Relating Product Line Adoption Mode and Transition Process

PLEES Workshop
23. September, 2003

Fraunhofer

Institut
Experimentelles
Software Engineering

Klaus Schmid

schmid@iese.fhg.de

Overview

1. **Product Line Adoption Situations**
2. **Adoption Strategies**
   - Adoption Patterns
   - Resulting Economic Patterns
3. **Product Line Planning Techniques**
   - **PL Planning and its Relation to Adoption and Evolution**
4. **An Economic Model to Optimize Product Line Adoption**
5. **Summary**

Product Line Adoption Situations

Independent
- No previous systems
- Entering a new market / domain / sub-domain
- Willingness to develop systems from scratch

Project-Integrating
- Systems exist
- System-Development needs to continue

How to introduce product lines?

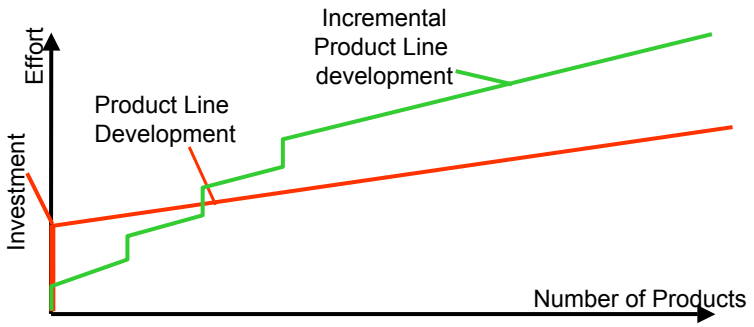
Reengineering-Driven
- Systems exist - but no basis for PL
- Knowledge is insufficient / other means are too costly

Leveraged — characteristics of previous ones
- Product Line in Place
- Entering a new market / domain / sub-domain

Product Line Adoption Patterns

Big Bang
- You plan it — You do it.

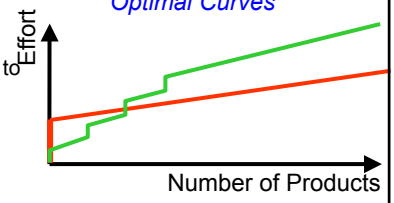
Incremental
- You build it on your way
- *Dimensions of Incrementality:*
  - products
  - functional areas (sub-domains)



Product Line Adoption Economics

Big Bang ⇔ Incremental

Optimal Curves



- Jumps in incremental
  - correspond to investments in migrating to product line assets
  - total sum higher than in big bang
- Lines
  - correspond to products built with partial infrastructure
- Angle
  - reduced in each investment
  - final angle still steeper
  - go for best ROI first (if risk controllable) — best reduction of angle
- Endpoint
  - higher for incremental

Why to go for incremental anyway?

- Risk control!!

How Do I Plan For Product Line Adoption?

Adoption Situation	PL Planning Look-Ahead	Approach
Independent	Broad Portfolio of Future Systems	Big Bang
Project-Integrating	Medium-Size Portfolio of Future Systems	Incremental (by functional area or component) *may be combined with* Incremental (by product)
Reengineering-Driven	Broad Portfolio of Future Systems and Legacy Products	Incremental (by functional area or component) *or* Big Bang (packaging legacy as a whole)
Leveraged	Broad Portfolio of Future Systems	Big Bang

Product Line Evolution Situations

Relation to Evolution?

- How much deviation do I allow?
- Deviation ↔ Re-adoption

How to evolve product lines?

Types of Deviation

Infrastructure-Based

- No deviation
- Everything goes through reuse infrastructure

Ideal

Branch-and-Unite

- Systems are split of the basis and re-integrated after being built

Few systems, implications unclear

Bulk Integration

- Product Line in Place
- Large diversions occur

Don't Do It!

How Do I Plan For Product Line Evolution?

Evolution Approach	PL Planning Look-Ahead	Approach
Infrastructure-based	A small number of products	Incremental by product
Branch-and-unite	Single product	Incremental by product
Bulk Integration	Small number of products (perhaps a market segment)	Incremental by product group

Product Line Planning Techniques

Product Portfolio Planning

- Define what the products are
- Interface with product management / Market concerns
- Typically workshops

Domain Potential Analysis

- Identify benefits and risks related them to functional sub-domains
- Assessment Approaches (e.g., PuLSE-B&R)

Reuse Infrastructure Scoping

- Identify parts that should be packaged as reusable assets ($\Rightarrow$ architecture impact)
- Rather fine-grained analysis
- Closest to implementation

PL Planning and Its Relation to Adoption

Mode of PL Extension	Portfolio Definition	Domain Potential Analysis	Reuse Infra-structure Scoping
Partial Big Bang / Evolution by Product Group	Very important	Recommended but mainly for risk analysis	Recommended to support architecture definition
By (single) product	Not necessary	Only needed if extension requires restructuring	Only needed if the extension requires restructuring
By component or functional area	Should be performed	Key for identifying the next component for product line extension	Should be applied to support architecture definition

Value-Based Adoption and the Architecture

Impact of product characteristics on the architecture:

- How often will we need a certain variation?
- How certain is it, we will need it?
- What are the costs of variation mechanisms?

Assumptions:

- Adding variabilities costs effort
  (variability mechanism + effort capability)
- More generic code is more complex, thus more costly to maintain
- Late implementation is more costly than if it was planned right away
- The less „places“ a change impacts, the less costly
- Architecting for a functionality reduces the number of impacted positions
- *Discounted Cash-Flow* Analysis

Value-Based Adoption and the Architecture

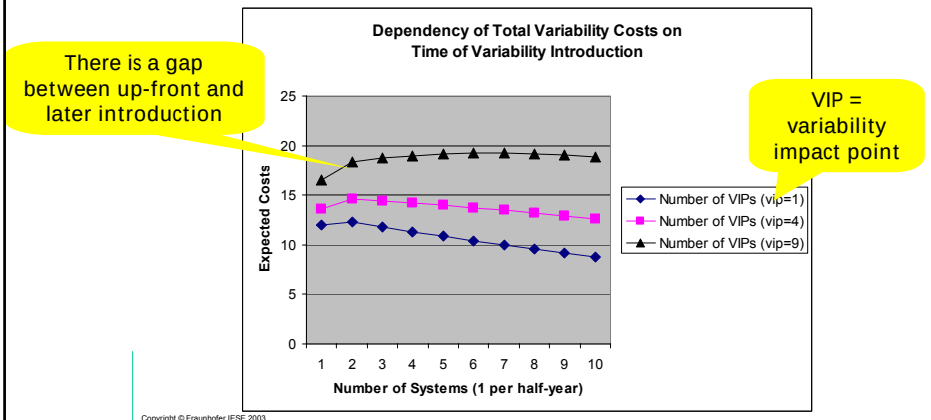
Example

- Some functionality (in our case distribution support) *can* be required:
  - We are not sure
  - The support is costly to build
  - It can not always be present
- What to do?
- The numbers are taken from the example, but the basic characteristics of the functions relate only to their structure

Value-Based Adoption and the Architecture

Is up-front introduction of variability better?

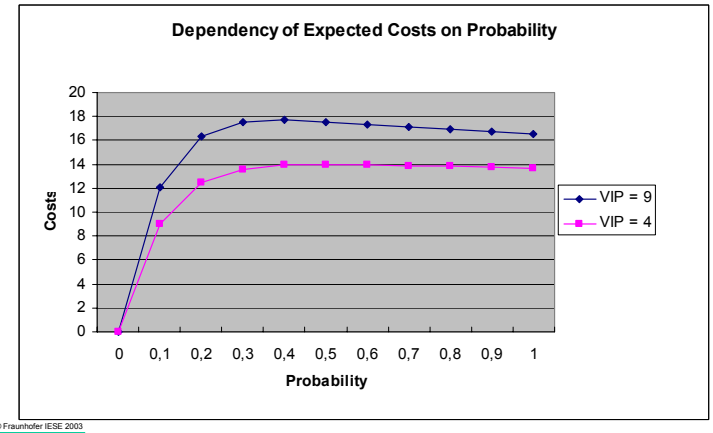
– The number of impacted positions (i.e., the architecture) is key to answer this question!



Value-Based Adoption and the Architecture

The effect of probability

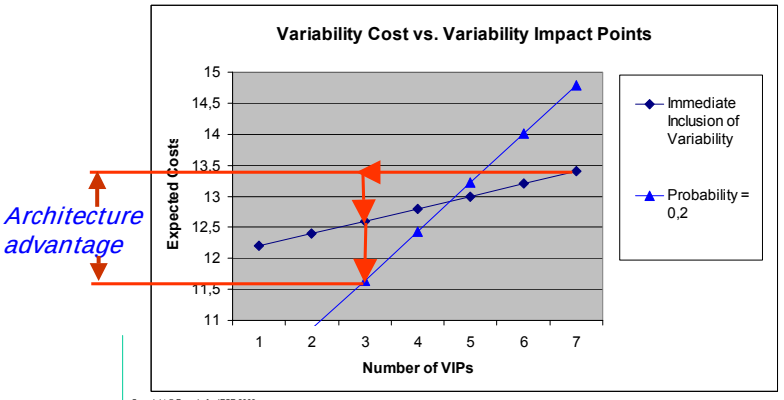
– Only for very low probabilities the total costs are reduced



Value-Based Adoption and the Architecture

Changing the architecture changes the approach

- Up-front architecting might be appropriate even if up-front implementation is not!!



Conclusion and Further Work

Conclusions

- Categorizations of product line adoption situations were given and different approaches for dealing with them were discussed
- Detailed recommendations for product line planning were given
  - Planning Look-Ahead
  - Relative Importance of Scoping Techniques
- A quantitative model was proposed that allows to derive more detailed guidelines
  - Amount of effort required for variability
  - Probability that it is needed
  - Characteristics of underlying development process (e.g., change effort)