



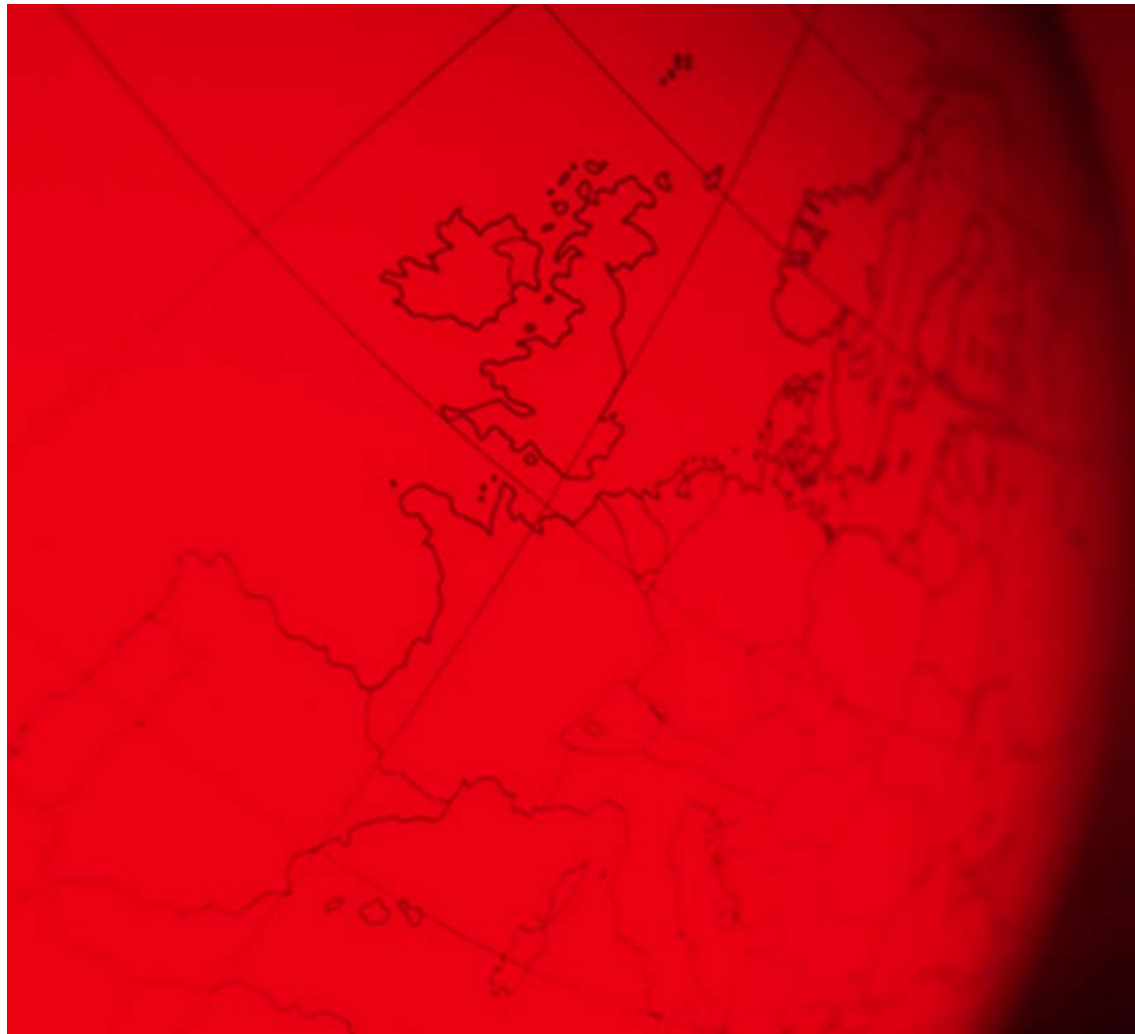
NUCLEUS
RESEARCH

April 2008

Document **119**

GUIDEBOOK

MICROSOFT PATTERNS AND PRACTICES



Corporate Headquarters
Nucleus Research Inc.
36 Washington Street
Wellesley MA 02481
Phone: +1 781.416.2900
Fax: +1 781.416.5252

Nucleus Research Inc.
NucleusResearch.com

TOPICSApplication development
and integration

Enterprise applications

THE BOTTOM LINE

Companies use Microsoft patterns and practices to make their development environments more productive, consistent, and standardized. Whether used by an independent software vendor, consultancy, or enterprise, patterns and practices can accelerate project cycle times, reduce costs, and shift application development efforts away from code building and toward maximizing business benefits.

SITUATION

Organizations are increasingly challenged to cost effectively create, maintain, update, and integrate productivity-enhancing applications that are broadly adopted by end users. Some of the challenges include:

- **Integration.** Organizations are seeking better integration, both internally and externally. Companies often pursue competitive advantage and higher levels of operational efficiency by integrating their value chains — and portions of their applications — with those of their partners, vendors, and customers.
- **Cost.** Skilled developers, especially those with expertise in specialized languages, are difficult to find. Cost reduction strategies such as offshore development and service-oriented architecture (SOA) can be effective, but have their own drawbacks, such as governance and control issues.
- **Obsolescence.** Software development languages tend to have relatively short life cycles and can sometimes be short-lived fads, making development projects more complex and increasing training costs.
- **Evolution.** IT departments must update their applications for compatibility with new technologies such as mobile devices, software as a service, and service-oriented architecture (SOA). Because of the prevalence and rapid evolution of browsers, on-demand applications, and mobile devices, many enterprise applications have become incompatible with how knowledge workers use technology.
- **Performance.** End users and customers have become used to sub-second response times from many applications. This level of performance is becoming steadily more difficult as companies' databases grow in volume and their applications increase in complexity.

MICROSOFT PATTERNS AND PRACTICES

Microsoft patterns and practices provides guidance that helps companies more rapidly design, develop, deploy, and operate architecturally sound applications using the Microsoft application platform. This group of tools is openly available for free over the Web and compatible with a wide variety of software solutions and development tools. The source code underlying these tools is open for modification by end users. This openness, combined with the fact that is provided free of charge, causes many users to associate it with open source development practices. Patterns and practices supports development in a variety of application types, including Web, Windows, SOA, and mobile devices.

Microsoft patterns and practices provides organizations and developers with guidance and development source code in the following forms:

- Application blocks, which are reusable pieces of source code that provide solutions to common and frequently repeated development tasks, and typically eliminate the need to repeatedly create custom code from scratch. There are application blocks for caching, cryptography, data access, exception handling, logging, policy injection, security, and validation.
- Software factories, which help architects and developers create instances of specific types of applications.
- Guides, which provide conceptual architecture and design guidelines for using application blocks, architecting solutions, and using other patterns and practices assets.
- Reference implementations, which are executable sample applications that demonstrate use cases that apply patterns and practices assets.

Although Microsoft provides patterns and practices free of charge, no development-related application is truly cost free, because its adoption will significantly impact productivity, development practices, and software quality. This guidebook is designed to help companies, independent software vendors, and system integrators identify the benefits of using Microsoft patterns and practices and how to maximize those benefits. The findings in this guidebook are based on Nucleus Research's analysis of the experiences of a number of users of Microsoft patterns and practices, including Magenic, Aviva Solutions, Infragistics, Amphire, and Red Prairie.

FIVE BEST PRACTICES

Nucleus found that companies that use Microsoft patterns and practices followed a number of best practices in order to not only increase developer productivity but also to improve the quality, benefits, and deliverables of their projects. These best practices focus first on achieving the productivity benefits available from patterns and practices, then on converting those benefits into higher value strategic improvements such as improved partnering with customers.

Start with the guides and reference implementations

Nucleus found that organizations in general, and developers in particular, were most able to rapidly achieve productivity improvements if they first spent time using the guides and reference implementations. These provide documentation, use cases, and best practices that enable developers to determine what patterns and practices assets are most appropriate for a given development task and how they should be utilized.

Companies successfully using patterns and practices pointed out that the documentation in patterns and practices is far better than is typically available in the software industry. It therefore helps to point this out to developers, who are typically biased against turning to documentation for knowledge.

The more familiar a developer is with object-oriented design and software reuse, the less the guides and reference architectures will be needed. However, developers consistently stated the importance of using these assets:

- *"Early on, we just started working with one of the application blocks. But 30 research hours later it was a mess and we realized there was a much simpler way explained in the documentation."*
- *"Most documentation treats software like a widget, explaining what it is and does. But in patterns and practices, the application blocks are explained in terms of their overall goals, use-case scenarios, and how to modify them."*
- *"It's easy to get going right away. But there's almost too much functionality that you can access. People should spend time in the instructional tools, blogs, and Webcasts so that you can get the full benefit of the functionality."*

Anticipate reuse opportunities

Reuse of software results in significant productivity benefits. However, this doesn't happen automatically, because the creation of productivity-improving reusable assets involves a tradeoff. The larger the core functionality a reusable asset has, the less customization will be required for reuse. On the other hand, the smaller the core functionality an asset has, the more broadly it can be reused. When developers use templates to customize and reuse application blocks or software factories, they should resolve this tradeoff by anticipating what assets will be needed for their project or department, so that the minimum number of reusable assets can accomplish the maximum number of tasks anticipated. Once reuse is underway, these assets should be continually refined and improved.

Nucleus found that companies were able to improve developer productivity by up to 40 percent by reusing assets from patterns and practices.

Users readily identified both the ease of reuse, and its impact on productivity:

- *"The first time we reused the composite application block we spent 400 hours creating an application, mostly because we decided the reusable core needed to be bigger. We've reused it three times, and saved 150 hours each time."*

"We've reused something we created with the enterprise library at least three times. Every time it eliminated about 90 hours from the project."

- *"Our productivity improvement from reuse is around 40 percent. We get to the basic requirements of an application in two or three days rather than a month."*

Shorten project cycle times

Nucleus found that organizations were able to rapidly improve developer productivity and reduce project cycle times by using the pre-built application blocks in patterns and practices rather than building code from scratch. This was particularly true of the Enterprise Library, which contains pre-built and instantly deployable code required by most applications for basic tasks, such as caching, cryptography, data access, exception handling, logging, security and validation.

Nucleus found that companies were able to shorten project cycle times by up to 25 percent — and by even more in the case of vertically-focused development organizations — by using and reusing pre-built patterns and practices assets.

Organizations readily identified productivity benefits from using the pre-built assets in patterns and practices:

- *“Every application requires things like authentication, data access, and exception handling. But I have a library of configurations for these things. So I can have an application up and running in base form in a day rather than a couple of weeks.”*
- *“Everyone has their own take on how to architect things like data access, and people tend to spend too much time on it. With patterns and practices, you shouldn’t bother with this. The application blocks for this are already built, tested, and highly evolved.”*
- *“We used the enterprise library to cut the cycle time for the server part of the project by almost 25 percent.”*
- *“When you reuse code, there is still some decision making. You have to think about how you’ll combine the components rather than what custom code you’ll develop. This way is far less complex.”*

Spend more time on business requirements

Using patterns and practices to eliminate custom code development for low-value development tasks means there is more time to ensure that the application being built actually meets the originally agreed upon business requirements. Most of the companies Nucleus spoke to improved productivity and accelerated project cycle times so they could dedicate more time to solution testing and end-user acceptance:

- *“The functionality is so rich and reliable that people should spend more time thinking about the business requirements and functionality than development.”*
- *“I’ve shifted from fixing bugs to fine tuning user acceptance. This is good for the client. At the end of the project, you’re improving the most important parts of an application.”*
- *“Be prepared not to do lots of analysis and architectural planning in the beginning. You can spend more time looking at the business problem driving your application rather than its wiring and architecture.”*

Build your business, not code

Nucleus found companies got the most from patterns and practices when they were able to convert its benefits — improved productivity, shorter cycle times, and more predictable projects — into more RFP wins and more time to build their business. When development staff spend less time developing code, key people in the development team — whether it is an in-house IT staff or an external software developer — can spend more time building their business. This means getting new customers, finding more RFPs to participate in, and learning more about existing customers’ operating needs. Companies also used increased project predictability — patterns and practices can automate the least predictable development tasks — to better control project timelines, shift to fixed cost bids, and improve on-time delivery statistics.

Companies got the most benefit from patterns and practices when they used productivity improvements to focus their time on more value-added tasks.

Users identified benefits that ranged from faster solution delivery to having more time for marketing activities:

- *"I have lower time to delivery. Projects get more predictable, and with enterprise library alone, I cut two to three weeks from project cycle time."*
- *"The productivity boost is great. When a customer asks for a particular application, I already know how I'll create it. I have a reputation for being faster and less expensive, even with developer turnover between projects."*
- *"Instead of just churning out code, I now spend more time learning new technologies and dealing with customers."*

FIVE FINE TUNINGS

Utilizing Microsoft patterns and practices can improve an application development environment by adding standardization, consistency, and reuse. Nucleus found that companies were most successful when they both recognized these improvements and maximized the benefits from them.

Prepare for adoption resistance

Because patterns and practices introduces standardization and its benefits increase with the rate of reuse, the more broadly Microsoft patterns and practices is adopted, the greater will be the returns. However, changes to development practices typically encounter cultural issues, and Microsoft patterns and practices is no exception. Developers sometimes resist adoption because of its emphasis on reuse, since developers typically prefer to create their own work rather than customize someone else's. Other sources of resistance include biases against shared source, open source, and Microsoft.

Companies used a mixture of championing, word of mouth, training, and organizational mandates to promote adoption:

- *"Developers tend to be anti-Microsoft, or consider patterns and practices to not be real open source. Here, we were a bit hierarchical about adoption. We told developers to come to us with something better if they thought they could convince us — but otherwise we were going with patterns and practices."*
- *"We found that sometimes there would be lead developers who adopted it, and really understood how it worked. But their staff sometimes used it without understanding why or how it had benefits. If the lead quit, the knowledge didn't always get passed on. This makes training — especially the documentation — really important."*
- *"We found sometimes people loved open source but hated Microsoft, or thought patterns and practices was heavy handed. It's especially important for these people to get trained, or there will be some backlash."*
- *"It's definitely better for enterprise organizations. The fewer people using it, the less benefit you get."*

Configure for openness

Because the application blocks and software factories in patterns and practices are highly configurable in order to maximize reuse, they enable developers to create applications that are highly flexible and interoperable. For example, a reusable core can be designed so that it can be interoperable with one or more of several databases with which it might interact. Or an application block can be designed so

that one or two of its features can be adjusted after it is deployed — possibly by non-IT staff.

Organizations using patterns and practices used its flexibility to deliver customers solutions that were not only flexible and readily adapted, but also capable of integration with a range of other applications:

- *“Integrating or adding a new feature is less expensive and complicated. In the data access application block, we can write the data layer so that it is database independent. So it’s easy and fast to change the configuration depending on whether it will access SQL or Oracle.”*
- *“We use the configurability to make the software more adaptable to changes in the business environment.”*
- *“As long as you’ve used the software factory right, it only takes ten seconds to change a configuration. This is a huge benefit — you can enable an application to target multiple databases and it makes the software more adaptable to changes in the business environment.”*

Update your cost structure

Nucleus found companies got the most benefit from patterns and practices if they updated their cost structure to reflect improved developer productivity and shorter cycle times. Companies also stated that because the application blocks enable elimination of source code development for the most labor-intensive development tasks, their project timelines and costs became more predictable.

Lower costs and more predictable projects enable companies to respond to RFPs more aggressively and confidently, and possibly shift to fixed-price bids:

- *“The application blocks can eliminate the most code-intensive and unpredictable parts of a project. So we are more often submitting fixed bids — and lots of times we finish it under budget.”*
- *“There are lots of developers who know patterns and practices. People are more familiar with it than other open source platforms such as Hibernate. So recruiting is easier and less expensive.”*
- *“We can talk to a customer with a lot more confidence, whether the project is CRM or ERP. We can work with any authentication source or provider. So we can cost better and are more aggressive and confident responding to RFPs.”*
- *“It enables development in a disconnected environment, which means you can choose from a variety of development assets and reduce costs.”*

Update your development approach

Beyond reducing costs and cycle times, patterns and practices also enables companies to shift from linear development practices with one testing cycle to more agile development practices, in which the core version of an application is created earlier on in the project and then steadily refined and improved through successive and iterative test-debug-refine cycles. The more an application is tested and refined during its development, the better will be its performance and the greater will be its end-user acceptance and adoption.

Organizations also used patterns and practices to decouple their projects from individual developers, teams, and geographic locations. The software assets and

support tools in patterns and practices are highly standardized and uniform. So when they are adopted broadly in a development organization, it means that more developers can work on more projects, which gives organizations flexibility and scale that enables greater utilization of their development staff and higher returns on investments in training and personnel.

Patterns and practices users readily identified ways to improve the agility and flexibility of their development staff:

- *"Patterns and practices is really good for large development teams. It gives them the ability to develop in a modular fashion while maintaining architectural control. Disparate teams on a project don't need to know much about one another in order to work together."*
- *"I leverage it to get more consistency and independence. I outsource a lot less, so I have more control."*
- *"We're shifting from waterfall to agile development — we create and test a release every two weeks, and the customer sees it more times before it's done. For this, you need a lot of technical issues settled upfront, and patterns and practices lends itself to this."*

Create applications that are less costly to maintain

The application blocks and software factories in patterns and practices are built with fewer lines of code than if they were built with custom code by a developer. Fewer lines of code means end users achieve performance advantages over applications developed in other platforms:

- **Stability.** Applications built using patterns and practices are less complex, have less downtime, and are easier to change and trouble shoot.
- **Testing.** By building an application with patterns and practices, developers will have fewer lines of code to test, analyze, and debug in the final stages of a deployment.
- **Maintenance.** The fewer lines of code an application has, the less labor intensive it is to upgrade, modify, or integrate with other applications.

Users readily identified benefits to creating less bulky applications for their customers:

- *"What may take 15 lines in .NET may only take one if you use the patterns and practices enterprise library. Having less to look at is important when it comes time to implementing or modifying applications."*
- *"If your end user is upgrading or modifying an application and there's a bug, there are a lot fewer lines of custom code to look through and a lot fewer unit tests to complete."*

FIVE MISSTEPS TO AVOID

Patterns and practices is supported by both a dedicated team within the Microsoft Developer Network, as well as well as a robust developer community via forums. Nucleus found that companies were most successful if they understood the value of this support and avoided a few common missteps.

Don't assume it's not supported

Organizations typically resist using open source because it is usually not proprietary owned or supported by a large software vendor and has been released to the public domain to generate billable hours for developers and consultants skilled in the language. Companies should be aware that these standard concerns about open source do not apply to Microsoft patterns and practices.

Microsoft created patterns and practices to broaden its footprint at companies and accelerate adoption of applications such as SQL Server, .NET, and Biztalk Server. It therefore provides more documentation and developer support than is typical for open source solutions.

When considering adoption of Microsoft patterns and practices, companies should anticipate resistance based on the standard open source model, and be prepared to make the case for patterns and practices based on the level of support provided by Microsoft and the breadth of the developer community:

- *"Microsoft's patterns and practices group is very helpful in assisting with problems. I'm impressed by the amount of influence you can have as a member of the community. I'm an advisor to them for the server factory and the new versions are better because of my input."*
- *"Microsoft is very approachable and responsive about our recommendations about changes. Each component we use has a wiki page set up with forums where you get information from technical champions for that tool."*

Don't overlook the community

Microsoft patterns and practices has an extensive network of developer-focused assets, such as communities, portals, and blogs that developers can use to download software, suggest features, report bugs, as well as learn about and properly use the assets within patterns and practices. Codeplex, Microsoft's project hosting Web site, is a portal where developers can start, join, or manage a project; download software created by the community, or learn more about patterns and practices.

Patterns and practices users readily identified the benefits of turning to the developer community:

- *"The community is where the real wealth of knowledge is, so don't restrict yourself to just the documentation. There are lots of 'MVPs' providing knowledge on the Web."*
- *"The support that's out there is a big success factor. The community is huge, there are lots of training materials, and lots of people use it."*
- *"People should go to Codeplex. It hosts lots of discussion groups. You can get valuable information about how to use patterns and practices, learn about potential pitfalls, and solve common problems."*

Don't ignore the documentation

Although virtually all applications and development tools have extensive documentation, end users and developers rarely rely on documentation for meaningful help or guidance. Rather, people tend to learn to use applications with

a combination of experimentation, ad-hoc informal training, and word-of-mouth advice from colleagues. However, end users of patterns and practices readily indicate that its documentation is far better written, informative, and useful to developers than documentation typically is. Additionally, since improper customization of a software factory or application block can impede productivity improvements or application maintenance, it's particularly important for developers to utilize the documentation provided in patterns and practices. One user said, *"You have to push the developers towards the documentation. They assume it will be sparse and they are used to learning things by word of mouth and trial and error."*

Don't just dive in

Most users that Nucleus analyzed said using the pre-built software assets in patterns and practices is easy — but only after some self training with the various documentation assets, including tutorials, hands-on labs, and non-standard documentation assets such as Webcasts. Generally, the more experience a developer has in reuse-intensive or object-oriented development practices, the more readily they will adopt patterns and practices. However, developers accustomed to more language-based development platforms — such as C, C++, or visual basic — should read about the design and use cases of an asset before using or customizing a software factory or application block. Users often recommended that developers start with the enterprise library, since it is relatively straightforward, and then move to more complex tools such as the composite application blocks.

Users were fairly consistent in recommending research and self training prior to developing with patterns and practices:

- *"It's easy to get going and there's lots of functionality you can access right away. But you should spend time in things like hands-on labs, instructional blogs, and Webcasts before you start building."*
- *"The software factories come with lots of guidance packages, for example it can provide automation within Visual Studio."*

Customize – but not too much

Companies that achieved significant productivity benefits by using templates to customize and reuse software factories or application blocks stressed the need to leave the source code within patterns and practices unchanged. Microsoft is continually making enhancements to these tools, so any changes to the core of one of these assets may cause it to require customization or rework if Microsoft later makes modifications to it.

Although users readily identified benefits from creating reusable assets, they also cautioned others not to change the source code:

- *"Be careful in customizing the software factories. The source code can be changed. But if you do, and Microsoft releases a new version, you may have a hard time keeping your customized version compatible."*
- *"Don't customize the libraries and other pre-built assets, even though Microsoft gives you the source code. It can lock out some benefits. Instead, take advantage of the configurability or extension points."*

CONCLUSION

Application development organizations can improve the applications they create by using Microsoft patterns and practices to make their development environments more productive, consistent, and standardized. Whether used by an independent software vendor, consultancy, or enterprise, patterns and practices can be used to create reusable software assets, accelerate project cycle times, and reduce costs. Nucleus has found that companies were most successful when they used these operational benefits to shift organizational efforts toward higher-value tasks such as improving application end-user acceptance, improving development practices, and winning more RFPs. Although patterns and practices is a powerful tool for improving development practices and reducing costs, companies considering using it should be prepared to carefully manage cultural issues in order to maximize consistency of use, breadth of adoption, and benefits from reuse.

Nucleus Research is a global provider of investigative technology research and advisory services. Building on its unique ROI case study approach, for nearly a decade Nucleus Research has delivered insight and analysis on the true value of technology and strategies for maximizing current investments and exploiting new technology opportunities. For more information or a list of services, visit NucleusResearch.com, call +1-781-416-2900, or e-mail info@NucleusResearch.com.