Introducing Select Perspective - Delivering Component Based Solutions

Whitepaper
Version: 2.0

Disclaimer and Trademarks

Information in this document is subject to change without notice and does not represent a commitment on the part of Select Business Solutions, Inc. to provide or continue providing said functionality. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without prior written consent of Select Business Solutions, Inc. Company names, data and other information appearing in examples are fictitious in nature unless otherwise designated.

The software described in this document is furnished under license or non-disclosure agreement and may be used or copied only in accordance with the terms and conditions of that agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement.

Select Enterprise is a Registered Trademark of Select Business Solutions, Inc., and Select Component Factory, Select Component Architect, Select Component Manager, Select Component Portal, Select JSync, Select CSync, Select C#Sync, Select ForteSync, Select VBSync, Select DBSync, Select XMLSync, Select ORSync, Select Document Generator, Select Object Animator, Reviewer for Select Component Architect, Reviewer for Select Enterprise, Reviewer for Rose, Select Process Director, Select Process Director Plus, Select UDDIServer, Select Application Composer, Select Scope Manager, Select Perspective, Select SE, Select SSADM, Select Yourdon are all Trademarks of Select Business Solutions, Inc.

Because of the nature of the material, numerous hardware and software products are mentioned by their trade names in the publication. In most, if not all, cases these designations are claimed as trademarks by their respective companies. It is not the publisher's intent to use any of these names generically, and the reader is cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

For more information about Select Business Solutions Inc., or to download an electronic copy of this document please visit the Select Website: http://www.selectbs.com/

Table of Contents

Introduction

For CIOs, IT Directors, Project Managers, and solution developers who need a software development process that meets the many demands of modern solution delivery, Select Perspective is a comprehensive development lifecycle for component based solutions that supports parallel development activities in order to reduce time-to-market.

Unlike other development processes, Select Perspective is focused on a small number of key deliverables within an organizational framework of suppliers and consumers of solution driven components.

Service-Oriented Architecture

Component based design and development has become the modern approach for delivering solutions that meet the needs of today's businesses. This is no accident. Components offer important business, technical, and economic benefits for any organization whether large or small. Components and Services can be combined in the design of a service-oriented architecture – a process fully supported by Select Perspective

To realize these benefits it is critical that your software development lifecycle is designed to support the contemporary approach. Whilst it may be tempting to consider components and services simply as a software construction technique and modify any existing processes - assuming you have them defined - this will not leverage the power of components and services. Key benefits will remain unrealized.

Containment of Complexity

Components have a strong boundary that is protected during the many stages of development, e.g. analysis, design, construction, testing and deployment. This encourages a development process that supports parallel development, including activities distributed across multiple sites.

Parallel and Distributed Development

Parallel development leads to new organizational dynamics: the interaction between suppliers and consumers of components. Solution builders consume components and services that come from a number of sources: constructed by other project teams, reused corporate software assets, provided by component brokers, or supplied by third party partners.

A Proven Process

Proven in real world, small to large scale IT efforts, Select Perspective has been refined over several years into a collection of highly practical, tightly focused best practices for solution development. It enables IT organizations to **take safe, low-risk steps into CBD** for both small and large-scale projects. Select Perspective supports **mature** CBD organizations that wish to realize the high-levels of reuse that components can offer.

Select Perspective has three principles:

- **Small set of key deliverables** e.g. a use case diagram or a class model. If any diagram is not necessary to a particular solution then you do not produce it; anything not on the delivery line has been ruthlessly pruned.
- **Based on experience**; experience gained by many consultants working over many years with numerous customer teams to deliver successful solutions; team sizes ranging from a few developers to large globally distributed teams, all with different skill levels.
- **Designed to fit most organizations**: recognizing that each organization is different. So it can be adapted to work within existing processes used by an organization or it can be a new foundation from which to start and then adapt as experience grows.

These principles make Select Perspective fit for real-world solution delivery.

Broadly Based

Some methodologies tend to revolve around one technique, such as Use Case modeling or database design. This is simply too limiting for IT organizations that need to build sophisticated applications. To more fully support design activities, Select Perspective seamlessly integrates the **three major kinds of visual modeling techniques** - Business Process Modeling (BPM), Unified Modeling Language (UML), and Data Modeling. This allows designers to start anywhere and use the right tool for the right job, producing a better result.

About Select Business Solutions

For more than 10 years, Select Business Solutions has created a successful track record with tools and process solutions and is generally recognized as being one of the early adapters of Service and Component Based Development (CBD) worldwide. The technology behind the Select tools is continually recognized as being innovative, ensuring that customers' demands are met to the fullest satisfaction. It is this three-way focus on development, customers and emerging markets that makes Select Business Solutions a leader in its field.

The ADT division of Select Business Solutions provides a comprehensive suite (named Select Component Factory™) of business software development solutions comprised of Select Component Architect™, Select Component Manager™, Select Component Portal™, Select UDDI Server™, Select Process Director™, Select Reviewer™, Select JSync™, Select VBSync™ Select C#Sync and C++Sync™. These tools are supported by a full complement of professional services (support, training, consultancy/mentoring) in addition to the development method, Select Perspective™.

Component and Services

Components have numerous and varied descriptions. Despite this perceived confusion there is a small set of characteristics that component developers agree upon. Each component:

- Is a *unit of deployment* - executing within computer systems;
- Is an *implementation mechanism* for one or more services;
- Offers *published service interfaces* each defining a set of service operations;
- *Interacts* with other components according to defined communications standards;
- Is *assembled* with other components and services to realize solutions.

Component based development (CBD) combines the construction and testing of components with other new or reused components to form the system solution.

Business Benefits

Massively parallel development and the ability to outsource development resulting in **faster development cycles** and delivery; you simply get there quicker. There is more work perhaps, in locating or buying suitable components and certainly more effort in testing and integrating reused assets into a possible solution. But this effort is less than building everything from scratch.

Solution delivery becomes **more assembly, less development** and new recruits to the team can be productive earlier as they have service "building blocks" already available.

Specialist knowledge can be leveraged more easily. Take the example of a tax component, tax experts will work with the component developers to **encode their rare expertise** into the final component; developers do not have to become taxation experts!

Economic Benefits

In the language of the markets, a commodity is any item that may be freely bought and sold; the key phrase here is "freely bought and sold".

Components and services used to be restricted to the confines of an organization. Now there are **market places and brokers** that buy and sell components. These market forces are lowering costs through multiple sales and raising the quality of the supplied component as greater numbers of buyers reuse it within their solution. As a result it has become increasingly uneconomic to build a complete system from scratch.

Technical Benefits

Technical benefits spring from the underlying nature of a component: its boundary, defined as the service interfaces offered by the component and the encapsulated functionality that implements them.

A strong boundary **encloses complexity**. If there are complex functions to perform such as tax calculations, then these can be enclosed within the boundary of a tax component. Component developers focus solely on the tax formulae, rules, etc. Focus gives more benefit than a small scope of interest it also increases productivity by ignoring irrelevant topics.

Containment of complexity also leads to **assembly of solutions** rather than full-scale development. Components can be plugged together to quickly assemble complex functionality and may be substituted later if necessary.

When you have a clear delineation of a boundary and can specify the services – the component with its interface contract – then it becomes possible to **distribute the work** necessary to construct the component itself. This may be distributed within the organization or outsourced to another reliable partner. Maintenance for the component may then be the responsibility of the supplier – the trusted partner – or may be undertaken by the component consumer.

As the supplier improves the component by, for example, including up-to-date tax rules, all the consumers of the component benefit.

Service Oriented Architecture

Today's hype is all about web services. Has the advent of web services made CBD a thing of the past? To the contrary, the market-place concept lies at the heart of the web service concept and at the heart of CBD too.

Service Market Places

Suppliers make their services available through "markets" – the registries of web services supported by companies such as IBM, Microsoft and Sun. Offering the computer service equivalents, of white, yellow and green pages in telephone directories, consumers of services can search the registries for the functionality they require.

Within an organization, the management of service resources is as vital as it is for components. Components and services represent long-lived software assets; investment in these assets must be protected. A management regime of search, acquisition, quality assurance and deployment is the only way to ensure that solutions using components and services can be successfully assembled and deployed.

The concept of the market-driven acquisition of services is embodied in the processes and techniques of Select Perspective. Select Component Manager provides tool support, with its ability to view UDDI registries, commercial market places (such as ComponentSource) as well as internally managed catalogues of software assets. Select Component Architect supports the active inclusion of these assets into an effective solution design based on a service-oriented architecture.

Technology Driven

It is arguable that the web service phenomenon is technologically based. There is no reason why the current family of acronyms (HTTP, UDDI, WSDL and SOAP) will persist for any prolonged period. There is considerable speculation in the industry press about the durability of the current UDDI specification. Many articles point out that while the use of HTTP and SOAP is ubiquitous; it is in no sense mandatory.

In contrast, the concept of service (independent of technology) is persistent. The focus of CBD from its first inception has been the identification of services forming the boundary to each component. This emphasis continues in its latest incarnation – web services.

Service-Oriented Architecture

Underlying the hype about web services is a more genuine movement towards the concept of a service-oriented architecture[1]. Solutions are assembled on an architecture that uniquely provides functional services. These services may be implemented as components (usually core functionality providing competitive advantage) or consumed from third party suppliers as "web services" (peripheral functionality necessary to all solutions such as user identification and verification). Of course, organizations can choose to expose their own core functionality as web services. These services can become an additional revenue source by being micro-rented to other service consumers.

Select Perspective is unique in offering processes and techniques that fully support the creation of service-oriented architectures. It supports practitioners when they think about and answer the hard questions such as "what services do I need". Select Component Factory provides the tool support necessary to make the delivery of solutions based on service-oriented architectures a practical reality. It allows practitioners to discover, retrieve and reuse the software assets needed to assemble solutions.

[1] Refer to the Select Business Solutions white paper "Components and Services, Towards a Service Oriented Architecture".

Consumers and Suppliers

"We think globally, specification is in Europe, construction is in India, and assembly is in the USA; Select Perspective supports this completely" Program Manager, ERP product supplier.

All successful economic systems are based on a market place of suppliers and consumers; this supply chain is the engine of economic success (Figure 1). We can enjoy that success by applying the same model to delivering software solutions.

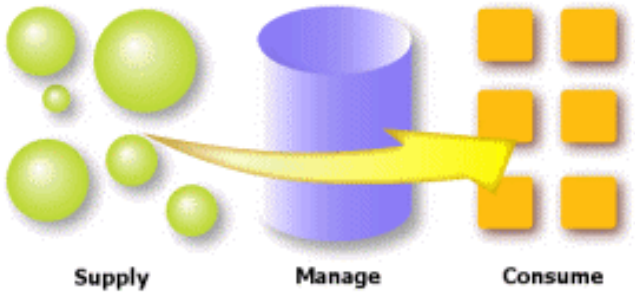

Figure 1 - Suppliers and Consumers

Obviously business solutions are responses to the needs of the business. This is worth restating because to build solutions without a need, is a waste of precious capital and scarce human resources. Solution delivery is a consumer of components and services. It may construct other software parts, such as the user interface, but generally it assembles particular services in specific patterns to form the solution.

The component suppliers have their own delivery process that responds to requests for services and supplies matching components. They are driven by a need for a particular set of services rather than a business need.

To connect suppliers to consumers there is a person, team, or technology that acts as a broker to the parties. This 'broker' facilitates the flow of requests and responses between the different groups.

The implication of this model is that the suppliers and consumers are different organizations, groups, teams etc. This may be so, but they are in fact different roles that can be fulfilled by people within a small team on their particular project. For larger projects the roles may be filled by other teams, groups etc.

Select Perspective encourages and supports the community of component suppliers and consumers and the brokerage role between them.

Consume Workflow

"We were very clear on our roles and what to specify for the developers" Business Analyst, major insurance organization.

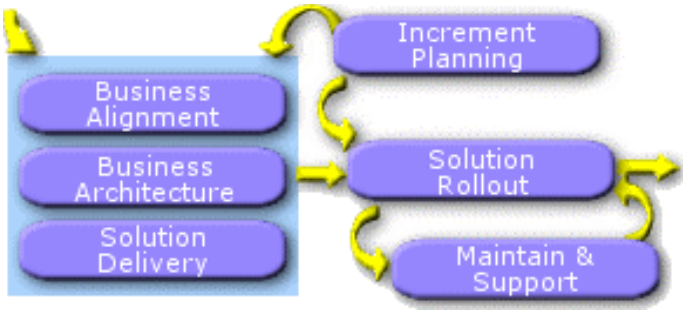

Figure 2 - Consume Workflow

The consume workflow (Figure 2) delivers the solution then maintains and supports that solution. This workflow contains four major workflows with two administrative workflows for planning and support.

Activities in business alignment, business architecture, and solution delivery are such that these workflows can start at the same time. For example, business alignment captures the business requirements that may require developing some prototype screens to aid user understanding; prototyping the user interface is part of solution delivery.

Business alignment delivers the business process models, the use cases describing the functional requirements, and the user acceptance test scripts.

Business architecture delivers the component specifications together with their test specifications. Solution delivery consumes the components from the component suppliers. It adds the user interface components and then merges with the technical architecture components to deliver the solution. This solution may or may not be rolled out depending on the delivered functionality e.g. incremental delivery.

Key to this parallel working is increment planning. This determines which activities within these workflows are to be undertaken for each new increment.

Select Perspective recognizes the essential need for alignment with the business and delivery of solutions within a service-oriented architecture.

Supply Workflow

"Even though we are remote from the business users, the component specifications made it so easy" Systems Developer, major insurance organization.

The supply workflow (figure 3) delivers and maintains components. When a request for services is received there is a negotiation on the specification e.g. splitting some services, supplying two components, altering the service specification to take advantage of existing components. This results in an agreed component specification - a form of contract between the component developers and their consumers.

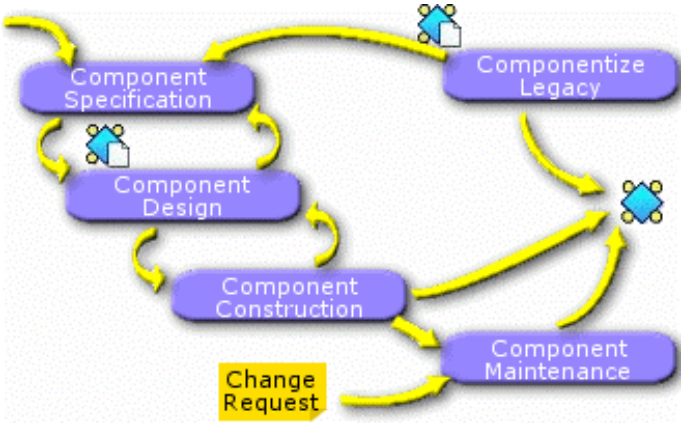

Figure 3 - Supply Workflow

The component is designed, constructed, and fully tested before being delivered to the consumers. Subsequent change requests cause the normal impact analysis, updates, and regression testing activities before the update is issued.

So far this workflow description has been about new components, but components can also be extracted from existing legacy systems. The supply workflow assesses the componentization strategy—wrapping, reengineering, reverse engineering—and tackles the changes without disrupting the operational system. Some areas of the legacy system will no longer be suitable, so component specifications, derived from the existing code, need to be drafted and used to construct new

components. This typically happens when the existing functionality lacks the performance for a new solution.

Select Perspective integrates new functionality, legacy harvesting and package solutions into one homogenous, service-oriented architecture.

Component Management Workflow

In the component management workflow (figure 4) there are two distinct streams of activity. One stream is concerned with the acquisition, certification, and publication of components. The other stream focuses on the location and retrieval of candidate components for reuse.

Components can be acquired from a number of sources. First there has to be a definition of the acquisition strategy e.g. core business components to be built, all others to be bought. Based on this strategy component specifications can be accumulated and issued to the organization's development teams, to trusted partners, or sought from commercial sources.

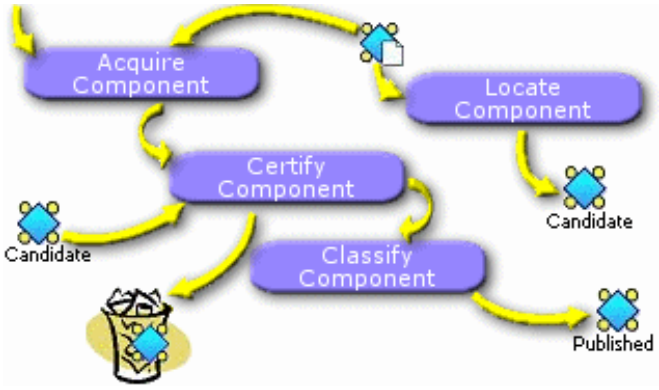

Figure 4 - Component Management Workflow

When the candidate component arrives it undergoes formal testing and certification; it can, of course, be rejected. If the component is certified then after classification and storage in the component repository, it is published ready for subsequent reuse.

When solution or component developers have specified the services required for their construction work, they then search the component repository for matches to their requirements. Whenever candidates are discovered they can be retrieved and examined, perhaps even tested, for their suitability before they are finally reused.

Select Perspective supports all forms of component and service acquisition.

Flexible and Agile

Select Perspective has been designed to deal with a broad range of solution development situations. As a development method, it recognizes the false nature of the "one-size-fits-all" philosophy. A danger with most process descriptions is that they fit only one kind of project, most commonly a **Greenfield** development. Such projects deliver new solutions utilizing newly constructed components and services.

In place of this, the definition of Select Perspective provides a core of processes, best practices, techniques, product definitions and risk descriptions. These best of breed resources can be customized to fit with an organization's culture and to support many different profiles of development project.

Brownfield development projects use a superset of the normal (Greenfield) process; there will be legacy systems and other existing assets to be factored into the design.

Web-driven development has extra activities associated with the Web content delivery, technical architecture and deployment of the solution e.g. to 'Web farms'.

Legacy rejuvenation projects need extra care in key areas to ensure a successful rejuvenation.

Integrating packaged solutions into the business and systems environments involves changes to the basic process.

Simple projects can avoid the administrative and management overhead associated with larger projects and are able to exclude certain processes and products entirely.

It is unlikely that any project will fit neatly into one of these profiles. Select Perspective allows a mix-and-match approach to defining a project process. An optimal plan can be created which minimizes effort and exposure to risk.

Agile Software Development

Throughout the history of software development there has been an ongoing debate between the advocates of formalized waterfall-based methods such as the "V" lifecycle and SSADM, and the proponents of iterative, prototype-based approaches such as Rapid Application Development (RAD) and Boehm's Spiral Method. Time and again, the debate between formal methodologies and simpler approaches has reached the same conclusion: *that elements of both are needed to meet the requirements of the superset of all projects*. It is only by considering the specific needs of a project at its inception that decisions about a development approach can be made.

The same debate is raging anew. The enemy is complexity, embodied in heavyweight methods such as the Rational Unified Process (RUP). The rebels fly the banner of Extreme Programming, (xP), eschewing complex processes and modeling approaches by returning to the roots of development: the programming code.

xP proponents reluctantly admit however, the need for some kind of core approach aimed once again, at a balance between the fast-moving requirements of development teams and the longer-term needs of the enterprise. There is recognition that any processes, methods, and tools need to be flexible not just at the start of a project but throughout its length, as new needs arise and new risks have to be mitigated. These are the key elements of facilities for what is termed *agile software development*.[2]

Agile software development requires *agile processes* and one thing these processes can never be, is an "off-the-shelf" methodology. Any textbook methodology that expresses sequencing of activities, inevitably constrains the execution of those activities. If it is large, it will appear monolithic. If small, it will be inadequate for more complex needs. In reality, we can make things even simpler by stating: "a defined process can neither be agile nor inflexible; it is how it is applied that counts".

---

[2] The term *adaptive* is sometimes used in place of *agile*. For more information refer to the white paper "Objective Based Planning – Managing Projects With Agility".

If Select Perspective is only delivered as a hard-copy manual then it could not be considered an agile process. It includes a multitude of activities that may be applied to every conceivable software development situation. It also incorporates best practice principles that may be applied by team members and project managers alike. To be agile Select Perspective requires mechanisms to enable the process to react to changing circumstances. Process Management tools, such as Select Process Director, provide these mechanisms. These tools enable managers and planners to:

- Define the most appropriate process for the project in hand often providing a number of process patterns, for example for simpler "green field" projects, for legacy integration, or for Web application development
- Customize the process to meet the needs of the project. In any process, some activities are optional and may be included or excluded. Other processes have alternatives, for example to cope with greater or lesser complexity, whilst others are required but have already been carried out and do not have to be repeated. The tool will guide the manager through the process, enabling informed decisions on what to include, exclude, or replace.
- Combine tasks from different workflows into the plan to ensure that milestones established during the project are met in an optimal manner.
- Assess the risks and make adjustments to the process at any stage. Potential risks are presented to the project manager together with explanations and mitigations. The latter may include recommendations to include or exclude activities from the process.
- Incorporate actual project data. Synchronization with project plans enables initial plans to be generated and resourced, and metrics to be harvested automatically over the life of the project, saving time to be spent on real project issues rather than administration.

Proponents of extreme agile development are often opposed to the use of design techniques, in particular modeling of requirements or design features prior to coding. For example, use cases have been criticized as a key cause of "analysis paralysis" in which the models become a drain on development resource, reducing the time available to develop the application. From its inception Select Perspective has advocated the use of multiple, best of breed modeling techniques[3] both text and diagram-based. These techniques exist as a kit bag that simplifies the lives of business analysts and developers alike. Select Perspective does not advocate the use of all the techniques, all the time; only the essential models need to be used to understand an application and its context.

[3] For example, the Select Perspective was first to link Ivar Jacobson's use cases and interaction diagrams with the core of Rumbaugh's OMT which together form much of the core of UML.

Summary

Select Perspective is invaluable for CIOs, IT Directors, Project Managers, Analysts and Solution Designers and Developers who need a contemporary software development process that meets the many demands of modern solution delivery. Select Perspective is a comprehensive development lifecycle for delivering component-based solutions operating in a service-oriented architecture. It supports business-aligned, scalable, parallel development in order to reduce time-to-market.

To find out more about how Select Component Factory and Select Perspective can solve your problems, visit: http://www.selectbs.com/

Or call our Sales teams who will be pleased to help with any questions you may have:

North America
Englewood, Colorado
Tel: +1.800.717.7761 (US only)

Email: info@selectbs.com
Trumbull, CT
Tel: +1.888.472.7347 (US Only)

France
Tel: +33.(0)141.481034

Email: infofrance@selectbs.com

Germany, Austria, Switzerland
Tel: +49.(0)721.98653.0

Email: infogermany@selectbs.com

United Kingdom
Tel: +44.(0)1452.638800

Email: infouk@selectbs.com