Microsoft application lifecycle evaluation

Bola Rotibi

July 2006

Microsoft application lifecycle evaluation

The Ovum view

The industry has waited a long time to see the results of announcements made by Microsoft about the next iteration of its technology stack and the tools that will leverage it. Microsoft has not disappointed: collaboration, productivity and greater organisational control and relevance are the foundations of Microsoft's Visual Studio 2005 development platform, particularly its aptly named, team-focused product, Visual Studio Team System.

With its Team System solution Microsoft is trying to recreate the value-add that Visual Basic brought to windows programming for millions of enterprise developers – only this time to the application lifecycle. Considering the resources available to the company and its success with Visual Basic, there is little reason to question Microsoft's ability to succeed in its strategy.

Visual Studio 2005 Team System provides a framework that maps architectural design with operational and hardware architecture and management. The Visual Studio development platform is a single unified server that incorporates version control, work item tracking and build management.

The creation of the Expression products and XAML programming script that intends to bring presentation and graphic designers into the lifecycle process is both commendable and visionary.

Microsoft has set out an aggressive pricing and licensing model for Visual Studio 2005, and in the process unveiled a host of changes in the Microsoft Developer Network (MSDN) subscription service. The result of this saw a major rebrand and a shift in emphasis of product coverage and support services. For the most part we welcome any simplification in pricing and product nomenclature that enables users to get to the relevant and most appropriate solution quickly. However, in practice, Microsoft has not been consistent with the clarity of all of its messaging surrounding its development platform. There still remains some confusion around which components are available on general release and which ones are beta or community test products.

For existing Microsoft customers the products, when delivered in full, will offer a greater collaborative environment that, if used intelligently, will

enable the development of applications that will raise the experience of the user interactions and the potential of delivering real value to the organisation. For non-Microsoft customers it will be more difficult to ignore the company, while providing a set of yardsticks to prod other vendors prominent in both the software development and application space.

Strengths

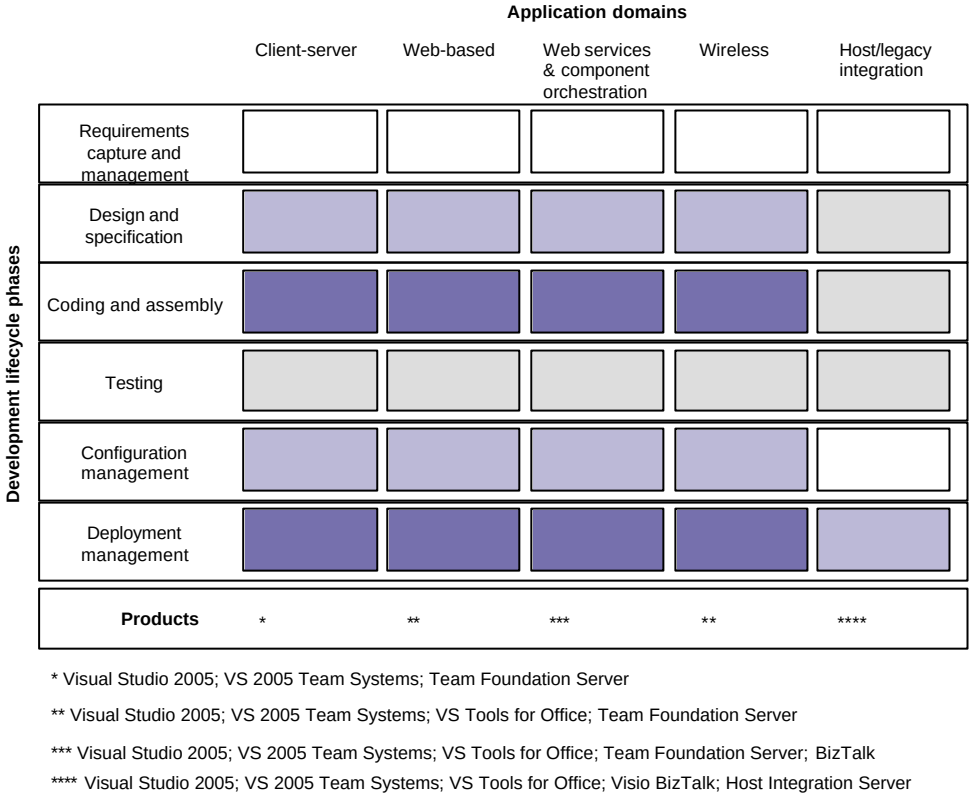
- Single unified programming model leveraging developer and designer skills, along with the productivity and collaboration to build applications across a number of platforms.
- Strong development background with a good understanding of the development and architectural issues facing businesses.
- Powerful strategy and roadmap for its Windows platform infrastructure and the development, delivery and management environment, which will see greater control and ease of use for all stakeholders.

Points to watch

- Development, delivery and management strategy is perceived to only focus on a Microsoft-centric environment with little consideration for other platforms.
- While Microsoft is offering a comprehensive development, delivery and management lifecycle solution, users may find it confusing to identify all the relevant pieces for effective design and deployment.
- Information around future versions of the Visual Studio platform and Windows operating system needs to be more effectively managed to prevent further confusion with delivery schedules and availability of new features.

For Ovum's view of Microsoft, please see the Microsoft vendor analysis.

Products



Microsoft offers a comprehensive development platform that directly supports most of the key phases of the software development lifecycle. More importantly, it does so within a framework that looks to provide full lifecycle and end-to-end coverage, touching on all parts of the system – from development, deployment through to operations and finally application retirement.

Since establishing .NET, Microsoft has focused on strengthening its technology and development capabilities. The .NET strategy advocates a single architectural approach and programming model, which allows it to offer the developer the ability to write an application, regardless of which Microsoft platform it will finally be deployed to – client or server, wireless or fixed. It is with this productivity benefit of transferable developer skill and reduced complexity that Microsoft markets its Visual Studio Suite of products.

Visual Studio continues to be the development powerhouse for Microsoft's product portfolio, with increasing development links into many of the company's other products. It offers strong reliable software development

tools for the construction of client-server, web and web services applications. For wireless developers, Visual Studio offers a good proposition for users who are more interested in functionality and less concerned with factors such as style and battery life. Embedded developers are catered for with specific tooling, but Microsoft has still to fully grasp how this group traditionally works. The toolset for host and legacy integration application is a mixed bag with development being spread across a number of products.

While the main development environment is the Visual Studio platform, the .NET Framework is Microsoft's programming model for building, deploying and running web services, web, Windows and wireless-based applications. It has three parts:

- the common language runtime (CLR) for executing applications, built on top of operating system services
- a hierarchical set of unified class libraries, through which developers can access services in runtime, and which can be used from many programming languages (for example, C#, C++, Visual Basic and J#)
- a component-based version of Microsoft Active Server Pages (ASP). This provides higher-level services specifically for XML web services.

Microsoft's .NET platform remains a powerful alternative to Java and the Java tool providers.

Testing facilities have been greatly improved, especially for developers. While there are good unit testing, code analysis and profiling tools, along with stress testing and functional testing of web applications, they are still no match for the more powerful testing solutions offered by companies such as Compuware, Mercury, Empirix and IBM Rational.

As usual, Microsoft has not been afraid to take on a strategy that it has not been the first to develop. The 'user experience' and usability angle has very much been the domain of the likes of Macromedia and Apple. Evidence of further 'borrowing' is littered throughout the Microsoft product set and technology stack.

The Microsoft Expression product range is a family of professional tools for the design and production of enhanced user experiences and rich content for the Web and Windows Vista platform. Microsoft supports developers – hobbyist and professionals – and designers alike but, more importantly, it wants to own the middle ground where a hybrid of both roles is seen as key to delivering experience based applications.

The Expression tools leave no doubts about the seriousness with which the company takes the importance of the user experience, both in the creation of applications and the use of applications. More importantly, the company's product and marketing strategy acknowledges what we believe

to be the new battle lines for environments tasked with delivering user-facing applications.

Competitive pricing has been a significant factor for Microsoft, which has resulted in a major rebranding of its maintenance and software assurance model, and a restructure of the pricing model and delivery mechanism of the various role-based components that make up Visual Studio Team System.

Requirements capture and management

Microsoft does not offer a requirements management solution within its product portfolio. Microsoft relies on ISVs in its Visual Studio Integration Program (VSIP) to provide additional functionality in this area. A notable partner product is CaliberRM, the requirements definition and management tool from Borland. In November 2005, the company announced full integration between its requirements management system for business and technical analyst and Microsoft's Visual Studio 2005 Team System. As a result, development teams will be able to gather, track and manage requirements directly within their Visual Studio Team System environment.

Integration with other requirement management tools is supported by other ISV partners. More information is available on Microsoft's partner website.

Design and specification

Modelling and design centred on the outdated Visio modelling product has not traditionally been a strong point of Microsoft. Visio has lacked the ability to support more in-depth enterprise modelling requirements. However, today the product's design capabilities have been superseded by the more sophisticated and comprehensive capabilities of the Domain Specific Language (DSL) toolkits and the offerings from ISV partners that provide deeper modelling and process extensions to the DSL APIs. Once again, partners such as Borland have provided UML 2.0 adapters for Microsoft's DSL within its more sophisticated Together modelling facility.

Key design capabilities are provided within Visual Studio Team Systems for Software Architects, a comprehensive solution providing facilities for distributed system design. Team Systems for Architects brings together a number of designers that leverage Microsoft's System Definition Model (SDM), an XML-based format that stores the model definition and provides a common language in which both application systems and data infrastructure can be described. SDM is a key constituent of Microsoft's Dynamic Systems Initiative (DSI), a commitment by Microsoft and its partners, as well as an architecture for the company's platform, to deliver self-managing dynamic systems that help IT teams capture and use

knowledge to design more manageable systems and automate ongoing operations. DSI underpins Microsoft's ethos for design and build for operations.

The designers within Team Systems for Software Architects are:

- **Logical Datacentre Designer** – this allows for a logical depiction of the infrastructure of a data centre. The diagram does not depict physical machines but encapsulates the specific configuration of application/middleware servers and any interconnections with other server software. Logical servers can then be grouped into zones to define logical communication boundaries
- **Application Designer** – allows for the visualisation of application solutions in terms of units of code that can be deployed as services, smart clients or websites. Fundamentally, it provides a service-based view, along with the ability to manage the relevant metadata. A good example is that you can add constraints and conditions to a service definition that specifies requirements for deployment and determines the logical server types to which it can be deployed
- **System Designer** – provides the ability to create systems from groups of individual services or applications and apply a system-wide configuration, constraint or deployment policy
- **Deployment Designer** – provides the ability to map system designs to logical data centre diagrams created by the Logical Datacentre Designer in order to both determine and assess the logical deployment. The Deployment Designer will check whether the system design can by deployed to a specific server according to the constraints or configuration data.

The model validation facilities certainly offer more sophisticated facilities that ensure configuration policies for data centres are adhered to and any impact of change is well understood.

Another key design facility is the recently announced Visual Studio Team Edition for Database Professionals. This 'skew' of the Team Systems offering provides very good and much-needed change management facilities for database design and development. The product centres more on bringing data development into the lifecycle process. As a result, it is very much more of a tool for modification and manipulation of existing database schemas. Fundamentally, it provides a sandbox environment for database development. A current failing of the product is that it does not yet possess the depth of database modelling and architecture facilities when starting out developing a database solution that some of the more traditional tools provide in this area.

In general, the Visual Studio platform has been beefed up to offer much improved facilities for the design and specification of components and interface artefacts for web, client and wireless-based applications.

Microsoft's EAI platform, BizTalk, offers a toolset that works in conjunction with Visual Studio to do the same for targeted EAI solutions.

The Expression products targeting designers are built on the same programming model and technology framework as the rest of the Microsoft product set, with the level of collaboration and interaction that offers. This should cause many to pause for thought irrespective of their previous affiliations. Technologies like XAML, an XML-based declarative mark-up language for describing content and user interface, allow for greater collaboration between developers and designers, for both to coexist and work without impinging on the effects made by the other.

Coding and assembly

With the .NET Framework, Microsoft provides a managed code environment with strong tooling support for the many different stakeholders. Visual Studio facilities for coding and assembling application artefacts for different application domains are capable and focus on ease of use. There is excellent support for both developing client, web, services and handheld device-based applications.

The key tools for code development in the Visual Studio range are:

- **Visual Studio Team System** – an integrated and extensible suite of lifecycle tools that expands the Visual Studio product line to enable greater communication and collaboration between software development teams
- **Visual Studio 2005 Professional Edition** – a comprehensive development environment intended for individual developers to build high-performance, multi-tier applications for a wide variety of Windows, Web, mobile, and Office-based solutions
- **Visual Studio 2005 Tools for the Office System** – tools to empower IT professionals, ISVs and system integrators to build robust smart client solutions for the Microsoft Office System
- **Visual Studio 2005 Standard Edition** – a highly focused development environment intended for individual developers to build departmental client-server Windows applications, websites, and web and consumer device-based applications
- **Visual Studio Express** – Microsoft's entry-level tool, targeting hobbyist, students and the power user.

Within the Visual Studio development environment greater attention has been provided in creating tools to improve individual productivity and greater collaboration with the rest of the development team. The ability to create and track work items is a feature.

Microsoft has announced .NET Framework 3.0, which was formally known as WinFX. WinFX was the name for the managed code model for the

Windows platform, as well as being the next version of the developer framework. WinFX is composed of:

- **Windows Communication Foundation** – unified framework for rapidly building services-based applications
- **Windows Presentation Foundation** – unified framework for building next-generation experiences with user interface, media and documents
- **Windows Workflow Foundation –** providing a programming model, engine and tools for building workflow-enabled applications
- **Windows CardSpace** – a federated digital identity solution.

Microsoft plans to deliver tooling support for building .NET 3.0 (WinFX) applications using Visual Studio. Tooling support for Windows Workflow Foundation will be available with the release of .NET Framework 3.0. The rest of the .NET Framework 3.0 tooling will be part of the Windows Visual Studio 'Orcas' release. Visual Studio 'Orcas' is Microsoft's next planned version of the Visual Studio platform, expected sometime in the second half of 2007.

The development tools will provide developers with support for building .NET 3.0 applications using the final released version of Visual Studio 2005. Future tools include three proposed tools under the Expression brand, which focuses on bridging the world of design and development, and bringing the designer into the application lifecycle process. The three tools will provide coding facilities aimed at improving the user experience and targeting web design, interactive design and graphical design.

Microsoft Visual Studio 2005 and Microsoft SQL Server 2005 were designed to help users build data-driven applications more productively. On top of this integration, SQL Server Management Studio is a new offering inside of Microsoft SQL Server 2005, which provides an integrated environment for accessing, configuring, managing, administering and developing all components of SQL Server. SQL Server Management Studio combines a broad group of graphical tools with a number of rich script editors to provide access to SQL Server to developers and administrators of all skill levels.

The coding of host and legacy integration applications is not so well catered for and must rely on a combination of:

- the BizTalk toolset
- Microsoft's outdated Host Integration Server integration components (covering data and application integration, and system network architecture connections)
- Visual Studio.

Visual Studio Tools for Office (VSTO) 2005 has been completely overhauled and Microsoft has announced Cyprus, which is a recompiler for Visual Studio Tools for Office 2005. It will take code written for Office 2003

and recompile it for Office Systems 2007. With the release of Office System 2007, there will be a set of upgrades for VSTO but not a new release. This is because VSTO is owned by the Visual Studio team, not the Office team.

Testing

Testing has not been one of Microsoft's strong points, but it has improved with the release of Visual Studio 2005 Team System. Although not up to the sophistication and advance of test environments provided by the likes of Compuware, Mercury and Empirix, the features have come a long way up the value chain.

Both the new developer and tester skew of Team System provide much-needed development testing facilities that tie into the collaborative team environment making interaction more productive. In the developer skew of Visual Studio Team System the following are provided.

- **Code Analysis** – provides the ability to analyse code while it is being built. Many rules are provided out of the box, but custom ones can also be added.
- **Dynamic Analysis** – provides detailed code profiling capabilities to measure the performance of applications at runtime.
- **Unit Testing** – provides unit testing application functions and usage scenarios.
- **Code Coverage** – provides an indication of the amount of code that is covered by the unit test.

The testing skew of Visual Studio Team Systems provides further testing facilities and a management console to help manage, execute and track tests. The specific testing facilities are:

- Unit Tests – with test code and method functionality
- Web Tests – to test and record activity against a web page
- Ordered Tests – to enable grouping to unit and web tests
- Load Tests – allows the replay of any combination of web or unit test
- Manual Tests – defines a set of manual steps that must be executed in order to validate some functionality in an application
- Generic Tests – an existing program wrapped to function as a test in Visual Studio.

Configuration management

Team Foundation Server is Microsoft's newly released software configuration management (SCM) tool that forms a central part of Visual Studio 2005, and relies on many components of that product. This common repository enables the collaboration and relational

interdependencies of software artefacts, created and managed by the various roles supported within the Team System suite.

It should also be the long-awaited and long overdue replacement for Microsoft's Visual SourceSafe product. Sadly, Microsoft has not yet scrapped this woefully deficient product, having recently released Visual SourceSafe 2005.

Team Foundation Server does not use conventional file-based configuration management. Rather, it is centred on changes in the form of 'changesets' and 'work items'. This takes some time to become familiar with, but the effort is worthwhile. The server has a single, unified interface, bringing process and workflow, versioning and requirements management into a single screen display. It has strong features for logging and tracking issues, work instruction authorisations and team communication via email notification, along with a wide range of reports and graphics. The Project Portal has a good dashboard, which allows managers to tailor metrics and charts on application development activities to their needs, and also supports drill-down for more detail where required. In general, the graphical user interface (GUI) is simple and easy to understand, and makes good use of colour and icons to convey status information. The web interface uses the same colours and icons, and the command line interface supports all the functionality available through the other user interfaces.

However, there is a ludicrous lack of a Microsoft web client for check-in/out, although there is a web client for project management (the Project Portal) and reporting.

Team Foundation Server is an attractive option for organisations engaged in offshore or outsourced development, as all source code is maintained in a single, centrally located archive without the need for replication or periodic synchronisation. Development across geographically separated sites is supported via a Source Control Proxy, which means file versions are only copied to the remote site once and thereafter are served from the proxy. The Team Foundation Server model minimises the need for conflict resolution and other administrative activities because there is only one repository to update and manage.

Deployment management

Deployment is managed well within Visual Studio for web- and wireless-based applications. Once again, ease of use is a key strategy of the tool. Integration applications are deployed – depending on their target platform – through the BizTalk toolset and Host Integration Server management tools, in conjunction with Visual Studio.

Within Microsoft's own portfolio of development tools there is good integration and interoperability. However, interoperability outside the

Microsoft toolset, while reasonable, is dependent on your skill in writing COM or ActiveX components, or your ability to license the Visual Studio Integration Program (VSIP) for deeper integration with Visual Studio platform.

Microsoft's online development support facilities are rich and in-depth, offering good support in the form of best practices, forums and technical information. While considerable effort has gone into streamlining navigation around the site, users will still require time, effort and patience to extract the relevant information.

Lifecycle management

Microsoft has come a long way in providing support for the software lifecycle management. The advantages of Microsoft being the sole provider and manager of the Windows environment has translated into the company's strategy for the software development lifecycle and its wider remit for operations and business alignment.

Collaboration and productivity for the individual, team and organisation are the core driving themes of its lifecycle story, as is governance and compliance and general management across services-oriented infrastructure and applications. This collaborative approach is enhanced by the way that Microsoft has built its collaboration solutions under the Microsoft Office brand. Having been through the pain cycle by trying collaboration out on the general user population, Microsoft is now deploying it for development teams. It is long overdue. Too many developers work in isolation from system architects and often neither really share data with the testers. Previous attempts to solve this have been characterised by the 'round trip' approach. This failed because of the problems of maintaining control over the 'round trip' process and because few development shops had the time and resources to devote to it.

Lifecycle management can be seen in the company's Visual Studio development platform, Office System, System Center, the management platform, and future releases of the Windows platform.

Microsoft is also introducing a lot of collateral changes to the development environment that need to be seen as part of the application lifecycle management (ALM). Initiatives and products such as the Dynamic System Initiative (DSI), System Definition Model (SDM) and even the Security Development Lifecycle (SDL) are all part of unifying application operations with the creation and management process. However, they are taking time to bed down and be adopted by developers.

Like others, role-based, process and management-centric strategies underpin Microsoft's lifecycle management offerings. Aside from support for classic roles such as architects, developers, testers and project

managers, the company is adding support for other roles such as database developers and administrators.

Microsoft has been working hard to persuade corporate developers and ISVs that its Team System suite of products can deliver real ALM across the entire development process. What is urgently needed, however, is more emphasis on how Team System works with the data centre. There is already the ability for the ops team to produce virtual data centres, which can be used by architects to plan deployment. What is missing is the ability to fully interrogate the servers and see their security status. This would improve software security by enabling architects to see the security impact of their software pre-deployment.

However, bear in mind that this is still a version 1.0 product from Microsoft. There are a lot of bugs to iron out and a lot of 'proof of concept' yet to appear. Simply deploying collaboration tools does not make for a lifecycle for software artefacts, whether it is Word documents, Excel spreadsheets or code and components. Microsoft has recognised this and as a result has been working on developing maturity models for both the infrastructure and application platforms. The company is working on providing an Infrastructure Optimisation Maturity Model and an Application Platform Maturity Model that covers infrastructure, development, service-oriented architecture (SOA) and business process, data and user experience capability.

However, Microsoft needs to do more to make the overall process more intuitive, to lead the development from one phase to another. This is all about workflow, and despite Microsoft talking about how it can design and implement it, it will continue to remain a black art for some time to come.

While Microsoft is focused on adding more 'roles' to Team System, it must not forget the underlying basics of a solid workflow on which everything can be built.

Client re-use disclaimer

- This is a verbatim reproduction of independent material that has previously been published by Ovum within the last 6 months
- Ovum operates under an Independence Charter. For full details please see www.ovum.com/about/charter.asp
- Neither Ovum nor the analysts were paid by the client to write any part of the material
- Ovum may have been paid by the client for the right to re-use the material
- Ovum may have a deal with the client to supply research or consultancy. However, no other relationship exists between the 2 companies (e.g. shareholdings, loans, non-executive directorships etc)
- Ovum does not endorse companies or their products
- While we take every care to ensure the accuracy of the information contained in this material, the facts estimates and opinions stated are based on information and sources which, while we believe them to be reliable, are not guaranteed. In particular, it should not be relied upon as the sole source of reference in relation to the subject matter. No liability can be accepted by Ovum Limited, its directors or employees for any loss occasioned to any person or entity acting or failing to act as a result of anything contained in or omitted from the content of this material, or our conclusions as stated
- This material is the copyright of Ovum Ltd.