# Software product lines, feature modelling, analysis and configuration

David Benavides
benavides@us.es
Evolución y Gestión de la Configuración

The main goal of this lesson is to give an overview of “software product lines” from a practical and research point of view

SPL

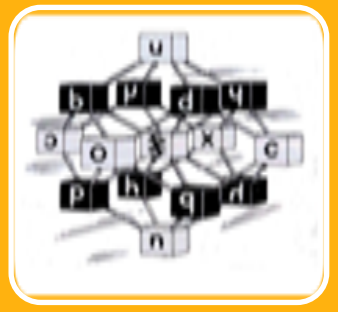
FM

AAFM

FOP

# Part I

# Software Product Lines



# Variability Modelling

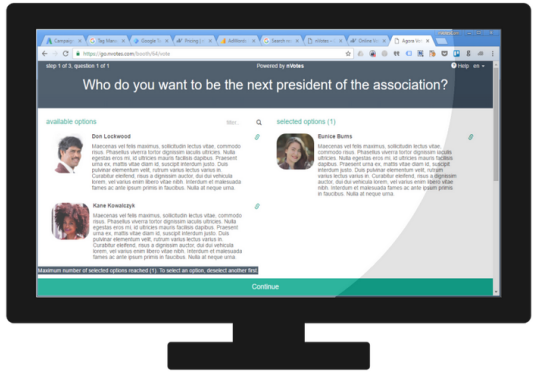
# Some real cases

# Real cases

nVotes

Features & Plans  Contact us

## Secure & easy online voting




Secure, robust and affordable internet election management solution that makes it easy for voters to cast their vote online

UNIVERSIDAD Ð SEVILLA

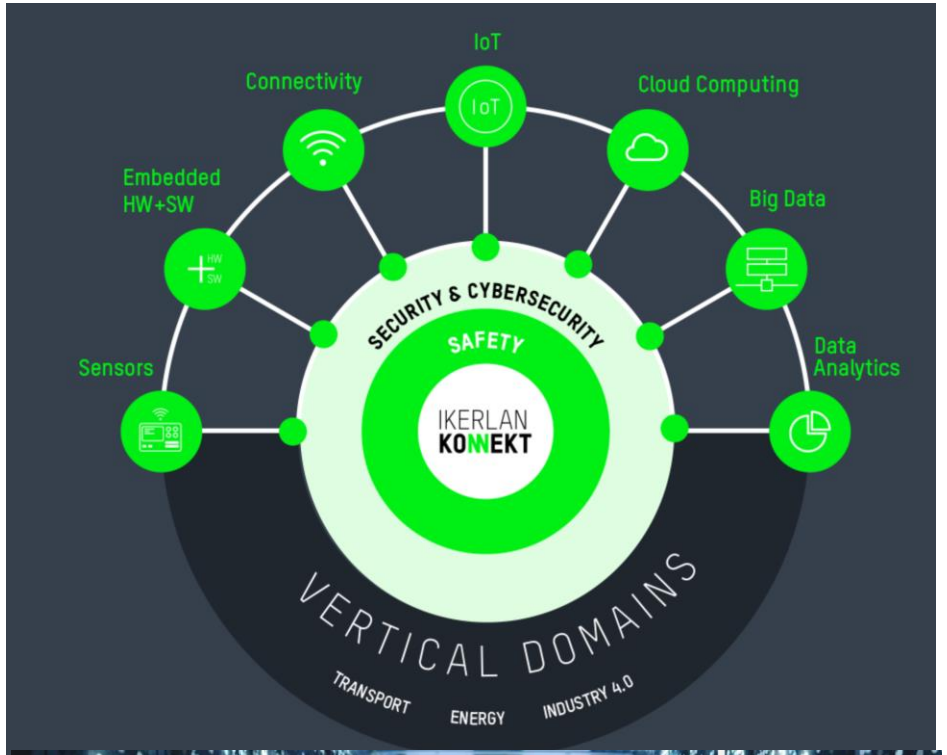
# Real cases

DIPUTACION DE SEVILLA
INPRO
DIPUTACION
buscar...
Zona Empleados
RSS | Directorio | Mapa Web | Contacta
Sociedad de Informática Provincial
INPRO
Inicio
La Empresa
Noticias
Novedades
Nuestros Usuarios
Productos y Servicios
I+D

# Real cases

Equipo
Carpeta personal
Instalar Guadalinex v9
Todas las aplicaciones
Accesorios
Educación
Juegos
Gráficos
Internet
Oficina
Sonido y vídeo
Acceso universal
Wine
Preferencias
Administración
Lugares
Archivos recientes
Actualización de software
Analizador de uso de disco
Aplicaciones preferidas
Applets
Archivos
Ayuda
Bloc de Notas
Bloqueador de pantalla
Bluetooth
Calculadora
Capturar pantalla
Cliente de correo Thunderbird
Color
Conexiones de red
Menú
22 de nov, 00:45

# Real cases


IoT
Connectivity
Cloud Computing
Embedded HW+SW
Big Data
Sensors
Data Analytics
SECURITY & CYBERSECURITY
SAFETY
IKERLAN KONNEKT
VERTICAL DOMAINS
TRANSPORT
ENERGY
INDUSTRY 4.0

IKERLAN KONNEKT
CONOCE LAS SOLUCIONES DE DIGITALIZACIÓN QUE SE ADAPTAN A TU EMPRESA

# Real cases

**What configurations should I test to defend my users from bugs?**

# Software product lines

# Industrial Trends

| | |
|---|---|
| Organizations are evolving | • *Project* Centric Software Engineering<br>• *Product* Centric Software Engineering |
| Software ***variability*** constantly increasing: | • Variability goes from hardware to software<br>• Variations points grows by thousands |
| Assets' ***Reuse*** is shifting | • from ad-hoc to ***systematic*** |

# What is a software product line?

# Real example

# Real example

# Mass production

producing efficiently a large amount of standardized products

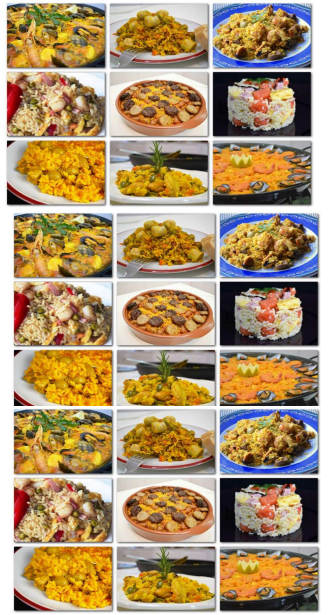
# Software product lines

No customization
-
one product

## oduction

acing efficiently a large amount of
standardized products

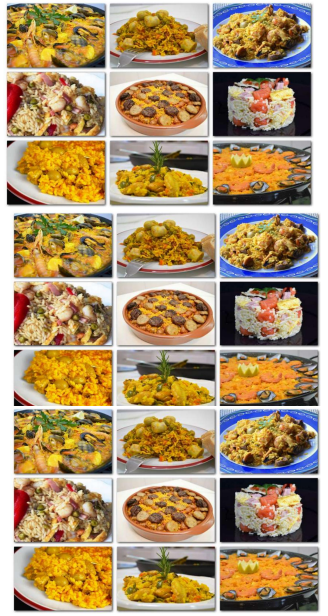
# Software product lines



# Mass customization

“a paradigm shift for the enterprise to offer products and services best catering to individual customer's needs whereas keeping near-mass production efficiency “

[Tseng, M.M., Jiao, J. (2001)]



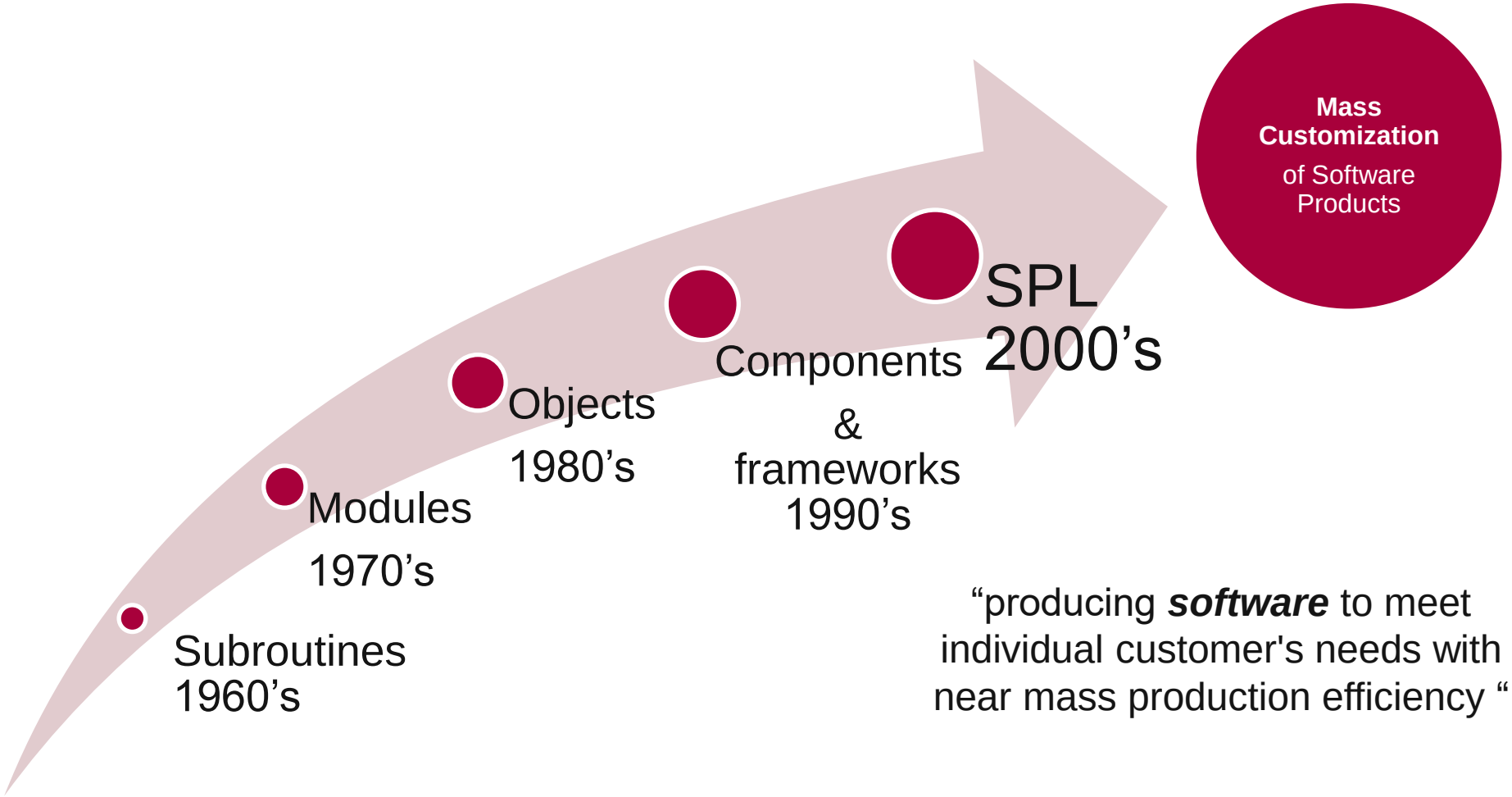
Customization
-
A set of products

# Mass customization

“a paradigm shift for the enterprise to offer products and services best catering to individual customer's needs whereas keeping near-mass production efficiency “

[Tseng, M.M., Jiao, J. (2001)]

# Software product lines




"producing ***software*** to meet individual customer's needs with near mass production efficiency "

# Software product lines

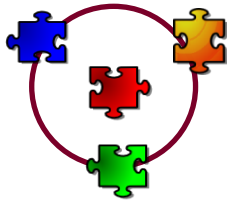
Common features

- Alarm clock
- Calls
- Messaging

Variable features

- Media
- Games
- Connectivity

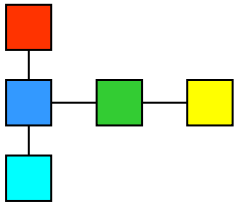
## Variability Model



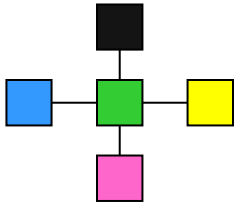
- Documents the variability of SPL
- Enable managing the variability

# Software product lines

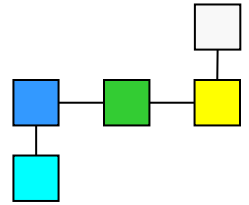
Traditional Approach (*mass production*)



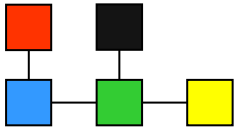
Product 1



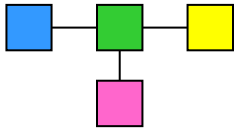
Product 2



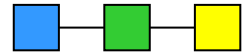
Product 3



Product 4

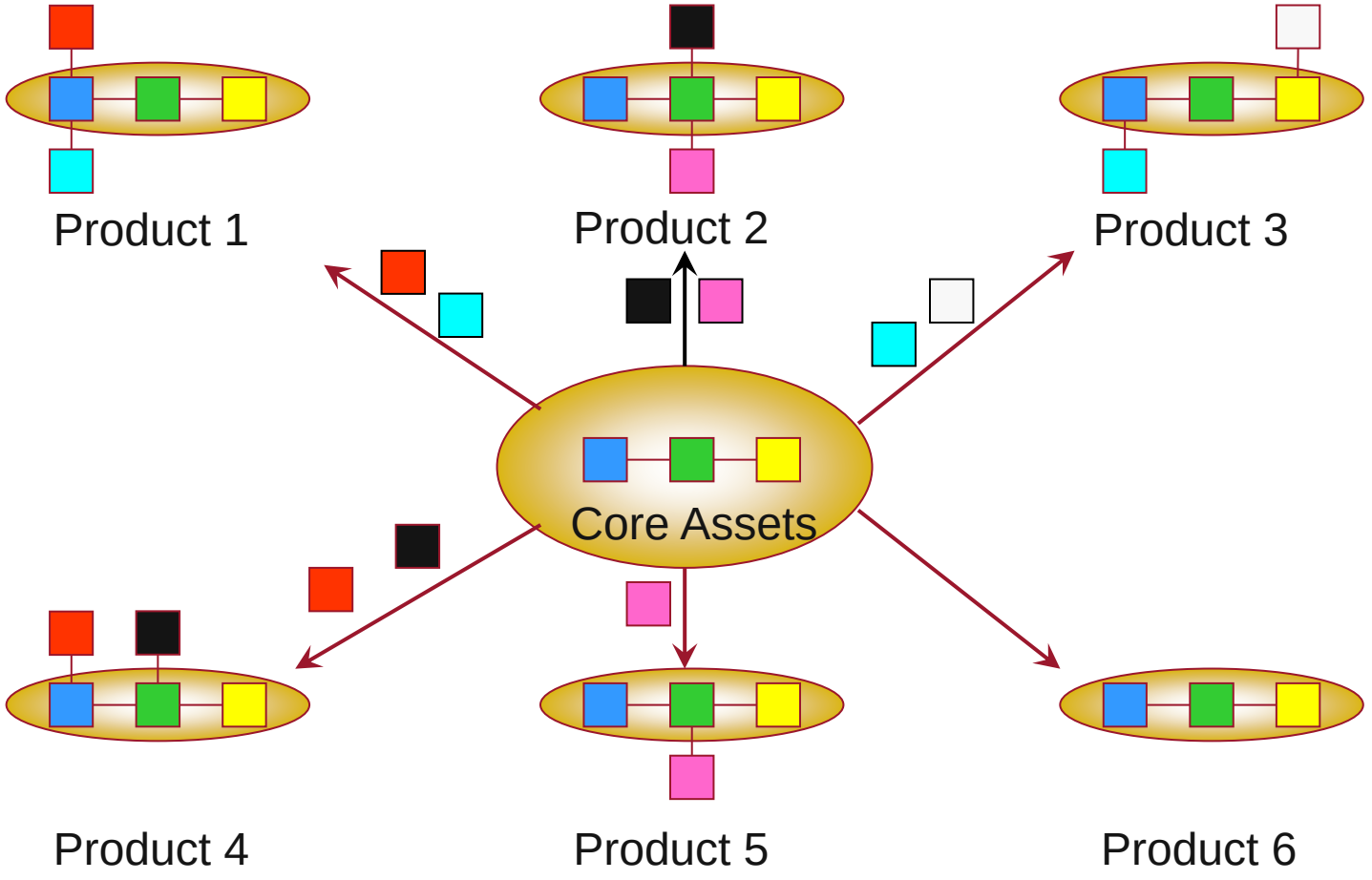


Product 5

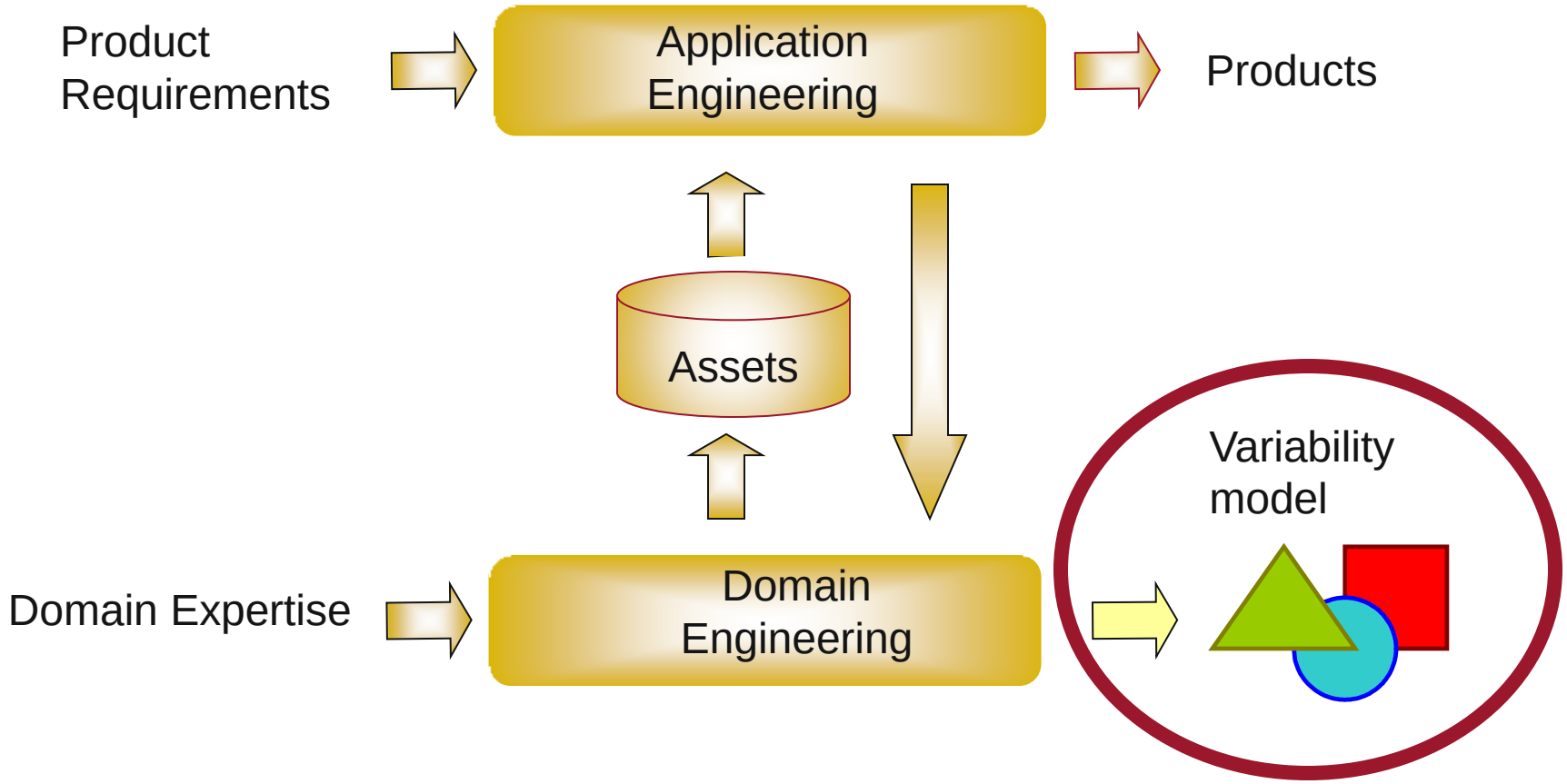


Product 6

Product Lines Approach (*mass customization*)



# SPL: Activities



# SPL framework

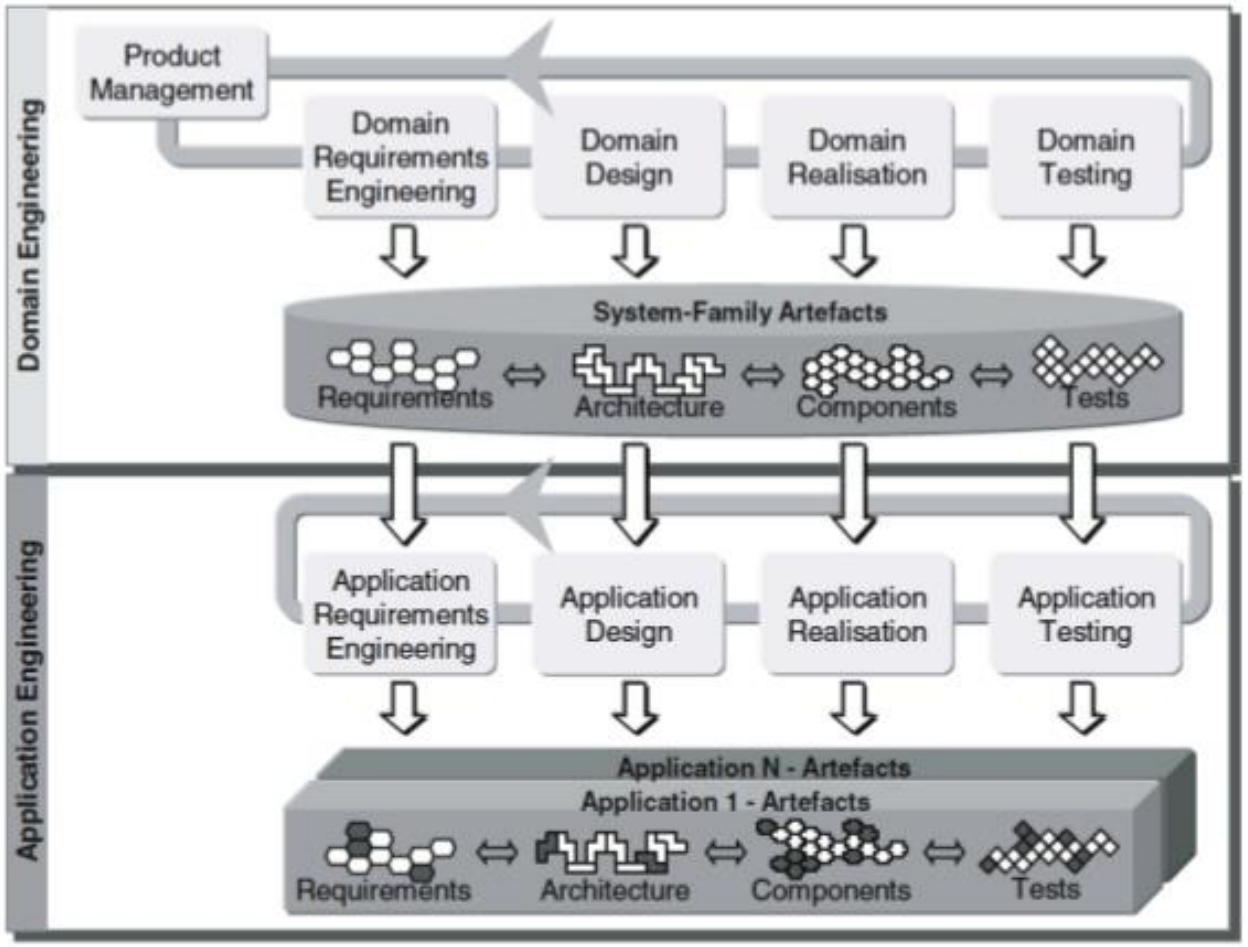



**Fig. 1.2.** The two-life-cycle model of software product line engineering

From "Software Product Line Engineering" by Phol et al.

# A more practical view of the SPL framework

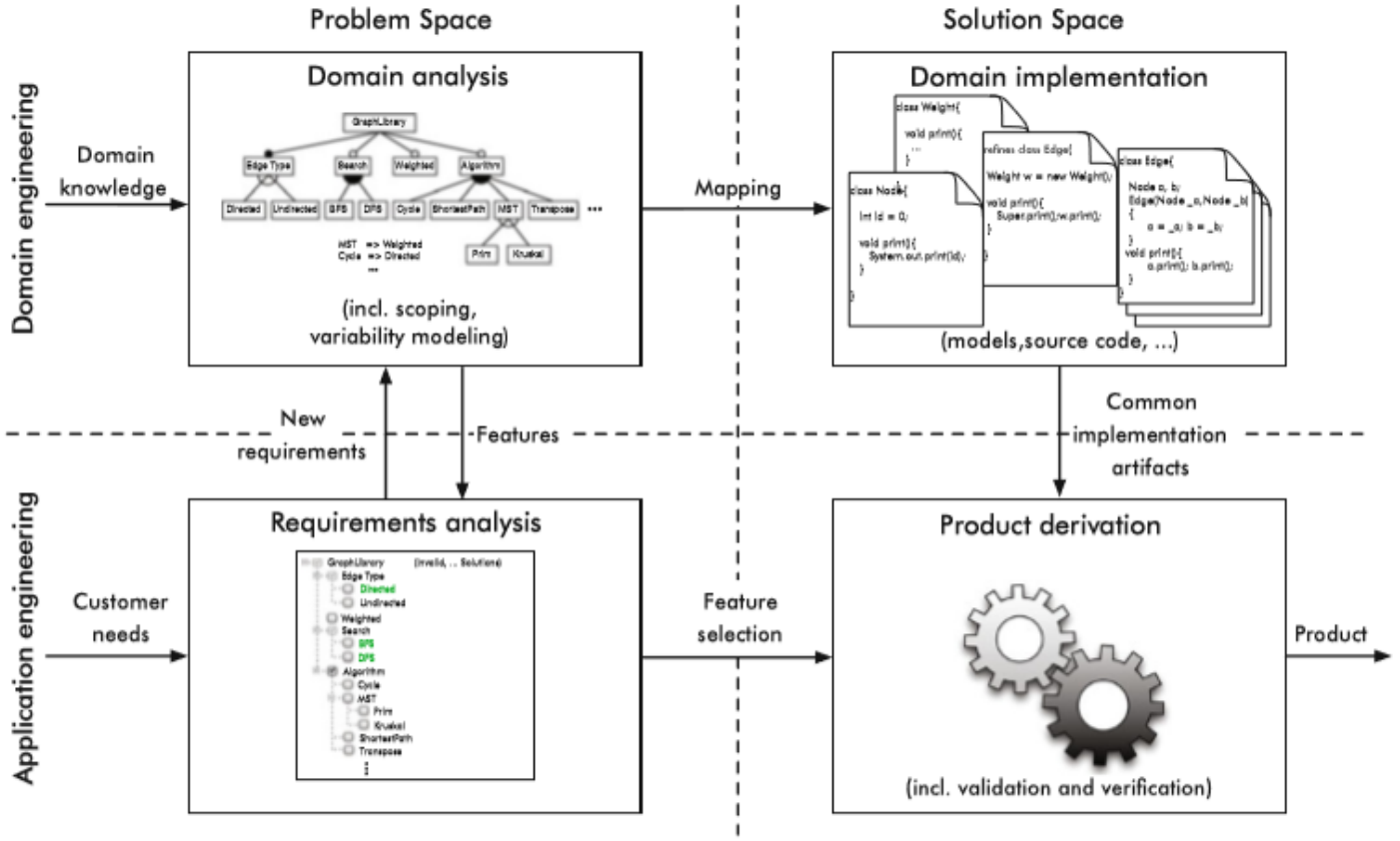



**Fig. 1.1** An overview on software product-line engineering

From “Mastering Software Variability with FeatureIDE”

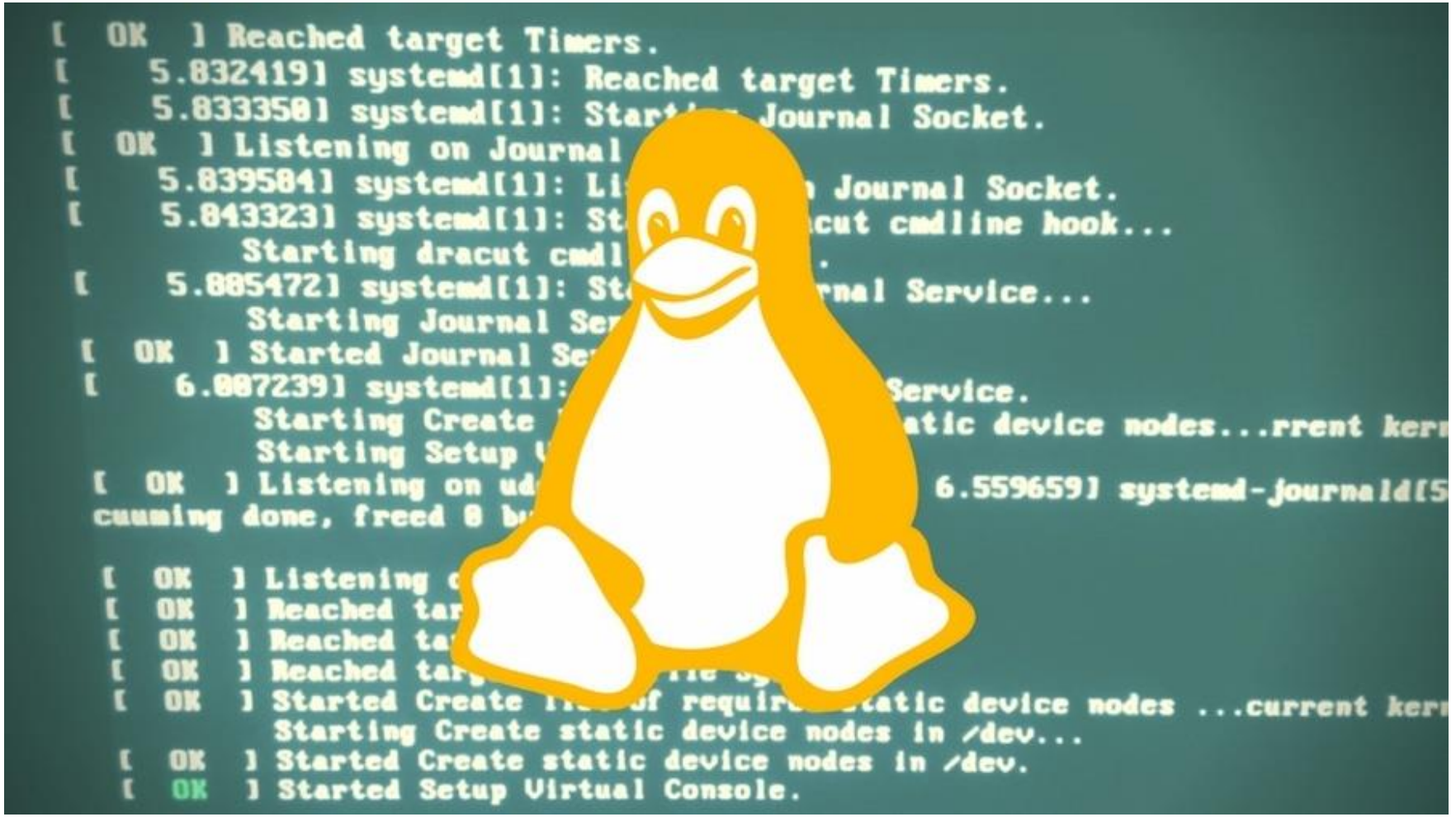
# What are the reasons for SPLE "tentations"?

# Product explosion

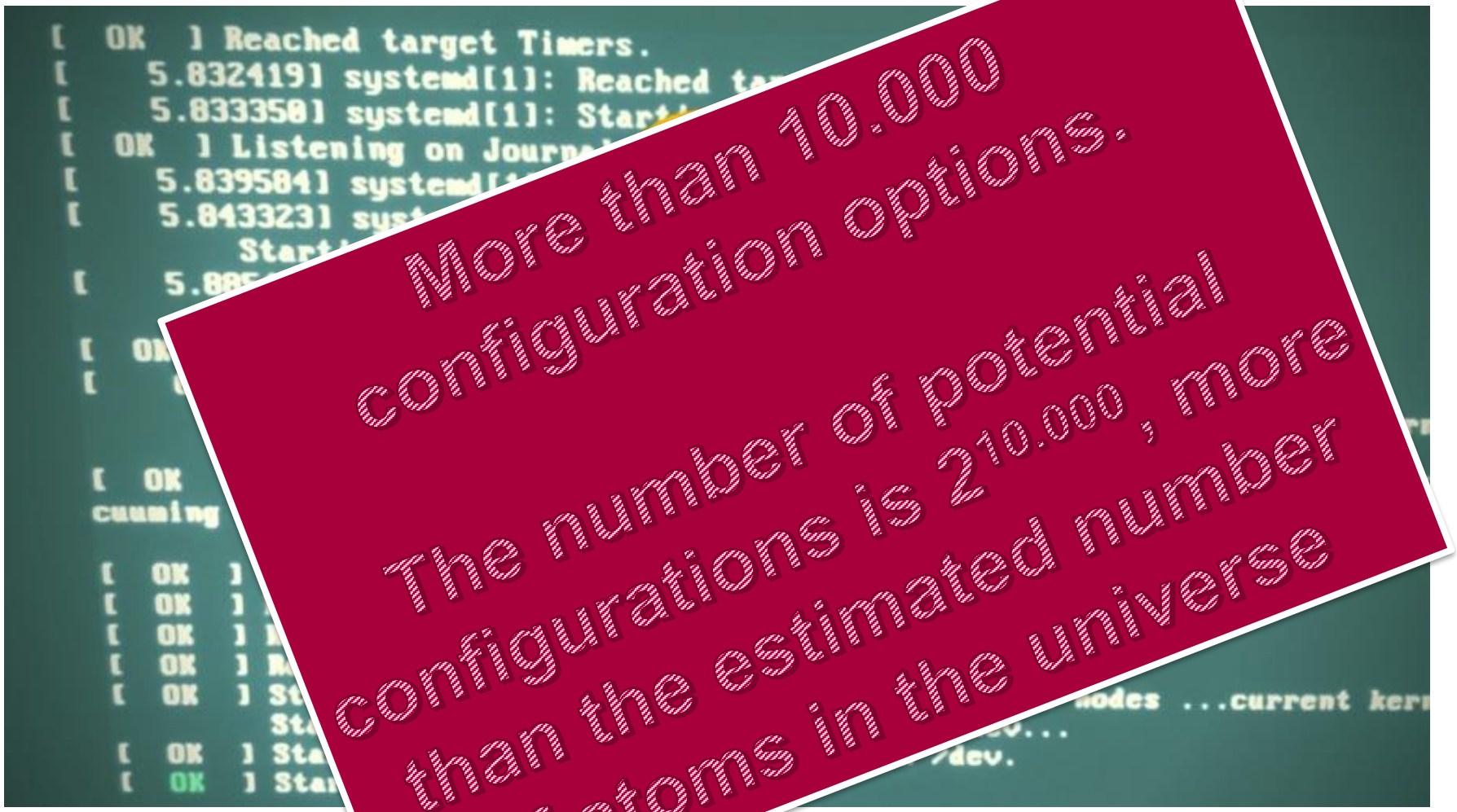
# Customers explosion

# Technology explosion

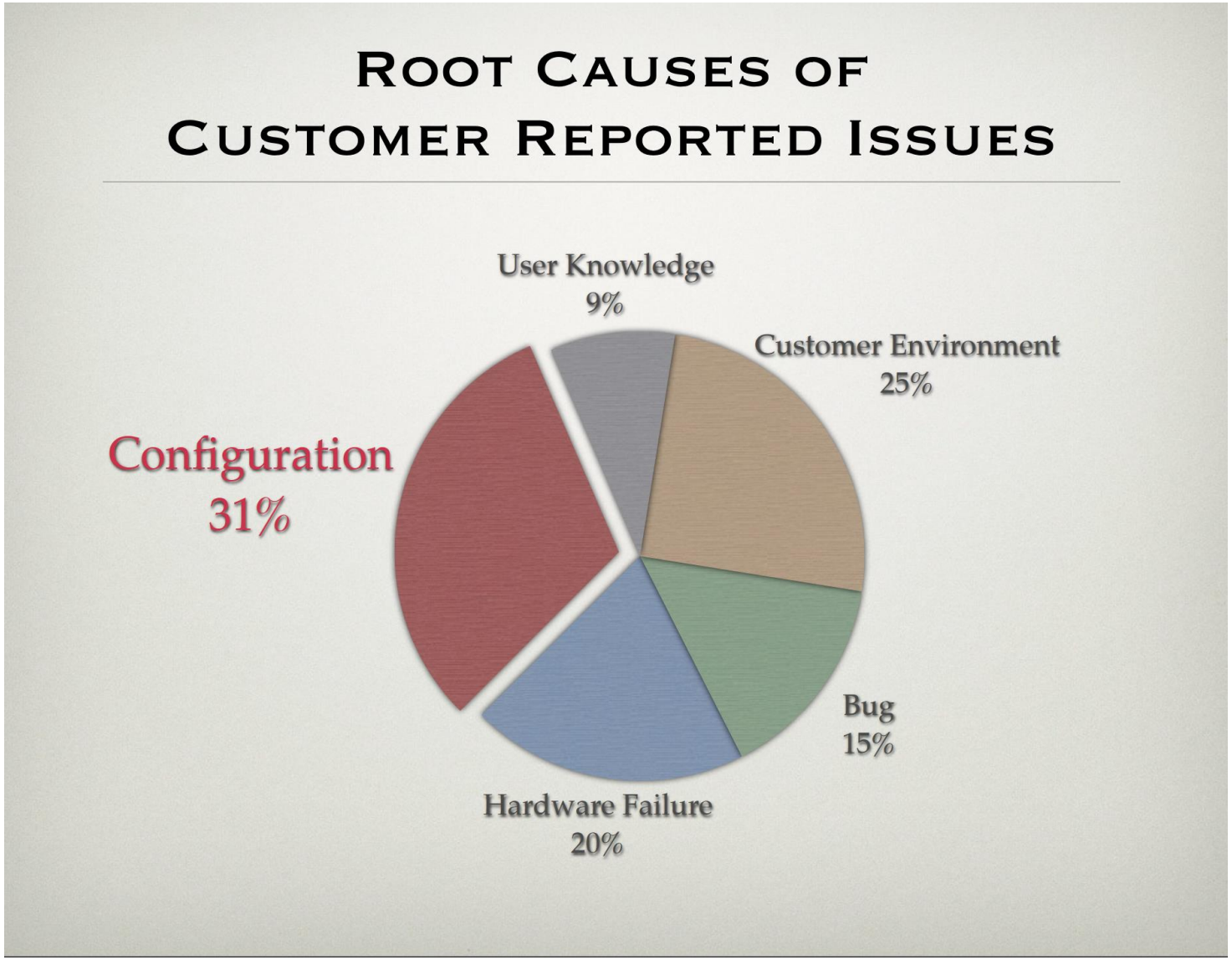
# Configuration explosion



# Configuration explosion



# Configuration explosion




Taken from http://sigops.org/sosp/sosp11/current/2011-Cascais/12-yin-slides.pdf

# Explosions consequences

- Product oriented development
- Fire-fighting mode
- Opportunistic reuse

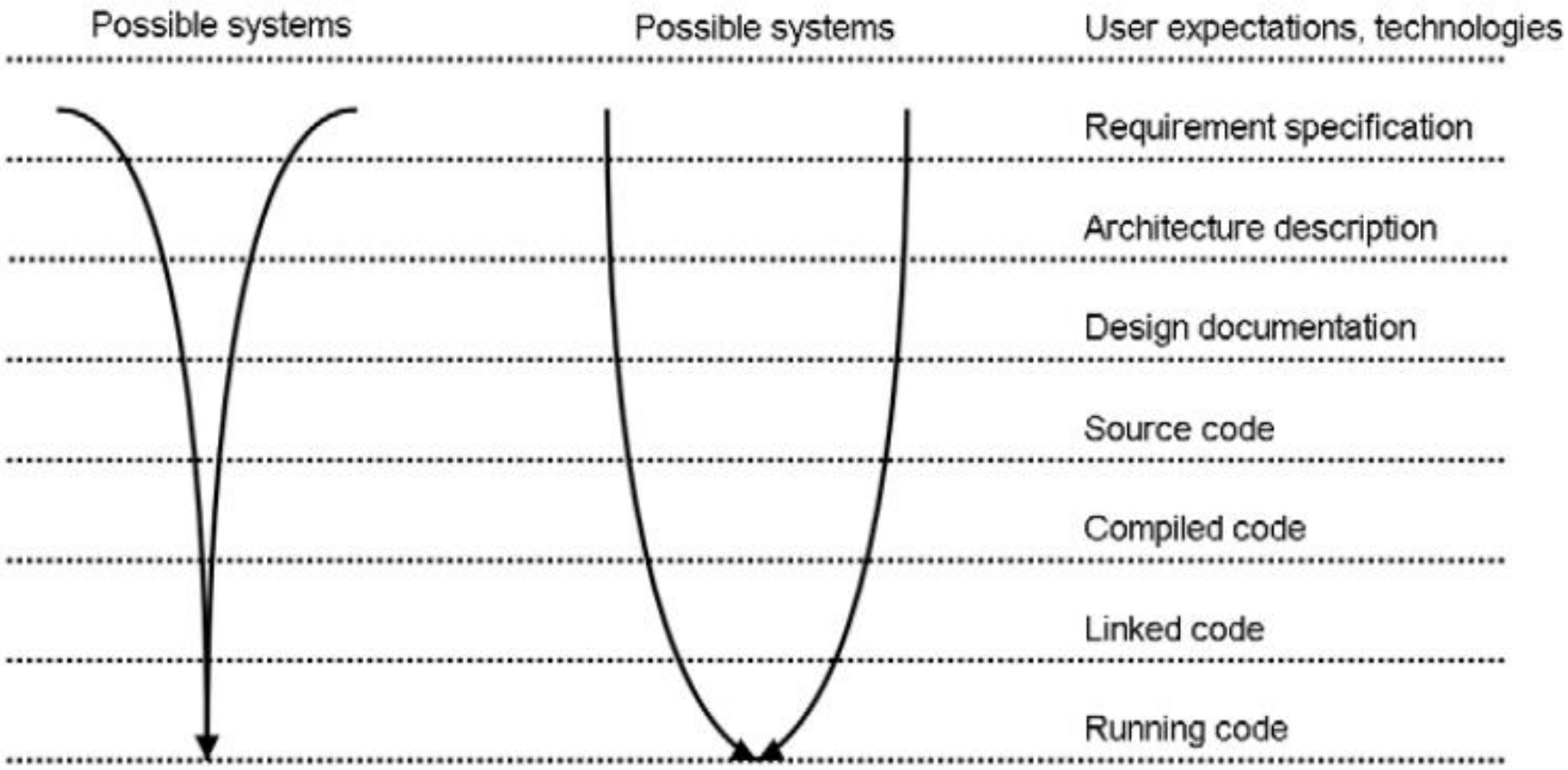
- Lack of innovation
- Quality degradation
- Knowledge lost

## Some “tentations”

- Product portfolio diversity
- Common user experience for product in the portfolio
- Customization of products

# What are the goals?

# SPL metaphors




Svahnberg M., van Gurp J., Bosch J., *On the Notion of Variability in Software Product Lines*. Proceedings of IEEE/IFIP Conference on Software Architectures, 2001.

# SPL metaphors

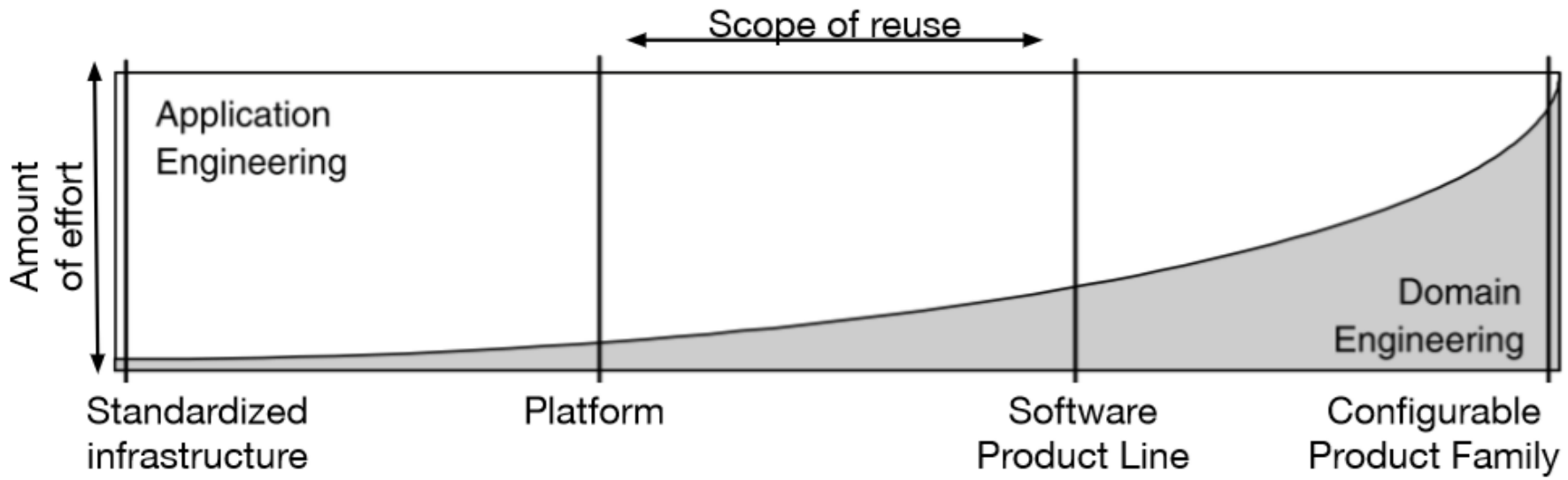



Figure 1.1: SPL maturity stages: from less mature (left) to more mature (right)[DSB05]).

# Evolution of an SPL



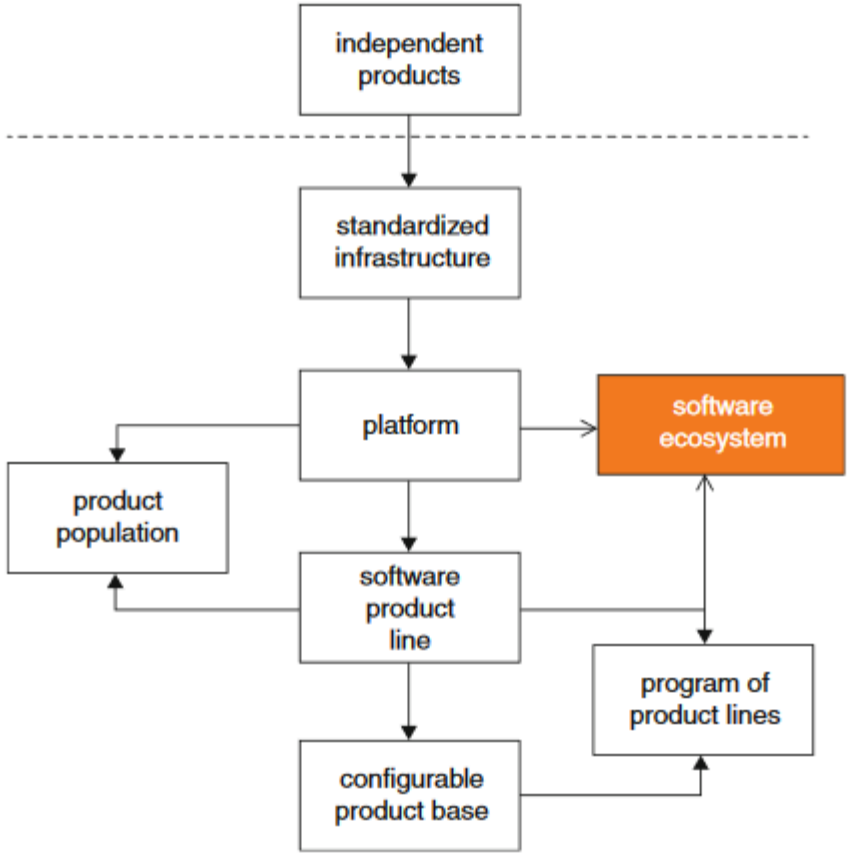
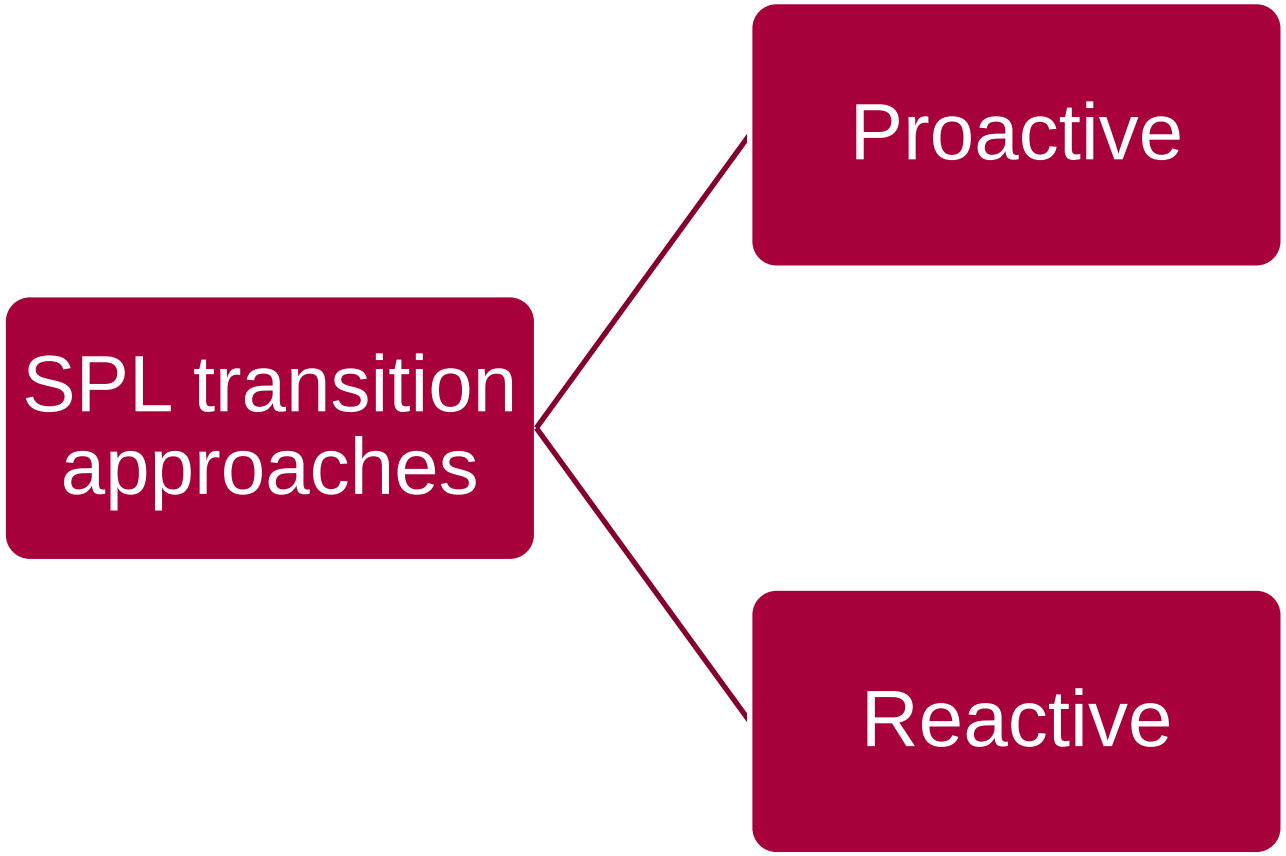



**Fig. 1.1** Evolution of a software product line

Taken from “Systems and Software Variability Management”

# OK, I trust you but...
# how shall I transition?

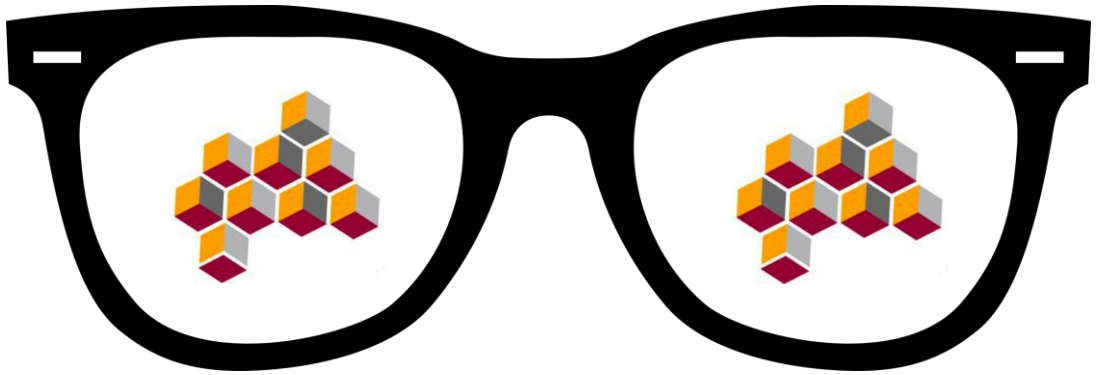

SPL transition approaches
Proactive
Reactive

# Some barriers

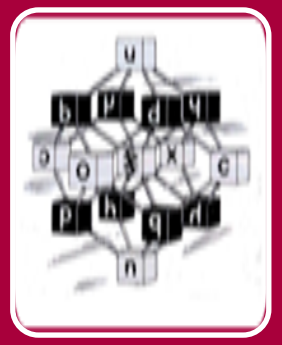
# Business strategy

# Variability, a new degree of complexity

DISCUSSION
QUESTIONS



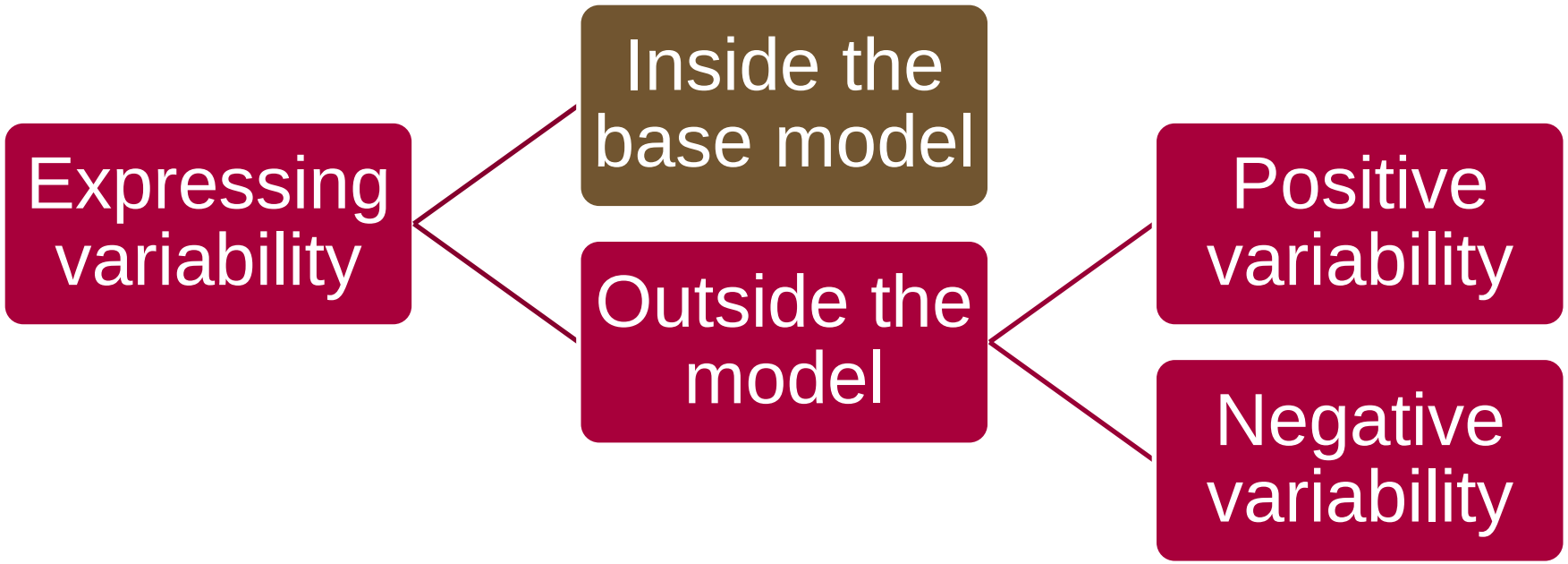
# Software Product Lines



# Variability modelling

# How to model variability?

# How to model variability



# Inside the model

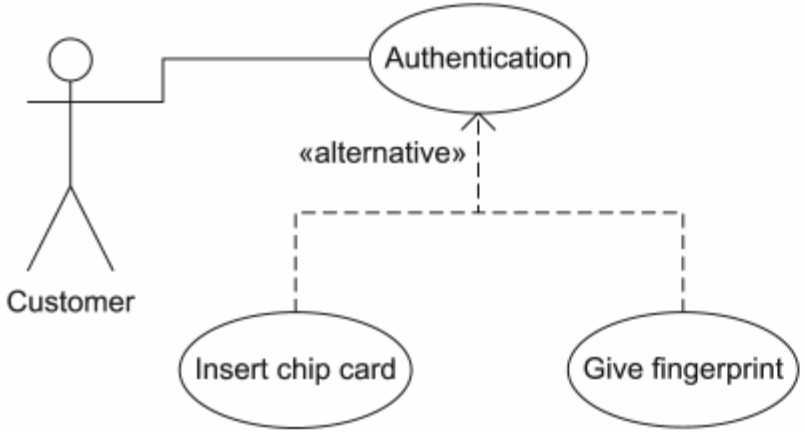



**Figure 5: Example of an alternative relationship**

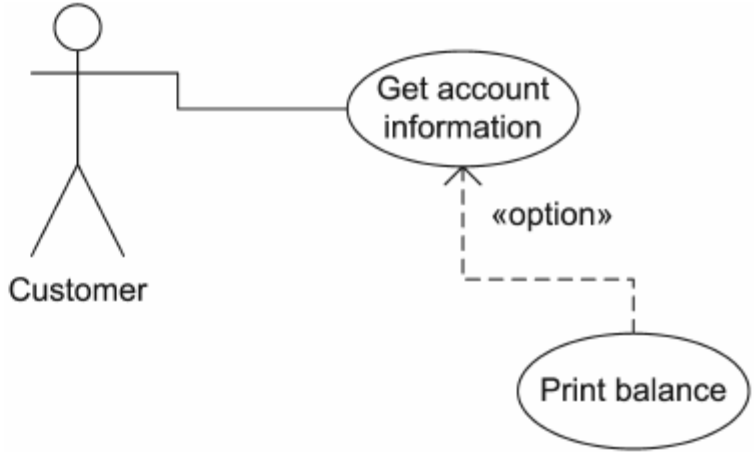
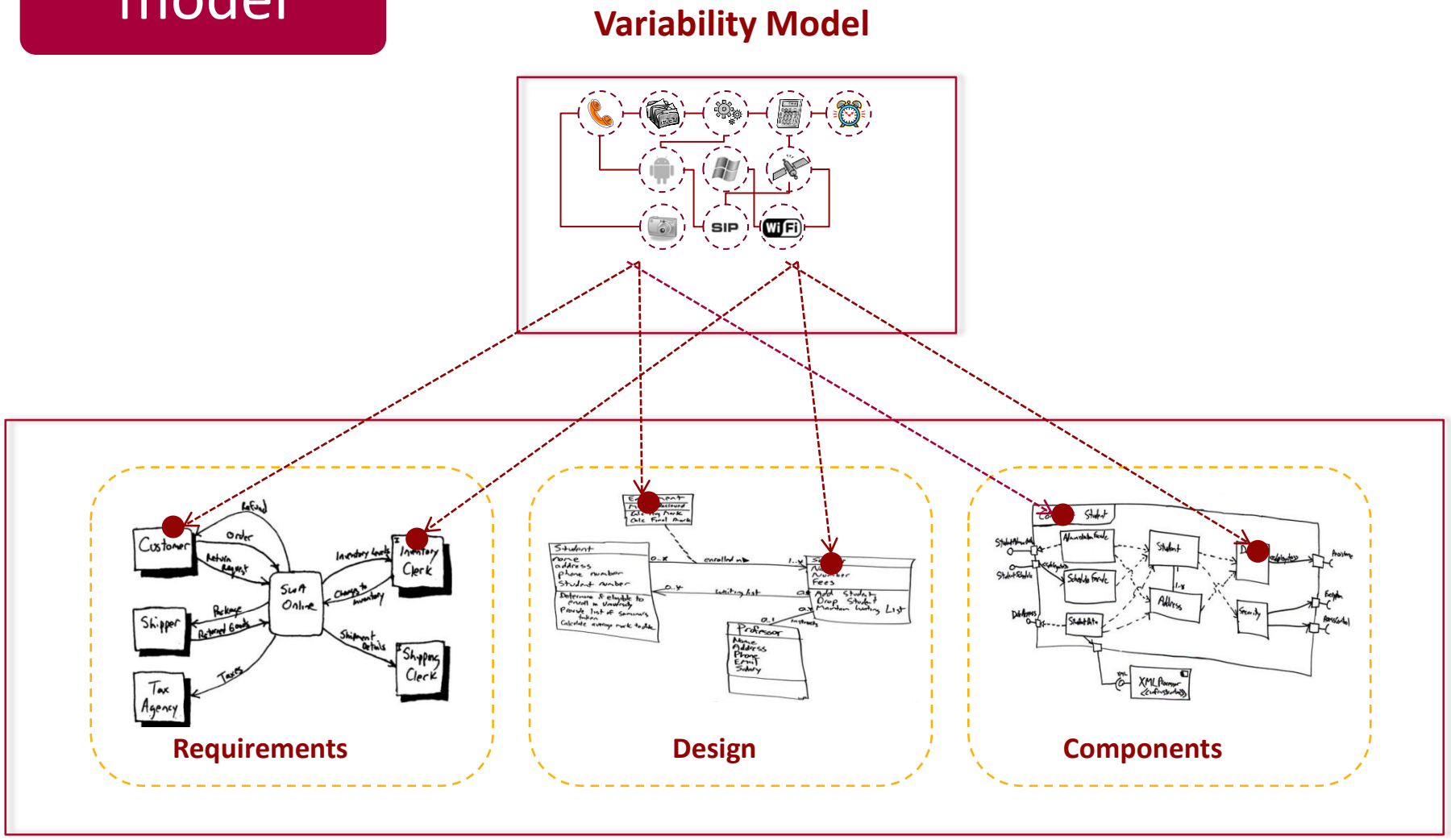


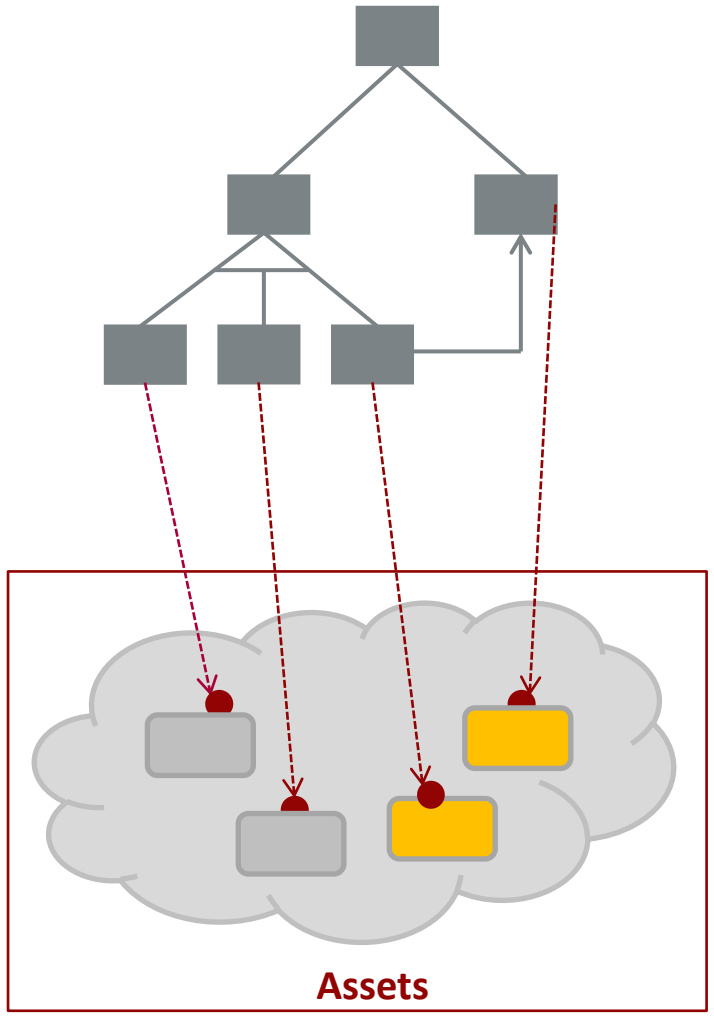

**Figure 6: Example of an optional relationship**

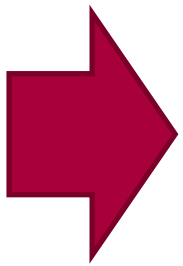
Taken from Modeling variability by UML use case diagrams by Von der Massen et al.

# Outside the model

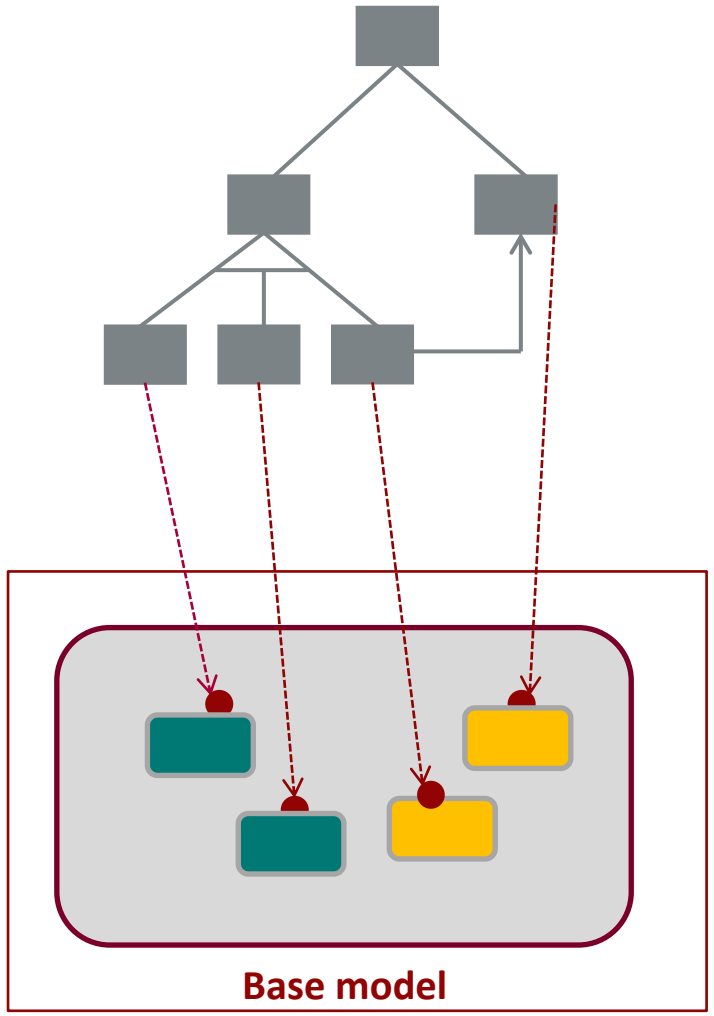


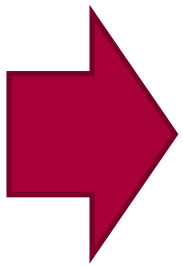
# Positive variability

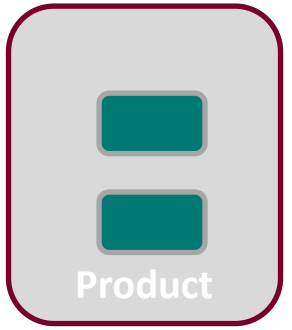


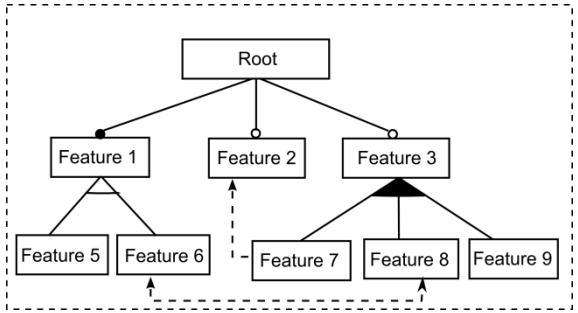
# Negative variability



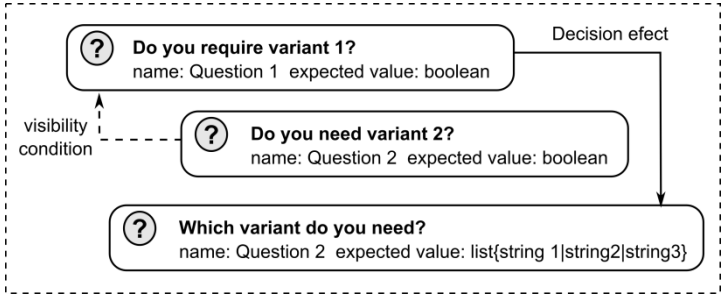





# How to model variability




**Feature modelling**

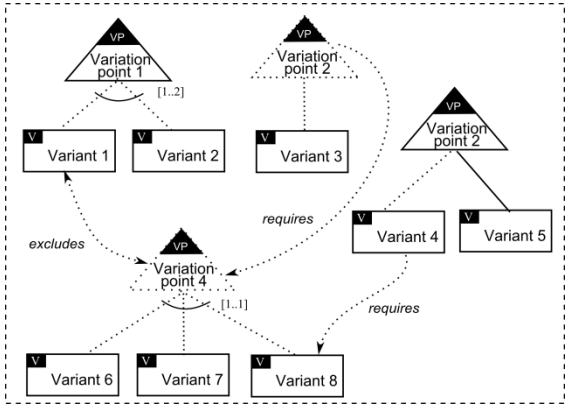



**Decision modelling**

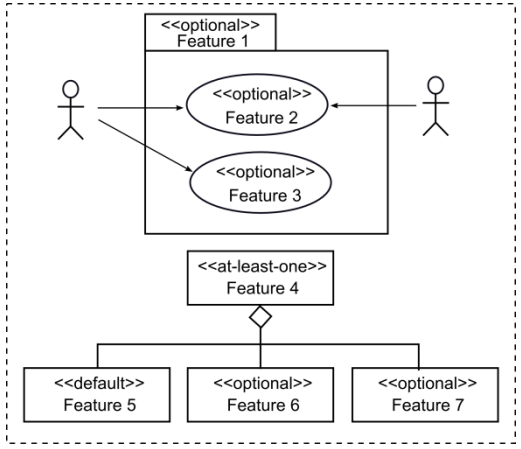
**Techniques**

**Ad-hoc solutions: tables, textual docs, ...**

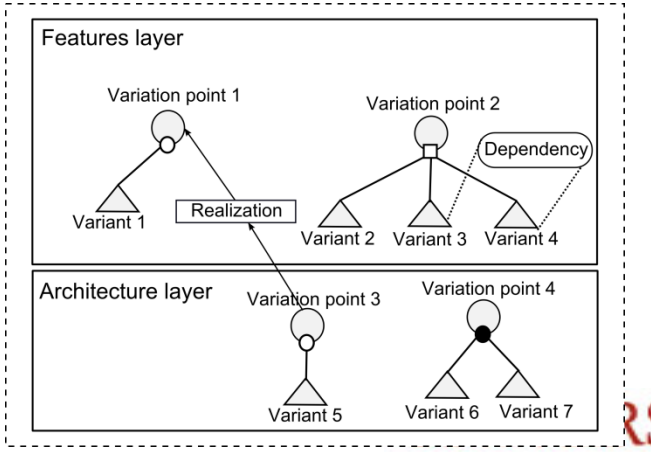
**Orthogonal variability modelling**




**UML-based**

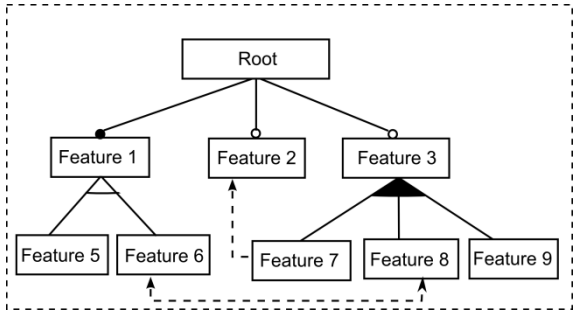



**COVAMOF**

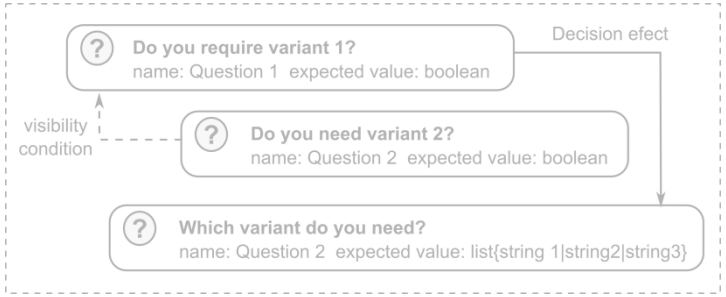



RSIDAD Ð SEVILLA

# How to model variability



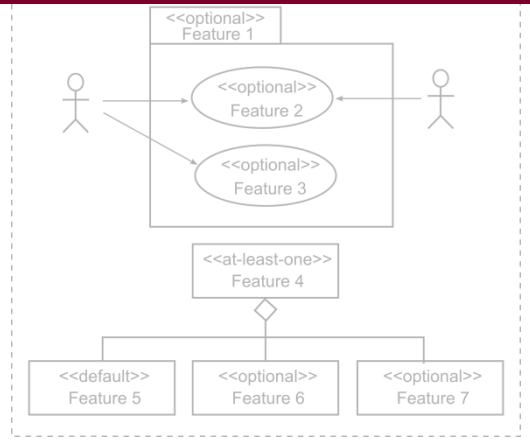

**Feature modelling**

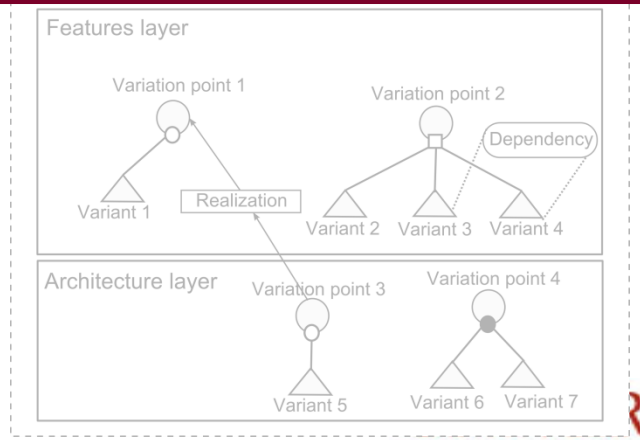



Decision modelling

Ad-hoc
tables,
...

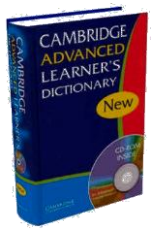
Feature models were first introduced by Kang et al. in 1990



# How to specify a particular product?

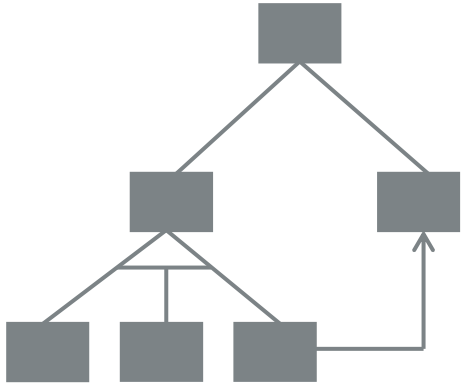
FEATURE




*"An important part of something"*

*"A prominent or distinctive characteristic of a software system"*

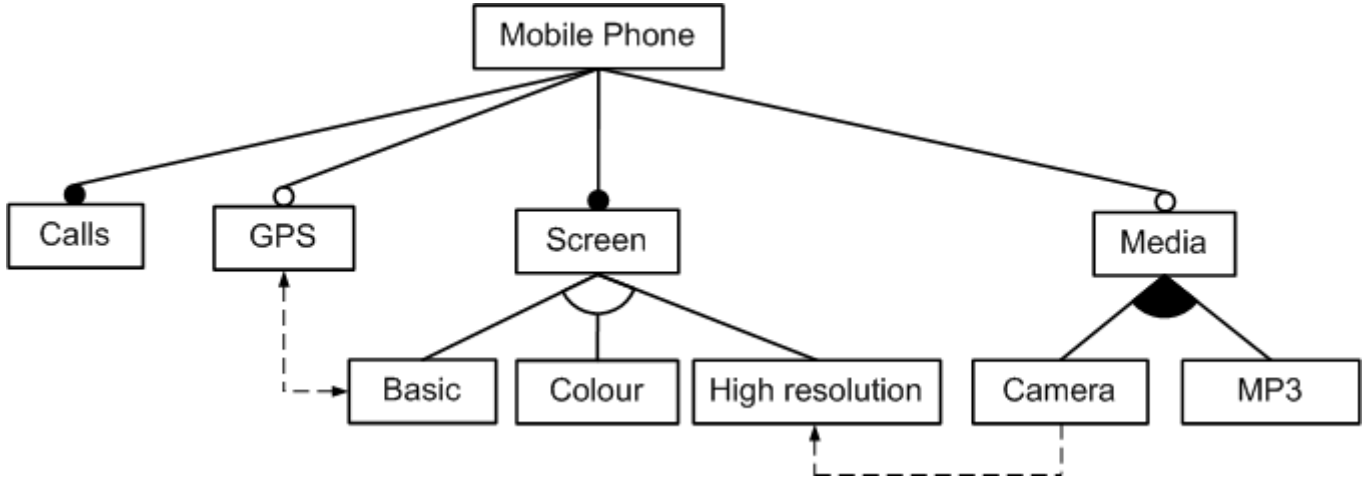
# How to specify an SPL?



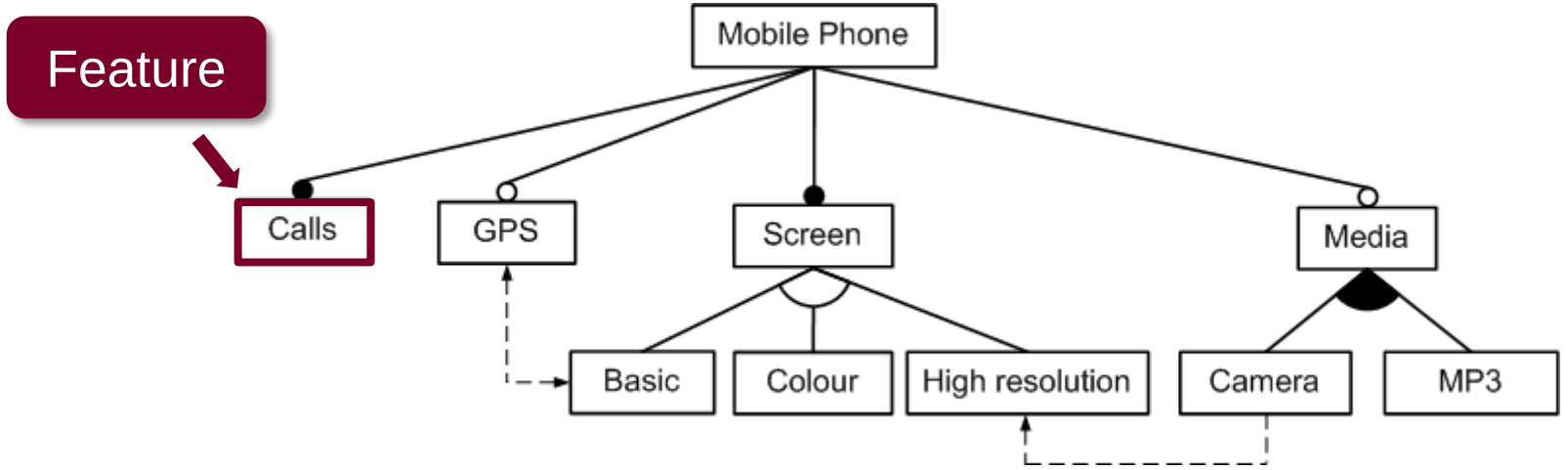
*“Feature Model: A hierarchically arranged set of features to represent all possible products of an SPL”*

Design a feature model for your own SPL!

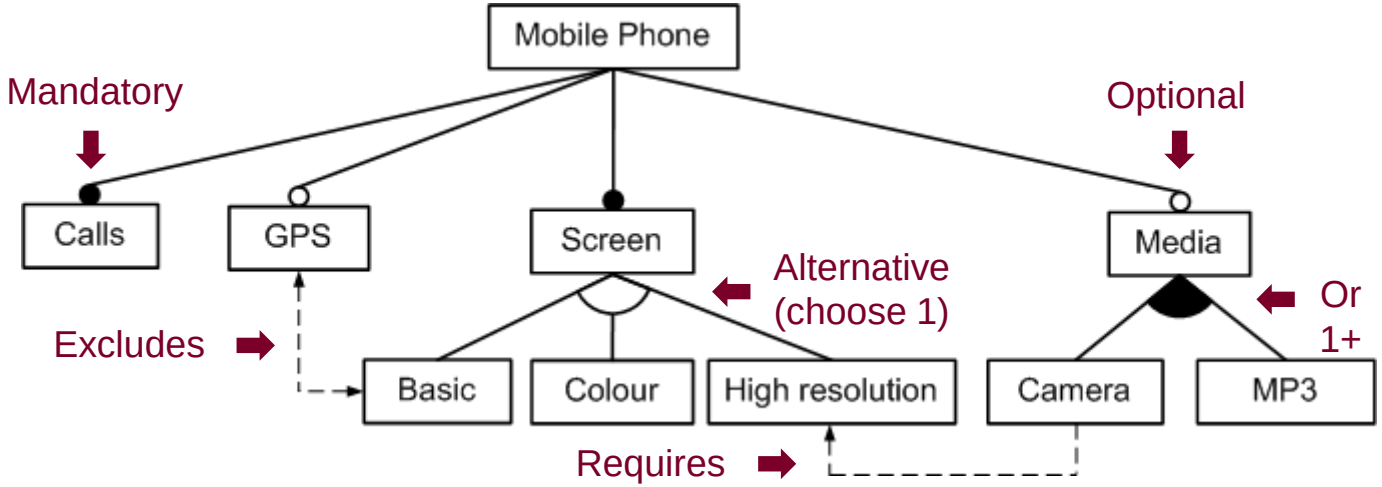
# Feature models



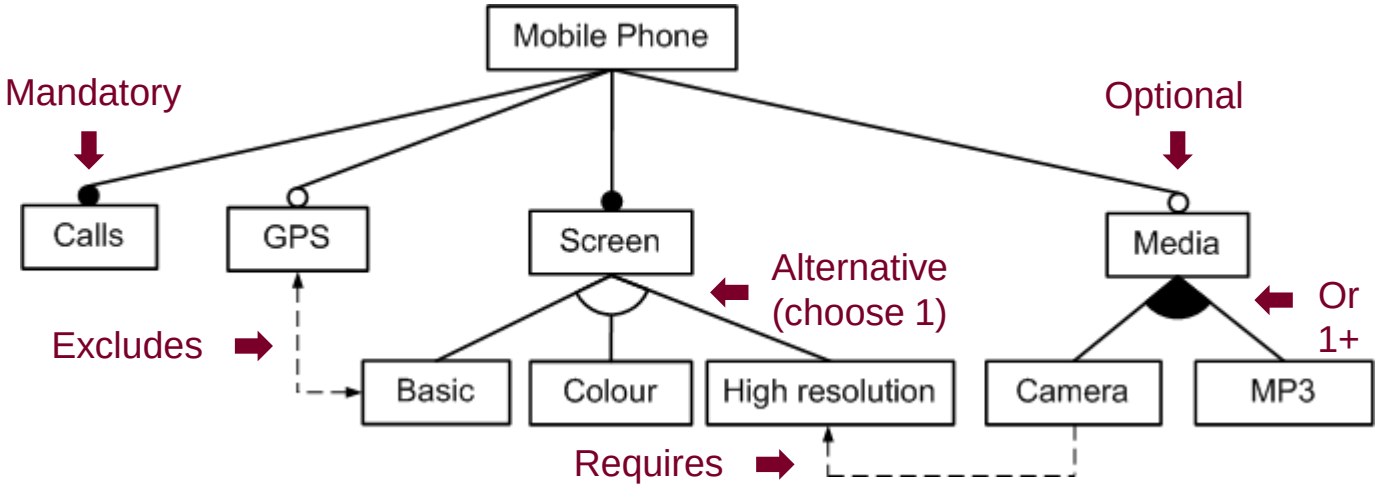
# Feature models



# Feature models

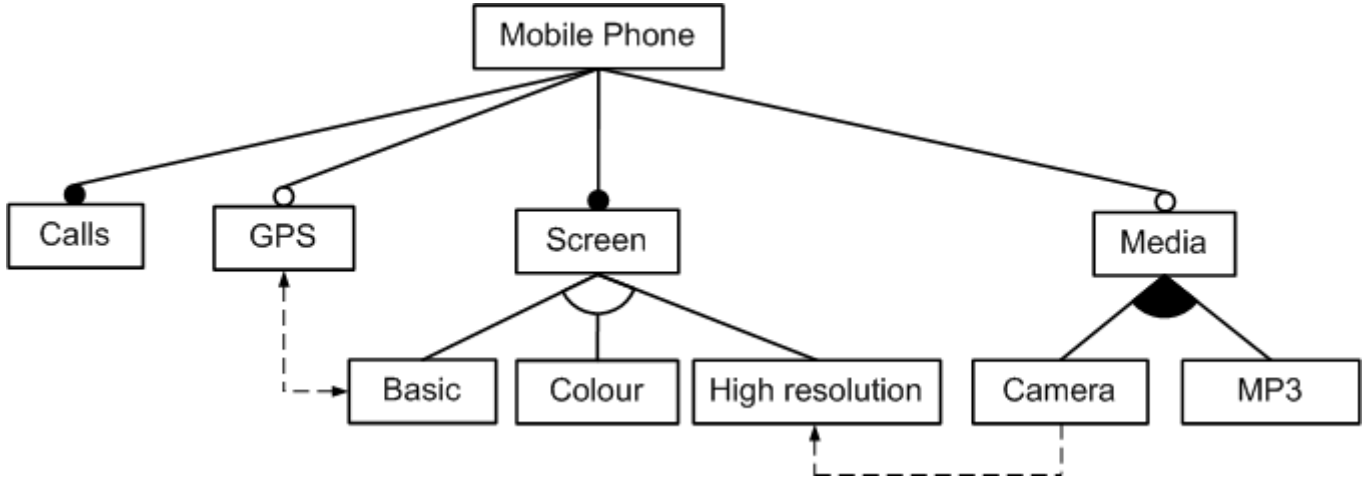


# Feature models



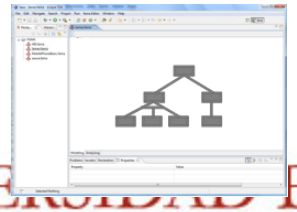

- **Mandatory**: A -> B && B-> A
- **Optional**: B -> A
- **Alternative**: sum (B_i) = 1 && B_i -> A
- **Or**: sum (B_i) > 1 && (B_i) <= count(B)
- **Requires**: A -> B
- **Excludes**: A -> !B && B -> !A

# Automated analysis of feature models:
## Computer-aided extraction of information from FMs

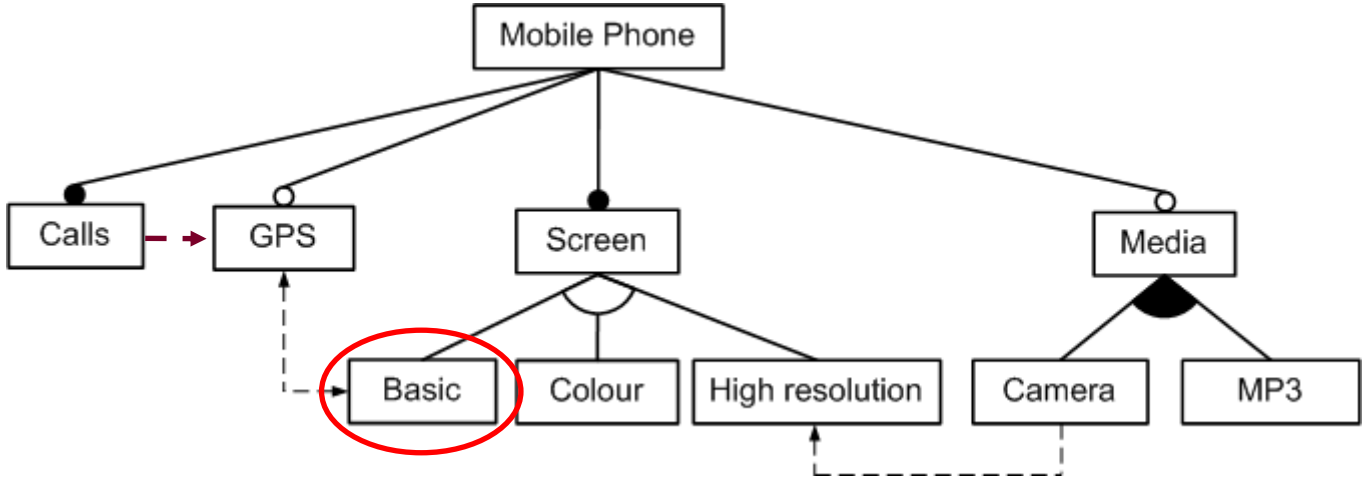



How many products?

14

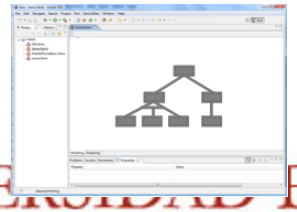


# Automated analysis of feature models:
Computer-aided extraction of information from FMs

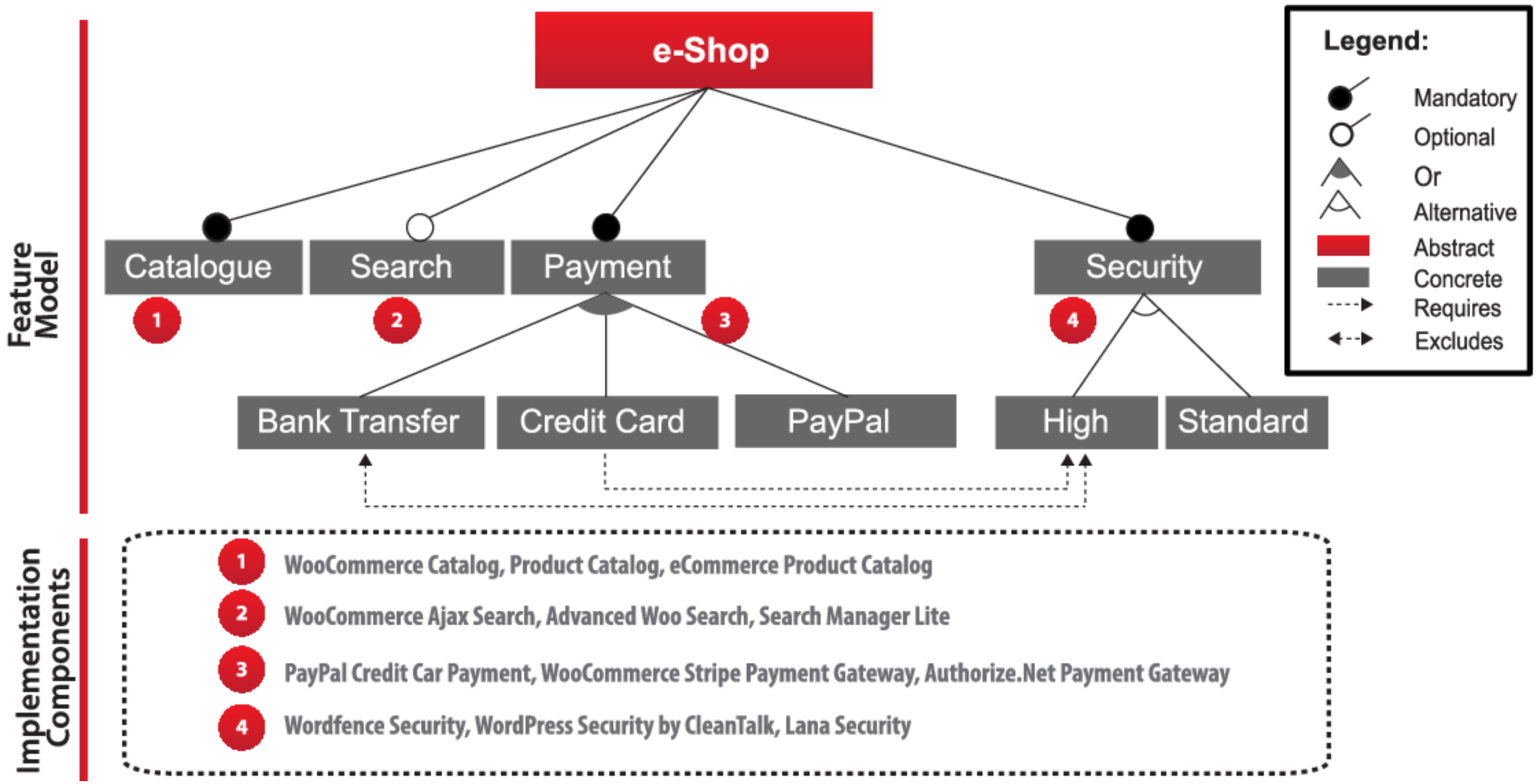


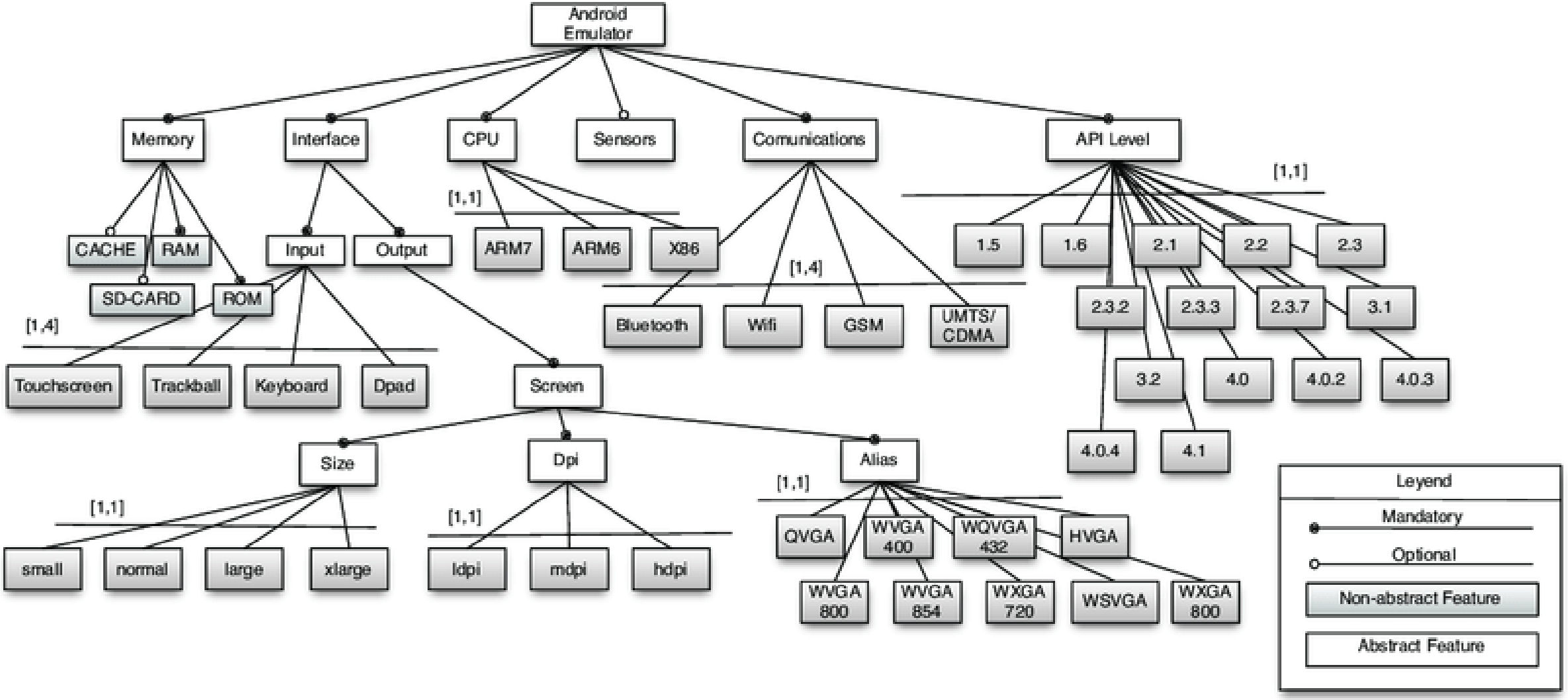

Any error?

Yes, feature "Basic" is dead



# Other problems!




Android Emulator
Memory
Interface
CPU
Sensors
Comunications
API Level
CACHE
RAM
SD-CARD
ROM
Input
Output
[1,1]
ARM7
ARM6
X86
[1,4]
Bluetooth
Wifi
GSM
UMTS/ CDMA
[1,1]
1.5
1.6
2.1
2.2
2.3
2.3.2
2.3.3
2.3.7
3.1
3.2
4.0
4.0.2
4.0.3
4.0.4
4.1
[1,4]
Touchscreen
Trackball
Keyboard
Dpad
Screen
Size
Dpi
Alias
[1,1]
small
normal
large
xlarge
[1,1]
ldpi
mdpi
hdpi
[1,1]
QVGA
WVGA 400
WQVGA 432
HVGA
WVGA 800
WVGA 854
WXGA 720
WSVGA
WXGA 800
Leyend
Mandatory
Optional
Non-abstract Feature
Abstract Feature

# Bibliografía

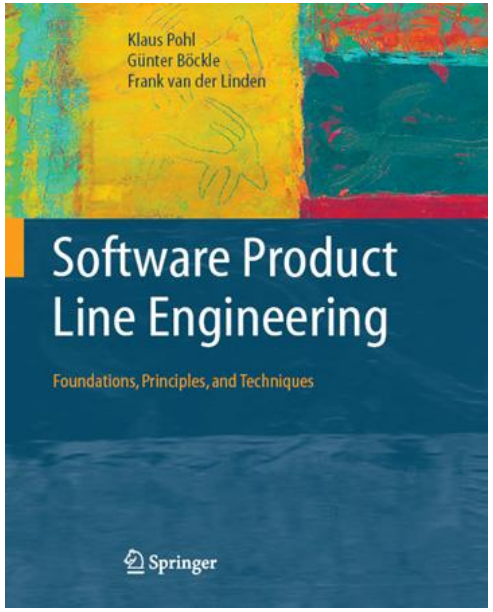
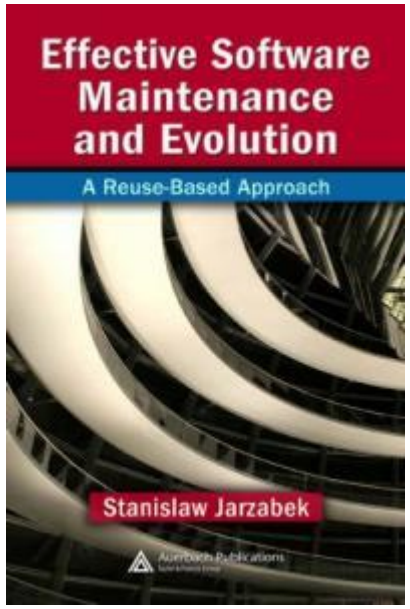
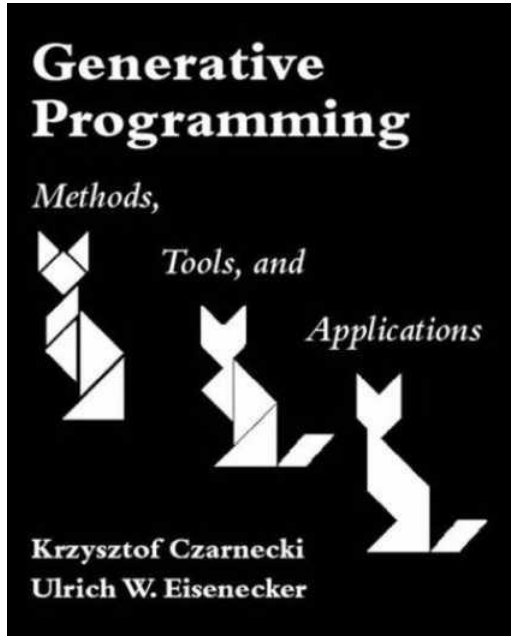
JOURNAL OF OBJECT TECHNOLOGY
Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering. ©JOT, 2009
Vol. 8, No. 6, September–October 2009

## Virtual Separation of Concerns – A Second Chance for Preprocessors

**Christian Kästner,** School of Computer Science, University of Magdeburg, Germany
**Sven Apel,** Department of Informatics and Mathematics, University of Passau,

SOFTWARE—PRACTICE AND EXPERIENCE
*Softw. Pract. Exper.* 2005; **35**:705–754
Published online 1 April 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/spe.652

## A taxonomy of variability realization techniques‡

Mikael Svahnberg[1,*,†], Jilles van Gurp[2] and Jan Bosch[3]

# Bibliografía