Product_G, Power-Up_FastFuncs
	F	team lead	8.5	Wobbler, Field_Bus, Reset_Fix, Board-Support_Package

[1] participant (interviewee) [2] experience with the product line in years

selection of

RESULTS

classification rationales

GOOD AND BAD FEATURES

individual features

good feature 

popular with customers

popular with developers

well implemented

error-free

thoroughly tested

architecture-conform

distinct functionality

bad feature 

customer complaints

duplicate features

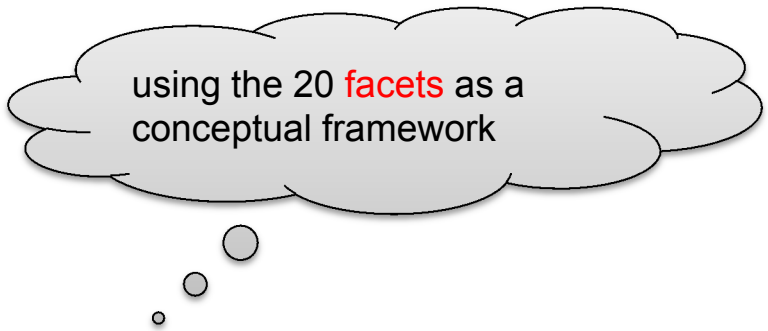
workaround (“hack”)

defect features

test challenges

optional feature

highly volatile



selection of results

CROSS-CASE ANALYSIS

distinct functionality

What is problematic is when it's too little specific.

graspable / distinct features are good features

vague features are bad features

Customers did not know what to expect.

position in hierarchy

- interviewees preferred to talk about leaf or top-level features

outlier features

O1: Features do not only address functional or non-functional concerns that end up in a product. Features are also used for atypical purposes, such as supporting a system's lifecycle.

target dedicated lifecycle purpose (build, startup, QA)

incomplete process sufficient

restricted to some organizational levels

coordinated by subset of roles (responsibility)

examples of outliers

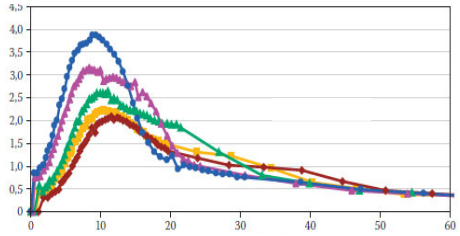
We didn't really know how to improve it.

Keba.UserGuidance

placeholder for future functionality (usability improvement)

Keba.HeatUpOptimization

prevent startup power peaks



power peaks

Danfoss.BoardSupportPackage

improve maintainability of Hardware Abstraction Layer

Danfoss.PowerUpFastFuncs

move functions from flash to RAM

We can tweak the product to indirectly fulfill the customer requirements.

features and parameters

O2: Parameters are not treated in the same way as features.

nature of features

- almost every feature came with configuration parameters

large parameter databases exist

parameters have

- no process attached
- no architectural responsibility
- no dedicated responsible role

cross-cutting features

O3: Scattered feature implementations do not necessarily lead to problematic features.

scope of a feature is not a differentiator between good and bad

- half of the studied features were cross-cutting

There can be good reasons for a scattered feature implementation.

O5: Scattered features that have to be tested are problematic.

cross-cutting features problematic with manual testing processes

- only testable at integration time
- potentially require hardware

Passos, Padilla, Berger, Apel, Czarnecki, Valente, “Feature scattering in the large: A longitudinal study of linux kernel device drivers,” MODULARITY, 2015.

immature features

O4: A rushed development process causes problematic features.

observed a diversity of different processes

process is not a differentiator between good and bad

but usually time pressure

There was an extreme pressure from the customer side.

Keba.ManualConfiguration

Danfoss' time-boxing experiment

We were told not to think, just to implement.

nature and use of features

We need that for the […] controlling.

features are primarily a unit of functionality

- used for communication among developers and customers
- used for scoping and creating awareness for software reuse

can serve as a unit of variability when necessary

It rather felt like it's a bug from the perspective of the customer.

Keba.DataManager

- introduced to provide low-level machine-data access
- making it optional caused significant effort

It went back and forth: it's a bug, it's a feature, it's a bug, it's a feature; and then we said OK it's a bug.


what is a feature?
we studied 23 real features
elicited key characteristics (facets)
studied good and bad practices
theory-building from cases
contributions for practitioners and researchers
future work
study feature lifecycles
create a model of what features are
design a prediction model
study other companies

thanks for your time!

What is a feature?

Thorsten Berger, Daniela Lettner, Julia Rubin, Paul Grünbacher,
Adeline Silva, Martin Becker, Marsha Chechik, Krzysztof Czarnecki