

# Clearing the Way for Software Product Line Success

*Lawrence G. Jones and Linda M. Northrop*

*Software Engineering Institute*

# The authors

- **Lawrence G. Jones** is a senior member of the technical staff and leads the Product Line Practice Initiative at Carnegie Mellon University's Software Engineering Institute. He received his PhD in Computer Science from Vanderbilt University. He's a senior member of the IEEE and ACM, a Fellow of ABET and the Computing Sciences Accreditation Board, and a member of the ABET Board of Directors.
- **Linda M. Northrop** is the director of the Research, Technology, and Systems Solution Program at Carnegie Mellon University's Software Engineering Institute. Her research interests include software architecture, software product lines, and ultra-large-scale systems. She is coauthor of *Software Product Lines: Practices and Patterns* (Addison-Wesley Professional, 2001). She is a member of the IEEE Computer Society and the ACM.

# Introduction

- Over the past fifteen years, we have gained valuable insights into product line triumphs and struggles
- *SEI Framework for Software Product Line Practice*
  - [http://www.sei.cmu.edu/productlines/frame\\_report/index.html](http://www.sei.cmu.edu/productlines/frame_report/index.html)
- SEI Product Line Technical Probe<sup>SM</sup> (PLTP<sup>SM</sup>)
- We have used the PLTP to examine a variety of government and commercial organizations, ranging in size from fewer than 50 people to global corporations.

# SEI Product Line Technical Probe

- The PLTP is a diagnostic method for examining an organization's readiness to adopt, or ability to succeed with, a software product line approach.
- A PLTP compares an organization's software engineering and management practices against the practice areas in the SEI *Framework for Software Product Line Practice* and produces a set of findings that portray organizational strengths and challenges with respect to a product line approach.
- Following a preliminary phase to establish context, review documentation, and focus the data collection, a team spends a week on site conducting structured interviews, consolidating the information and presenting the findings.
- The results, typically documented in a final report, help in planning how to "get from here to there."

# Common Pitfalls

- We have discovered some pitfalls that seem to challenge all would-be product line organizations. In addition, there are downfalls that are frequently associated with certain organization types. Though individual organizations may defy complete categorization, we have found trends that are idiosyncratic to what can be typified as either engineering-based or developer-focused organizations\*.

# Common Pitfalls

- Based on these results, we characterize two strategic pitfalls that we have repeatedly seen:
  - 1) failure to recognize that a software product line approach is a business and technical strategy
  - 2) failure to manage the product line unique aspects of governance and roll-out appropriately

# Common Pitfalls

- We provide a set of diagnostic questions and remedies for problems related to these failures.
- The two pitfalls lead naturally to four high-level diagnostic questions, the answers to which are best discovered by answers to subordinate questions

# Diagnostic Question 1

- Does Your Software Product Line Approach Include a Sound Business Strategy?
  - Do you have a business case with incentives at all levels?
  - Have you involved the key players from the business side of the house?
  - Are your business practices aligned with the product line approach?
  - Are all levels of managers appropriately involved? Do they walk the talk?

# Diagnostic Question 2

- Does Your Software Product Line Approach Have a Sound Technical Strategy?
  - Do you have a dominant architecture focus?
  - Does your technical staff have software engineering skills and do they know how to apply them in a product line?
  - Have you chosen variation mechanisms based on your business goals and business case and have you considered the possible impact of emerging technologies?
  - Do you have a production strategy and a production plan that ensures repeatable and predictable application of the chosen variation mechanisms?
  - Do you have non-code core assets?

# Diagnostic Question 3

- Do You Have Product-Line-Appropriate Governance in Place?
  - Do you have an organizational structure and an operating concept that support the product line?
  - Are you coordinating and managing both the business and technical dimensions of the product line effort?
  - Is there open communication and knowledge transfer throughout your product line organization?

# Diagnostic Question 4

- Do You Have an Effective Product Line Roll-out Strategy?
  - Do you know what *you* need to do to move to a product line approach?
  - Are you effectively orchestrating all of the product line activities?
  - Have you included change management mechanisms in your roll out?
  - Have you determined how to stay in business while effecting the necessary changes?

# Problems related to Specific Organizations

- We also provide additional downfalls that tend to occur in either **engineering-based** or **developer-focused** organizations

# Engineering-based Organizations

- This most often, but not exclusively, describes organizations where the products in the product line have both hardware and software components and those involved in the product line effort have an engineering background.
  - Core assets lack business relevance
  - Lack of organizational support and visibility
  - Technical thrashing among the software engineers
  - Cost overruns for the product line ramp-up

# Developer-based Organizations

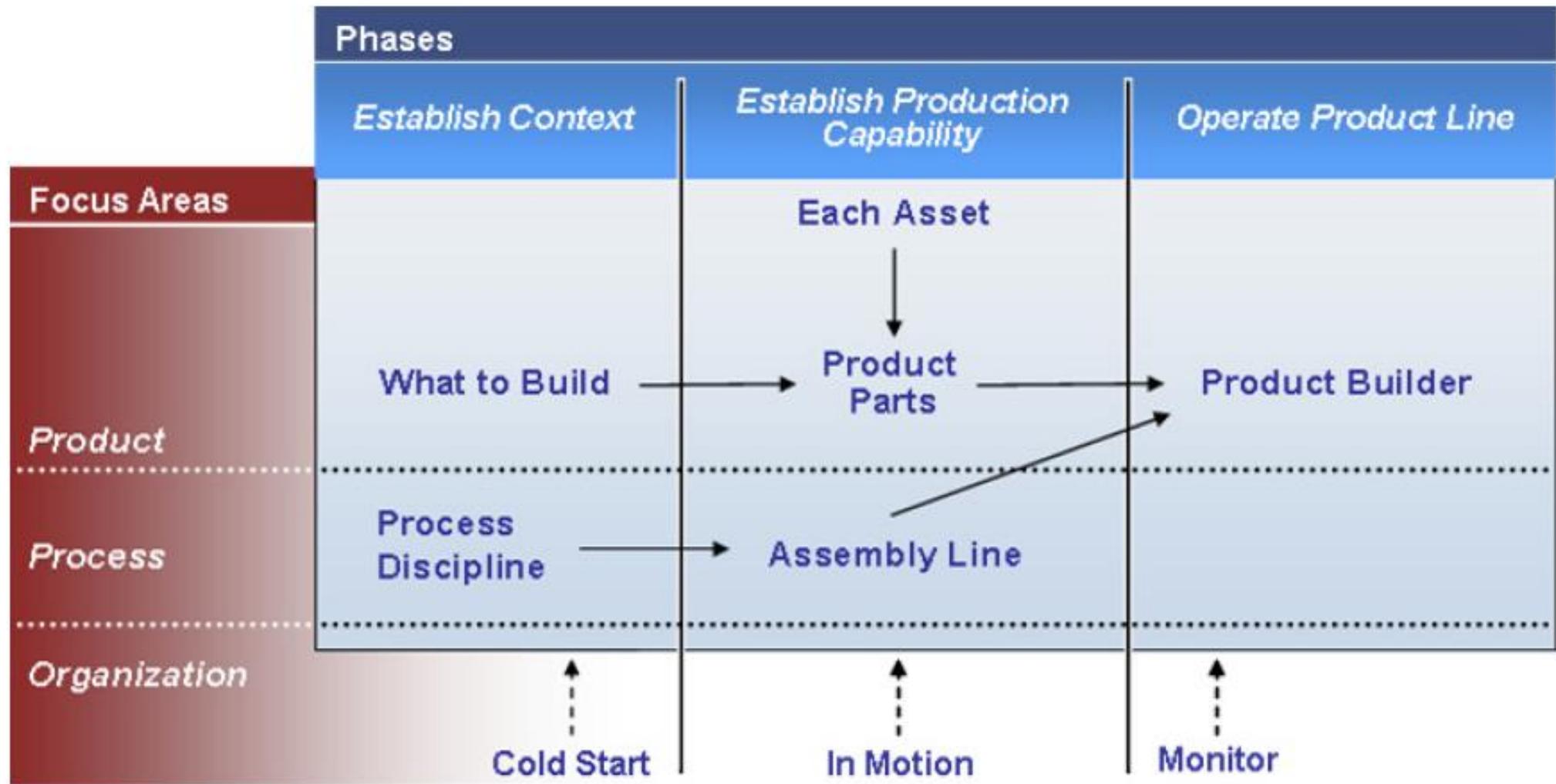
- Organizations that produce software applications and enterprise systems. These organizations likely have not grown out of a hardware or engineering culture but rather an IT one.
  - Lack of strategic technical focus
  - Lack of consistently applied software engineering practices across teams
  - Stagnation of integration and deployment efforts
  - Lack of technical agreement across team boundaries
  - Waffling commitment to the product line endeavor

# Developer-based Organizations

- Usually the product line is being driven by the business leaders and usually the technical work is done by an assortment of fairly autonomous technical teams with a code-level focus. These teams tend to:
  - Vary in software engineering skills and practices
  - Focus on the code and functionality
  - Have different definitions for key concepts such as release, build, tested, architecture
  - Love their autonomy
- More disciplined technical practices and more attention to organizational change management are the remedies to these problems

# The Adoption Factory Pattern

- In the spirit of software design patterns, product line practice patterns show how practice areas from the *Framework for Software Product Line Practice* can be combined and coordinated to accomplish useful outcomes.
- The Adoption Factory pattern shows the complete landscape of product line adoption.
- It maps out the broad coordination among the many practices needed to get the SPL development strategy going.



 *Informs and information flow*  
 *Supports*

	Establish Context	Establish Production Capability	Operate Product Line
Product	<ul style="list-style-type: none"> <li>• Marketing Analysis</li> <li>• Understanding Relevant Domains</li> <li>• Technology Forecasting</li> <li>• Building a Business Case</li> <li>• Scoping</li> </ul>	<ul style="list-style-type: none"> <li>• Requirements Engineering</li> <li>• Architecture Definition</li> <li>• Architecture Evaluation</li> <li>• Mining Existing Assets</li> <li>• Component Development</li> <li>• Using Externally Available Software</li> <li>• Software System Integration</li> <li>• Testing</li> </ul>	<ul style="list-style-type: none"> <li>• Requirements Engineering</li> <li>• Architecture Definition</li> <li>• Architecture Evaluation</li> <li>• Mining Existing Assets</li> <li>• Component Development</li> <li>• Using Externally Available Software</li> <li>• Software System Integration</li> <li>• Testing</li> </ul>
Process	<ul style="list-style-type: none"> <li>• Process Discipline</li> </ul>	<ul style="list-style-type: none"> <li>• Make/Buy/Mine/Commission</li> <li>• Configuration Management</li> <li>• Tool Support</li> <li>• Measurement and Tracking</li> <li>• Technical Planning</li> <li>• Technical Risk Management</li> </ul>	
Organization	<ul style="list-style-type: none"> <li>• Launching and Institutionalizing</li> <li>• Funding</li> <li>• Structuring the Organization</li> <li>• Operations</li> <li>• Organizational Planning</li> <li>• Customer Interface Management</li> <li>• Organizational Risk Management</li> <li>• Developing an Acquisition Strategy</li> <li>• Training</li> </ul>	<ul style="list-style-type: none"> <li>• Launching and Institutionalizing</li> <li>• Funding</li> <li>• Structuring the Organization</li> <li>• Operations</li> <li>• Organizational Planning</li> <li>• Customer Interface Management</li> <li>• Organizational Risk Management</li> <li>• Developing an Acquisition Strategy</li> <li>• Training</li> </ul>	<ul style="list-style-type: none"> <li>• Measurement and Tracking</li> <li>• Technical Risk Management</li> <li>• Organizational Risk Management</li> <li>• Customer Interface Management</li> <li>• Organizational Planning</li> </ul>

# Conclusion

- While all this may sound daunting, take heart. The first step to getting out of a jam is to recognize it. More formally, you should
  - 1. Determine where you are by diagnosing the strength of your product line practices. (Use the *Framework* and the advice in this article.)
  - 2. Determine where you want to be. (You can use the Adoption Factory pattern as a guide.)
  - 3. Plan how to bridge the gap.
  - 4. Execute your plan.
- Expect to iterate and make incremental progress towards your goal
- Product line adoption is an on-going journey, not a discrete step

# Clearing the Way for Software Product Line Success

*Lawrence G. Jones and Linda M. Northrop*

*Software Engineering Institute*

Dúvidas?

Obrigado