Optimalizácia vývoja softvéru pomocou softvérových produktových linii

Prednáška 10

20.11.2024

MSOFT

Real-time	Fault-tolerant	70 separate procesors
30 different local area network nodes	Radar interface and other sensors	Missile and torpedo launchers
Complex and highly demanding human-computer interface	Quality is everything	Robust
Reliable	Avoid host, performance, distribution, communication and other errors	…

Real-time

Fault-tolerant

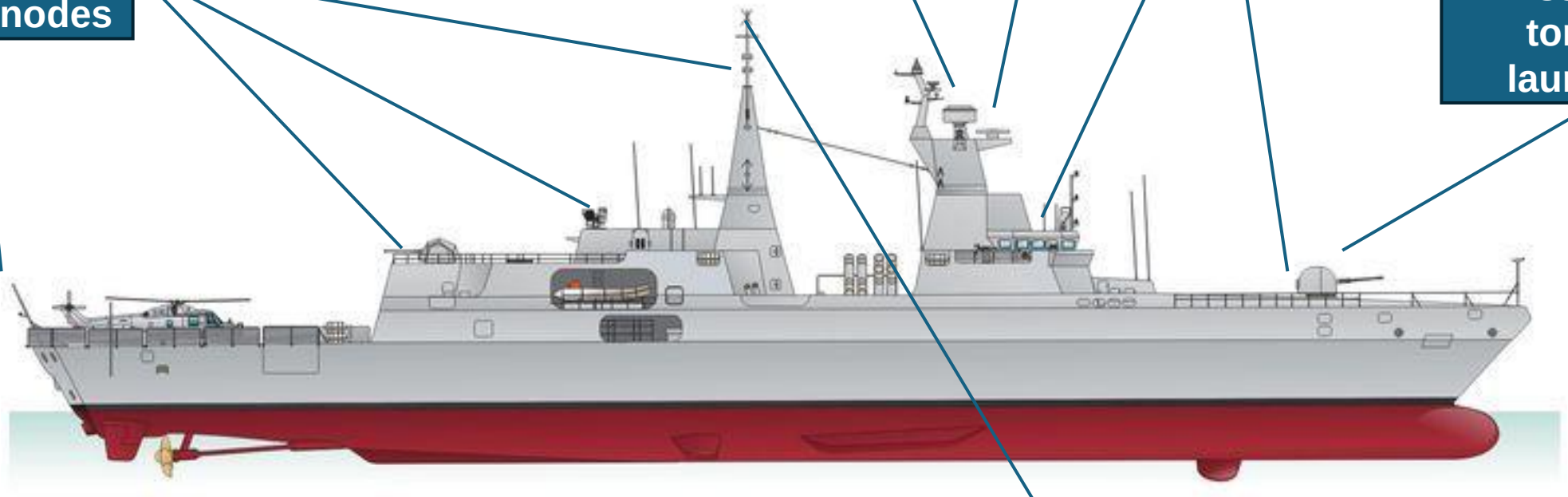
Radar interface and other sensors

70 separate procesors

Robust

30 different local area network nodes

Missile and torpedo launchers

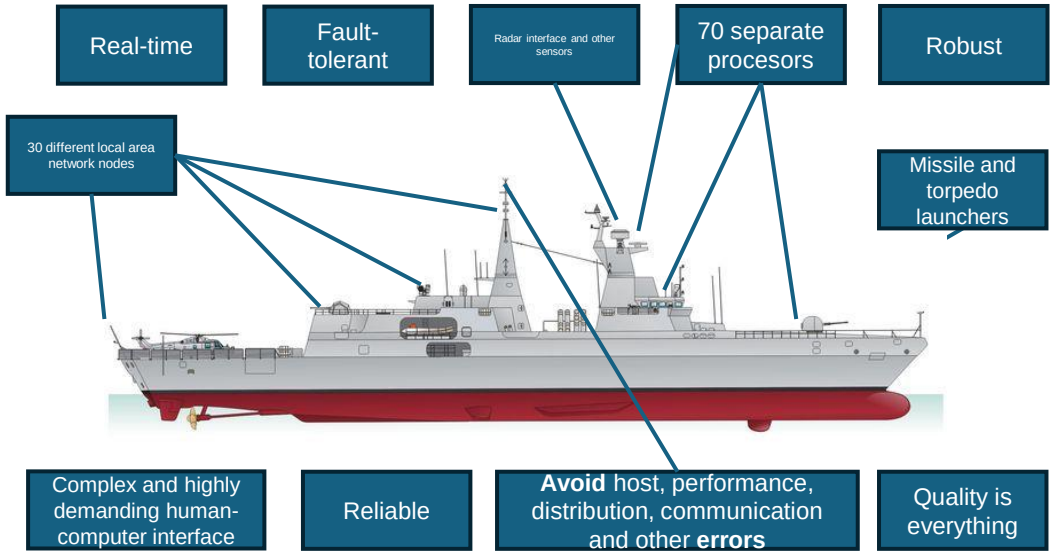


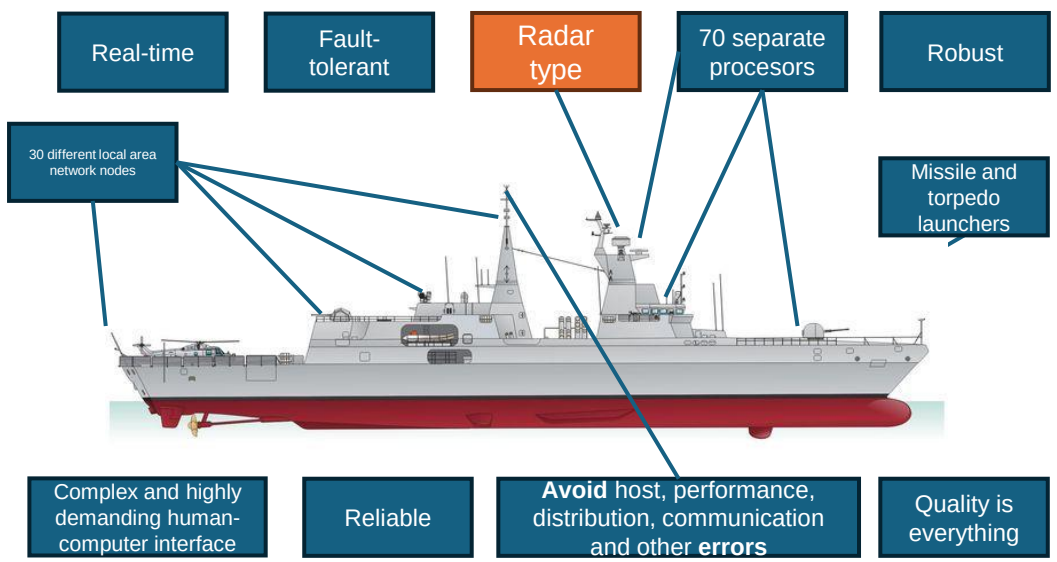
Complex and highly demanding human-computer interface

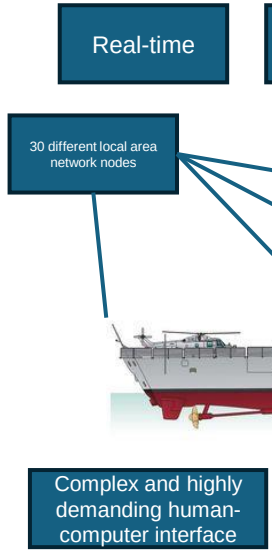
Reliable

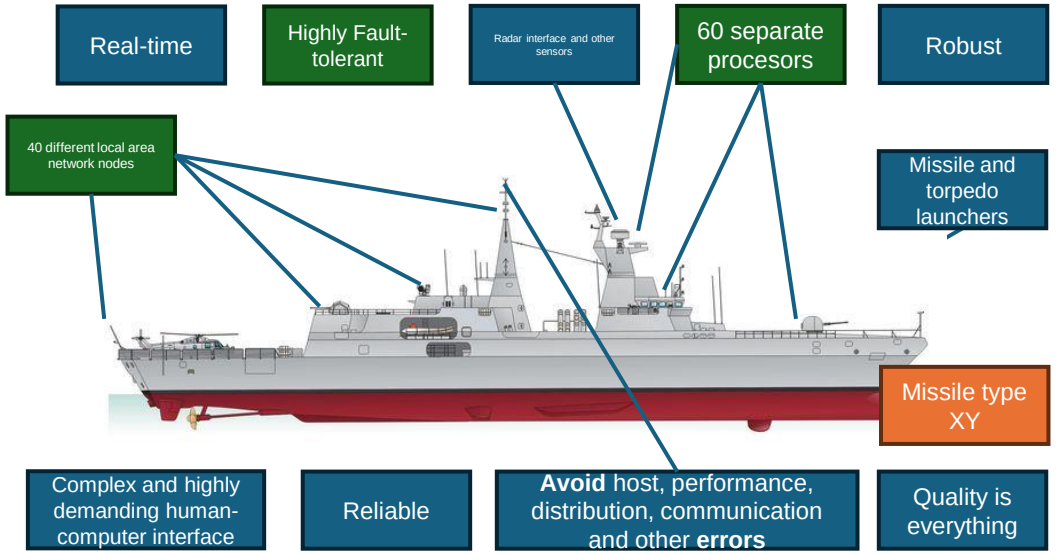
Avoid host, performance, distribution, communication and other errors

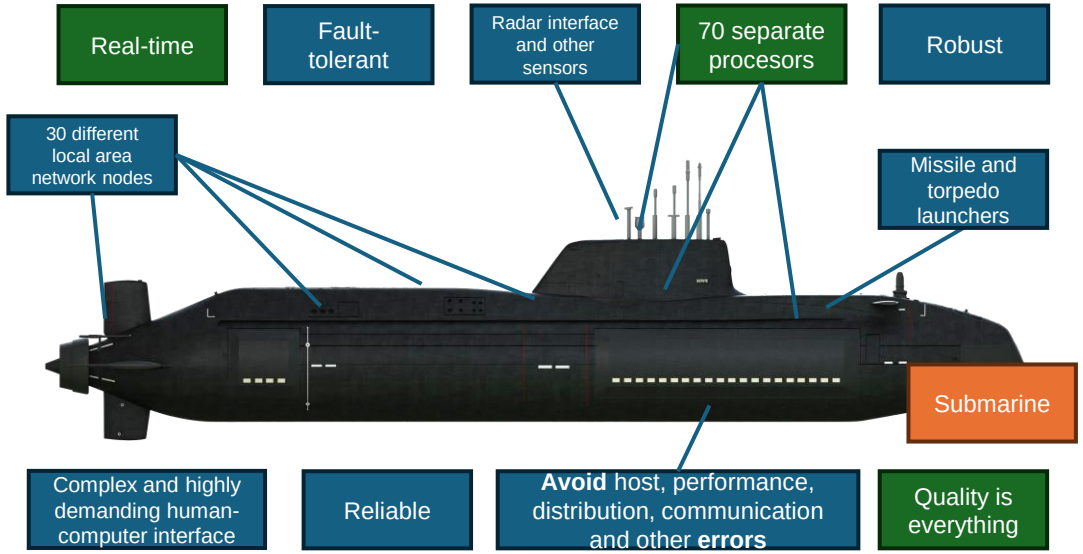
Quality is everything

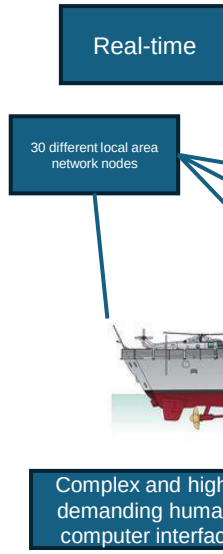

Real-time
Fault-tolerant
Radar interface and other sensors
70 separate procesors
Robust
30 different local area network nodes
Missile and torpedo launchers
Complex and highly demanding human-computer interface
Reliable
Avoid host, performance, distribution, communication and other errors
Quality is everything


Real-time
Fault-tolerant
Radar type
70 separate procesors
Robust
30 different local area network nodes
Missile and torpedo launchers
Complex and highly demanding human-computer interface
Reliable
Avoid host, performance, distribution, communication and other errors
Quality is everything


Real-time
30 different local area network nodes
Complex and highly demanding human-computer interface


Real-time
Highly Fault-tolerant
Radar interface and other sensors
60 separate procesors
Robust
40 different local area network nodes
Missile and torpedo launchers
Missile type XY
Complex and highly demanding human-computer interface
Reliable
Avoid host, performance, distribution, communication and other errors
Quality is everything


Real-time
Fault-tolerant
Radar interface and other sensors
70 separate procesors
Robust
30 different local area network nodes
Missile and torpedo launchers
Submarine
Complex and highly demanding human-computer interface
Reliable
Avoid host, performance, distribution, communication and other errors
Quality is everything


Real-time
30 different local area network nodes

Vy ste manažér/kou tohto projektu

Čo urobíte?

Panika?

Výpoveď?

Krajina tretieho sveta?

Lacnejšie a rýchlejšie

Vyššia kvalita

Spokojnosť zákazníkov

Len správna kombinácia komponentov

CelsiusTech Systems AB

CelsiusTech začal vyvýjať oba systémy súčasne použíl na to rovnaké zdroje

nerobili

Vytvorili si *Software product line*

Spôsob ako sa veci vyrábajú sa výrazne zmenil

Minulosť

Na mieru – ručná výroba

- Individuálny prístup
- Nedá sa vyrobiť veľké množstvo

Nedávno

Jeden typ - Pásová výroba

- “One size fits all”
- Masová výroba

Mass Customisation

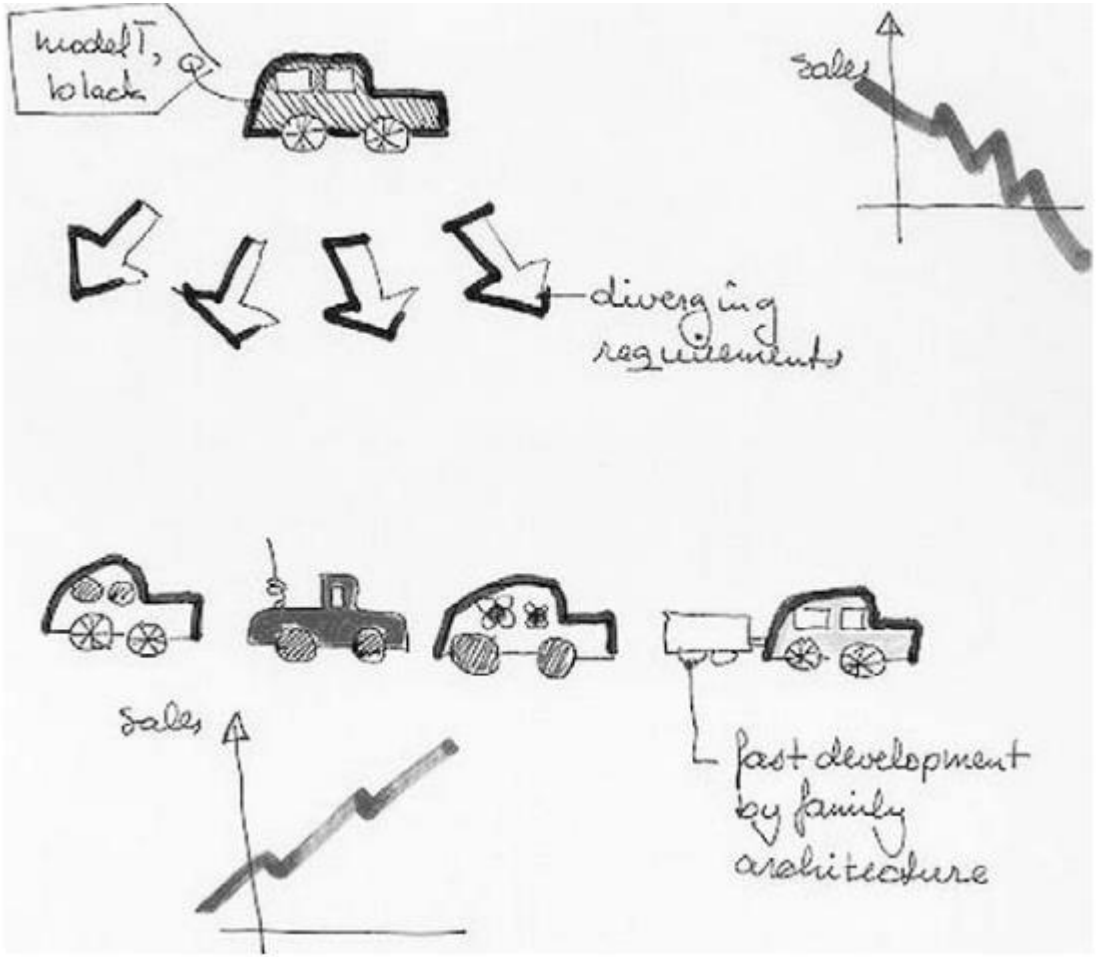
Mass customisation is the large-scale production of goods tailored to individual customers needs

Autá

Pre jednotlivcov/rodinné

Mesto/vidiek/terénne

Rýchle/úsporné



Fuji vs. Kodak

Fuji 1987

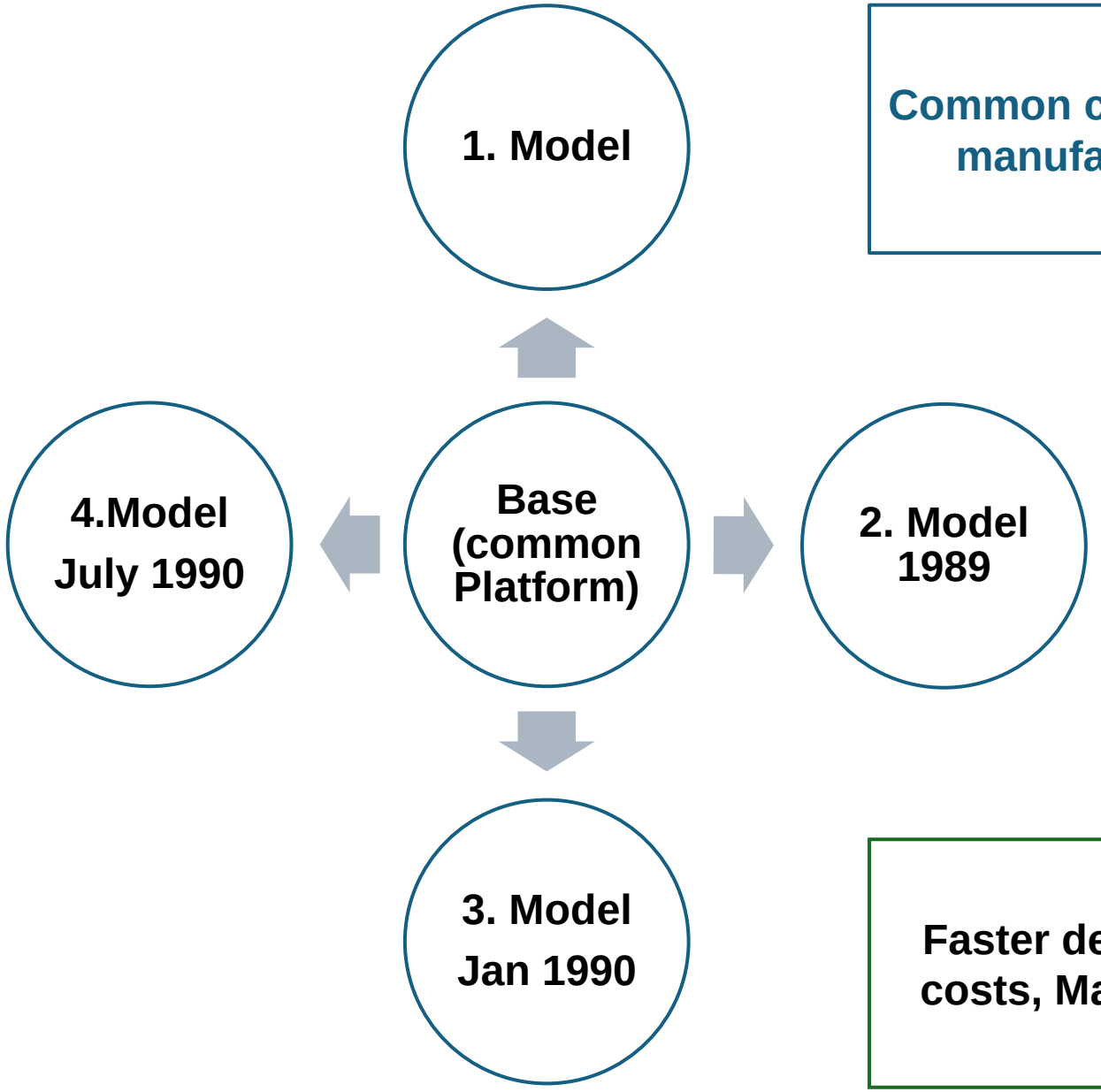
- Quicksnap – single-use camera

Fuji

- 1988 => 3 millióny
- 1994 => 43 milónov

Kodak 1994

- Won the market share back by 70%
- How?



Common components + same manufacturing process

Faster development, lower costs, Mass customisation

Platform

A platform is any base of technologies on which other technologies or processes are built

Mass Customisation and platform-based development

Allows to reuse a common base of technology and at the same time bring out products in close accordance with customers wishes.

Software Product Line Engineering

Software Product Line Engineering

Software product line engineering is a paradigm to develop software application (software-intensive systems and software products) using platforms and mass customisation

Aké časti OS vášho telefónu umožňujú mass customisation v rámci platfromy?

Zníženie nákladov na vývoj

Keď sa znovu používajú komponenty v rôznych ďalších systémoch, znižujú sa výdavky

Znovu použiteľnosť musí byť plánovaná dopredu a je potrebné vynaložiť úsilie pre vznik platformy

Lower development cost

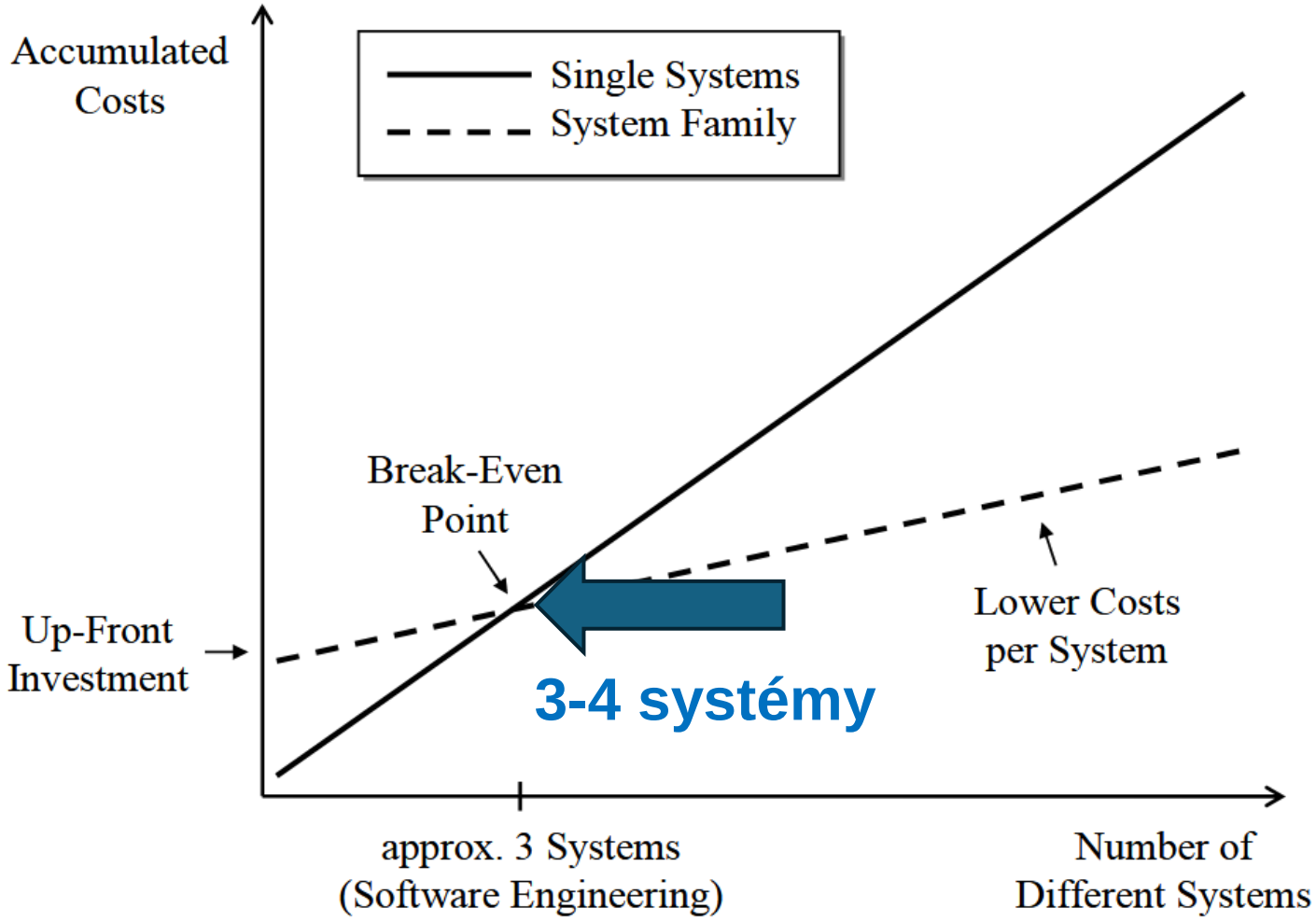


Fig. 1-2: Costs for developing n kinds of systems as single systems compared to product line engineering

Domain engineering

Domain engineering is the process of software product line engineering in which the commonality and the variability of the product line are defined and realised.

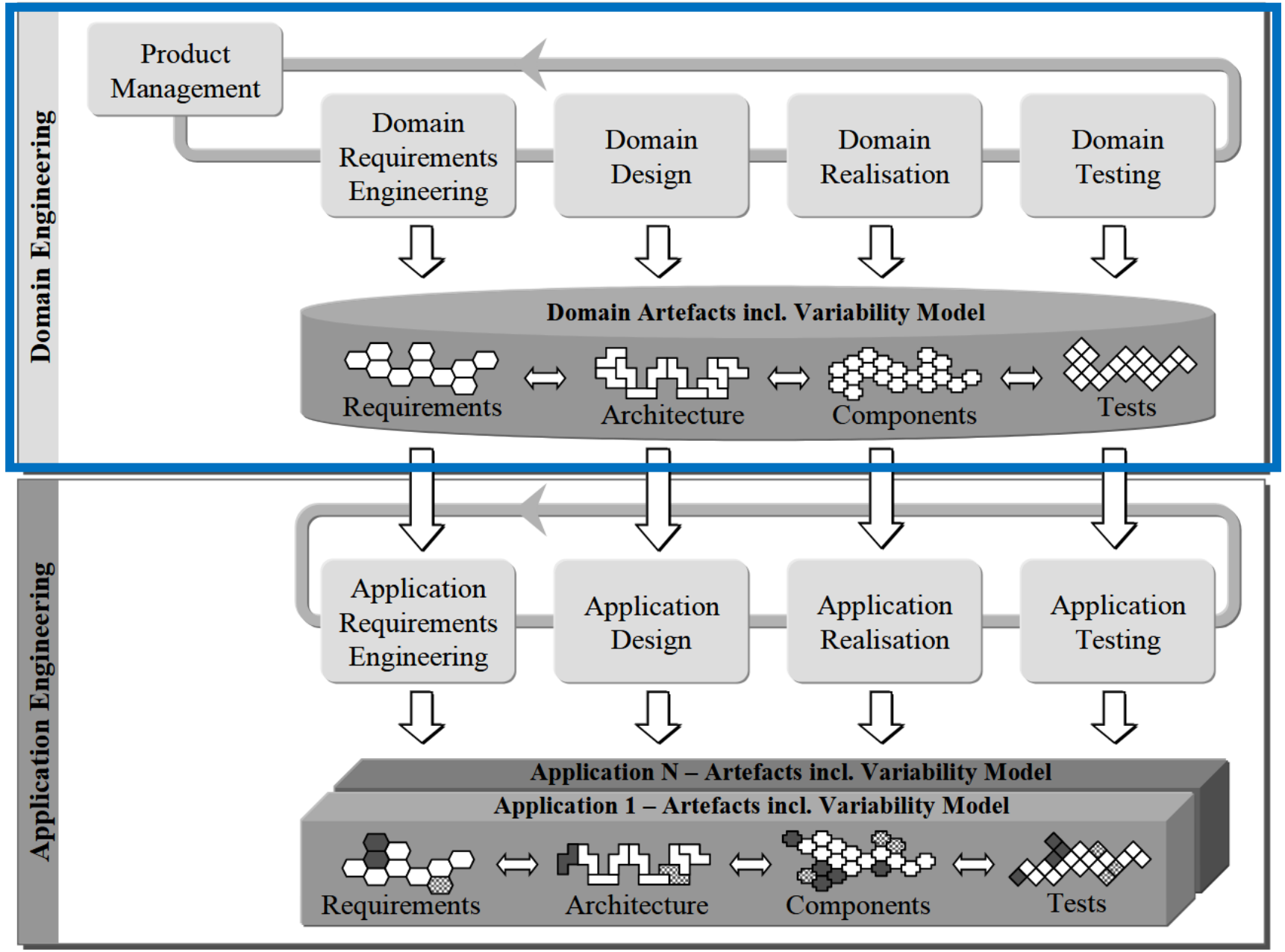


Fig. 2-1: The software product line engineering framework

Application Engineering

Application engineering is the process of software product line engineering in which the applications of the product line are built by reusing domain artefacts and exploiting the product line variability.

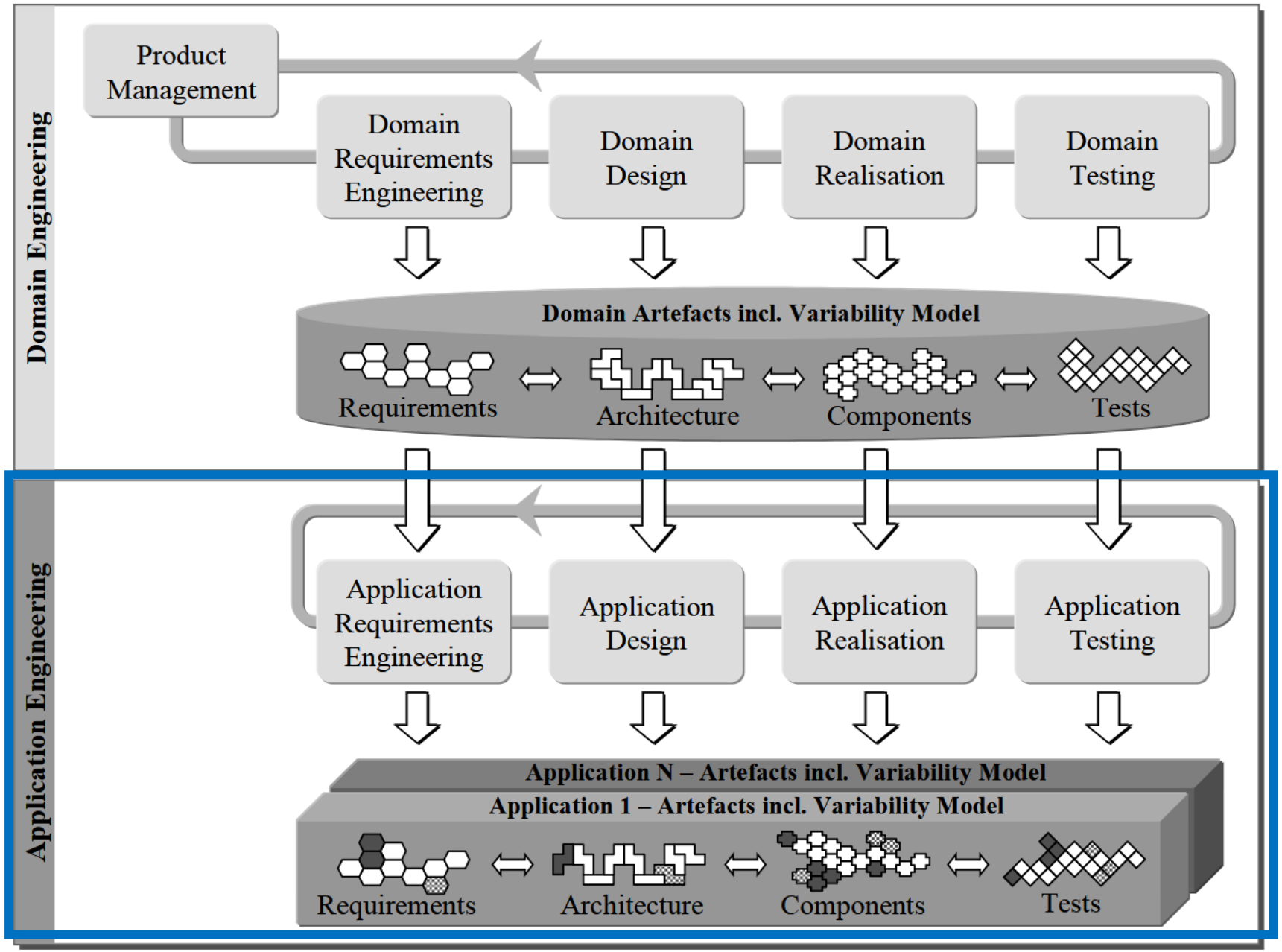


Fig. 2-1: The software product line engineering framework

Zlepšovanie kvality

Artefakty v rámci platformy sú testované a kontrolované vo viacerých produktoch

Vyššia šanca odhalenia chýb a problémov

Zníženie *Time to Market*

Pre single-product development je *Time to Market* relatívne konštantný

Pre SPL je *Time to Market* zo začiatku, áno, vyšší – najprv musia byť vyvinuté spoločné artefakty. Po tomto sa *Time to Market* výrazne zníži.

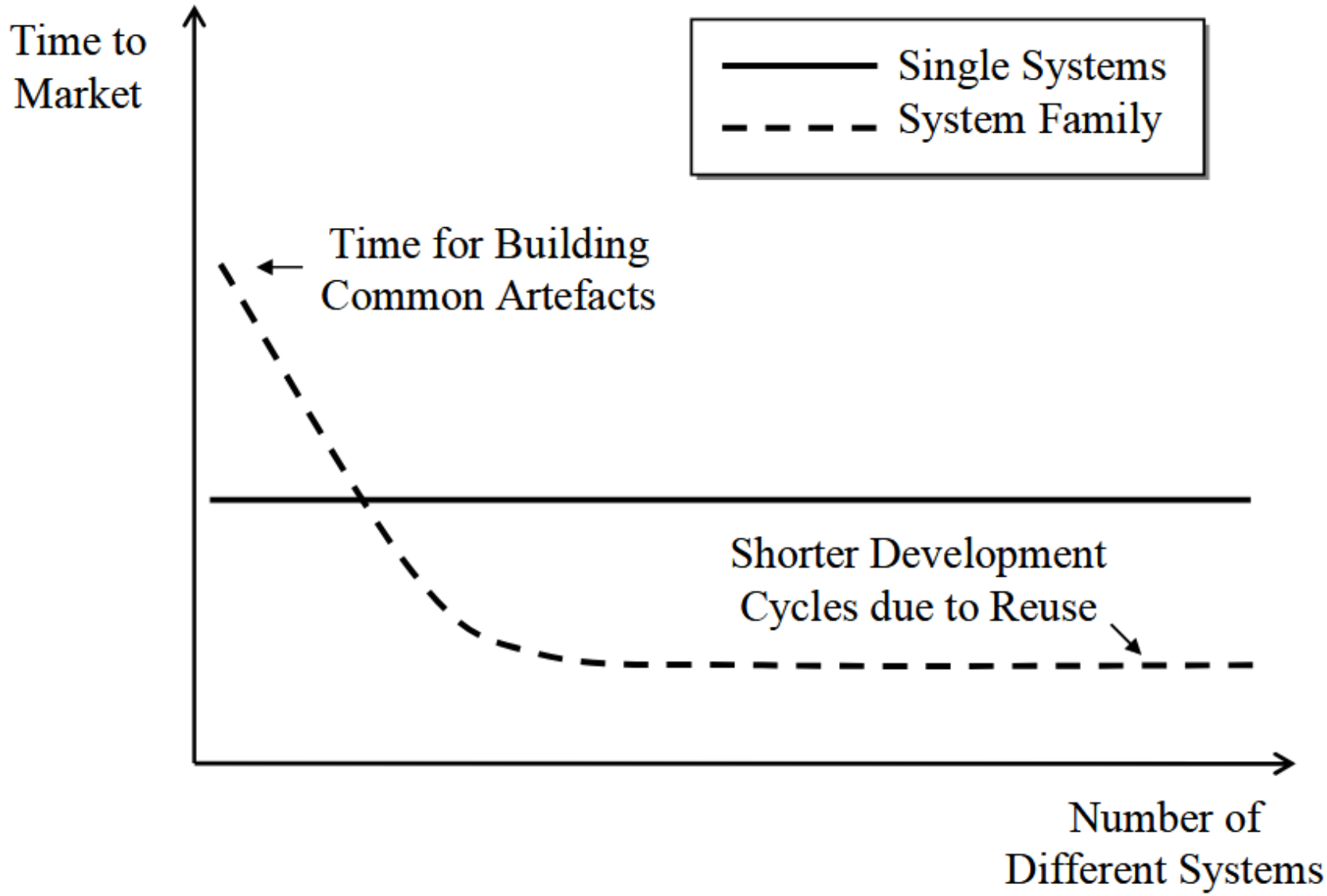


Fig. 1-3: Time to market with and without product line engineering

Zníženie úsilia na správu

Evolúcia a vývoj

Výhody pre zákazníkov

Komplexita

Zlepšenie odhadu nákladov

Variantnosť softvéru

Vývoj aplikácii využívajúc *Platform* znamená proaktívne plánovať pre znovu použiteľnosť,

Vývoj aplikácii ktoré používajú *Mass customisation* znamená použiť concept manažovanej variantnosti

Čo sa mení?

Variability Subject

A variability subject is a variable item of the real world or a variable property of such an item

Prečo sa to mení?

Ako sa to mení?

Variability Object

A variability object is a particular instance of a variability subject.

Variation Point

A variation point is a representation of a variability subject within domain artefacts enriched by contextual information.

Variant

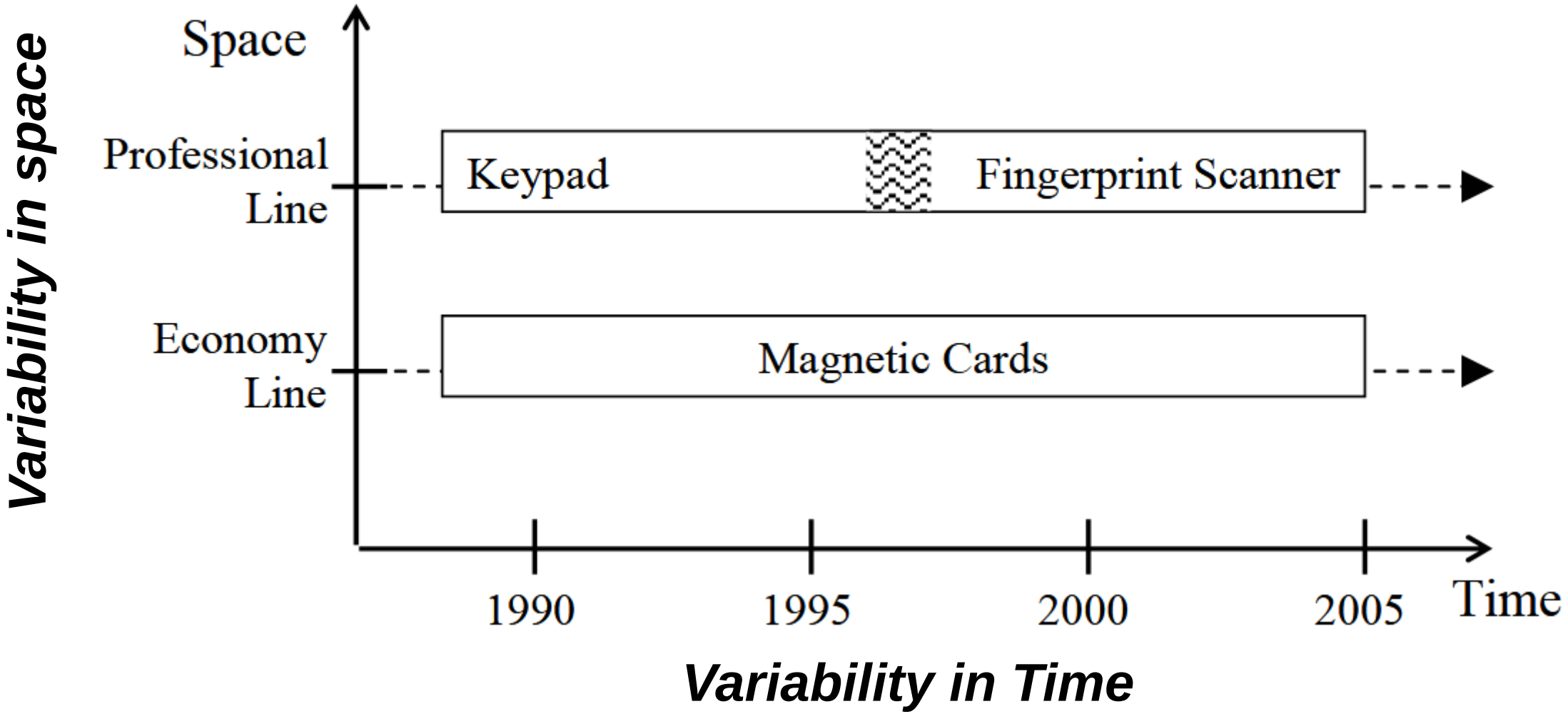
A variant is a representation of a variability object within domain artefacts

	Reálny svet	*Software product line (Context)*
Subjekt	**Variability Subject** **Colour**	**Variation Point** **Colour of a car**
Objekt	**Variability Object** **red, green, blue…**	**Variant** **red, blue**

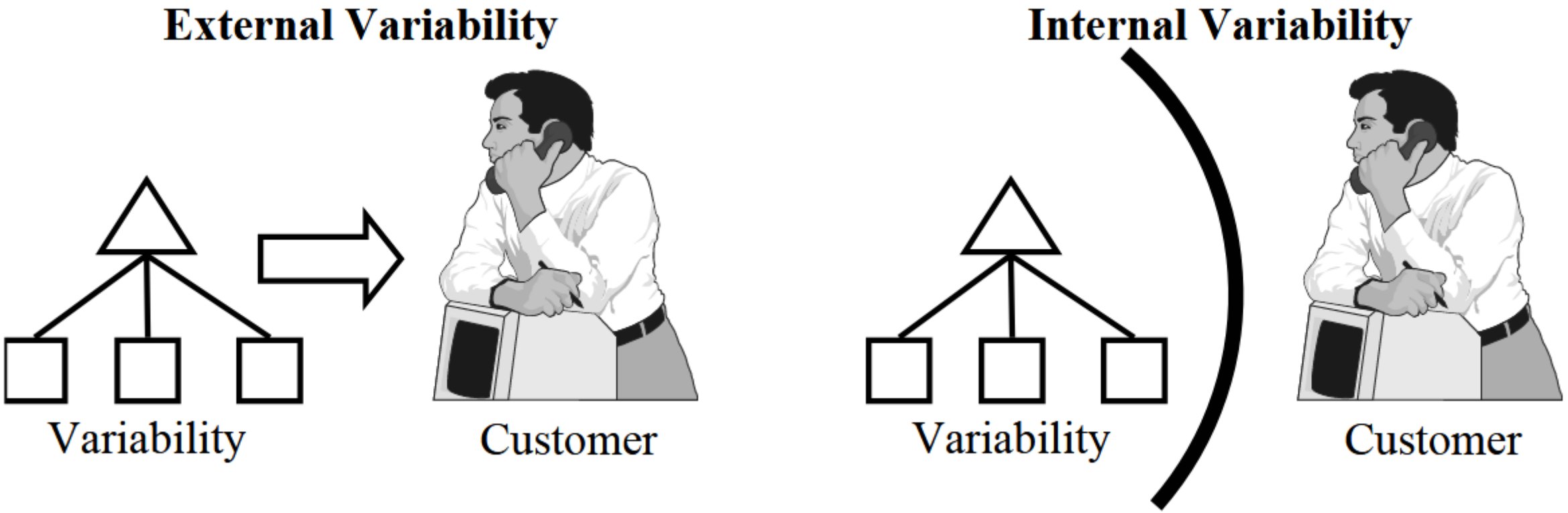
Napr.

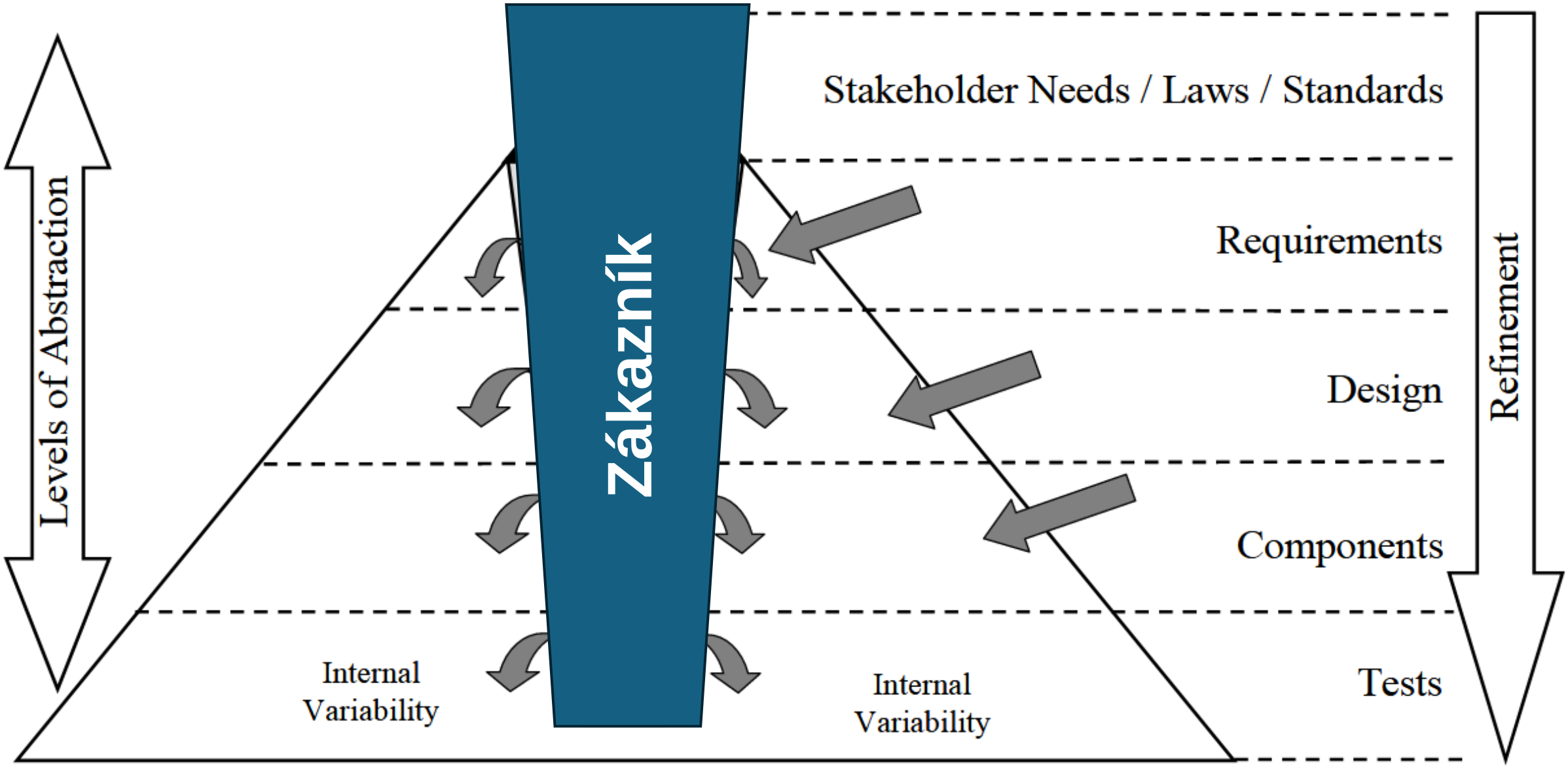
Variability Subject	Variability Object
Colour	Red, green, blue, yellow
Payment Method	Card, cash, check, bitcoin
Identification Mechanism	Keypad, tag, fingerprint scanner, face scanner

Variantnosť môže mať rôzne podoby

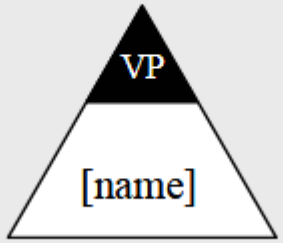


Variantnosť z pohľadu zákazníka

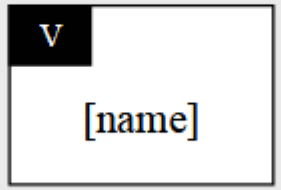



Levels of Abstraction
Zákazník
Stakeholder Needs / Laws / Standards
Requirements
Design
Components
Tests
Refinement
Internal Variability
Internal Variability

Variation Point



Variant



Variability Dependencies

optional - - - -

mandatory ——

Alternative Choice

[min..max]

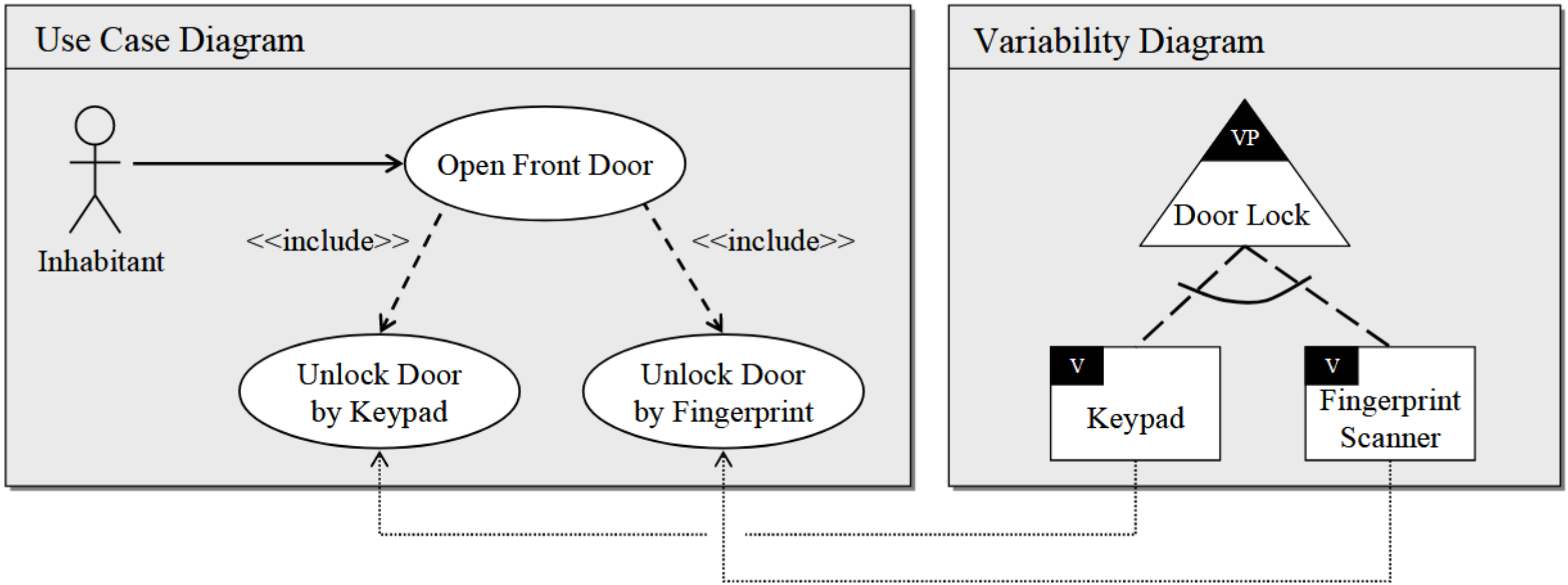
Artefact Dependencies

artefact dependency ·····→

VP artefact dependency -·-·-·-·→

Constraint Dependencies

requires_V_V	·····requires_v_v·····→	requires_V_VP	·····requires_v_vp·····→	requires_VP_VP	·····requires_vp_vp·····→
excludes_V_V	·····excludes_v_v·····→	excludes_V_VP	·····excludes_v_vp·····→	excludes_VP_VP	·····excludes_vp_vp·····→


Use Case Diagram
Inhabitant
Open Front Door
<<include>>
<<include>>
Unlock Door by Keypad
Unlock Door by Fingerprint
Variability Diagram
VP
Door Lock
V
Keypad
V
Fingerprint Scanner

Ďalšie čítanie

POHL, Klaus; BÖCKLE, Günter a LINDEN, Frank van der. *Software Product Line Engineering: Foundations, Principles and Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. Dostupné z: https://doi.org/978-3-540-28901-2.