

Towards Predictable Assembly of Certifiable Assets

Cobus Smit^{1,2}, John Muller^{1,2}, Jay van Zyl² and Judith Bishop¹

¹ Computer Science, University of Pretoria, Pretoria, 0002
{csmit, jmuller, jbishop}@cs.up.ac.za

² SystemicLogic Innovation Center 222 Grosvenor Road
Bryanston 2194, Johannesburg, South Africa
jay@systemiclogic.com

Abstract. With business drivers such as innovation and the constant influence of changing requirements, organizations seek ways to increase time-to-market capability, enhance overall productivity and maintain or improve product quality. At the same time these organizations strive to leverage legacy investments. Assets unbundled from these legacy systems, acquired through vendors or developed in-house form the building blocks of products that promote consumer trust. The characteristics of quality attributes on different levels of abstraction differ substantially and are influenced not only by technical perspectives, but also business-oriented perspectives. Research on Predictable Assembly of Certifiable Components (PACC) [1] has proved to be an important consideration in enabling trust on higher levels of abstraction. In this paper we consider primary influences in moving towards predictable assembly and define a high-level reasoning framework for predictable assembly of certifiable assets in the construction of the application portfolio.

1 Introduction

With business drivers such as innovation and the constant influence of changing requirements, organizations seek ways to increase time-to-market capability, improve product quality and enhance overall productivity. At the same time these organizations strive to leverage legacy investment. Reuse and associated product line benefits [9] address many of the aforementioned aspects. Closely related to product line benefits and general organizational goals are the issues surrounding confidence and predictability in the combined application of these assets to promote business sustainability.

Research such as PACC (Predictable Assembly of Certifiable Components) and an implementation (PECT) [1] [2] apply the concept of property theories (analytic models) to component technologies to enable predictable assembly prior to composition. Similar applications in real-time product line environments have also been implemented [3].

In this paper we investigate the characteristics of predictability as dictated by lower component level assemblies and apply the prominent concepts to asset assemblies as they are used in products across product lines that form part of application portfolios. We define a reasoning framework for predictable asset level assembly. The assets we refer to are coarser grained and used in the context of products in large organizations with multiple users and millions of lines of code. Employing fine grained low level reuse is impractical in these environments.

In order to define a higher level reasoning framework, we consider prominent confidence affecting influences that include predictability on component abstraction, asset granularity and design influence. Fig 1 shows high level considerations for discussion presented in this paper. These combined influences form a conceptual framework that can be used to explore aspects related to achieving predictability and higher degrees of confidence on application portfolio levels from a business perspective.

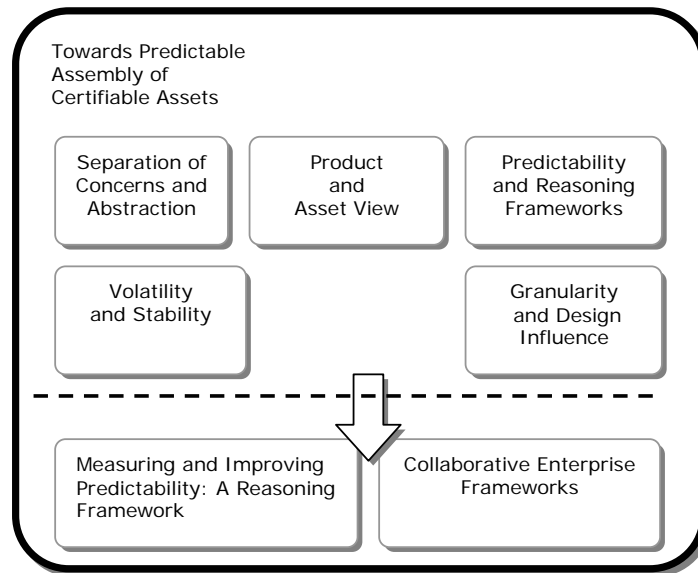


Fig 1. High level considerations.

Many of the aspects are closely interrelated, yet are still significant individually to be regarded as separate core influences. We consider most of these aspects separately to form a basis for defining a reasoning framework in the context of predictable assembly of certifiable assets. Our aim is not to focus on the direct requirements for certification, but rather investigating the holistic view on independent assets assembled into products across product lines. The primary work presented in this paper has been performed in financial industry.

2 The Separation of Concerns

To effectively characterize predictability and higher levels of confidence in asset assemblies, we identify different concerns that constitute products and product lines. The Separation Continuum (Fig 2) [5] separates different vertical and horizontal concerns by defining a layered view on a common architecture (e.g. as used by financial industry). Vertical concerns range from business abstractions to technical architectures that realize business models. Horizontal concerns address customer-facing aspects to business-facing aspects (visual to non-visual).

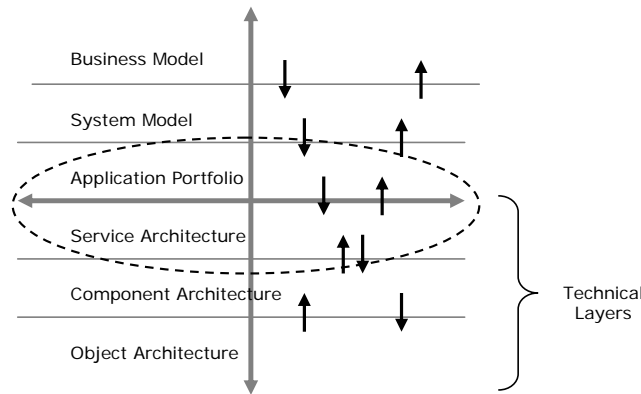


Fig 2. The Separation Continuum.

Both bottom-up and top-down views on the continuum are important in the context of certifiable asset assemblies.

Top-down: business models drive strategic initiatives. Interpretations of how business systems must function are realized through the application portfolio specification.

Bottom-up: Architectures are already existent and present enablers or inhibitors to the business model. [5]

The top-down and bottom-up views are significant. It can be argued that predictability in a bottom-up view means that components and services and their subsequent assemblies comply with some criteria (dictated by quality attributes). When the so-called products are then assembled from these individually certifiable components, services and respective predictable assemblies, top-down factors influence the adequacy of the product in the market context – again influencing the confidence in the product (and constituent assets) themselves.

We now take a closer view on the product and product line constituents i.e. products and asset views in the context of the Separation Continuum.

3 Granularity and Design Influence

The product line approach does not make use of small grained and opportunistic reuse – assets and product are planned. Planned assets are subsequently used to construct different products across product lines. Both assets and products-as-assets are stored in a common repository.

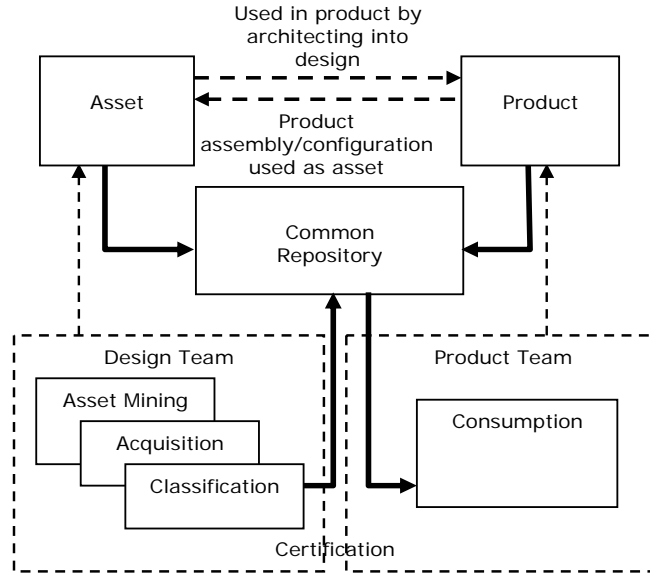


Fig 3. Products and Assets Perspectives

The assets that are fit for product construction, map to the application portfolio. To ensure a predictable or increased confidence level on the portfolio level, the courser grained layers of the continuum present a perspective on already available industry solutions for reuse i.e. components and services. It is the source of supply of these components, their applicability in the product line context and quality that will prove important in the context of a product assembly.

Certification must be applied to the source of the assets as well as the process used for consumption. Products themselves also have to go through a product release and certification phase. Building assets in-house is seldom a viable option and decisions are made to either acquire or mine assets out of existing resources, thereby leveraging investment to the greatest extent possible. Unbundling monolithic systems into components and services present many subtleties. Mapping out functional areas in monolithic systems is an important step in determining these assets. These systems also have their own execution environments with built-in quality characteristics.

But it should be realized that the separate systems in themselves are of different levels of granularity with functionality and qualities that are subject to redundancy or inadequacy. We define a model that explores design influence and granularity.

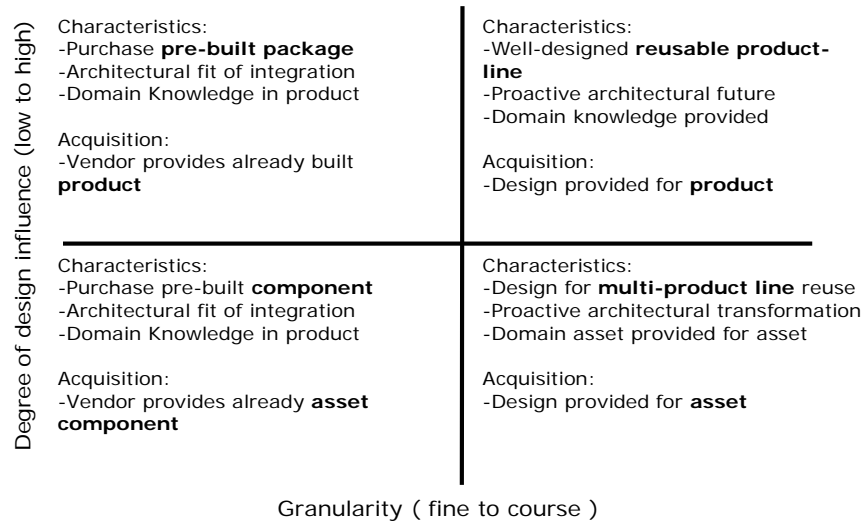


Fig 4. Design Influence and Granularity

The importance of design in architecture cannot be over stipulated. Traversing the details of design is not in the scope of this paper. It is important however to study the influence of design in the organization and the granularity of software assets before considering predictability- this is particularly true where control and the effect of external dependencies are concerned. Planning is affected by internal strategy i.e. if all assets are acquired then clear guidelines should be defined for vendors to understand integration abilities, whereas a higher degree of design control translates into better results and an ‘up-front’ view of architecture.

External dependencies influence overall confidence. Component-Off-The-Shelf (COTS) [17] related issues and requirement aspects are typically applicable here. Bass et al. [8] describes the influence of granularity on defining quality attributes for architectures as being a difficult task. Attribute analysis does not lend itself to standardization. Patterns and pattern frameworks also function on different levels of abstraction with different relationships among them – it becomes cumbersome to decide on which patterns to analyze for quality.

We have defined Fig 4 to show that with more design influence more control and subsequent confidence is derived. More control subsequently enables increased capability to manage products. The alternative (from a pure acquisition perspective) is to promote standards and embrace collaborative enterprise principles. We explore aspects surrounding collaborative enterprise principles in section 6.

4 Predictability and Reasoning Frameworks

With important influences including the separation of concerns, products and assets, and granularity,

the predictability and the basis for reasoning frameworks can now be established. In order to install higher levels of confidence and enable levels of predictability in asset assemblies (products in product lines), it is required to consider the characteristics of predictability as dictated by lower implementation levels. PECT [2] enables predictable assembly from certifiable components by combining component technology with analysis models. Similar work is described in [18].

Predictable assembly in PECT means:

1. Assemblies of components are known, by construction to be amenable to one or more method for predicting their emergent properties.
2. The component properties that are required to make these predictions are defined, certified and trusted.

Hissam et al. regard component certification and predictable assembly as correlates. It is this same way that certifiable assets must contribute to predictable behaviour in products. Van Zyl [4] describes how the vertical implementation layers are used to form core assets that collectively yield product lines in product families in the application portfolio.

The set of qualities (forming a reasoning frameworks) imposed on the component assemblies are often ambiguous and overlapping [8]. Typical examples include performance, reliability, security and modifiability. Our recent work ([16]) highlights architectural considerations in enabling predictability and higher levels of confidence on the service architectural layer, where services are loosely coupled, standards-based and message-oriented (e.g. web services) and functions in cross-organizational contexts. Qualitative considerations such as security, performance and reliability were explored. The cross-organizational context does pose certain challenges with regard to the variance in reasoning frameworks. Each organization, possessing their own set of products and execution environments, will have different standards and interpretations of quality attributes. Bass et al [8] describes difficulties relating to adequately defining quality attribute primitives for architectures – aspects such as granularity and attribute specificity apply. Standards in reasoning frameworks will play an important part. Organizations functioning in certain domains (e.g. financial domains) all share similar characteristics emphasizing the importance of collaborative initiatives.

Hissam et al. [2] purports that it would be meaningless if predictions could not be validated. Model and empirical validation is used to accomplish validity on the component assembly level. Establishing validity on higher levels of abstraction is subject to multi-dimensional and increasingly complex and dynamic factors. We investigate customer behaviour, product manageability capability and market opportunities as top-down factors that aid validation on higher levels of abstraction (section 5).

One of our primary concerns now lies with establishing higher levels of predictability on the products and product line level via the assembly of assets. Fig 6 illustrates the levels of volatility and stability in software asset creation. Volatility repre-

sents behaviour that is imposed on stable technology platforms. Section 3 highlighted the occurrence of volatility in patterns due to the influence of granularity.

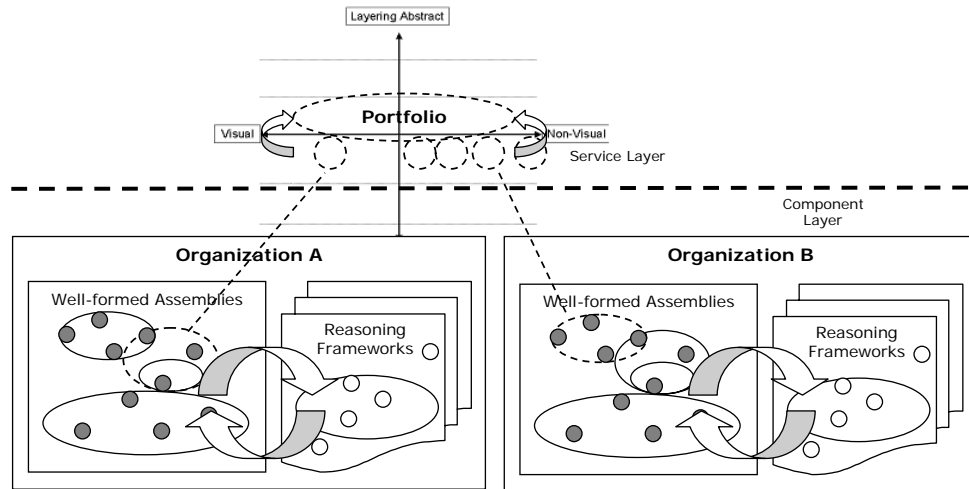


Fig 5. Different Reasoning Frameworks across Organizations

Each layer of abstraction has its own set of characteristics and is built on more fine grained and increasingly complex structures.

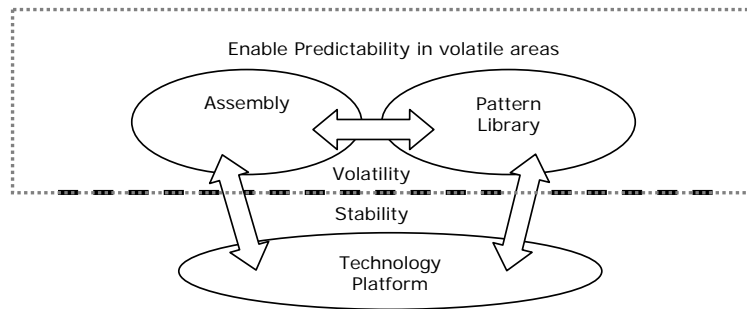


Fig 6. Software Asset Creation using a Pattern-Technology-Assembly Method

Predictability on technical layers does not have the same characteristics as business-oriented layers – section 5 defines qualitative and quantitative attributes that collectively help characterize analytic models on the portfolio level. Therefore, we must also consider predictable behaviour for products and assets from the business-perspective (top-down).

Quality considerations on business-oriented abstractions can be viewed from a customer and market perspective. The appropriateness of products and assets used in

product lines are constantly being challenged by factors of change. The levels of confidence and predictability of behaviour is now dictated by the 'trust' of organizations and consumers in the product and assets themselves. A top-down perspective on predictability will now consider customer requirements and market behaviour as well as the predictability or forecasting of the product life cycle. This is done in order to establish how long products and their constituent assets will remain viable in the market context.

5 Measuring and Improving Predictability: A Reasoning Framework

An approach is required to measure product viability and means of understanding product diversity and improvement. The so-called current status of a product in terms of viability is referred to as product health. Two main focal areas include understanding the current product health and the impact of customers on the current product offering.

Product health can be viewed from two main perspectives ([12] and [13]):

1. Opportunity – represents an external view on a product where it is important to understand the demand for the product.
 2. Capability – refers to the capability of an organization to manage a product.
- We derive an opportunity-capability matrix in order to analyze products.

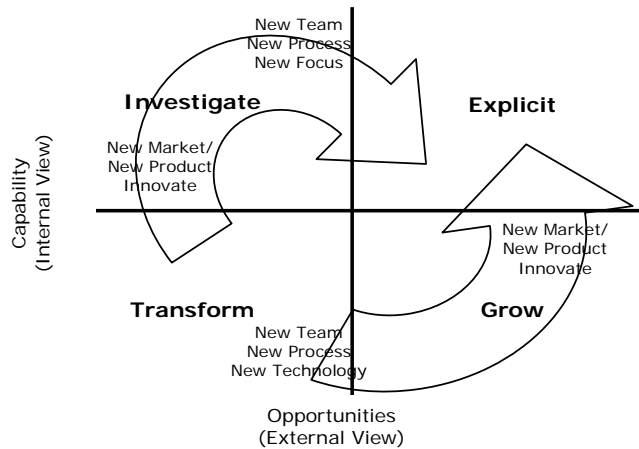


Fig 7. Product Improvement in the Capability-Opportunity Matrix

Portfolio oriented attributes are required in order to measure these products - capability and opportunity attributes (both qualitative and quantitative) are defined [13]: Opportunity attributes include:

1. Market size – the size of the market this product is aimed at. (Quantitative)
2. Market share – How much of this market does the organizations already own? (Quantitative)
3. Growth Market – Is the market size or demand for the product increasing? (Qualitative / Quantitative)
4. Barriers to Entry – The effort involved in generating sales in this market with this product. (Qualitative)
5. Strength of differentiation – Is this product perceived to offer a higher level of differentiation than competitive products? (Qualitative.)
6. Brand awareness – Is the brand that this product falls under well understood by the market. Does it add value to the product? (Qualitative)
7. Sales – Are sales targets being achieved and or exceeded for this product offering? (Quantitative)

Capability attributes include:

1. Profitability – Is this product profitable to the organization? (Quantitative)
2. Cost Control – Are costs easily controlled and recovered for this product offering? (Qualitative)
3. Innovation ability – Does this product offer additional levels of improvement or innovation? (Quantitative)
4. Variability – Is the product offered with a high number of variations? (Quantitative)
5. Technology – Is the product automated and to what extent. Is there scope for further automation? Does the technology limit the product? (Qualitative)
6. Management – Is the product well managed? Does it require a high level of human intervention? (Qualitative)
7. Compliance – Does the product comply with current legislation both organizational and regional? (Quantitative)
8. Risk profile – Does the product carry a high level of risk? (Qualitative).

Products are placed based on the measurements of these attributes. The four quadrants are subsequently defined as follows (as illustrated in Fig 7):

1. High Opportunity – High Capability, implies that this product can be exploited in its current state.
2. High Opportunity – Low Capability, implies an investigation into internal capability to fully realize the potential of this product should be undertaken.
3. Low Opportunity – High Capability, implies the need to grow the market it currently operates in or to identify new markets for this product offering.
4. Low Opportunity – Low Capability, would imply that this product line requires a complete review of all aspects or a decision on whether this product line should be retired in its current form.

Two opposing progressions now present themselves i.e. product degradation and product improvement (Fig 7). Product degradation is characterized by a loss of either opportunity, capability or both. Improvement does the opposite – increased opportunity can be achieved through product repackaging, or subsequently entering a new

market. Capability can be enhanced through improved process, more effective technology as well as positive changes in leadership.

We now consider customer behaviour. Behaviour analysis represents a response to the current product offering. Measurement of the response closely correlates with validation techniques used on the component and service assembly level. Customer's contribution to revenue against cost of service is now used the cost of service.

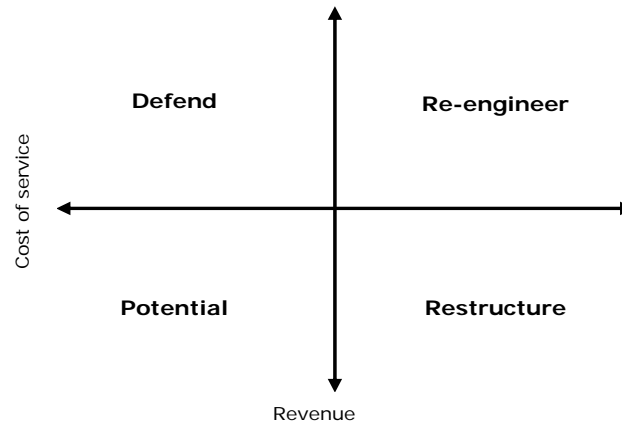


Fig 8. Cost of Service-Revenue Matrix

The four quadrants are defined as follows ([14] and [15]):

1. Customers in High revenue contribution – Low cost of service need their relationship/commitment strengthened, as they are open to persuasion from competitors.
2. Customers in High revenue contribution – High cost of service can be better serviced through improving the product offering.
3. Customers in Low revenue – High cost of service can be viewed as customers who may misuse services and can be improved through education and offering of alternate products to meet their needs. These customers may also have taken the higher revenue generating products up at a competitor.
4. Customers in Low revenue – Low cost can be viewed as customers, if managed correctly, carry a high level of potential for future business.

Understanding customer behaviour enables organizations to set up customer profiles and subsequently identify trends and patterns in the application portfolio. Scenarios where current product offerings meet the needs of the customer profile provide lower cost and yield higher revenues should be undertaken. New product offerings based on customer behaviour yielding improved service costs and revenue as well as meet customer requirements can be investigated. Based on these indicators, organizations can now measure and gain higher levels of confidence in the products that they provide to the consumer.

6 Collaborative Enterprise Principles

Much hype has surrounded the concept of ‘business-on-demand’. With indicators such as customer behaviour, opportunity and product management capability it is now imperative to establish means to achieve agility in order to keep products viable. At the same time it is important to consider the impact of agility on predictability in this context.

Different reasoning frameworks across different component and services environments in the same or differing organizational contexts pose the problem of inconsistency. Each separate environment functions almost in isolation. In much the same way, the monolithic approach to market dominance can be regarded as rapidly becoming obsolete. Establishing domain standards on the technical levels can be projected onto the portfolio layer of abstraction by considering the impact of collaborative partners on future competitiveness – thereby establishing ties across isolated business environments.

Collaboration with new partners will result in the assembly of new product, ultimately embracing new opportunities in response to consumer demands. Harvesting best practices and patterns will create a frame of reference that can be shared by collaborating partners.

We identify three focal areas for collaborative enterprise principles: collaborative enterprise patterns, collaborative enterprise principles and a collaborative enterprise framework (Fig 9) [11].

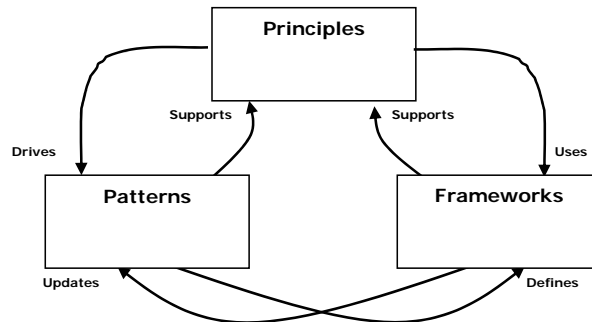


Fig 9. Collaborative Focus Areas

Collaborative Enterprise Principles defines its objective to be the means to enabling competitive advantage through the ability to respond quickly to market demands for quality products and services.

The impact on predictability can now be investigated. Different partners will typically have different reasoning frameworks defined on technical levels. The volatility of assembly (Fig 6) in the collaborative context can now be seen. Subsequent re-configuration is necessary with newly initiated collaborative initiatives. The collaborative framework considers the current organizational configuration, identifies areas of misalignment and gaps and defines a framework within which to re-configure or

improve the organization. Collaborative patterns are documented problems and their resolution in the context of applying Collaborative Enterprise Principles. Documenting these patterns improve both learning and implementation speed within organizations. Collaborative Enterprise Principles are technically composed of collaborative patterns and a collaborative framework. These principles take an alternative view on key drivers that stress the importance that organizations should embrace these principles.

7 Conclusions

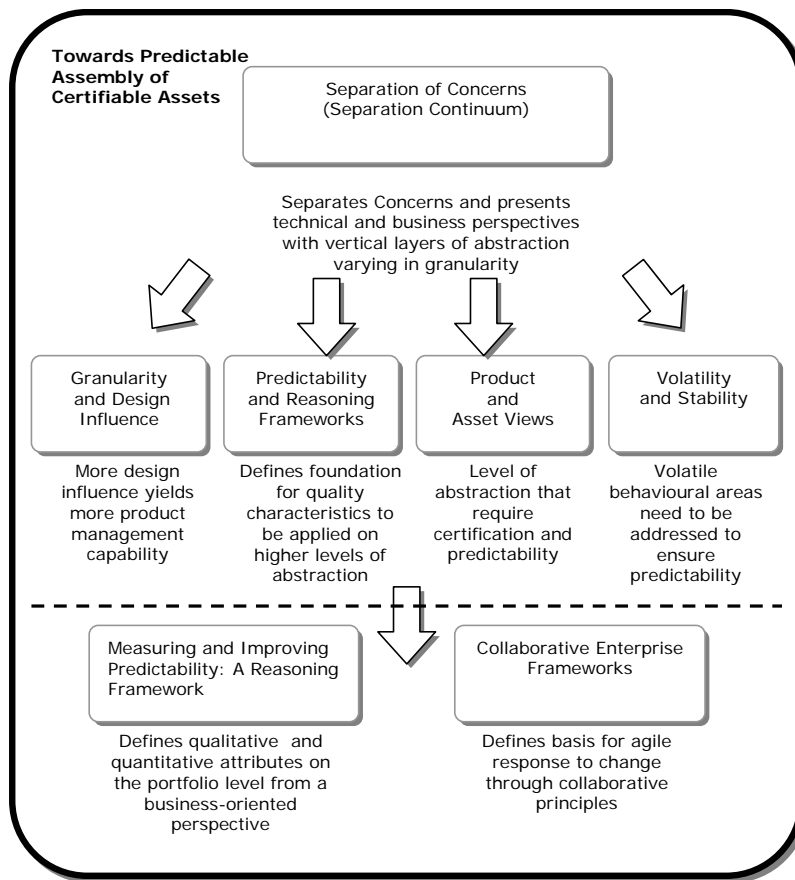


Fig 10. Towards-Predictable Assembly

In our research to date, working with large financial services institutions, we have found that the challenges of implementing predictable asset assemblies are vast and

complex in nature. This is particularly true when considering that these institutions can have more than 300 systems and 20+ strategic business units with millions of clients.

This paper covers a wide range of concepts addressing (and classifying) important concerns required to view large systems. Concepts ranging from separation of concerns, granularity and design influence, predictability in assemblies on technical layers to portfolio and business-oriented perspectives with the focus on collaboration are explored.

The main conclusions concern the impact of different influences on predictability. These are:

1. The separation continuum highlights vertical abstraction in order to promote awareness of the different characteristics inherent to each layer of abstraction. Bottom-up and top-down approaches are used to understand technical-oriented and business-oriented perspectives.
2. Granularity and design influence shows that more design influence increases an organizations ability to manage products, thereby promoting higher levels of confidence. The alternative is to consider how external dependencies should be resolved. We define a matrix that shows characteristics inherent to varying levels of design influence and granularity.
3. Predictability characteristics on the component level are used to understand the concept of quality attributes being applied on separate components to form predictable assemblies. The correlation between certifiable components and predictable assemblies form the foundation for establishing the same correlation on the portfolio level using a top-down perspective. We show that confidence needs to be installed in volatile areas i.e. assemblies and patterns frameworks (that are also influenced by granularity).
4. We explore influences on products and identify aspects that aid organizations in establishing product viability. We define quantitative and qualitative attributes constituting a high level reasoning framework for products (asset assemblies).
5. The need for standards on the lower levels of abstraction is then also projected onto the portfolio level and we consider collaborative principles as a means to promote both higher trust and agility in the product line context.

It is our future focus however to continuously investigate the correlation between predictability on technical and higher abstract behaviours in order to establish defining characteristics influencing certification and predictability in complex organizational environments.

8 Acknowledgement

Thanks to colleagues in the Polelo Laboratory for helpful discussions, especially Kathrin Berg and Basil Worrall. This work was funded by the NRF-DTI-THRIP grant no. 2788.

References

1. Wallnau, K.C.: Volume III: A Technology for Predictable Assembly from Certifiable Components (PACC). Technical Report (CMY/SEI-2003TR-009 ESC-TR-2003-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. (http://www.sei.cmu.edu/publications/documents/03_reports/03tr009.html)
2. Hissam, A.S., Moreno, G.A., Stanford, J.A., Wallnau, K.C.: Packaging Predictable Assembly. In: Bishop, J. (ed.): Component Deployment (Proceedings of the IFIP/ACM Working Conference, CD 2002). Springer-Verlag, Berlin Heidelberg New York (2002) 108-124.
3. Larson, M., Wall, A., Norström, C., Crncovic, I. Using Prediction Enabled Technologies for Embedded Product Line Architectures. Proceedings of the 5th ICSE Workshop on Component-Based Software Engineering: Benchmarks for Predictable Assembly. (<http://www.sei.cmu.edu/pacc/CBSE5/CBSE5-Proceedings.html>)
4. Van Zyl, J. Product Line Architecture and the Separation of Concerns. In: Chastek, G.J. (Ed.): Lecture Notes in Computer Science, Vol 2379. Springer Verlag, Berlin Heidelberg New York (2002) 90-109.
5. Van Zyl, J. A Perspective on Service-Based Architectures. A Perspective on Service Based Architecture: The Evolutionary Concept that Assists Technology Providers in dealing with a Changing Environment. In: Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, Port Elizabeth, South Africa.
6. Van Zyl, J. Implications of using Life Cycle Wide Design on the Software Development Life Cycle. SystemicLogic© Research (work in progress) (<http://www.systemiclogic.com/>).
7. Kang, K.C., Donohoe, P., Koh, E., Lee, J., Lee, K. Using a Marketing and Product Plan as a Key Driver for Product Line Asset Development. In: Chastek, G.J. (Ed.): Lecture Notes in Computer Science, Vol 2379. Springer Verlag, Berlin Heidelberg New York (2002) 366-381.
8. Bass, L., Klein, M., Bachmann, F. Quality Attribute Design Primitives and the Attribute Driven Design Model. In: F. van der Linden (Ed.): Lecture Notes in Computer Science, Vol 2290. Springer Verlag, Berlin Heidelberg New York (2002) 169-186.
9. Clements, P., Nothrop, L. Software Product Lines: Practices and Patterns. Addison-Wesley Publishers, One Lake Street Upper Saddle River.
10. Kishi, T., Noda, N., Katayama, T. A Method for Product Line Scoping Based on a Decision-Making Framework. In: Chastek, G.J. (Ed.): Lecture Notes in Computer Science, Vol 2379. Springer Verlag, Berlin Heidelberg New York (2002) 348-365.
11. Kulenthal, N. An Introduction to Collaborative Enterprise Principles. SystemicLogic© Research (work in progress) (<http://www.systemiclogic.com/>).
12. Kulenthal, N. Product Health Instrument Introduction. SystemicLogic© Research (work in progress) (<http://www.systemiclogic.com/>).
13. Kulenthal, N. Understanding Product Health. SystemicLogic© Research (work in progress) (<http://www.systemiclogic.com/>).
14. Kulenthal, N. Opportunity focus on Product or Market. SystemicLogic© Research (work in progress) (<http://www.systemiclogic.com/>).
15. Kulenthal, N. The Impact of Customer Behavior on Product Health. SystemicLogic© Research (work in progress) (<http://www.systemiclogic.com/>).
16. Smit, C., Muller, J., Bishop, J., Van Zyl, J. Architectural Consideration in Predictable Component Assembly: A Service-Oriented Perspective. Technical Report submitted to Component Deployment, 2004.

17. Maiden, N.A.M., Ncube, C., Moore, A. Lessons Learned during Requirements Acquisition for COTS systems. In: Communications of the ACM, Vol. 40, No. 12 (December 1997), 21-25.
18. Klein, M., Kazman, R. Attribute-Based Architectural Styles. Technical Report (CMU/SEI-99-TR-022). Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University. (<http://www.sei.cmu.edu/publications/documents/99.reports/99tr022/99tr022abstract.html>)