



Variability and Software Product Line Architecture

University of São Paulo
Institute of Mathematics and Computer Science

Danillo Ferreira dos Reis
Pedro Henrique Dias Valle

SSC5944 – Software Architecture || 1st Semester 2016



Contents

- ❖ **Introduction**
- ❖ **Software Product Line (SPL)**
- ❖ **Variability**
- ❖ **Software Product Line Architecture (SPLA)**
- ❖ **Example**
- ❖ **Final Considerations**



Introduction

Introduction

- Increase of society and consumption
- Henry Ford
- Product Line
 - ✓ Reduced prices
 - ✓ Large scale production



Credits: <https://goo.gl/zfLBXD>



Credits: <http://goo.gl/81ejh5>

[Pohl et al. 2005]

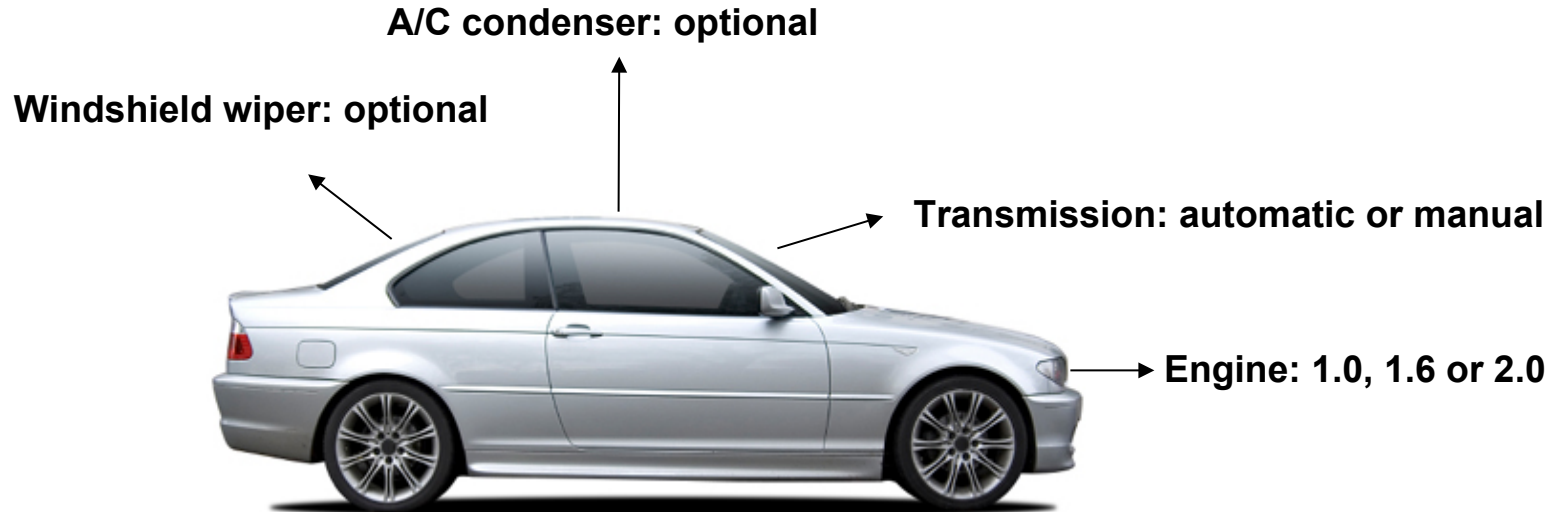
Introduction

- Example



Introduction

- **Example**



Introduction

- Enterprises

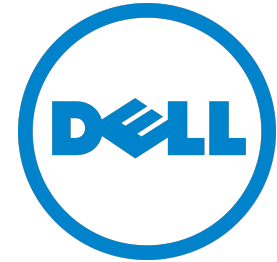
PHILIPS

sense and simplicity



TOSHIBA

Leading Innovation >>>



NOKIA

Connecting People

Introduction

■ Benefits

✓ **Decrease of delivery time**



✓ **Increase in quality**



✓ **Reduction of costs**



✓ **Reduction of time to market**



✓ **Increased productivity in large scale**



[Pohl et al. 2005]



Software Product Line (SPL)

- **Definition**

“Software product line engineering is a paradigm to develop software applications (software-intensive systems and software products) using platforms and mass customisation” [Pohl et. al, 2005].

“A software product line is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way” [Clements, Northrop, 2002].

- **Software Product Line**

- ✓ **Domain engineering:**

- Commonality and variability in software product line

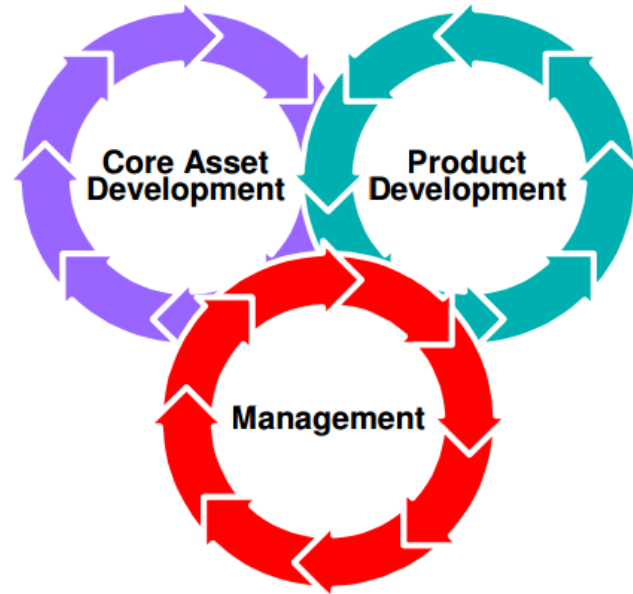
- ✓ **Application engineering**

- Development of applications derived from the platform established in the domain engineering

[Pohl et al. 2005]

SPL

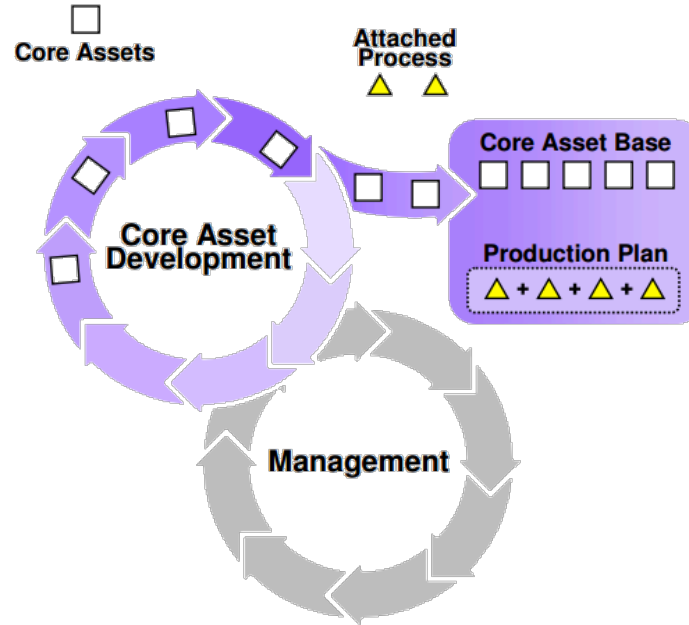
- Software Product Line



[Clements, Northrop, 2002]

SPL

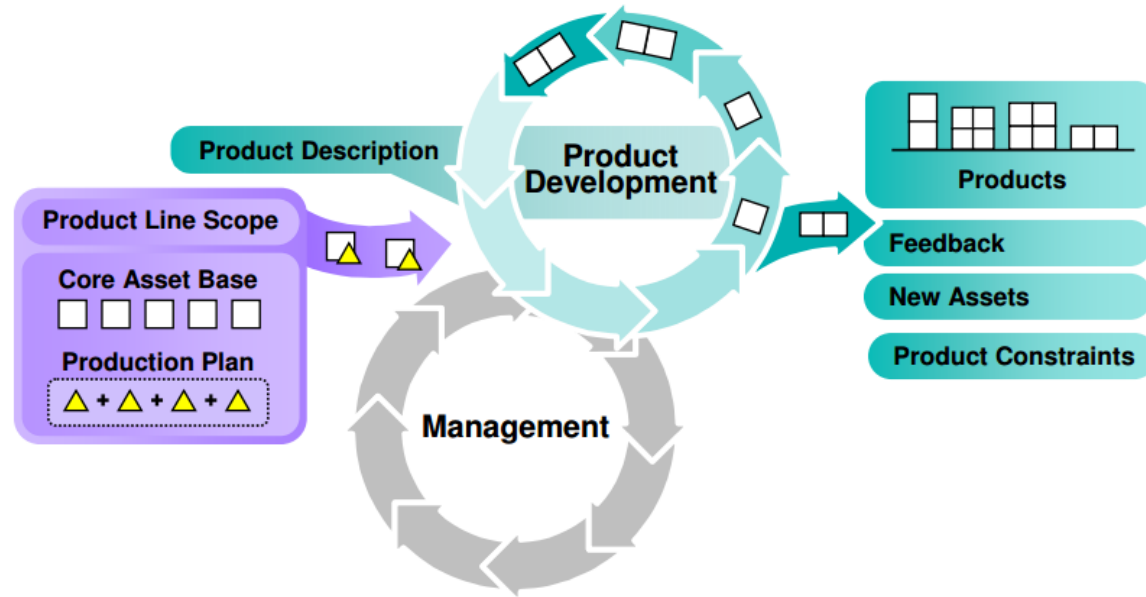
- Software Product Line



[Clements, Northrop, 2002]

SPL

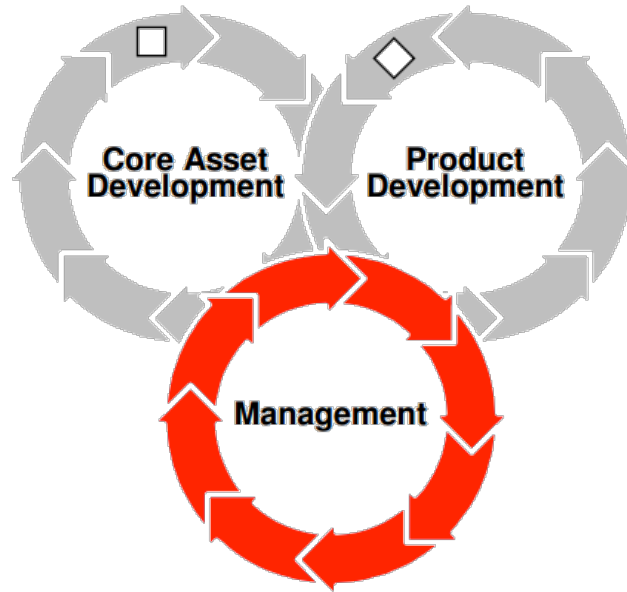
- Software Product Line



[Clements, Northrop, 2002]

SPL

- Software Product Line



[Clements, Northrop, 2002]



Variability

Variability

- **Definition:**

Variability of a software product line is variability that is modelled to enable the development of customized applications by reusing predefined, adjustable artefacts [Pohl et al. 2005].

Variability

- **Software product line are described by representations:**
 - ✓ **Variability Point:** Specific location where will have variabilities
 - ✓ **Variant:** A set of possible alternatives variability

[Pohl et al. 2005]

Variability

- Example

The customers of a home automation system can decide on the language of the user interface before the system is installed. Moreover, for an additional charge, a multilingual version is offered that allows selecting the user's favorite language at any time (e.g. by selecting a flag symbol on a touch screen) [Pohl et al. 2005].

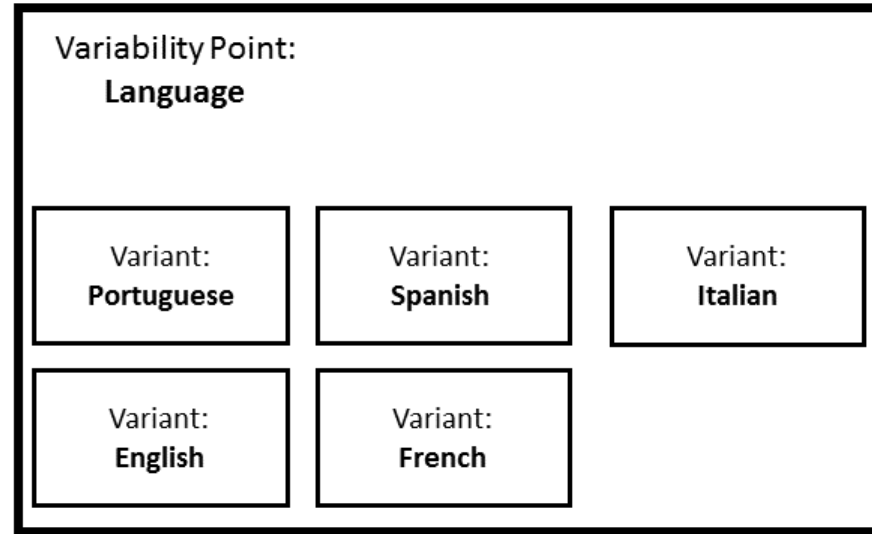
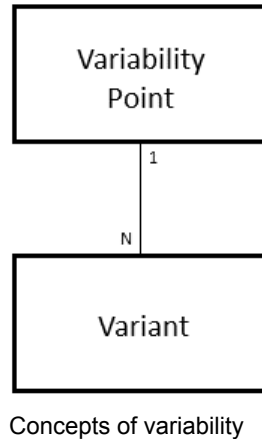


Credits: http://startsollution.com.br/?navega=paginas_interna&id_pag=28&interna=102

[Pohl et al. 2005]

Variability

- **Possibility to change or customize a system**



Variability point and variants

Variability

- **The variability points can be presented in various levels of abstraction**
 - ✓ Architecture Description
 - ✓ Documentation by Diagrams
 - ✓ Source Code
 - ✓ Compiled Code

[da Silva et al, 2011]

Variability

▪ Variability Management

- ✓ Identify Variabilities
- ✓ Introduce the Variability in the System
- ✓ Grouping Variants
- ✓ Including the System in a Variant

[da Silva et al, 2011]

Variability

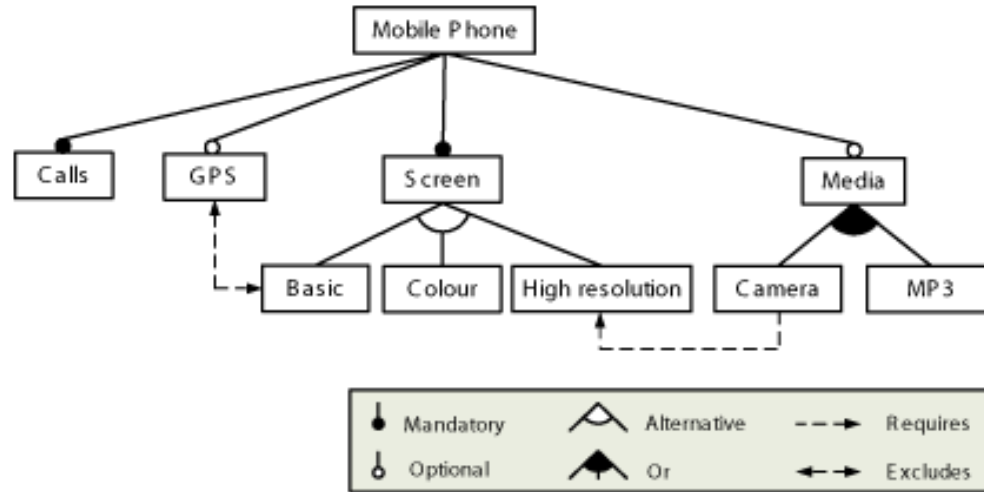
▪ Feature Model

- ✓ **Feature** is a characteristic of the system visible to the final user
- ✓ Used to group a set of related requirements
- ✓ Overview of the main common characteristics and variables of a product line
- ✓ Feature-Oriented Domain Analysis (FODA)

[da Silva et al, 2011]

Variability

■ Feature Model



Credits: <http://goo.gl/7Esd9o>



Software Product Line Architecture (SPLA)

- **Definition:**

A product-line architecture is an abstraction: it is a specification of the high level structures of a family of applications. These structures reveal complementary facets of an architecture (static structure, dynamic structure, etc...) and contain elements like components, connections, data, processes [Bass *et al*, 1998].

SPLA

- **A product-line architecture has to meet three fundamental requirements:**
 - ✓ It has to drive the architectural design of new applications in the product-line;
 - ✓ It has to facilitate the reuse of components at the product-line level;
 - ✓ It has to permit various analyses in order to assess the impact (cost, performance, etc...) of specific requirements for the development of new applications in the product-line.

[Bass et al, 2012]

SPLA

- **A product line architect needs to consider three things that are unique to product line architectures:**
 - ✓ Identify variation points
 - ✓ Support variation points
 - ✓ Evaluate the architecture

[Bass et al, 2012]

Variation Mechanisms

- Modifying code is undesirable
- Three primary architectural variation mechanisms are following
 - ✓ Inclusion or omission of elements
 - ✓ Inclusion of a different number of replicated elements
 - ✓ Selection of different versions of elements that have the same interface, but different behavioral or quality characteristics

[Bass et al, 2012]

Variation Mechanisms

- Sophisticated techniques of variation mechanisms
 - ✓ Extension points
 - ✓ Reflection
 - ✓ Overloading

[Bass et al, 2012]

Variation Mechanisms

- The choice of variation mechanism affects product quality:
 - ✓ The impact of the variation mechanism on quality, such as possible performance penalties or memory consumption
 - ✓ The impact on the mechanism's maintainability

[Bass et al, 2012]

Variation Mechanisms

- Choosing the right variation mechanism affects numerous costs:
 - ✓ The skill set required to implement, learn, and use the specific variation mechanism
 - ✓ The one-time costs of building or acquiring the tools
 - ✓ The recurring cost and time to exercise the variation mechanism

[Bass et al, 2012]

Variation Mechanisms

- The documentation for a product line architecture differs from that of a conventional architecture
- The architecture documentation should also describe how its variation points are exercised

[Bass et al, 2012]

Evaluation

- The architecture for a software product line should be evaluated for fitness of purpose
- Robustness and generality
- To make sure it meets the specific behavioral and quality requirements of the product at hand

[Bass et al, 2012]

What and How to Evaluate

- The focus of evaluation will be variations points
- They offer sufficient flexibility to cover the product line's intended scope
- They allow products to be built quickly
- They do not impose unacceptable runtime performance costs

[Bass et al, 2012]

What and How to Evaluate

- Evaluation is scenario based
- Different products in the product line may have different quality attribute requirements
- The architecture will have to be evaluated for its ability to provide all required combinations
- Try to elicit scenarios that capture the quality attributes required of family members

[Bass et al, 2012]

When to Evaluate

- Instance or variation of the architecture
- The evaluation artifacts will have potential reuse
- Provide useful feedback to the product line architects and fuel architectural improvements
- The product line architecture can be reevaluated

[Bass et al, 2012]

When to Evaluate

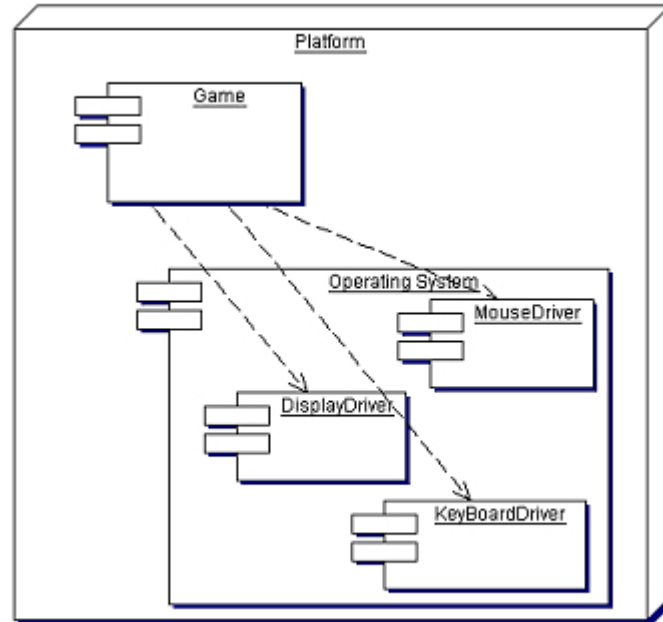
- The evaluation can determine how the architecture will have to be modified to accommodate the new product
- To determine architectural risks and to understand economic consequences

[Bass et al, 2012]



Example

Example



Process Allocation for AGM Games



Final Considerations

Considerations

- SPL is a trend of how to produce software product lines, joining productivity, quality and reuse
- A software product line architecture is an abstraction, in other words, it is a specification of the high level structures of a family of applications
- SPLA can help to identify more than similarities and variations management. It can also help non technical issues must be addressed including how the organization adopts the model, structures itself and maintains its external interfaces



References

References

- ❑ Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley Professional, 3rd edition.
- ❑ Clements, Paul, and Linda Northrop. "Software product lines: practices and patterns." (2002).
- ❑ Pohl, Klaus, Günter Böckle, and Frank J. van Der Linden. Software product line engineering: foundations, principles and techniques. Springer Science & Business Media, 2005.
- ❑ da Silva, F. A. P., Neto, P. A. D. M. S., Garcia, V. C., & Muniz, P. F. . "Linhas de Produtos de Software: Uma tendência da indústria." (2011).



Thank you!