

## Success with Variants

How to establish product line engineering that lasts

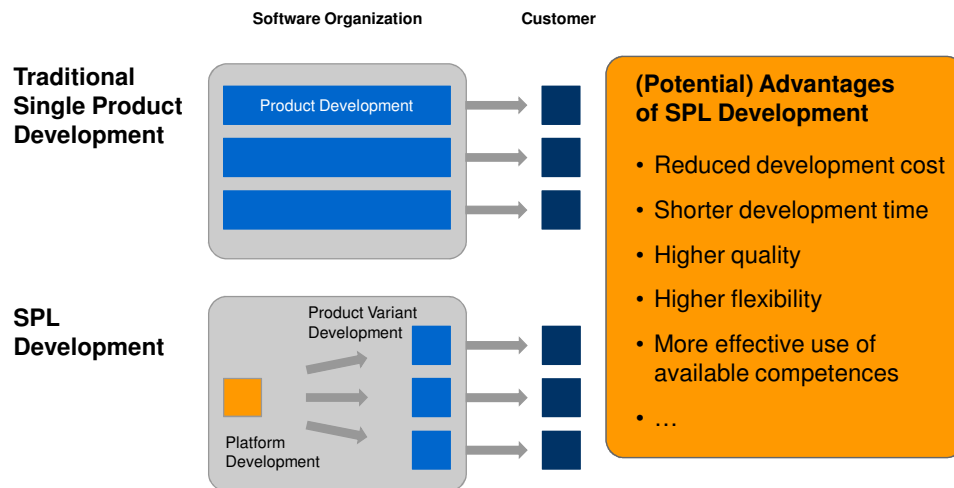
Dr. Andreas Birk  
SABRE MDD/PL Conference, Leipzig  
24 September 2007

## Software Product Line (SPL) Definition

A software product line (SPL) is a set of software-intensive systems sharing a common, managed set of features that satisfy 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.

*(P. Clements ,L. Northrop, CMU SEI, 2002)*

## What is Software Product Line (SPL) development? What are its advantages?



## Lasting SPL success requires strong attention to business, organization, and process aspects

- SPL can be highly productive and beneficial
- However, there are still obstacles to broad SPL acceptance and success
  - Inherent complexity of SPL development
  - Relative newness of the approach
- SPL is most often approached from a technical perspective
- We need stronger attention to business, organization & process aspects of SPL
- Focus of this presentation:

What organizational and process measures help mastering SPL development?

## Agenda

- Introduction
- **Challenges of SPL**
- **Perspectives on SPL**
- **SPL Success Factors**
- **Summary**

## Challenges of SPL

### Results of a comparative case study on SPL challenges

Category	Challenges
Organization and Management	Justification of the platform approach as a process model by a cost / benefit-analysis
	Independent platform team
	Difficult cooperation between platform and product development teams
	Proof of justification of the platform team
	High communication overhead
Requirements engineering	Poor configuration management
	Influence of the architecture on requirements negotiation is not taken into account
	No description of variability for domain analysis
	Missing domain analysis and domain description
	Discussions on design and not on requirements level
Product- vs. platform-specific	No explicit requirements process
	Missing tool support
	Sequence of integrating requirements into the platform
	No explicit prioritization of requirements
Architecture	Realization of platform requirements in products
	Strong influence of the pilot client
	No use of the architectural advantages
	Poor description of the generic architecture

Source: A. Birk, G. Heller, J. Leht, T. von der Maßen, K. Müller, K. Schrick, Product Line Engineering: The State of the Practice, IEEE Software, 20(6), 2003.

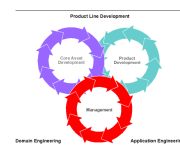
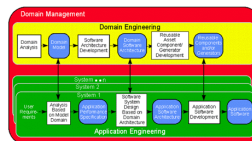
## The five core challenges of SPL development

- Complexity**
  - SPL business decisions and technical solutions are particularly complex
- Competence**
  - High competence required in business, technical, and organizational matters
  - Blend specialized competence profiles to gain interdisciplinary decisions
- Extent**
  - Simultaneously pay attention to business, technology, and management
  - SPL affects many diverse aspects of organization and processes
- Discipline**
  - Conduct processes, practices, and communication in a disciplined manner
  - Lack of discipline can severely challenge the complex SPL processes
- Duration**
  - SPL tend to last long and include several long-term strategic decisions
  - Anticipate future business needs and reliability of technical trends

Mastering the SPL challenges requires a sound understanding of SPL approaches.  
 What are the important perspectives on SPL to gain the required understanding?

## We need new perspectives on SPL development in order to approach the challenges of SPL

Conventional models of SPL are necessary but not sufficient for understanding practical issues of SPL development



We need additional perspectives that help us distinguish between different kinds of SPL development, focusing on aspects such as ...

**Business      Technology      Organization      Process**

Adopting the right perspective for a particular SPL challenge is a great step towards mastering the challenge.

## Overview of important SPL perspectives

<b>Business rationale and SPL structure</b>	Product Suite		Customized Products	
<b>Business decision „buy or build?“</b>	Core Asset Development	Core Asset Acquisition	Combination	
<b>Technical approach to product instantiation</b>	Constructive	Compositional	Generative	Combination
<b>Scope of product line: Assets</b>	Requirements	Architecture	Components	Test Cases
<b>Organizational approach to SPL development</b>	Separate Platform Team	Platform and Product	Mixed	Phased Combination
<b>Organization-wide technical SPL structure</b>	One SPL	Hierarchy of SPL	Interwoven System of SPL	
<b>Overall SPL life cycle process</b>	Instantiation and Preparation	Set-up and Operation	Closure	

24 September 2007

Copyright © 2007, Software.Process.Management, Dr. Andreas Birk

9

## Business rationale and SPL structure

### Product Suites



- Existing applications are integrated over time
- Customer uses more than one product from SPL
- SPL motivation: Reduced cost of ownership
- Example: Microsoft Office

### Customized Products







- New applications are instantiated from the framework
- Customer uses only one product from SPL
- SPL motivation: Delivery efficiency
- Examples: Cummings, Bosch

24 September 2007

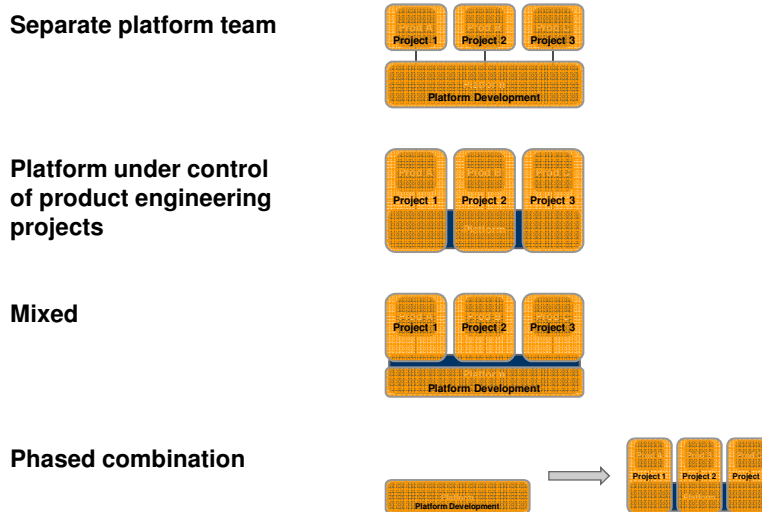
Copyright © 2007, Software.Process.Management, Dr. Andreas Birk

10

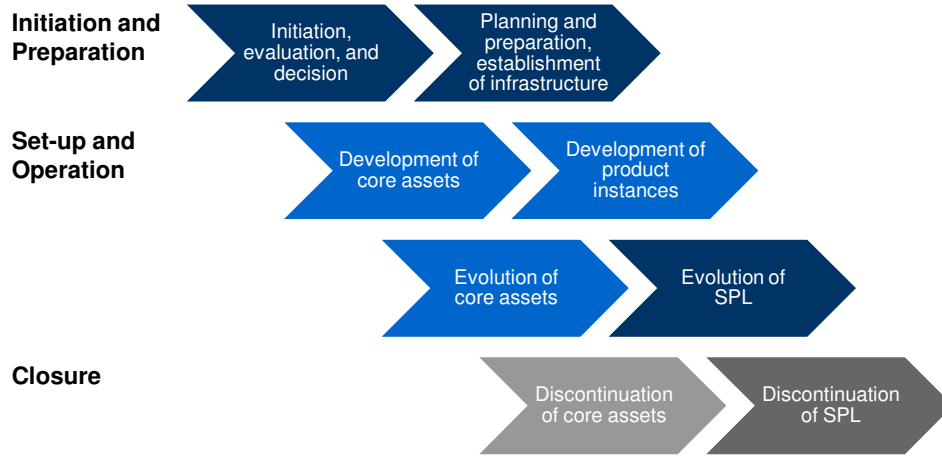
## Technical approach to product instantiation

- |                      |   |  |
|----------------------|---|--|
| <b>Constructive</b>  |  | <ul style="list-style-type: none"> <li>• Customization based on framework, library, or shared components</li> </ul>  |
| <b>Compositional</b> |  | <ul style="list-style-type: none"> <li>• Fixed building blocks with parametrization</li> <li>• Parametrization during build process or during installation; never exposed to end user</li> </ul> |
| <b>Generative</b>    |  | <ul style="list-style-type: none"> <li>• Model-driven development with artefact generation</li> </ul>  |
| <b>Combination</b>   |  | <ul style="list-style-type: none"> <li>• For instance, generation of shared components with subsequent customization</li> </ul>  |

## Organizational approach to SPL development



## Overall SPL life cycle process



Different life cycle phases require different organizational structures and processes.

## Example: Perspectives-based characterization of SPL

<b>Business rationale and SPL structure</b>	Product Suite		Customized Products	
<b>Business decision „buy of build?“</b>	Core Asset Development	Core Asset Acquisition	Combination	
<b>Technical approach to product instantiation</b>	Constructive	Compositional	Generative	Combination
<b>Scope of product line: Assets</b>	Requirements	Architecture	Components	Test Cases
<b>Organizational approach to SPL development</b>	Separate Platform Team	Platform and Product	Mixed	Phased Combination
<b>Organization-wide technical SPL structure</b>	One SPL	Hierarchy of SPL	Interwoven System of SPL	
<b>Overall SPL life cycle process</b>	Instantiation and Preparation	Set-up and Operation	Closure	

**Project A**

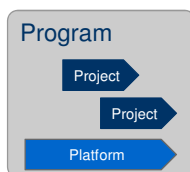
**Project B**

## SPL perspectives help mastering SPL success

<b>Business rationale and SPL structure</b>	Product Suite		Customized Products	
<b>Business decision</b>	Core Asset Development	Core Asset Acquisition	Combination	
<b>Support products and Assets</b>	Requirements	Architecture	Components	Test Cases
<b>Organizational approach to SPL development</b>	Separate Platform Team	Platform and Product	Mixed	Phased Combination
<b>Organization-wide technical SPL structure</b>	One SPL	Hierarchy of SPL	Interwoven System of SPL	
<b>Overall SPL life cycle process</b>	Instantiation and Preparation		Set-up and Operation	Closure

The right SPL perspective helps selecting, characterizing, and applying SPL success factors – i.e., organizational measures, processes, and practices.

## Establish program management for SPL development



A program is a group of related projects managed in a coordinated way to obtain benefits and control not available from managing them individually. [...] (PMBOK Guide, 3<sup>rd</sup> Ed., PMI, 2004)

SPL raises the need to establish program management in a software organization, in order to manage platform and product engineering projects in a coordinated way.

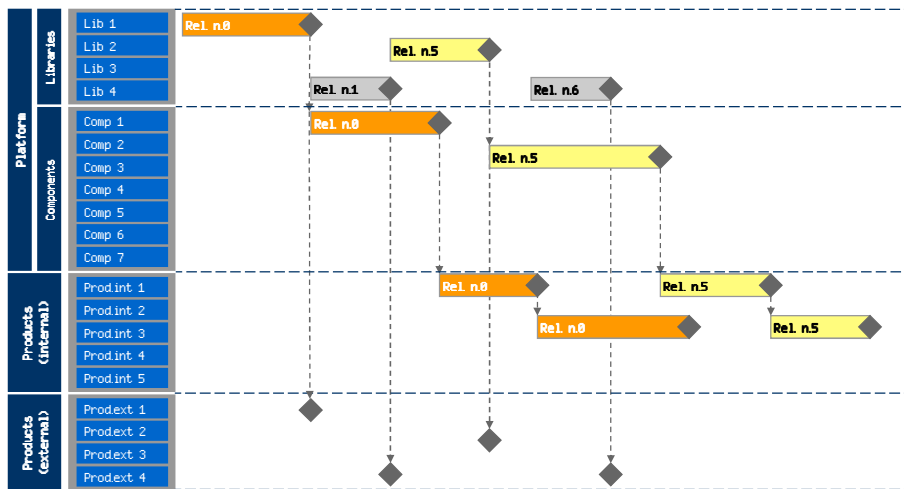
### Key tasks of program management in SPL

- Project scope definition and planning
- Time and release management
- Portfolio planning
- Human resource planning
- ...

### Practices of program management in SPL

- Visual project / program landscape
- Uniform SPL-wide project management practices
- Continuous requirements management
- Continuously maintained and disseminated estimation base
- ...

## Example program schedule visualization for SPL

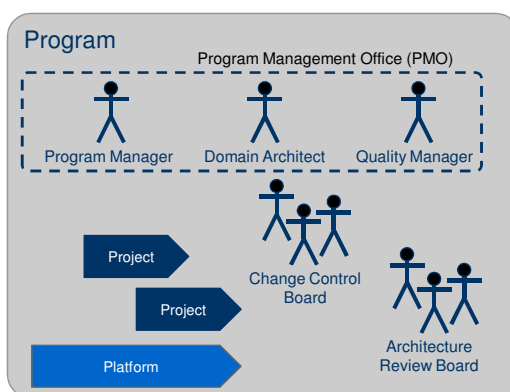


24 September 2007

Copyright © 2007, Software.Process.Management, Dr. Andreas Birk

17

## Organizational entities for effective SPL



### Program Manager

- Coordinate platform, projects, and additional tasks of program

### Domain Architect

- Coordinate requirements processes and artefacts
- Develop and maintain domain model

### Quality Manager

- Coordinate quality management on program level
- Coordinate standards, templates, and central core asset management

### Architecture Review Board

- What architectural changes to be implemented?
- Develop and maintain platform architecture

### Change Control Board

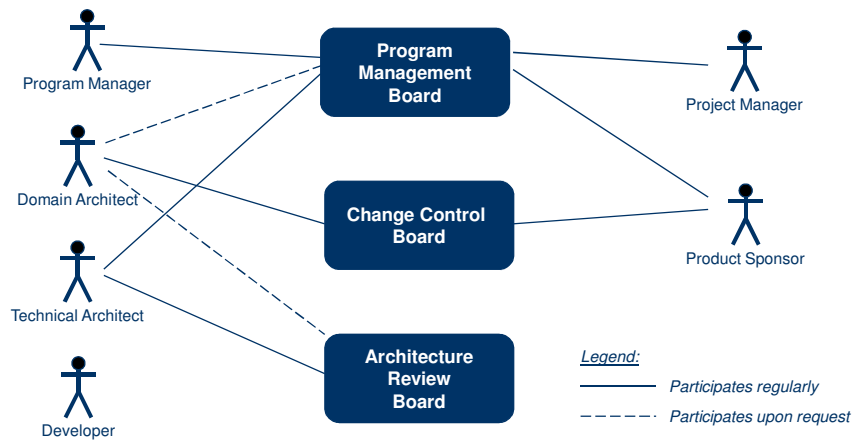
- What to be implemented when?
- Where? – Platform or products?

24 September 2007

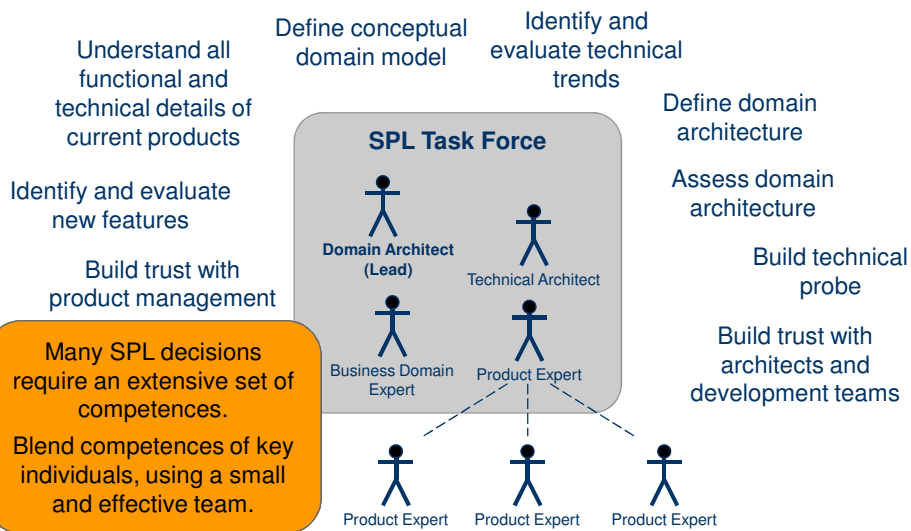
Copyright © 2007, Software.Process.Management, Dr. Andreas Birk

18

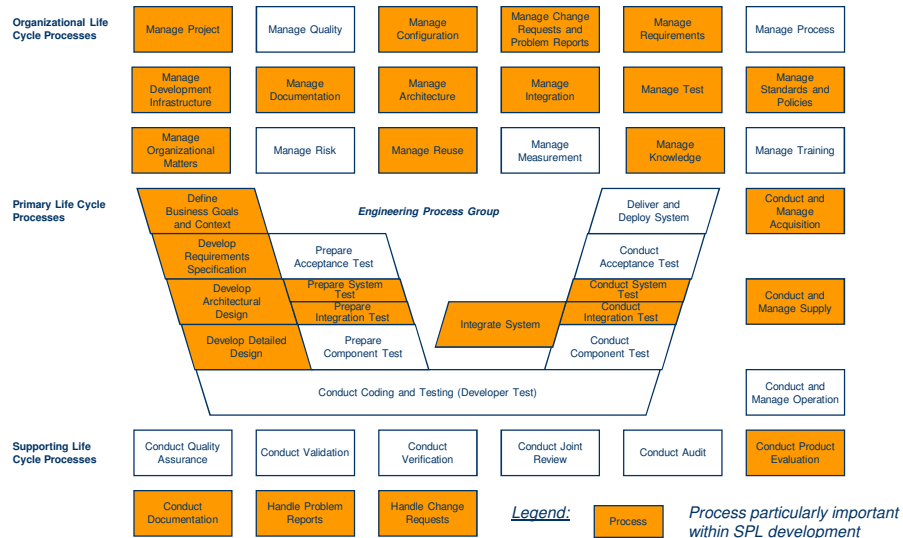
### Example structure of organizational entities



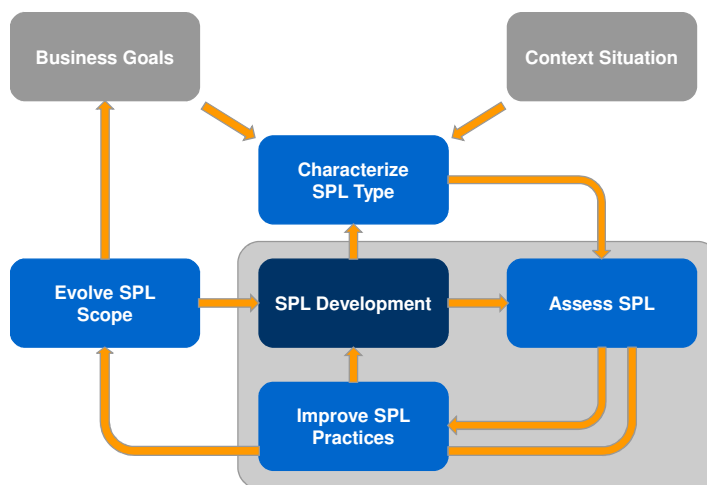
### Blend key competences required for SPL decisions



## Establish effective processes and practices



## Regularly assess and improve SPL development



## Other SPL Success Factors – Last but not least ...

- Approach SPL gradually with incremental steps
- Define overall vision very early, maintain vision
- Establish and keep motivation, commitment, and support
- Establish attractive task profiles and career paths in platform development
- Establish financial valuation of platform contributions to product success
- Prepare for heterogeneous SPL structure with different organizational structures
- Establish effective tool support

## Summary

- Adopting the **appropriate SPL perspective** helps approaching and mastering the challenges of SPL development
- Successful SPL development must combine technological considerations with attention to **business, organization, and process** aspects
  - Establish program management for SPL
  - Establish effective organizational structures
  - Enhance SPL-related processes and practices
  - Regularly assess and improve SPL development
- There is no silver bullet / one-size-fits-all solution: Understanding and mastering the **variants of SPL development** builds the foundation for lasting **success with software variants** of SPL



## Contact Information

**Dr. Andreas Birk**

Software.Process.Management

Gutenbergstraße 99

70197 Stuttgart, Germany

Phone. +49 711 6645 324

Fax. +49 711 6645 325

Mobile. +49 171 9394 908

E-Mail. [andreas.birk@swpm.de](mailto:andreas.birk@swpm.de)

<http://www.swpm.de>