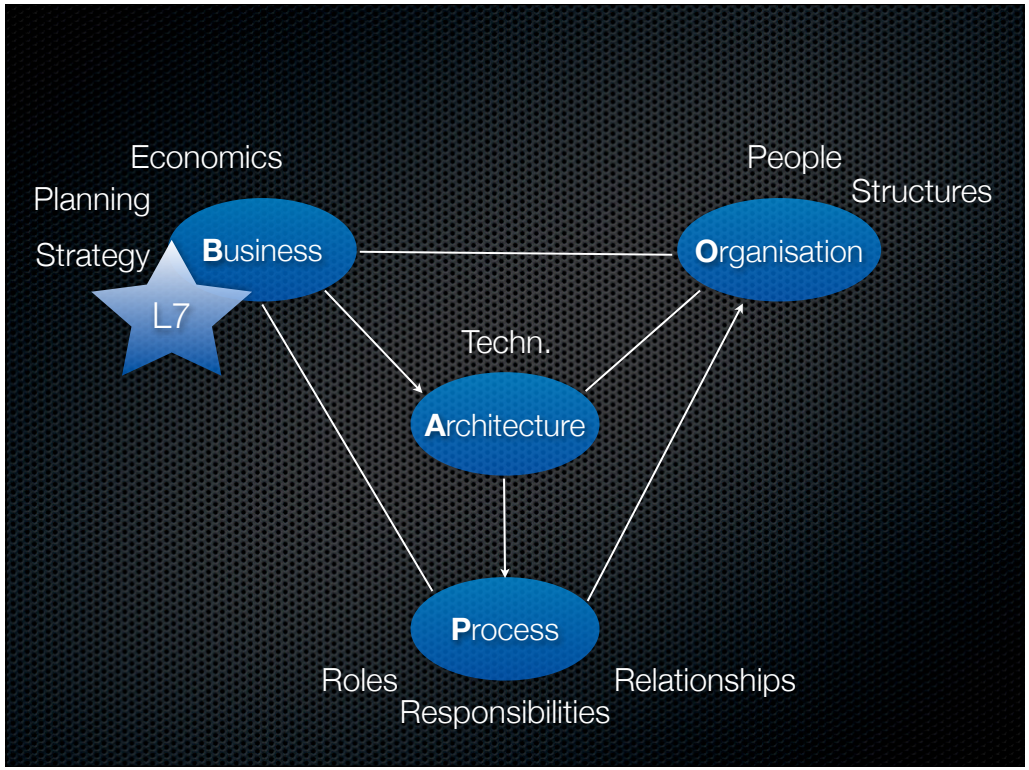
Software Product Line Engineering

L7: Business and SPLE

Tony Gorschek - tony.gorschek@gmail.com


Economics
Planning
Strategy
Business
L7
People
Structures
Organisation
Techn.
Architecture
Process
Roles
Responsibilities
Relationships

Business

- Motivation and Economics of SPLE
  - introduction/transition to PL
  - why should we...
  - experience reports
  - SPLE economics
- SPLE Planning
  - product line markets
  - strategies and roadmaps
  - what should we do (e.g. what requirements to select)
  - product portfolio planning

Markets and strategy

- Product definition strategy
- Market strategy
- Product line life-cycle

Product definition strategy

- How new products are defined
  - **customer-driven**
    - demands/requirements from present of future customers drives the development of products
    - final product individualized to the specific needs of the customer (mass-customization, large amounts of req.)

hard to identify all req. upfront

= need a very flexible PL platform

Product definition strategy

- **producer-driven**
  - developing organization defines the product
  - mass-market development where each product variant is sold to large groups/segments
  - **market-oriented (market-pull)**
    - products in portfolio based on market-analysis of potential market segments
    - new products defined to satisfy newly formed segments or changes in market segments
  - **technology-oriented (technology-push)**
    - evaluate the technological capabilities that is developed by the development org. and bring this to the market

lower risk
short term
-
higher risk
long term

big risk
-
big reward?

Product definition strategy

- **producer-driven vs. customer-driven**
  - in reality often mixture (need and opportunity)
  - e.g. a platform developer that lets other companies do the last customization might become technology-driven (tech push) to introduce a new piece of technology, and in this situation opt to support some end-customers to make sure the new technology takes off...

short-term losses => long term (larger) gains

often companies sway as opportunities arise...

Market strategies

- **Market strategies**
  - how an organizations wants to be known by the market
  - Cost leadership — Lowest prices
  - Differentiation — product sets itself apart through specific features or attributes
    - Improving — improve aspects irt competitors (from cust. perspective) e.g. quality, price, etc
    - Newer — e.g. innovation
    - Faster — time-to-customer
  - Focusing — company focuses on a specific niche

Product line life-cycle

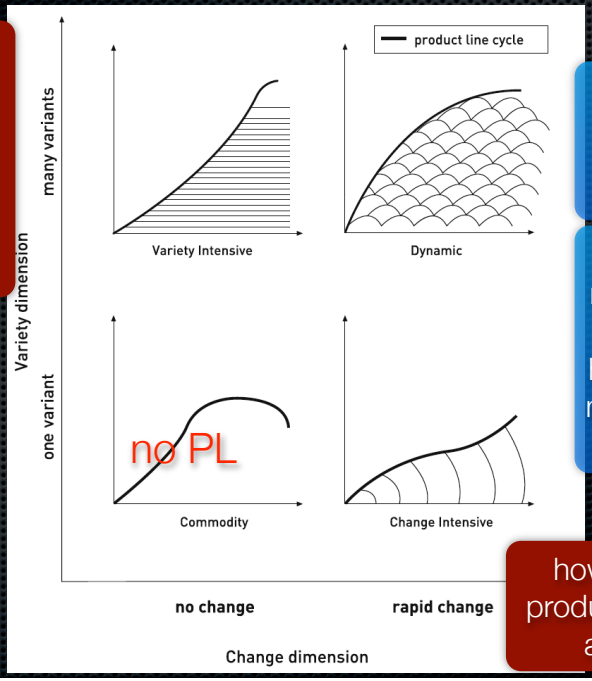
- Individual products
  - Introduction — launch, low sales as product unknown
  - Growth — the product (and its competitive features/qualities) become known to market, sales increases
  - Maturity — sales increase diminishes, prices (cost!) has to be reduced to win market share
  - Saturation — maximum sakes is achieved (often in environment of hard competition)
  - Degeneration — substitution, diminishing profits

Product line life-cycle

- Combination of products (time and variation)
  - Multiple products fighting for market-share (also true for features within products) can cannibalize on each other
  - Different products complement each other by supplanting each other over time
- The mixture of these two perspectives can determine what the overall PL dynamics looks like...

Product line life-cycle

how many different products available on market at same time



in a young market variety demands are lower

in mature markets variety demands are higher, but new market entrants might be fewer

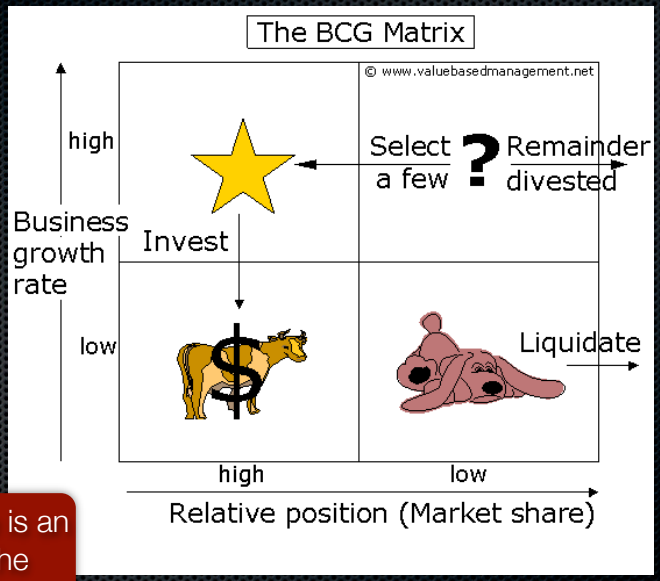
how fast a new products supplants an old one

Product line product portfolio

even spread

cows = cashflow
stars = future
? = potentials

make sure there is an evolution of the spread



Strategy and PLE

- PLE can support a product line marketing strategy
  - reduced time-to-market, reduced cost and effort to develop a product (that fits in the PL)
  - Well suited for *cost leadership strategy* (>3 products)
  - The production of an additional product is faster and lower => enable *differentiation* by offering more products on the market (despite potential cannibalization)
    - *Faster*
    - *Focusing* (not that expensive to focus on specialized product)

PLE and market strategy MUST FIT!
Product management is the executive arm (e.g. scoping)

Examples

- How can strategies be used to select what to do (and how can strategies be formulated and changed based on input from e.g. requirements)

MERTS

- **Specify goals and objectives**
  - Where do we want to go?
    - direction of movement (e.g. towards profit, growth, market share)
  - How to get there?
    - targeting customer segments and assigning them priority (e.g. asian market prioritized over european)
    - targeting specific competitors and giving them priority (by targeting a specific competitor their strategy and offering influences yours)
    - differential advantage (in relation to technology, pricing, strategic alliances and non-functional aspects and giving them priorities which are then weighed against incoming requirements)
  - What to do?
    - addresses specific programs/tactics to be used to achieve goals and objectives established in the light of “how to get there”. This deals with the product, pricing, promotion, distribution, and service. Selection of strategic drivers (technology-push or market-pull or both)

MERTS example

Questions	Factors	Weightings of Factors	Sub classification	Sub classifications Weightings	Normalized Weightings	Requirement (Japanese language support)	Normalized
Where	Market growth	60			60	70	42
	Market share	20			20	70	14
	Profit	20			20	50	10
		100			***100***		
How	Customer segments	40	*USA market*	20	8	0	0
			European market	30	12	0	0
			Asian market	50	20	100	20
				100			
	Competitors	30	*ABC company*	70	21	100	21
			HIJ company	20	6	0	0
			Others	10	3	0	0
				100			
	Differential advantage	30	*Innovation*	60	18	70	12.6
			Pricing	5	1.5	0	0
			Strategic Alliances	10	3	0	0
			Non functional requirements	25	7.5	0	0
		100		***100***	***100***		
What	New Technology	40			40	80	32
	Use of core assets	11			11	0	0
	Architecture stability	20			20	40	8
	Market pull	10			10	80	8
	Technology push	15			15	0	0
	Customization flexibility	2			2	0	0
	COTS	2			2	40	0.8
		100			***100***	**Total**	**168.4**

growt!

asia!

ABC

Innovation

new tech

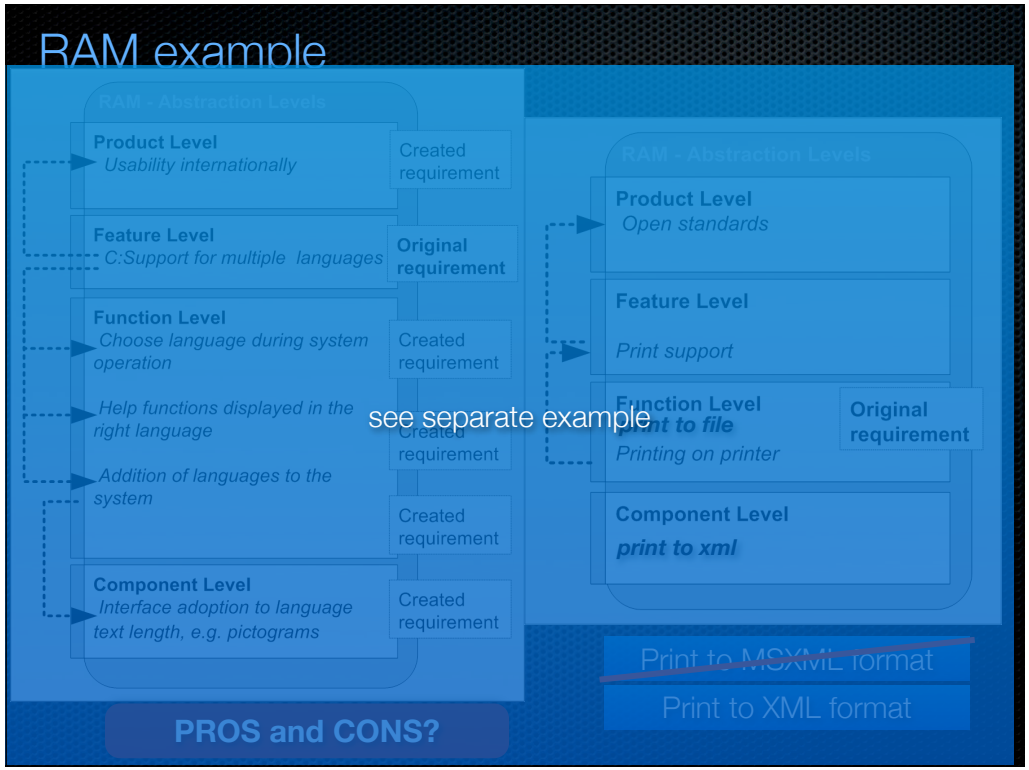
PROS and CONS?

requirement got 168 out of possible 300

RAM

- Utilize abstraction levels to trace from strategic goals to implementational details
- Any requirement coming in has to be worked-up to product level -> compared to the strategies
  => YES / NO, if YES -> requirement is broken down, if NO -> dismiss (fast triage)

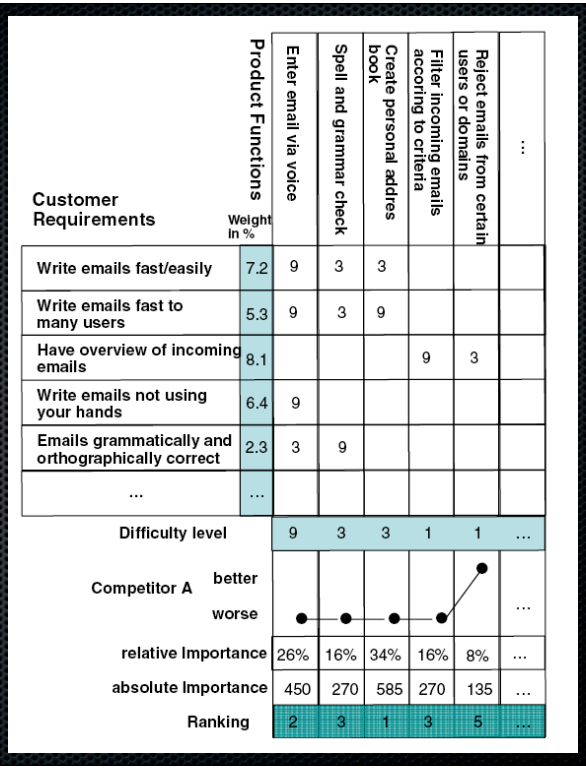
RAM example


RAM - Abstraction Levels
Product Level
Usability internationally
Created requirement
Feature Level
C:Support for multiple languages
Original requirement
Function Level
Choose language during system operation
Created requirement
Help functions displayed in the right language
Created requirement
Addition of languages to the system
Created requirement
Component Level
Interface adoption to language text length, e.g. pictograms
Created requirement
PROS and CONS?
see separate example
RAM - Abstraction Levels
Product Level
Open standards
Feature Level
Print support
Function Level
Printing on printer
Original requirement
Component Level
print to xml
Print to MSXML format
Print to XML format

QFD for Planning (selection)

- Quality Function Deployment (and HoQ) can be used to bring requirements and business togehter
  - e.g. compare features offered with requirements
  - e.g. satisfaction analysis, competitor analysis etc
- Pre-requisites
  - have priority (from e.g. customers)
  - have overview of customers and their relative importance
  -etc

QFD example



Customer Requirements	Weight in %	Enter email via voice	Spell and grammar check	Create personal addres book	Filter incoming emails accoring to criteria	Reject emails from certain users or domains	…
		Product Functions					
Write emails fast/easily	7.2	9	3	3			
Write emails fast to many users	5.3	9	3	9			
Have overview of incoming emails	8.1				9	3	
Write emails not using your hands	6.4	9					
Emails grammatically and orthographically correct	2.3	3	9				
…	…						
Difficulty level		9	3	3	1	1	…
Competitor A	better						
	worse						…
relative Importance		26%	16%	34%	16%	8%	…
absolute Importance		450	270	585	270	135	…
Ranking		2	3	1	3	5	…

QFD example

Customer Requirements	Product Line Member/ Customer Segment #1	Product Line Member/ Customer Segment #2	Product Line Member/ Customer Segment #3	...
Write emails fast/easily	7.2	5.3	6.8	
Write emails fast to many users	5.3	6.3	7.5	
Have overview of incoming emails	8.1	11.2	8.1	
Write emails not using your hands	6.4			
Emails grammatically and orthographically correct	2.3	7.4	5.3	
...				

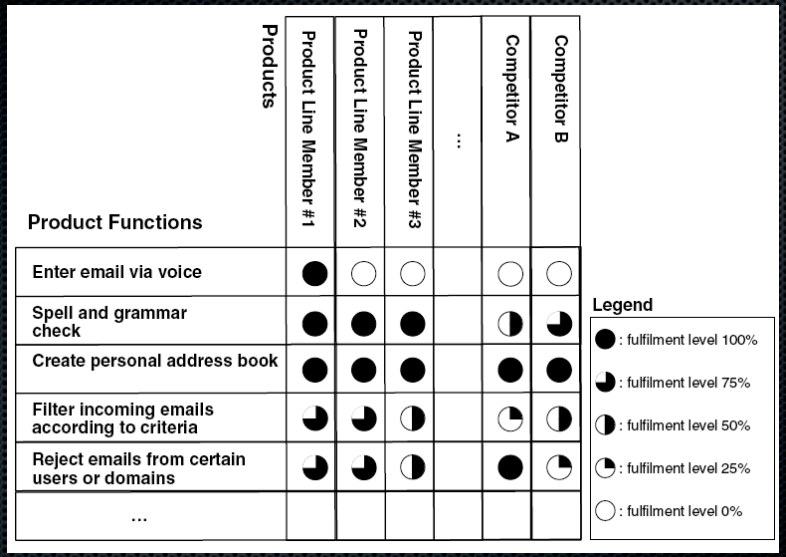
Customer Segments /Products (in %)

Legend

: 10% most important customer requirements

: 25% most important customer requirements

QFD example


Products
Product Functions
Product Line Member #1
Product Line Member #2
Product Line Member #3
...
Competitor A
Competitor B
Enter email via voice
Spell and grammar check
Create personal address book
Filter incoming emails according to criteria
Reject emails from certain users or domains
...
Legend
: fulfilment level 100%
: fulfilment level 75%
: fulfilment level 50%
: fulfilment level 25%
: fulfilment level 0%

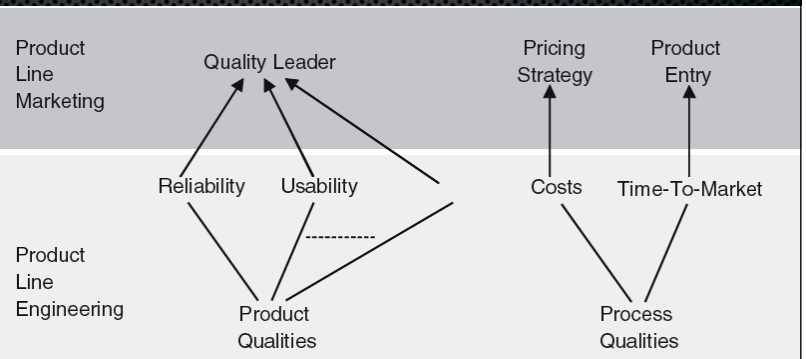
others...

- How do you measure your products in relation to competitors
  - GAP analysis, relative CVA analysis
- How do you measure customer value
  - Customer Value Analysis
- How do you check that your company (and product development) is following your strategies
  - Internal Value Analysis

Product line economics

- **PLE influences the properties of the products being developed**
  - product qualities
    - properties of product execution (e.g. security, reliability, usability)
  - process qualities
    - properties of the process (e.g. innovation, cost, time-to-market)

the characteristics of the qualities influence marketing (strategic) decisions



Product line economics

- **Development costs**
  - large parts of the functionality for the development of new products is provided by the platform reducing size and complexity (reduce costs to same extent)
- **Development time**
  - same as above
- **Reliability**
  - reuse of components that are well proven...
- **Usability**
  - consistency among UI
- **Portability**
  - a type of variability (= easy IF product line is prepared for this)
- **Maintenance**
  - maintain platform to a large extent (and not every individual product)

asset control costs?

prediction accuracy?

what about innovation?

what about customer value

what about WOW

Andreas Helferich, Georg Herzwurm, and Sixten Schockert, “QFD-PPP: Product Line Portfolio Planning Using Quality Function Deployment”, SPLC 2005, LNCS 3714, pp. 162 – 173, 2005.

QFD, HoQ

Khurum M., Khurum A., and Gorschek T., “A Model for Early Requirements Triage and Selection (MERTS) Utilizing Product Line Strategies“, 11th International Software Product Line Conference, Kyoto, 2007, pp. 97-104.

MERTS

Gorschek T. and Wohlin C., "Requirements Abstraction Model", Requirements Engineering journal, vol. 11, 2006, pp. 79-101.

RAM

Günter Böckle, Paul Clements, John D. McGregor, Dirk Muthig,and Klaus Schmid, “A Cost Model for Software Product Lines”, PFE 2003, LNCS 3014, pp. 310–316, 2004.

Examples of how to calculate cost

Scenario #	Description
1	An organization has a set of products in the marketplace that were developed more or less independently. It wishes to explore the possibility of converting them to a software product line, built from a common set of core assets.
2	An organization has a set of products that it plans to bring to the marketplace, but which are not yet under development. It wishes to explore the possibility of building them as a software product line on top of a common set of core assets.
3	An organization has a set of products that it plans to bring to the market, and is planning to build them as a software product line. It wishes to explore building a core asset base to support a larger set of the products than the ones currently planned. This scenario is similar to #2 except that here the organization intends to perform the product line analysis without knowing all of the products that will be in the product line.
4	An organization has two or more software product lines that appear to have some overlap with each other. It wishes to know if it would be economical to merge them – that is, to merge the core assets bases and to build the affected products from the new combined platform.
5	An organization is planning to bring a new product to market, and wishes to know what the cost implications are of developing this product under the auspices of an already-existing product line, or building it in a stand-alone fashion.
6	An organization wishes to start a new software product line on the basis of already existing ones. It wishes to know the best strategy from a cost perspective: Should it build a new asset base? Should it reuse one of the asset bases and extend it as needed? Should it "scavenge" one or more of the asset bases to produce the new one? And should it keep or abandon the old asset bases once the new product line is deployed?
7	An organization currently has at least one product line. Due to changing marketing conditions, the organization is considering dropping a product that was previously planned as part of the product line but has not been built yet. The manager would like to know the cost benefit of dropping the product.

Scenario #	Formula
1	$C_{cab} + \sum_{i=1}^{s1} C_{unique}(p_i) + \sum_{i=1}^{s1} C_{reuse}(p_i)$
2	$C_{cab} + \sum_{i=1}^{k} C_{unique}(p_i) + \sum_{i=1}^{k} C_{reuse}(p_i)$
3	$C_{cab} + \sum_{i=1}^{k} C_{unique}(p_i) + \sum_{i=1}^{k} C_{reuse}(p_i)$
4	$C_{cab} + \sum_{i=1}^{n_1+n_2} C_{unique}(p_i) + \sum_{i=1}^{n_1+n_2} C_{reuse}(p_i)$
5	$C_{cab} + C_{unique}(p) + C_{reuse}(p)$
6	$C_{cab} + \sum_{i=1}^{n_i} C_{unique}(p_i) + \sum_{i=1}^{n_i} C_{reuse}(p_i)$
7	$(C_{cab} + \sum_{i=1}^{sum} C_{unique}(p_i) + \sum_{i=1}^{sum} C_{reuse}(p_i)) - (C_{cab} + \sum_{i=1}^{sum-1} C_{unique}(p_i) + \sum_{i=1}^{sum-1} C_{reuse}(p_i))$ $\because sum = \sum_{i=0}^{s1} n_i$

$C_{unique()}$ is a function that, given the relevant parameters, returns the development cost to develop unique software that itself is not based on a product line platform. The result might be a complete product, or it might be the unique part of a product whose remainder is built atop a product line core asset base.

$C_{cab()}$ is a function that, given the relevant parameters, returns the development cost to develop a core asset base suited to satisfy a particular scope. Ccab differs from Cunique in that it must take into account the cost of performing a commonality/variability analysis, the cost of designing and then evaluating a generic (as opposed to one-off) software architecture, and the cost of developing the software so designed. Ccab may be invoked to tell us the cost of developing a core asset base where none currently exists, or it may be invoked to tell us the cost of deriving a desired core asset base from one or more already in place.

$C_{reuse()}$ is a function that, given the relevant parameters, returns the development cost to reuse core assets in a core asset base. Creuse includes the cost of locating and checking out a core asset, tailoring it for use in the intended application, and performing the extra integration tests associated with reusing core assets.