Valuing Flexibility in Software Product Line Architectures

Rami Bahsoon
School of Computer Science,
The University Of Birmingham,
Edgbaston, B15 2TT, Birmingham, UK
r.bahsoon@cs.bham.ac.uk

Wolfgang Emmerich
London Software Systems, Dept. of Computer Science,
University College London,
Gower Street, WC1E 6BT, London, UK
w.emmerich @cs.ucl.ac.uk

Abstract

We highlight some of the challenges in valuing flexibility and investment decisions in Software Product Line Architectures (SPLA). We explain the limitations of traditional approaches, which fail to address uncertainty, while valuing flexibility in SPLA. We motivate the use of Real Options to address these limitations. We suggest a multi-perspectives valuation approach for valuing flexibility in SPLA and report on its formulation using some examples.

Keywords. *Economics-Driven Software Engineering, Software Product Line Architectures, Real Options Theory.*

1. Introduction

Software Product-Line architectures are systematic approaches for managing the change and guiding the evolution of a software system. This is achieved through anticipating the major evolutionary milestones in the development of the product, capturing the properties that remain constant through the evolution, and documenting the variability points from which different family members may be created. The approach gives a structure to the product's evolution and possibly rules out some unplanned evolutions, if the architecture is respected [Jazayeri, 2000]. In practice, product-line analysis often benefits from technology roadmapping to anticipate future requirements, and likely future product variations (which may include combinations of features not supported in current products). In the new communication industries, companies use roadmapping to plan and envision possible paths for "perfecting" their services and offering, as the rapid advances in the technology or the infrastructure enabling these enhancements materialize. This is necessary for catching up with the market, generating wealth, and improving the value of what is offered to the end users. Moreover, industrial practice shows that companies are investing part of their resources in envisioning the future of the stakeholders' requirements and the environment, the evolution of technology and its supporting infrastructure. This is apparent through the related investments in research and development, the increasing number of personnel recruited in technology roadmapping, and aligning the company's future performance with its ability to execute the set roadmap.

In managing the core and the variability points, architects often face the challenges of translating the roadmaps into architectures, which are stable yet flexible enough to accommodate changes in requirements across various product families. Means for achieving architectural flexibility are typical architectural mechanisms or strategies built-in or adapted into the architecture with the objective of facilitating evolution and future growth, in response to new features for given products or in response to changes in functional (e.g., changes in functionality) or non-functional requirements (e.g., changes in scalability demands) supported by the products derived from the core. Unfortunately, built-in or adapted flexibility comes with a price. Questions of interest, however, are how worthwhile is it "buying" flexibility to support new features and variability points? How can we select an architecture, which maximizes the yield of such flexibility relative to the core and the new features? How we can systematically manage investments around the core in supporting variability points and new features?

In this paper, we highlight some of the challenges in valuing flexibility and investment decisions in Software Product Line Architectures (SPLA). We explain the limitations of traditional approaches, which fail to address uncertainty, while valuing flexibility in SPLA. We motivate the use of Real Options, as reported in [Bahsoon 2005], to value the flexibility in Software Product Lines. We suggest a multi-perspectives valuation approach for valuing investments in SPLA and report on its formulation using some examples.

2. Valuing Flexibility in Software Product Lines: A Real Options Perspective

Classical financial valuation techniques, such as Discounted Cash Flow (DCF) analysis and Net Present Value (NPV), fall short in dealing with flexibility and uncertainty [Trigeorgis 1995]. The main problem with these techniques is that they are best valid when valuing an ongoing business or an immediate investment. However, in the case of valuing the flexibility and investment decisions in SPLA in the face of evolutionary changes and variability points, the nature of the investment is long-term and strategic. For example, assume that an investment in an architecture appears to be unattractive, as it would have a negative NPV in the first instance: unless the enterprise makes the initial investment, subsequent generations out of the core architecture or other applications will not even be feasible. The value of the investment, thus, may derive not only from the direct measurable cash flows of the investment, but also from the ability of an architecture to unlock future *growth opportunities* (e.g. case of reuse, exploring new markets, expanding the range of services while leaving the architecture intact).

Real options theory is well suited to address many Software Engineering problems from a value-based engineering perspective. To value the flexibility of software architectures using an economic approach, we need a valuation technique that is suitable for strategic and long-term valuation, accounts for flexibility, and makes the value of the options created by flexibility tangible. Real options satisfy these requirements. First, real options theory provides an analysis paradigm that emphasizes the value-generating power of flexibility under uncertainty [Erdogmus et al., 2002]. In traditional applications, real options analysis recognizes that the value of the capital investment lies not only in the amount of direct revenues that the investment is expected to generate, but also in the future opportunities flexibility creates. The flexibility may take the form of abandonment or exit, delay, exploration, learning, and growth options. In an evolutionary context, the change is uncertain as the demand on the future changes in requirements is uncertain. Thus, the value-generation of the architectural flexibility in accommodating the change is a powerful heuristic for analyzing investment decisions. In particular, flexibility is a strategic architectural quality that adds to the architecture values in the form of *growth options.* A growth option is a real option to expand with strategic importance [Myers 1987]. Growth options are common in all infrastructure-based or strategic industries with multiple-product generations or applications [Trigeorgis 1995]. Obviously, investments in software architectures are infrastructure-type of investments. As many early investments can be prerequisites or links in chain of interrelated projects [Myers 1987]. *growth op*tions set the path for the future opportunities. This is the case of Software Product Lines Architectures as the core is planned to set the path to accommodate variability features. The architecture may provide both the system and the enterprise the potentials for growth. In the architectural context, growth opportunities are linked to the flexibility of the architecture to respond to future changes that may target new products, requirements, markets, etc. Flexibility has a value under uncertainty [Ross et al., 1996]. Since the future changes are generally unanticipated, the value of the growth options lies in the enhanced *flexibility* of the architecture to cope with uncertainty; otherwise, the change may be too expensive to pursue and/or opportunities may be lost. We have extensively studied Growth Options in relation to software architectures [Bahsoon and Emmerich 2005, Bahsoon 2005, Bahsoon and Emmerich 2004]. Second, the selection of cost-effective SPLA requires finding an architecture that maximizes the yield in the added value, relative to some new future features, changes in market requirements, etc. As we are assuming that the added value is attributed to flexibility, the problem becomes maximizing the yield in the embedded or adapted flexibility in the core architecture relative to these anticipated or unanticipated changes. A Real options approach is a value-maximizing paradigm and suited to address this problem as it involves valuation under uncertainty.

3. Multiple Perspectives Valuation in Software Product Lines

The problem of valuing the flexibility of an architecture in accommodating new features and variability requirements necessitate a comprehensive solution that is flexible enough to incorporate multiple valuation techniques; some with subjective estimates and others based on market data, when available. This is because of the following reasons:

First, the valuation activity is a *human-centered activity.* The participants in the valuation activity may include developers, architects, project mangers, market analysts, product analysts etc. Interviews, meetings, or surveys could be conducted to gather qualitative and quantitative costs and benefits information. The participants often rely on experience and subjective judgments in valuation. Describing the valuation as human-centered activity implies subjectivity and introduces different perspectives to the valuation problem.

Second, upon building on the core architecture to accommodate new features and variability requirements, the change may impact one or more architectural qualities, such as performance, maintainability, availability and so forth. For example, new scalability requirements, which could be related to a particular product, derived from the core, may affect both behavioral and structural qualities of the architecture. Linking the impact of such new requirements to value, as a way for valuing flexibility in responding to the change, requires a valuation solution that is comprehensive enough to account for the economic ramifications of the change and its global impact on the architecture. The aim is to provide the architect/analyst with a comprehensive tool for understanding the extent to which the change can “ripple” to impact other qualities and its economic implications.

Third, the valuation is often relative to the evaluation objectives and the primary drivers motivating the change and/or introducing new features in realization of the variability points. The drivers could be, for example, future cost savings, shorter time-to-market, entry to new markets, service enhancements, etc. It is often the case that there is more than one driver behind the change. This necessitates a valuation solution that is flexible enough to capture the value relative to the said drivers.

Fourth, we have advocated the use of real options valuation in software product lines. Real options valuation uses twin asset to the valuation of the asset in question. If the twin asset is not directly observable, it is reasonable to use estimates of return on the asset in question to estimate value or market-calibrated value [Trigeorgis, 1995]. In some cases, the flexibility of the architecture to change in requirements and/or requirements associated with new market products can be valued in terms of directly observable cash flows linked to future operational benefits or market value, making it easy to use the return to value the options. In other cases, the flexibility of an architecture to the new features may not be directly observable through cash flows. Consequently, the analyst may then need to rely on experience for estimation. If the analyst relies on experience and judgment in his/her estimation, the estimates tend to be subjective but could make an implicit use of market information. Note that back-of-the-envelope calculations, which are based on value estimates (rather than on market value), continue to be acceptable and revealing [Sullivan et al., 2001]. It is often the case that both market and subjective value estimates are available. That is, in real options, values are often estimated by inspecting a relevant experience or by using subjective estimates. Hence, this brings a need for a solution that comprises both value and accounts to the different perspectives to the valuation.

The problem of valuing the flexibility of SPLA, therefore, necessarily requires a comprehensive solution that is flexible enough to capture the options from different perspectives and to incorporate multiple valuation techniques; some with subjective estimates and others based on market data, when available. The problem of how to guide valuation and introduce discipline in this setting, we term as the *multiple perspectives valuation problem*. To address this problem, we outline a conceptual *valuation points of view* framework. The framework aims to capture and value the flexibility of the architecture to change from different *points of views*. A *point of view, P,* is a perspective used by an analyst/architect to assess the architectural potential to the change. The perspective could be either technically related (e.g., structural such as development, configuration, deployment; behavioral such as performance, availability, reliability etc.), market-related (e.g., market potential of a product), and/or related to the organization business objectives. Therefore, the corresponding value of an architectural potential to a change and/or accommodating new features may be relative to the market, to one or more technical dimension of the system, or to the organization, as sketched in Figure 1. The purpose is to reach a comprehensive value of growth options from different perspectives. In addition, the aim is to promote flexibility through incorporating both subjective estimates, which may implicitly use market information and/or explicit market value, when available. Furthermore, it remains an open challenge to strongly justify precise estimates for real options in software. Part of the problem stems in the absence of frameworks that capture the options on the software from different perspectives. The outlined valuation point of view framework is promising to address these shortcomings.

4. Formulation

We build on ArchOptions[Bahsoon 2005] to value the architectural potential to supporting new features relative to several points of view. We sketch and discuss two valuation points of view: these are technical and market valuation points of view. For a valuation point of view p_j and a new feature *i*, the constructed call options could be expressed in *(1)*, where x_iVp_j corresponds to the value of the architectural potential of the change relative to p_j, with an exercise cost of $C_{ei}p_j$.

$$E\,[\max\,(x_iVp_j - C_{ei}p_j,\, 0)] \qquad (1)$$

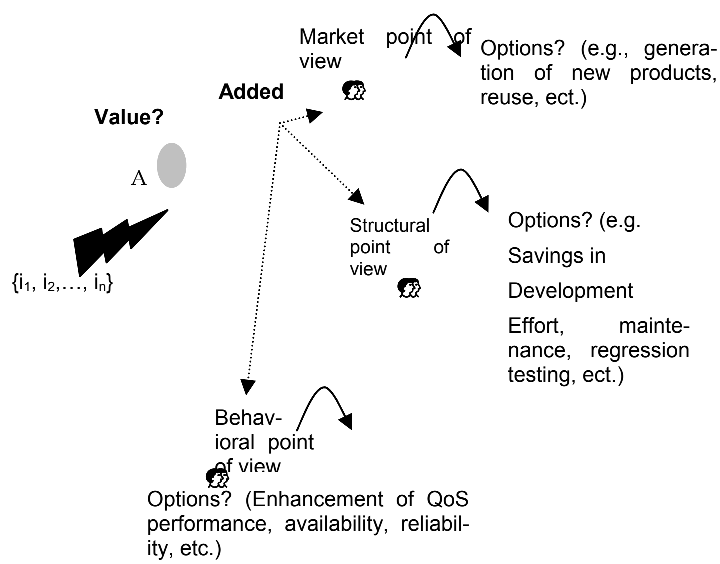

Figure 1. Valuing the options using valuation points of view for features $\{i_1, i_2, \ldots, i_n\}$ on architecture A.

For a valuation point of view p_j: we estimate the cost of accommodating the change and/or the new features. This cost corresponds to the exercise price (Options Theory). We value the potential of the architecture to support the new feature/change. We analyze ways for computing the fluctuation in the estimated value. The major inputs of the ArchOptions model would have been identified: These are x_iV_{pj} (i.e., Value of the "architectural potential" in supporting the change), σ_{pj} (i.e., the "fluctuation" in the return of value of x_iVpj), and C_{epj} (i.e., the estimate of the likely cost to accommodate the change) and relative to the valuation point of view pj. Interested reader may refer to [Bahsoon and Emmerich, 2004]. Having these parameters, we can then construct calls to value the flexibility of the architecture in supporting the new features.

Estimate C_{eipj}. This corresponds to an estimate of the cost of the architectural strategy, mechanisms, and/or the associated implementations, which realize the new feature. The cost corresponds to the exercise price. Generally speaking, ArchOptions is flexible to incorporate either coarse-grained or fine-grained cost estimation. Note that the ArchOptions model is complementary to expert estimation, where expert estimates of the change can be fed into our model. To help experts come up with estimates that are more precise, they can inspect relevant effort, past projects, associated design patterns, and so forth. Alternatively, techniques such as COCOMO II [Boehm et al., 1995] may be used if the key predictors, such as size of the change can be reliably estimated. As with expert-based estimation, the estimates for change could be fed into the model. Note that by inspecting a previous valuation experience to satisfy the concept of "twin asset" and by identifying the key predictors to COCOMO II, we end up applying a "composite" approach to cost estimation. An approach which combines both expert knowledge and parametric estimation is said to be more precise than approaches which solely rely on either expert knowledge or parametric models to estimation [Briand and Wieczorek, 2002]. However, the real-world usefulness of models such as COCOMO II has been questioned for constant and unexplained calibration, which often leads to inaccuracy in the prediction. It could be also argued that in iterative development, when estimations are continuously recalibrated (e.g., in the Unified Process), it is possible to come up with estimations that are more accurate than COCOMO II, as they will take into account factors, such as the skills of the developers, the project maturity, and other organizational factors.

Estimate X_iV_{pj}. Upon the application of the model, the problem that the analyst/architect faces is that the cost is often tangible, but the value is hard to grasp. For example, refactoring a system of a given SPLA incurs up-front design costs; but the value is so elusive and long-term. Part of the value may materialize if the refactoring exercise is planned so the structure can be utilized to create future value such as future savings in maintenance and regression testing. Such a value may span several dimensions such as ease of future maintainability, extensibility, modularity, reusability, complexity, and efficiency. Alternatively, the architectural potential could be valued in relation to the market, as it is the case with product line-architectures. For example, the architecture could "pull" the options by responding to changes in the market requirements, while leaving the architecture of the software system intact or by requiring minimal changes to the architecture. In many cases, the value crosscuts many dimensions ranging from market to technical leading to both technical and market benefits. Hence, the valuation is relative to the evaluation objectives and the primary business drivers motivating the change. The business driver could be for example, future cost savings, shorter time-to-market, entry to new markets, service enhancements, and so forth. In many cases, we consider that the right to claim future cost savings as a result of the architecture supporting the change is a value. In other cases, the value of the architectural potential is a consequence of an upfront investment to facilitate future changes, which in turn will create value. The payoff occurs in the future, contingent on uncertain future conditions. It is worth noting that valuing the architectural potential is case dependent and there is no generic off-the-shelf solution to such valuation. The valuation activity is a human-centered activity. Ideally, the valuation is done in connection with the product, strategy, and/or the marketing team. Below, we discuss how we can value an architectural potential to change relative to a point of view. We discuss two valuation points of view: these are technical and market valuation points of view.

Valuation using technical point of view. By using a technical point of view to assess the architectural potential to the change, we may aim at assessing the architectural potential of an architecture to the change relative to some structural or behavioural properties of the system of a given architecture. As an example of the structural properties, we may aim at assessing the expected savings (if-any) in development, configuration, and deployment efforts to be realised upon accommodating the change on the system of a given architecture. We may also be interested in assessing savings in licenses and hardware. For the behavioural properties, we may for example, aim at understanding the economics implication of the changes to accommodate a given feature. The impact may span one or more architectural qualities such as performance, reliability, availability, and so forth. In many other cases, the enterprise could focus the analysis on one technical dimension. For example, by using development point of view to assess the architectural potential to the change, we may aim at understanding the savings in development effort (if any) to be realized upon accommodating the change on the system of a given architecture.

In fact, the choice of the dimensions is dependent on how the enterprise defines its value proposition. As a result, there is no generic off-the-shelf formula. A range of metrics can be used. Typical measures may include cost savings; risk and losses avoidance; increased productivity; reduction in personnel required for integration; reduction in time-to-market; savings in regression testing effort; and/or enumeration of short-term (e.g., quarterly cycle) and long-term (e.g., two-years or more) benefits and so forth. Our assumption here is that the resulting value is cast into monetary value. Valuing the architectural potential to the change requires finding a twin asset with the similar risk characteristic of the one at hand. We have argued that reusing a past development experience such as previous design and its corresponding implementation to inform the valuation bear a resemblance to the concept of a "twin asset" [Bahsoon et al., 2005; Bahsoon and Emmerich, 2004]. Note that much of the valuation in software engineering is based effort measured in person-months. Such valuation is based on similar experience and may hold similar risk characteristics to the case in hand. The valuation does implicitly hold market information as effort valuation if often priced relative to the market. Valuation using the market point of view. The value of the architectural potential could be realized in relation to the market or the enterprise business objectives. This is true when the change is driven by purely market needs: this could be in response to market differentiators, assimilating and exploiting new technologies, in response to changes in standards, customer demands, and market competition. By using a market point of view to valuation, we may aim at assessing the market potential of the architecture upon supporting the change leading to new products, new services, etc. The market point of view may provide an insight on the profitability of evolution and consequently the success (failure) of evolution relative to the market upon accommodating the change. The analysis may highlight the role of the architectural flexibility in instantiating from the core architecture new market products. This gives the analyst/architect a way to think about this flexibility as being tangible. The analysis may provide an answer to when the payback will be realized upon investing in the change.

We have exemplified the use of the market valuation point of view to value the flexibility of a small product-line suite, xlinkit [www.systemwire.com], in responding to changes in the market requirements. The change is driven by a need to accommodate a new market standard. In summary, the xlinkit suite provides capabilities for checking the consistency of distributed and heterogeneous documents. xlinkit uses a built-in grammar-based Extensible Markup Language (XML) validation language, referred to as CliX, to the consistency checking and the validation of these documents. Being a grammar-based validation language, CliX has some limitations when validating complex documents, which are inconvenient and difficult to represent using grammar-based languages. Example of this category of documents is patterns of graph-structured data of scholarly research. Schematron is a grammar-free validation language that is suitable for validating this category of documents. The current xlinkit implementation does not support Schematron. As Schematron is undergoing ISO certification, Schematron is likely to become one of the most used XML validation languages in the market. For xlinkit, the support of Schematron is likely to enhance the product potentials for the capability of CLiX and Schematron are complementary. This is in turn may translate into long-term revenues for the enterprise due to likely penetration of new markets. In [Bahsoon and Emmerich, 2005], we have shown how ArchOptions can value the flexibility of the core xlinkit architecture in integrating Schematron. Upon valuation, we have appealed to the use of two valuation points of views: the maintenance and market valuation points of views. The analysis has shown a possible way of using ArchOptions to provide insights into investment decisions related to SPLA. The case has provided an idea on how ArchOptions can be employed to quantify the value of the architectural potential in supporting new market product while achieving a net benefit.

Using the *market point of view* to value the architectural potential has some shortcomings:

Limited applicability. The only time where an architectural potential can be assigned a market value is when the resulting product due to introducing new feature can be sold, create market revenues, or be correlated with the market.

The valuation is subject to manipulation and fairly subjective. This is because the valuation could be affected by variation in the market conditions such as supply and demand, market competition, contractual agreements etc. This often leads to subjectivity upon assigning a market value.

A question of interest, however, how could we capture such value? In real options, values are often estimated by inspecting a previous relevant experience or by using subjective estimates. The participant in the valuation activities may include the developers, the architects, the project mangers, the market analysts, and other stakeholders. Interviews, meetings, or surveys are often conducted to gather benefit information. It is the norm that enterprises construct business cases for justifying the upfront investment in a particular architecture. In some cases, a business case may include some probable evolutionary milestones in the lifetime of the architecture, forecast of possible revenues, enumeration of some benefits, risks, and so forth. The business case may also include estimates of costs and valuation scenarios for probable payback upon realizing the evolutionary milestones, such as instantiating from the core architecture a new market product. If this is the case, the use of valuation scenarios to capture the possible value of the architectural potential upon accommodating the change over a period of interest becomes feasible. The scenario valuation preserves the dynamisms entailed by the options approach and accounts for various possible and foreseen values.

Figure 2, For example, depicts an extract from Company Y's valuation of the probable payback upon instantiating from the core architecture a simplified new market product and in response to market requirements. The valuation uses five scenarios showing a likely payback value ranging from £-15,028(Scenario 3), £14,025(Scenario 1), £37,472(Scenario2), £40,472(Scenario 4), to £55,153(Scenario 5). Note that these values correspond to the present value:

$$x_i V_{\text{market point of view (scenario 1)}} = £14,025;$$

$x_i V_{\text{market point of view (scenario2)}} = £37,472;$
$x_i V_{\text{market point of view (scenario 3)}} = £\text{-}15,028;$
$x_i V_{\text{market point of view (scenario4)}} = £40,472;$
$x_i V_{\text{market point of view (scenario 5)}} = £55,153.$

Business Case for PAML						
Income						
		FTB	**Amex Bank**	**EFG**	**PAML1**	**Co-op**
Licence		$250,000	$243,000	$447,100	$250,000	£360,000
PS charge to customer		$100,000	$50,000	$300,900	$100,000	£460,000
PS income after costs (cost = $59,500 assuming 100 man day deployment)		$41,500	-$9,500	$241,400	$41,500	£400,500
Maintenance		$54,000	$52,488	$96,574	$54,000	£77,760
%Income		**FTB**	**Amex**	**EFG**	**PAML 1**	**Co-op**
Product Marketing Licence	10.2%	$25,500	$24,786	$45,604	$25,500	£36,720
PS	35.2%	$14,608	-$3,344	$84,973	$14,608	£140,976
Corp-License	36.0%	$90,000	$87,480	$160,956	$90,000	£129,600
Corp-Services	9.5%	$3,943	-$903	$22,933	$3,943	£38,048
Corp-Maintenance	8.5%	$4,590	$4,461	$8,209	$4,590	£6,610
TOTAL INCOME$		**$138,641**	**$112,481**	**$322,675**	**$138,641**	
TOTAL INCOME£		**£81,553**	**£66,165**	**£189,809**	**£81,553**	**£351,953**
Expenditure						
Goal total Man Days	848					
Daily Internal Charge Rate	£350					
Total Expenditure	**£296,800**					
Income						
Payback Scenario 1						
FTB+AMEX	£147,719					
PAML1	£81,553					
PAML2	£81,553					
Total income	**£310,825 Profit =**	**£14,025**				
Payback Scenario 2						
FTB+AMEX	£147,719					
PAML1	£81,553					
2 AMLE deals	£105,000					
Total income	**£334,272 Profit =**	**£37,472**				
Payback Scenario 3						
FTB+AMEX	£147,719					
PAML1	£81,553					
1 AMLE deal	£52,500					
Total income	**£281,772 Profit =**	-£15,028				
Payback Scenario 4						
FTB+AMEX	£147,719					
EFG	£189,809					
Total income	**£337,527 Profit =**	**£40,727**				
Payback Scenario 5						
Co-op	£351,953					
Total income	**£351,953 Profit =**	**£55,153**				

Figure 2. An extract from Company Y's valuation of the probable payback upon instantiating from the core architecture a simplified new market product

Another simplified solution would be using value estimates representing pessimistic, optimistic, and likely [Gilb, 1998] values of the architectural potential, over a specified period of interest.

Calculate σ_{pj}: In short, the volatility $\sigma_{\mathbf{pj}}$ tends to provide a measure of how the stakeholders are uncertain about the future value of the architectural potential relative to the change and relative to p_j; it tends to measure a fluctuation in value. In financial options, practitioners often rely on historical data of investment returns to estimate the volatility of the stock price. This is feasible because the valuation is done in span of the market where high volume of historical data is available. Yet, this is not the case in valuing software. For example, the case of valuing the architectural potential to the change may hint that the uncertainty and the fluctuation in value are private to the given project. Further, such case often occur in low volumes, therefore getting valid data, treating them consistently, and dealing with the non-quantifiable effects makes the valuation and estimating volatility different from market-traded options. Hence, unlike financial options where richly traded-market information on values and uncertainty are available, it is hard to provide reliable and justified estimates of volatility in real options. Note that real options practitioners often rely on subjective opinion to estimate the volatility. In many cases, real options practitioners make simplified assumptions by either using modeling assumptions or making educated guess. For example, one approach is to examine a range of estimates from say 30% to 60% and guess which might be the most appropriate. When the estimates are poorly justified, performing sensitivity analysis to verify the choice becomes essential. In modeling volatility, in some cases we adopt a simplistic solution to the problem. We use stakeholder judgment variation of the estimated x_iVp_j's as a way for estimating volatility. The evaluation team is asked to record their judgment of possible variation, ± % var, of the previously estimated x_iVp_j's. A +%var corresponds to an anticipated percentage increase in the x_iVp_j. A -%var corresponds to an anticipated percentage decrease in the x_iV_{pj}. Possible %var values may be then available for the optimistic, the pessimistic, and likely x_iV_{pj}'s respectively given by Optimistic $x_iVp_j \pm \%var_{o,}$ Likely $x_iVp_j \pm \%var_{l,}$ Pessimistic $x_iVp_j \pm \%var_{p.}$ In real options, σ calculates to the standard deviation of the rate of return on the asset. Intuitively, the %var is analogous to the rate of return on the architectural potential. Accordingly, we take the percentage of the standard deviation of the x_iVp_j variation estimates-the optimistic, likely, and pessimistic values to calculate σp_j.

Construct call options to calculate the option relative to this valuation point of view. Having estimated the major parameters of the model, it is now possible to compute the call options using (2) and (3) on the architecture in supporting change i. As we have noticed, several estimates for $C_{ei}p_j$ and x_iVp_j, ranging from optimistic to pessimistic or representing possible valuation scenarios, would have been computed at the end of the valuation and relevant to a valuation point of view P_j. Examples are depicted in Table 1. Based on the case and the evaluation objectives, the analyst may then compute optimistic, pessimistic, or likely options.

$$E\,[\max\,(x_iVp_j - C_{ei}p_j,\,0)] \qquad (2)$$

$$C = x_iVp_j\,N(d_1) - C_{ei}p_je^{-r(T)}\,N(d_2) \qquad (3)$$

where,
$$d_1 = \frac{\ln(x_iVp_j / C_{ei}p_j) + (r + \sigma_{pj}^2/2)(T)}{\sigma_{pj}\,(T)^{1/2}}$$

$$d_2 = \frac{\ln(x_iVp_j / C_{ei}p_j) + (r - \sigma_{pj}^2/2)(T)}{\sigma_{pj}\,(T)^{1/2}} = d_1 - \sigma_{pj}\,(T)^{1/2}$$

Table 1. Example of estimated parameters at the end of the valuation

Variable	Estimated Parameters
$C_{ei}p_j$	*Optimistic* $C_{ei}p_j$
	Likely $C_{ei}p_j$
	Pessimistic $C_{ei}p_j$
x_iVp_j	*Optimistic* x_iVp_j
	Likely x_iVp_j
	Pessimistic x_iVp_j
σ_{pj}	*Optimistic* $x_iVp_j \pm var_o$
	Likely $x_iVp_j \pm var_l$
	Pessimistic $x_iVp_j \pm var_p$

5. Conclusion

We have highlighted some of the challenges in valuing flexibility and investment decisions in Software Product Line Architectures (SPLA). We have explained the limitations of traditional approaches, which fail to address uncertainty, in valuing flexibility of SPLA. We have motivated the use of Real Options to address these limitations. We have suggested a multi-perspectives valuation approach for valuing flexibility in SPLA and have reported on its formulation using some examples.

6. References

[Bahsoon, 2005] Bahsoon, R. : Evaluating Architectural Stability with Real Options Theory. PhD Thesis. University of London (2005)

[Bahsoon and Emerich, 2004] Bahsoon, R. and Emmerich, W.: Evaluating Architectural Stability with Real Options Theory. In: Proc. of the 20th IEEE Int. Conf. on Software Maintenance, IEEE CS Press (2004) 443-447

[Bahsoon and Emmerich, 2005] Bahsoon, R. and Emmerich, W.: Using ArchOptions to Value the Flexibility of Product-Line Architectures. UCL Dept. of Computer Science, Research Note (2005)

[Bahsoon et al., 05] Bahsoon, R., Emmerich, W., and Macke, J.: Using ArchOptions to Select Stable Middleware-Induced Architectures. In: IEE Proceedings Software, Special issue on Relating Requirements to Architectures, IEE Press 152(4) (2005) 176-186

[Black and Scholes, 1973] Black, F., and Scholes, M.: The Pricing of Options and Corporate Liabilities. Journal of Political Economy. U. of Chicago Press (1973) 637-654

[Boehm and Sullivan, 2000] Boehm, B., and Sullivan, K. J.: Software Economics: A Roadmap. In: A. Finkelstein (ed.): The Future of Software Engineering. ACM Press (2000) 320-343

[Boehm et al., 1995] Boehm, B., Clark, B., Horowitz, E., Madachy, R., Shelby, R., Westland, C.: The COCOMO 2.0 Software Cost Estimation Model. In: International Society of Parametric Analysts (1995)

[Briand and Wieczorek, 2002] Briand, I. and Wieczorek, L.: Resource Modeling in Software Engineering, Second edition of the Encyclopedia of Software Engineering, Wiley (2002)

[EDSR 1-8] EDSER 1-8: Proceedings of the Workshops on Economics-Driven Software Engineering Research: In conj. with the 21st through 28th International Conference on Software Engineering (1999 - 2006)

[Erdogmus et al, 2002] Erdogmus, H., Boehm, B., Harrison, W., Reifer, D. J., and Sullivan, K. J.: Software Engineering Economics: Background, Current Practices, and Future Directions.Tutorial Summary. In: Proc. of 24th Int. Conf. on Software Engineering, ACM Press (2002) 683-684

[Gilb, 1989] Tom, G.: Principles of Software Engineering Management, Addison-Wesley Longman (1989)

[Jazayeri, 2002] Jazayeri, M.: On Architectural Stability and Evolution. Lecture Notes in Computer Science, Springer Verlag, Berlin (2002) 13-23

[MSR04-07] MSR 1-4: Proceedings of the ICSE Workshops on Mining Software Repositories, In conjunction with ICSE 2004-2007.

[Myers, 1987] Myers, S. C.: Finance Theory and Financial Strategy. Corporate Finance Journal. 5(1) (1987) 6-13

[Ross et al., 1996] Ross, S. A., Westfield, R.W., and Jaffe, J.: Corporate Finance (Fourth). Irwin, Chicago (1996)

[Sullivan et al, 2001] Sullivan, K.J., Griswold, W., Cai, Y., and Hallen, B.: The Structure and Value of Modularity in Software Design. In: the Proceedings of the ninth ESEC/FSE, Vienna, Austria (2001) 99-108

[Sullivan et al. 1999] Sullivan, K. J.: Chalasani, P., Jha, S., and Sazawal, V.: Software Design as an Investment Activity: A Real Options Perspective. Real Options and Business Strategy: Applications to Decision-Making. In: Trigeorgis L. (ed.) Risk Books (1999) 215-260

[Trigeorgis, 1995] Trigeorgis, L.: Real options in Capital Investment: Models, Strategies, and Appications. Praeger Westport, London (1995)